Documentation

OpenScape Office V3
myReports, Reference

Description

A31003-P1030-T100-01-7618

Siemens Enterprise Communications
www.siemens-enterprise.com

SIEMENS

Our Quality and Environmental Management Systems are implemented according to the requirements of the ISO9001 and ISO14001 standard certified by an external certification company.

Reference No.: A31003-P1030-T100-01-7618

Siemens Enterprise Communications
www.siemens-enterprise.com

Contents

1 Introduction . . . 6
1.1 Display Conventions . . . 6
1.2 Overview of the Documentation . . . 7

2 Introduction to Reporting . . . 8
2.1 Overview of the Predefined Report Templates . . . 8
2.2 Report Designer . . . 48
2.3 Report Parameters . . . 48
2.4 Output Values . . . 49
2.5 Creating Reports . . . 50
2.6 Definition of a Contact Center Call . . . 50

3 Predefined Report Templates in Detail . . . 55
3.1 Report Group Agent Activity . . . 55
3.1.1 Agent Activity Logged Times . . . 55
3.1.2 Agent Activity Missed Call Times . . . 58
3.1.3 Agent Activity On Break Times . . . 61
3.1.4 Agent Activity Status (All Agents) – Daily . . . 65
3.1.5 Agent Activity Status (By Agent) - Daily . . . 66
3.1.6 Agent Activity Work Times . . . 69
3.2 Report Group Agents . . . 73
3.2.1 Agent G.O.S. . . . 73
3.2.2 Agent G.O.S. (Daily) . . . 75
3.2.3 Agent Private Calls (All Agents) . . . 77
3.2.4 Agent Private Calls (Per Agent) . . . 79
3.2.5 All User Calls (By Agent) . . . 82
3.2.6 All User Calls (By Agent) 2 . . . 86
3.3 Report Group Call History . . . 90
3.3.1 External Calls Per User . . . 90
3.3.2 Incoming Calls (Free-Calls) - Per User . . . 93
3.3.3 Incoming Calls (International) – Per User . . . 97
3.3.4 Incoming Calls (Mobile/Cell) – Per User . . . 101
3.3.5 Incoming Calls (Other External Calls) – Per User . . . 105
3.3.6 Incoming Calls (Specific Calls) – Per User . . . 109
3.3.7 Incoming Calls Per User . . . 113
3.3.8 Incoming Calls Report – Group . . . 116
3.3.9 Incoming Calls Report – Group Summary . . . 119
3.3.10 Incoming Calls Report – User . . . 121
3.3.11 Incoming Calls Report – User Summary . . . 124
3.3.12 Internal Calls Per User . . . 125
3.3.13 Missed Calls (Incoming) Per User . . . 128
3.3.14 Missed Calls (Incoming) Per User 2 . . . 131
3.3.15 Missed Calls (Outgoing) Per User . . . 134
3.3.16 Missed Calls (Outgoing) Per User 2 . . . 137
3.3.17 Outgoing Calls (Free Calls) – Per User . . . 140
3.3.18 Outgoing Calls (International) – Per User . . . 144
3.3.19 Outgoing Calls (Mobile/Cell) – Per User . . . 148
3.3.20 Outgoing Calls (Other External Calls) – Per User . . . 152

3.3.21 Outgoing Calls (Pay Calls) – Per User 156
3.3.22 Outgoing Calls (Specific Calls) – Per User 160
3.3.23 Outgoing Calls Per User 164
3.3.24 Outgoing Calls Report – Group 167
3.3.25 Outgoing Calls Report – Group Summary 170
3.3.26 Outgoing Calls Report - User 172
3.3.27 Outgoing Calls Report - User Summary 175
3.4 Report Group Calls 177
3.4.1 Abandoned Calls Statistics 177
3.4.2 Abandoned Calls Statistics – Details 181
3.4.3 Answered Calls Alert Times 183
3.4.4 Answered Calls Alert Times (All Agents) 185
3.4.5 Answered Calls Alert Times – Details 187
3.4.6 Answered Calls Statistics 191
3.4.7 Answered Calls Wrap-up Information 195
3.4.8 Call Traffic All Agents – Per Hour Daily 200
3.4.9 Call Traffic All Agents – Per Hour Daily – Details 205
3.4.10 Call Traffic All Queues – Per Hour (Daily) 210
3.4.11 Call Traffic All Queues – Queue Time, GOS Per Hour Daily 213
3.4.12 Call Traffic By Queue – Per Hour Daily – Details 215
3.4.13 Call Traffic One Agent – Per Hour Daily 218
3.4.14 Call Traffic One Agent – Per Hour Daily – Details 223
3.4.15 Call Traffic One Queue – Queue Time, GOS Per Hour Daily 229
3.4.16 Callback Calls 232
3.4.17 Calls List Agent 236
3.4.18 Calls List Queue. 238
3.4.19 Contact Center (Per Agents) – Chart 240
3.4.20 Contact Center (Per Agents) – List 242
3.4.21 Contact Center (Per Queues) – Chart 244
3.4.22 Contact Center (Per Queues) – List 246
3.4.23 Contact Center Calls 248
3.4.24 Contact Center Summary 253
3.4.25 Contact Center Summary 2 255
3.4.26 Contact Center Summary – Answered Calls 257
3.4.27 Contact Center Summary – Details 259
3.4.28 Missed Calls Report 261
3.4.29 Missed Calls Summary (Per Agent) 265
3.4.30 Missed Calls Summary (Per Queue) 266
3.5 Report Group - Other 269
3.5.1 Calls History Per User 269
3.5.2 Default Break Information 271
3.5.3 External Directory User Details 272
3.5.4 Fax Journal – Received Faxes (By User) 273
3.5.5 Fax Journal – Sent Faxes (By User) 276
3.5.6 Fax Transmission Report 280
3.5.7 Incoming Calls Report – Hourly 282
3.5.8 Incoming Calls Report – Hourly Per Weekday 284
3.5.9 Internal Directory User Details 285
3.5.10 Voicemail Center (All Users) 286
3.5.11 Voicemail Center (By User) 288

3.6 Report Group - Performance . . . 291
3.6.1 Abandoned Calls Per Hour . . . 291
3.6.2 Agent Calls Percentage . . . 292
3.6.3 Agent Performance Details . . . 294
3.6.4 Answered Calls Per Hour . . . 297
3.6.5 Call Traffic By Queue Per Hour . . . 299
3.6.6 Call Traffic By Queue Per Hour (Daily) . . . 300
3.6.7 Contact Center Traffic Per Hour . . . 302
3.6.8 Contact Center Traffic Per Hour (Daily) . . . 303
3.6.9 Missed Calls Per Hour . . . 304
3.6.10 Summary of Details per Agent . . . 306
3.6.11 Summary of Details per Queue . . . 322
3.7 Report Group - Queues . . . 330
3.7.1 Agent Calls Queue Specific . . . 330
3.7.2 Agent Properties . . . 331
3.7.3 Agent Queue Load . . . 333
3.7.4 Avg. G.O.S Per Queue . . . 334
3.7.5 Avg. G.O.S. Per Queue (Daily) . . . 335
3.7.6 Missed Calls Per Queue . . . 337
3.7.7 Queue Summary Details. . . . 339
3.7.8 Queue Traffic Comparison . . . 342
3.8 Report Group - User Presence Status . . . 344
3.8.1 User Presence Status (All Users) – Daily . . . 344
3.8.2 User Presence Status (All Users) . . . 346
3.8.3 User Presence Status (By User) – Daily. . . . 349
3.8.4 User Presence Status (By User) . . . 352
3.9 Report Group - Wrap-up Codes . . . 356
3.9.1 Wrap-up Code Usage All Queues . . . 356
3.9.2 Wrap-up Code Usage Per Group . . . 359
3.9.3 Wrap-up Code Usage Per Queue . . . 362
3.9.4 Wrap-up Code Usage Per Wrap-up . . . 365

4 myReports User Roles . . . 369

5 myReports Software Architecture . . . 371

6 myReports Data Relationship Model . . . 372

Index. . . . 373

1 Introduction

myReports is an application for creating reports on Contact Center agents and their activities, including calls, queues, performance, GOS (Grade of Service) and wrap-up codes.

The myReports Reference Manual is intended for myReports users who wish to

- create reports using existing report templates
- define their own report templates.
  In order to use the Report Designer, familiarity with the application itself as well as database structures, SQL and Java are required.

1.1 Display Conventions

This documentation uses a variety of methods to present different types of information.

Purpose	Presentation	Example
User interface elements	Bold	Click **OK**.
Menu sequence	>	**File > Exit**
Special emphasis	Bold	**Do not delete** Name.
Cross-reference text	Italics	You will find more information in the topic*Network*.
Output	Monospace font, e.g., Courier	`Command not found.`
Input	Monospace font, e.g., Courier	`Enter` LOCAL as the file name.
Key combination	Monospace font, e.g., Courier	`<Ctrl>+<Alt>+<Esc>`
Work Steps and Substeps	Numbered and alphabetical lists	• Configure the DSL telephony stations with the associated DID phone numbers. – Click **Add**. – Enter the name of the Internet telephony station under **Internet Telephony Station**.
Alternative Work Steps	Enumeration	• If you want to output amounts, enable the check box **Display amounts instead of units**. • If you want to output units, clear the check box **Display amounts instead of units**.

1.2 Overview of the Documentation

The OpenScape Office Contact Center documentation is intended for different target groups.

The following documentation is available:

- OpenScape Office V3, myReports, Reference Manual
  This document describes how to create reports using existing report templates and how to define customized report templates.
- OpenScape Office V3, myAgent, User Guide
  This document describes the installation, configuration and operation of the integrated application myAgent and is intended for the user.
- OpenScape Office V3, myReports, User Guide
  This document describes the installation, configuration and operation of the integrated application myReports and is intended for the user.
- OpenScape Office V3, Administrator documentation
  This document provides a complete description of the hardware, installation, configuration, operation, features and administration and is intended for administrators.

2 Introduction to Reporting

Reports are used to determine the current status of the OpenScape Office Contact Center and to analyze the strengths and weaknesses of its associated components. This makes it possible to optimize the call center configuration, for example, and to thus use the call center resources more efficiently.

About 100 predefined report templates for creating reports are available to users via the myReports application. These templates are classified by subject area and assigned to the following report groups:

- Agent activity
- Agents
- Call history
- Calls
- Other
- Performance
- Queues
- User Presence Status
- Wrap-up Codes

If needed, the predefined report templates can be adapted to individual customer requirements via the integrated Report Designer and incorporated as new report templates in the Report Manager. Furthermore, additional sets of report groups can also be created and incorporated in the Report Manager.

The Report Manager is used to manage all report templates. The report templates can be optionally sorted by ID, report name and report group.

2.1 Overview of the Predefined Report Templates

About 100 predefined report templates sorted by subject area (report groups) are available to users for creating reports via the myReports application.

The following tables list the predefined report templates as a function of the different report groups.

Detailed descriptions of all predefined report templates can be found in the section *Predefined Report Templates in Detail*.

Report Group Agent Activity

The focus of these reports is set to agent activities. Other system users are not considered within these reports.

Report template	Description	Output	
		Values	Format
Agent Activity Logged Times	The report represents the login, logout and logged in times in the specified date interval for the selected agent.	• Login time • Logout time • Logged In time • Daily total Logged in time • Total Logged in time	Table
Agent Activity Missed Call Times	The report displays missed call times in specified date interval for selected agent.	• Start time • End time • Daily total Missed Call Time • Total Missed Call Time	Table
Agent Activity On Break Times	The report displays break times in the specified date interval for the selected agent.	• Start time • End time • Break Name • Default Break Interval (min.) • Actual Break Time • Daily Total Break Time • Total Break Time	Table
Agent Activity Status (All Agents) – Daily	The report displays daily status details (logged, on break, work, missed call) by agents for one specified day.	• Start time • End time • Status Name • Status Duration	Tables grouped by agents
Agent Activity Status (By Agent) - Daily	The report displays status details for selected agent and specified day.	• Start time • End time • User Status • Duration	Table & Graphic
Agent Activity Work Times	The report displays work times in specified date interval for selected agent.	• Start time • End time • Work Time • Daily total Work Time • Total Work Time	Table

Report Group Agents

This group of reports is contact center related. The included reports refer to agent issues in general. Other system users are not considered within these reports.

Report template	Description	Output	
		Values	Format
Agent G.O.S.	The report shows hourly average grade of service for specified agent in a specified date range. ***INFO***: The report template **Agent G.O.S. – Daily** has a different graphic for each day.	• N/A	Graphic
Agent G.O.S. (Daily)	The report shows hourly average grade of service for specified agent in a specified date range.	• N/A	Graphic (there is a different graphic for each day)
Agent Private Calls (All Agents)	The report shows details about the agent private calls in the specified date range.	• Agent • Agent extension • Department • Number of calls • Talk Time • Percentage of total talk time	Table
Agent Private Calls (Per Agent)	The report shows detailed information about the agent private calls for the specified agent in the specified date range.	• Call Date • Start time • Calling Number • Called Number • Direction I/O (Inbound / Outbound) • Talk Time • Daily Total Number of Calls • Daily total talk time • Total number of calls • Total talk time	Table

Report template	Description	Output	
		Values	Format
All User Calls (By Agent)	The report shows detailed information about all user calls for the specified agent in the specified date range.	• Call Date • Start time • End time • Calling Number • Called number • I/C – Incoming call (yes or no) • O/G – Outgoing call (yes or no) • Int – Internal call (yes or no) • Talk time • Daily totals for: number of I/C, number of O/G, number of internal calls, talk time • Grand totals for: number of I/C, number of O/G, number of internal calls, talk time	Table
All User Calls (By Agent) 2	The report shows detailed information about all user calls for the specified agent in the specified date range. ***INFO***: For the report template **All User Calls (By Agent)**, the report parameter **Business hours only** can be additionally selected.	• Call Date • Start time • End time • Calling Number • Called number • I/C – Incoming call (yes or no) • O/G – Outgoing call (yes or no) • Int – Internal call (yes or no) • Talk time • Daily totals for: number of I/C, number of O/G, number of internal calls, talk time • Grand totals for: number of I/C, number of O/G, number of internal calls, talk time	Table

Report Group Call History

These reports contain system-wide information and not only contact center related information.

Report template	Description	Output	
		Values	Format
External Calls Per User	The report shows information about the user external calls for the specified user in the specified date range.	• Call Date • Start time • End time • CLI (Calling or called number) • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Incoming Calls (Free-Calls) - Per User	The report shows incoming free calls details for the specified user in the specified date range.	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Incoming Calls (International) – Per User	The report shows incoming - international calls details for the specified user in the specified date range.	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table

Report template	Description	Output	
		Values	Format
Incoming Calls (Mobile/Cell) – Per User	The report shows incoming - mobile/cell calls details for the specified user in the specified date range.	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Incoming Calls (Other External Calls) – Per User	The report shows incoming other calls details for the specified user in the specified date range. Other calls means not international, free, pay, mobile/cell and specific calls.	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Incoming Calls (Specific Calls) – Per User	The report shows incoming specific calls details for the specified user in the specified date range. Incoming specific calls means incoming calls filtered by specific call number prefix.	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Incoming Calls Per User	The report shows information about the incoming calls for the specified user in the specified date range.	• Calling Number • Date of call • Start time • End time • Length of call • Daily total length of calls per calling number • Daily total number of calls per calling number • Total length of calls • Total number of calls	Table

Report template	Description	Output	
		Values	Format
Incoming Calls Report – Group	The report shows information about all incoming calls grouped by departments.	• Department • Users • Extension • Total number of calls per user • Total ring time per user • Total talk time per user • Total number of calls, ring time and talk time per department	Table
Incoming Calls Report – Group Summary	The report shows summary information about the incoming calls per departments.	• Department • Total number of calls per department • Total ring time per department • Total talk time per department • Total number of calls, total ring time and total talk time (all departments)	Table
Incoming Calls Report – User	The report shows information about the incoming calls for the specified user in the specified date range.	• Start time • CLI – Calling number • Ring Time • Talk Time • Daily total number of calls • Daily total ring time • Daily total talk time • Total number of calls • Total talk time	Table
Incoming Calls Report – User Summary	The report shows summary information about the incoming calls per users.	• User First name • User Surname • User Extension • Total number of calls per user • Total ring time per user • Total talk time per user • Total number of calls, total ring time and total talk time (all users)	Table

Report template	Description	Output	
		Values	Format
Internal Calls Per User	The report shows information about the internal calls for the specified user in the specified date range.	• Call Date • Start time • End time • CLI (Calling or called number) • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Missed Calls (Incoming) Per User	The report shows incoming - missed calls details for the specified user in the specified date range.	• Start time • End time • Missed Call Time • Daily total missed call time • Total missed call time • Total number of missed calls	Table
Missed Calls (Incoming) Per User 2	The report shows incoming - missed calls details for the specified user in the specified date range (including calling number details).	• Start time • End time • Calling Number • Missed Call Time • Daily total missed call time • Total missed call time • Total number of missed calls	Table
Missed Calls (Outgoing) Per User	The report shows outgoing - missed calls details for the specified user in the specified date range.	• Start time • End time • Missed Call Time • Daily total missed call time • Total missed call time • Total number of missed calls	Table

Report template	Description	Output	
		Values	Format
Missed Calls (Outgoing) Per User 2	The report shows outgoing missed calls details for the specified user in the specified date range (including called number details).	• Start time • End time • Missed Call Time • Called Number • Daily total missed call time • Total missed call time • Total number of missed calls	Table
Outgoing Calls (Free Calls) – Per User	The report shows outgoing - free calls details for the specified user in the specified date range.	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Outgoing Calls (International) – Per User	The report shows outgoing - international calls details for the specified user in the specified date range.	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Outgoing Calls (Mobile/Cell) – Per User	The report shows outgoing - mobile/cell details for the specified user in the specified date range.	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table

Report template	Description	Output	
		Values	Format
Outgoing Calls (Other External Calls) – Per User	The report shows outgoing - other calls details for the specified user in the specified date range. Other calls means not international, free, pay, mobile/cell and specific calls.	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Outgoing Calls (Pay Calls) – Per User	The report shows outgoing - pay calls details for the specified user in the specified date range.	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Outgoing Calls (Specific Calls) – Per User	The report shows outgoing - specific calls details for the specified user in the specified date range. Outgoing specific calls means outgoing calls filtered by specific call number prefix.	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls	Table
Outgoing Calls Per User	The report shows information about the outgoing calls for the specified user in the specified date range.	• Called Number • Date of call • Start time • End time • Length of call • Daily total length of calls per called number • Daily total number of calls per called number • Total length of calls • Total number of calls	Table

Report template	Description	Output	
		Values	Format
Outgoing Calls Report – Group	The report shows information about all outgoing calls grouped by departments.	• Department • Users • Extension • Total number of calls per user • Total ring time per user • Total talk time per user • Total number of calls, ring time and talk time per department	Table
Outgoing Calls Report – Group Summary	The report shows summary information about the outgoing calls per departments.	• Department • Total number of calls per department • Total ring time per department • Total talk time per department • Total number of calls, total ring time and total talk time (all departments)	Table
Outgoing Calls Report - User	The report shows information about the outgoing calls for the specified user in the specified date range.	• Start time • CLI – Called number • Ring Time • Talk Time • Daily total number of calls • Daily total ring time • Daily total talk time • Total number of calls • Total talk time	Table
Outgoing Calls Report - User Summary	The report shows summary information about the outgoing calls per users.	• User First name • User Surname • User Extension • Total number of calls per user • Total ring time per user • Total talk time per user • Total number of calls, total ring time and total talk time (all users)	Table

Report Group Calls

This group of reports is contact center related. The included reports refer mainly to the contact center calls within the queues or those handled by agents. In addition cumulated reports about contact center performance are included.

Report template	Description	Output	
		Values	Format
Abandoned Calls Statistics	The report represents details about the abandoned calls by queues.	• Queue • Count • Percentage of abandoned calls • Max Queue Time • Abandoned percentage of queue – number of abandoned calls per queue and percentage of all abandoned calls for that queue (per queue time: 0-30 s, 31-60 s, 61-90 s, 91-120 s, 121-300 s, 300+ s) • Totals for the columns: count, max. queue time and number of calls for all columns showing abandoned calls per queue time interval • Average totals in percents for all columns showing Abandoned Calls per Queue Time interval Queue Time means the amount of time a caller has been waiting to get connected to an agent.	Table
Abandoned Calls Statistics – Details	The report represents details about the abandoned calls.	• Call ID • Call arrived time • Queue • Queue time • Pickup time • CLI (Calling number) • Customer Company • Average pickup time • Average queue time	Table

Report template	Description	Output	
		Values	Format
Answered Calls Alert Times	The report represents the alert times of answered calls for a specific agent in the specified date range.	• Day • Alert time (call pickup time) - daily • Percentage of total alert time - daily • Total alert time	Table and Graphic
Answered Calls Alert Times (All Agents)	The report represents the alert times of answered calls for all agents in the selected date range.	• Agent • Alert time (call pickup time) - by agent • Percentage of total alert time - by agent • Total alert time	Table and Graphic
Answered Calls Alert Times – Details	The report represents the alert times of answered calls for a specific agent in the specified date range.	• Time of call • End of call waiting – time when the call is answered • Alert time (call pickup time) • Daily total alert time • Total alert time	Table

Report template	Description	Output	
		Values	Format
Answered Calls Statistics	The report represents details about the answered calls by queues.	• Queue • Count • Percentage of answered calls • Max Queue Time • Answered percentage of queue – number of answered calls per queue and percentage of all answered calls for that queue (per queue time : 0-30 s, 31-60 s, 61-90 s, 91-120 s, 121-300 s, 300+ s) • Totals for the columns: count, max. queue time and number of calls for all columns showing answered calls per queue time interval • Average totals in percents for all columns showing Answered Calls per Queue Time interval Queue Time means the amount of time a caller has been waiting to get connected to an agent.	Table
Answered Calls Wrap-up Information	The report displays details including wrap-up information for answered calls in the specified date range.	• Call ID • Arrived At • Queue • Agent login • Wrap up • CLI – calling number • Daily total number of calls	Table

Report template	Description	Output	
		Values	Format
Call Traffic All Agents – Per Hour Daily	Count of calls (call center calls, direct calls, outbound, inbound) and talk time by agents (for all available agents having calls in the specified date range). ***INFO***: The report template **Call Traffic All Agents-Per Hour Daily-Details** returns the following additional output values: `Talk time CC Calls`, `Talk time Direct Calls`	• Day • User/Agent • Time – hourly interval (e.g.: 09:00-10:00) • CC calls (number of contact center calls) • Direct calls (number of direct calls) • Inbound calls (number of inbound/incoming calls) • Outbound calls (number of outbound/ outgoing calls) • All calls (number of all calls = cc calls + direct calls) • Talk time - all calls (Total Talk time for the specified hourly interval) • Daily Totals and Grand Totals (CC calls, direct calls, inbound, outbound, all calls, talk time)	Table

Report template	Description	Output	
		Values	Format
Call Traffic All Agents – Per Hour Daily – Details	Count of calls (call center calls, direct calls, outbound, inbound) and talk time (for cc calls, direct calls and all calls) by agents, for all available agents having calls in the specified date range.	• Day • User/Agent • Time – hourly interval (e.g.: 09:00-10:00) • CC calls (number of contact center calls) • Talk time CC calls (CC calls: total talk time for the specified hourly interval) • Direct calls (number of direct calls) • Talk time direct calls (direct calls: total talk time for the specified hourly interval) • Inbound calls (number of inbound/incoming calls) • Outbound calls (number of outbound/ outgoing calls) • All calls (number of all calls = cc calls + direct calls) • Talk time - all calls (total talk time for the specified hourly interval) • Daily totals and grand totals (CC calls, Direct calls, Inbound, Outbound, All calls and talk time of CC calls, direct calls and all calls)	Table

Report template	Description	Output	
		Values	Format
Call Traffic All Queues – Per Hour (Daily)	Count of calls (all calls, answered calls and abandoned calls) for all available queues having calls in the specified date range.	• Day • Queue • Time – hourly interval (e.g.: 09:00-10:00) • All calls (number of calls per hour daily) • Answered calls (number of answered calls ...) • Abandoned calls (number of abandoned calls ...) • Daily Totals per queue and Grand Totals (all calls, answered calls and abandoned calls)	Table
Call Traffic All Queues – Queue Time, GOS Per Hour Daily	Number of calls, maximum queue time, minimum queue time and grade of service for all available queues – having calls in the specified date range.	• Day • Queue • Time – hourly interval (e.g.: 09:00-10:00) • All calls (number of calls per hour daily) • Max Queue Time (Maximum queue time in seconds...) • Min Queue Time (Minimum queue time in seconds...) • GOS (Grade of service...) • Daily totals per queue (number of calls, average maximum queue time, average minimum queue time, average grade of service) • Grand totals all queues (number of calls, max queue time, min queue time, average GOS)	Table

Report template	Description	Output	
		Values	Format
Call Traffic By Queue – Per Hour Daily – Details	Count of calls (all calls, answered calls and abandoned calls) for the selected queue and the specified date range.	• Day • Time – hourly interval (e.g.: 09:00-10:00) • All calls (number of calls per hour daily) • Answered calls (number of answered calls ...) • Abandoned calls (number of abandoned calls ...)	Table
Call Traffic One Agent – Per Hour Daily	Count of calls (call center calls, direct calls, outbound, inbound) and talk time for selected agent and the specified date range. ***INFO***: The report template **Call Traffic One Agent – Per Hour Daily – Details** returns the following additional output values: `Talk Time CC Calls`, `Talk Time Direct Calls`	• Specified agent details (first name, surname, email and department) • Time – hourly interval (e.g.: 09:00-10:00) • CC calls (number of contact center calls) • Direct calls (number of direct calls) • Inbound calls (number of inbound/incoming calls) • Outbound calls (number of outbound/ outgoing calls) • All calls (number of all calls = cc calls + direct calls) • Talk times (total talk time for the specified hourly interval) • Daily totals and grand totals (CC calls, direct calls, inbound, outbound, all calls, talk time)	Table

Report template	Description	Output	
		Values	Format
Call Traffic One Agent – Per Hour Daily – Details	Count of calls (all calls, call center calls, direct calls, outbound, inbound) and talk time (cc calls talk time, direct calls talk time and total talk time-for all calls) for selected agent and the specified date range.	• Specified agent details (first name, surname, email and department) • Time – hourly interval (e.g.: 09:00-10:00) • CC calls (number of contact center calls) • Talk time CC calls (CC calls: total talk time for the specified hourly interval) • Direct calls (number of direct calls) • Talk time direct calls (direct calls: total talk time for the specified hourly interval) • Inbound calls (number of inbound/incoming calls) • Outbound calls (number of outbound/ outgoing calls) • All calls (number of all calls = cc calls + direct calls) • Talk time - all calls (total talk time for the specified hourly interval) • Daily totals and grand totals (CC calls, Direct calls, Inbound, Outbound, All calls and talk time of CC calls, direct calls and all calls)	Table

Report template	Description	Output	
		Values	Format
Call Traffic One Queue – Queue Time, GOS Per Hour Daily	Number of calls, maximum queue time, minimum queue time and grade of service for the selected queue and the specified date range.	• Day • Time – hourly interval (e.g.: 09:00-10:00) • All calls (number of calls per hour daily) • Max Queue Time (Maximum queue time in seconds...) • Min Queue Time (Minimum queue time in seconds...) • GOS (Grade of service...) • Daily totals (number of calls, average maximum queue time, average minimum queue time, average grade of service) • Grand totals (number of calls, max queue time, min queue time, average GOS)	Table
Callback Calls	The report displays callback details for all calls in the specified date/time range.	• Call date • Queue Name • Time of call • Call ID • CLI – calling number • Agent • Callback Number • Daily total number of callback calls by queue • Daily total number of callback calls (all queues) • Total number of callback calls	Table

Report template	Description	Output	
		Values	Format
Calls List Agent	Call list for selected agent in the specified date/time range	• Start time • End time • Queue Name • Queue time • Talk time • CLI – Calling number • Grade of service • Total number of calls • Total Queue Time • Total talk time Queue Time means the amount of time a caller has been waiting to get connected to an agent.	Table
Calls List Queue	Call list for selected queue in the specified date range	• Start time • End time • Agent • Queue Time • Talk time • CLI – Calling number • Grade of service • Total number of calls • Total Queue Time • Total talk time Queue Time means the amount of time a caller has been waiting to get connected to an agent.	Table
Contact Center (Per Agents) – Chart	Number of calls (Total number of calls, answered and missed calls) by agents for the specified date range.	• Total number of calls • Total number of Answered Calls • Total number of Missed Calls	Graphics & Grids

Report template	Description	Output	
		Values	Format
Contact Center (Per Agents) – List	Number of calls (Total number of calls, answered and missed calls), percents of calls, average queue time and talk time by agents for the specified date range.	• Agent • Number of calls by agent (All calls) (Nc) • Percentage of total number of calls by agent • Number of answered calls by agent (Na) • Percentage of total number of answered calls • Number of missed calls by agent (Nm) • Percentage of total number of missed calls • Average queue time in seconds by agent • Average talk time in seconds by agent • Total number of calls (Ntc) • Total number of Answered Calls (Nta) • Total number of Missed Calls (Ntm) • Total average queue time in seconds (all agents) • Total average talk time in seconds (all agents)	Table
Contact Center (Per Queues) – Chart	Number of calls (Total number of calls, answered and missed calls) by queues for the specified date range.	• Total number of calls • Total number of Answered Calls • Total number of Missed Calls	Graphics & Grids

Report template	Description	Output	
		Values	Format
Contact Center (Per Queues) – List	Number of calls (total number of calls, answered and missed calls), percents of calls, average queue time and talk time by queues for the specified date range.	• Queue • Number of calls by queue (All calls) (Nc) • Percentage of total number of calls by queue • Number of answered calls by queue (Na) • Percentage of total number of answered calls • Number of missed calls by queue (Nm) • Percentage of total number of missed calls • Average queue time in seconds by queue • Average talk time in seconds by queue • Total number of calls (Ntc) • Total number of Answered Calls (Nta) • Total number of Missed Calls (Ntm) • Total average queue time in seconds (all queues) • Total average talk time in seconds (all queues)	Table

Report template	Description	Output	
		Values	Format
Contact Center Calls	The report displays call details (missed, answered and abandoned calls) in the selected date/time range.	• Missed Calls – Call ID – Arrived At – Agent – Queue – Missed Call Time (s) – CLI – calling number – Total number of missed calls – Average missed call time (s) • Abandoned Calls – Call ID – Arrived At – Queue – Queue Time (s) – CLI – calling number – Average queue time (s) – Total number of abandoned calls • Answered Calls – Call ID – Arrived At – Queue – Agent – Queue Time (s) – Talk Time (s) – Pickup Time (s) – CLI – calling number – Average queue time (s) – Average talk time (s) – Average pickup time (s) – Total number of answered calls	Table

Report template	Description	Output	
		Values	Format
Contact Center Summary	Number of calls, average queue time, talk time and pickup time by queues for the specified date/time range.	• Queue • Total number of calls per queue • Average pickup time (s) • Average talk time (s) • Average queue time (s) • Total number of calls • Total average pickup time, queue time and talk time	Table
Contact Center Summary 2	Number of calls, average queue time, talk time and pickup time, number of callback calls and queue time by queues for the specified date range.	• Queue • Total number of calls per queue • Average pickup time (s) • Average talk time (s) • Average queue time (s) • Callback calls per queue • Queue Time • Total number of calls • Total average pickup time, queue time and talk time • Total number of callback calls • Total Queue Time	Table
Contact Center Summary – Answered Calls	Number of calls, average queue time, talk time and pickup time, number of callback calls and queue time of answered calls by queues for the specified date range.	• Queue • Total number of calls per queue • Average pickup time (s) • Average talk time (s) • Average queue time (s) • Callback calls per queue • Queue Time • Total number of calls • Total average pickup time, queue time and talk time • Total number of callback calls • Total Queue Time	Table

Report template	Description	Output	
		Values	Format
Contact Center Summary – Details	Number of calls (total number of calls, answered and abandoned calls), average queue time, talk time and pickup time by queues for the specified date/time range.	• Queue • Total number of calls per queue • Average pickup time (s) • Average talk time (s) • Average queue time (s) • Callback calls • Answered Calls • Abandoned Call • Total number of calls • Total average pickup time, queue time and talk time • Total number of answered calls • Total number of abandoned calls	Table
Missed Calls Report	The report displays details for missed calls in the specified date/time range.	• Call date • Queue Name • Time of call • Call ID • CLI – calling number • Customer name • Customer company • Daily total number of missed calls by queue • Daily total number of missed calls (all queues) • Total number of missed calls	Table
Missed Calls Summary (Per Agent)	The report displays missed calls summary details (number of calls and percent of all missed calls) per agent for calls in the specified date/time range.	• Agent • Number of missed calls (per agent) • Percentage of all missed calls • Total number of missed calls (all agents)	Table

Report template	Description	Output	
		Values	Format
Missed Calls Summary (Per Queue)	The report displays missed calls summary details (number of calls and percent of all missed calls) per agent for calls in the specified date/time range; these details are grouped per queue.	• Queue • Agent • Number of missed calls (per agent) • Percentage of all missed calls • Total number of missed calls per queue • Grand total of missed calls (all queues)	Table

Report Group - Other

These reports contain system-wide information and not only contact center related information.

Report template	Description	Output	
		Values	Format
Calls History Per User	The report displays call history information for the specified user in the selected date range.	• Call Date • Start time • Calling Number • Called Number • Direction I/O (Inbound / Outbound) • Talk Time • Daily Total Number of Calls • Daily total talk time • Total number of calls • Total talk time	Table
Default Break Information	The report displays the default break information (Break name and default break interval in minutes)	• N/A	Graphic
External Directory User Details	The report displays information about the user external directory (User company, first name, surname, business phone 1, business phone 2, home phone and mobile phone)	• User Company • First name • Surname • Business phone 1 • Business phone 2 • Home phone • Mobile Phone	Table
Fax Journal – Received Faxes (By User)	The report shows details of the received faxes for a specified user in the selected date range.	• Time • Fax Group • Contact (last name, first name) • Company • CLI – fax calling number • Fax Status • Fax Pages • Total number of daily received faxes and fax pages • Total number of received faxes • Total number of received fax pages	Table

Report template	Description	Output	
		Values	Format
Fax Journal – Sent Faxes (By User)	The report shows details of the sent faxes for a specified user in the selected date range.	• Time • Fax Group • Contact (last name, first name) • Company • Destination • Status • Pages • Total number of daily sent faxes and fax pages • Total number of sent faxes • Total number of sent fax pages	Table
Fax Transmission Report	The report shows fax details including the fax itself.	• From date • Until (to date) • Business hours only (else 24/24) • Daily report	Text (fax details) + Fax (embedded picture)
Incoming Calls Report – Hourly	The report shows the number of incoming calls per hour and daily.	• From date • Until (to date) • Daily report	Table
Incoming Calls Report – Hourly Per Weekday	The report shows the number of incoming calls per hour and weekday.	• From date • Until (to date) • Daily report	Table

Report template	Description	Output	
		Values	Format
Internal Directory User Details	The report shows information about the user internal directory.	• Users • Email • External 1 • External 2 • Mobile phone • Home phone • Fax	Table
Voicemail Center (All Users)	The report shows voicemail details in the specified date range.	• Users • Call start time • Office Status (Office, Meeting, Sick, Break, Gone out, Holiday, Lunch, Home, DND) • Calling Number • Priority (Normal, Urgent, Private) • Duration • Total number of daily voicemail messages • Total number of voicemail messages	Table
Voicemail Center (By User)	The report shows voicemail details for the selected user in the specified date range.	• Call Start Time • Office Status (Office, Meeting, Sick, Break, Gone out, Holiday, Lunch, Home, DND) • Calling Number • Priority (Normal, Urgent, Private) • Duration • Total number of daily voicemail messages • Total number of voicemail messages	Table

Report Group - Performance

This group of reports contains contact center related reports. The focus of the reports is set to hourly performance values.

Report template	Description	Output	
		Values	Format
Abandoned Calls Per Hour	Hourly representation of all abandoned calls in the specified date range	• Number of abandoned calls • Percentage of all abandoned calls • Total number of abandoned calls	Table and Graphic
Agent Calls Percentage	The report displays information about the percentage and number of calls received by agents.	• Agent • Number of calls by agent • Percentage of total number of calls (all agents) • Percentage of total talk time (all agents) • Total number of calls for all agents	Table and Graphic
Agent Performance Details	The report shows agent performance details for the specified agent in a specified date/time range	• Queue Name • Start time • Pickup Time • Talk Time • Grade of Service • Daily Total Number of Calls, Pickup Time, Talk Time per queue • Daily Average Grade of Service per queue • Total number of calls • Total Average Pickup Time, Talk Time and Grade of Service	Table
Answered Calls Per Hour	Hourly representation of all answered calls for specified date range.	• Number of answered calls • Percentage of all answered calls • Total number of calls	Table and Graphic

Report template	Description	Output	
		Values	Format
Call Traffic By Queue Per Hour	Hourly representation of the number of calls for specified queue in the selected date range. ***INFO***: The report template **Call Traffic By Queue Per Hour (Daily)** has a different table and graphic for each day.	• Time • Number of calls • Total number of calls • Percentage of total number of calls	Table and Graphic
Call Traffic By Queue Per Hour (Daily)	Hourly representation of the number of calls for specified queue in the selected date range.	• Time • Number of calls • Total number of calls • Percentage of total number of calls	Table and Graphic (there is a different table and graphic for each day)
Contact Center Traffic Per Hour	Hourly representation of the number of calls in the selected date range. ***INFO***: The report template **Contact Center Traffic Per Hour (Daily)** has a different table and graphic for each day.	• Time of call • Number of calls • Total number of calls • Percentage of total number of calls	Table and Graphic
Contact Center Traffic Per Hour (Daily)	Hourly representation of the number of calls in the selected date range.	• Time • Number of calls • Total number of calls • Percentage of total number of calls	Table and Graphic (there is a different table and graphic for each day)
Missed Calls Per Hour	Hourly representation of all missed calls in the specified date range.	• Number of missed calls • Percentage of all missed calls • Total number of missed calls	Table and graphic

Report template	Description	Output	
		Values	Format
Summary of Details per Agent	The report contains a summary of the details (duration of agent activities, percentage of work, break and absence times during the logged in time, percentage of logged in time during business hours, calls, talk times) for a specific agent in the specified date range.	• Duration of agent activities: logged in, work, break and absence times • Percentage of work, break and absence times during the logged in time • Percentage of logged in time during business hours • (***) All calls, outgoing calls, incoming calls, direct calls, CC Calls (Contact Center calls), CC callback calls, answered CC calls, answered CC overflow calls, answered CC callback calls, missed CC calls, missed CC overflow calls, missed CC callback calls • Internal calls, external calls and calls during business hours for all (***) columns • Total talk time, average talk time and average number of calls per hour during business hours for all (***) columns, except for missed calls	Table

Report template	Description	Output	
		Values	Format
Summary of Details per Queue	This report contains a summary of the details (calls, call- and wait times, details for answered and abandoned calls, percentage of total number of all answered and abandoned calls) for a specific queue in the specified date range.	• All calls, internal calls, external calls, callback calls, answered calls, calls answered during business hours, calls answered outside business hours, calls answered by primary agents / overflow agents, abandoned calls, etc. • Number of calls, total talk time, average talk time, average queue time, max. queue time for all of the above columns • Details for answered and abandoned calls with respect to queue time: up to 30 s, less than 3 s, between 3 and 20 s, between 20 and 30 s • Percentage of all answered calls for internal calls, external calls and callback calls during business hours and outside business hours • Percentage of all abandoned calls for internal calls and external calls and with respect to the queue time • Other values: date, business hours, number of calls and Grade of Service (GOS) during business hours	Table

Report Group - Queues

This group of reports is contact center related. Reports are focused to calls and agents assigned to queues.

Report template	Description	Output	
		Values	Format
Agent Calls Queue Specific	The report displays information about the percentage and number of calls received by agents for selected queue in specified date range.	• Agent • Percentage of calls received by agents (by Queue) • Number of calls by agent • Percentage of total number of calls (all agents) • Percentage of total talk time (all agents) • Total number of calls for all agents	Table and Graphic
Agent Properties	The report displays the agent properties for all available agents.	• Agent • Queue • Agent type (primary or overflow) • Callback Calls (Yes or No) • Start Calls Overflow • Start Seconds Overflow - Seconds of call in queue before it is delivered to Overflow Agent • Work Time - in seconds • Grand Totals for Start Call Overflow, Start Seconds Overflow and Work Time	Table
Agent Queue Load	The report shows queue load information for the specified agent in the specified date range.	• Queue • Number Of Calls (by queues) • Percentage of total number of calls • Total number of calls	Table and graphic

Report template	Description	Output	
		Values	Format
Avg. G.O.S Per Queue	Hourly representation of the average grade of service for specified queue in the selected date range. ***Info***: The Report template **Avg. G.O.S. Per Queue (Daily)** has a different graphic for each day.	• N/A	Graphic
Avg. G.O.S. Per Queue (Daily)	Hourly representation of the average grade of service for specified queue in the selected date range.	• N/A	Graphic (there is a different graphic for each day)
Missed Calls Per Queue	Missed calls grouped by queues for call in the specified date range.	• Queue Name • Number of missed calls (per Queue) • Percentage of total number of missed calls • Total number of missed calls	Table and Graphic (Pie Chart)
Queue Summary Details	Queue summary details for selected queue and specified date range.	• Answered Calls • Abandoned Calls • Other calls • Maximum Queue time for answered calls • Minimum Queue time for answered calls • Average Queue time for answered calls • Maximum Queue time for abandoned calls • Minimum Queue time for abandoned calls • Average Queue time for abandoned calls • Maximum Talk time for answered calls • Minimum Talk time for answered calls • Average Talk time for answered calls • Total number of calls for all agents • Average grade of service for selected queue	Grid and Graphic

Report template	Description	Output	
		Values	Format
Queue Traffic Comparison	Queue traffic comparison by numbers of calls for selected date/time range.	• Queue Name • Number of calls (per Queue) • Percentage of total number of calls • Total number of calls	Table and Graphic (Pie Chart)

Report Group - User Presence Status

This group of reports is contact center related. The focus of the reports is set to the agent status.

Report template	Description	Output	
		Values	Format
User Presence Status (All Users) – Daily	The report shows the user presence status details for the “daily” office statuses : Meeting, Break, Lunch, Gone out, DND. The report data is selected for a specified day (date) and grouped by users.	• Start time • End time • Status (Meeting, Break, Lunch, Gone out, DND - Do Not Disturb) • Duration	Table
User Presence Status (All Users)	The report shows the user presence status details for the two “longest” statuses : Sick and Holiday. The duration of these statuses in most of the cases will be in days unlike the duration of the others office statuses (meeting, break, gone out, lunch and DND) usually measured in minutes and hours.	• Users • Start time • End time • Status (Sick or Holiday) • Duration • Total duration time for all users	Table
User Presence Status (By User) – Daily	The report shows the user presence status details for the “daily” office statuses : Meeting, Break, Lunch, Gone out, DND. The report data is selected for a specified user and the specified day (date)	• Start time • End time • Status (Meeting, Break, Lunch, Gone out, DND - Do Not Disturb) • Duration	Table and Graphic
User Presence Status (By User)	The report shows the user presence status details for the two “longest” statuses : Sick and Holiday. The report data is selected for a specified user in the specified date range. The duration of these statuses in most of the cases will be in days unlike the duration of the others office statuses (Meeting, Break, Gone Out, Lunch and DND) usually measured in minutes and hours.	• Start time • End time • Status (Sick or Holiday) • Duration • Total duration time for the selected user	Table

Report Group - Wrap-up Codes

The focus of these reports is set to display Wrap-up Code Usage for all queues, per queue, per wrap-up and per wrap-up group.

Report template	Description	Output	
		Values	Format
Wrap-up Code Usage All Queues	The report shows wrap-up code usage details in the specified date/time range.	• Wrap-up description • Count (Number of Calls) - per wrap-up • Percentage of total number of calls • Average Talk Time -per wrap-up • Talk Time – per wrap-up • Average Queue Time – per wrap-up • Totals for all previous values (Average Talk Time, Talk Time, Average Queue Time) Queue Time means the amount of time a caller has been waiting to get connected to an agent.	Table and Graphic (Pie Chart)
Wrap-up Code Usage Per Group	The report shows wrap-up details for the selected wrap-up group in the specified date range.	• Wrap-up description • Count (Number of Calls) - per wrap-up • Percentage of total number of calls • Average Talk Time -per wrap-up • Talk Time – per wrap-up • Average Queue Time – per wrap-up • Totals for all previous values (Average Talk Time, Talk Time, Average Queue Time) Queue Time means the amount of time a caller has been waiting to get connected to an agent.	Table and Graphic (Pie Chart)

Report template	Description	Output	
		Values	Format
Wrap-up Code Usage Per Queue	The report shows wrap-up details for the specified queue.	• Wrap-up description • Count (Number of Calls) - per wrap-up • Percentage of total number of calls • Average Talk Time -per wrap-up • Talk Time – per wrap-up • Average Queue Time – per wrap-up • Totals for all previous values (Average Talk Time, Talk Time, Average Queue Time) Queue Time means the amount of time a caller has been waiting to get connected to an agent.	Table and Graphic (Pie Chart)
Wrap-up Code Usage Per Wrap-up	The report shows wrap-up details for the specified wrap-up.	• Queue Name • Count (number of calls) - per queue • Percentage of total number of calls • Average Talk Time -per queue • Talk time- per queue • Average Queue Time – per queue • Totals for all previous values	Table and Graphic (Pie Chart)

2.2 Report Designer

If needed, the predefined report templates can be adapted to individual customer requirements via the integrated Report Designer and incorporated as new report templates in the Report Manager. Furthermore, additional sets of report groups can also be created and incorporated in the Report Manager.

The Report Designer is a separately started Open Source application (called the BIRT RCP Designer) for the professional creation of report templates. BIRT is an acronym for Business Intelligence and Reporting Tools.

INFO: The Report Designer can only be called by a myReports administrator.

Details on the different user roles and on the associated authorization rights to use the functions of myReports can be found in the section on *myReports User Roles*.

---

myReports supports the BIRT RCP Designer through

- the predefined database connection,
- the integration of report templates used in myReports.
- a data transfer program for integrating newly created report templates in the Report Manager.

In order to use the Report Designer successfully, familiarity with the application itself as well as database structures, SQL and Java are required.

Information on using the Report Designer can be found in the associated online help.

2.3 Report Parameters

Before creating a report, report parameters must be set to determine the period (date, time) and entities (agents, queues, etc.) should be measured.

The following report parameters are available:

- **From date**
  Date for the start of the reporting period
- **To Date**
  Date for the end of the reporting period
- **From Time**
  Time for the start of the reporting period
- **To Time**
  Time for the end of the reporting period
- **Queue Name**
  Relevant queue for the report
- **Agent**
  Relevant agent for the report

- **Users**
  Relevant internal subscribers for the report
- **Wrap up Description**
  Relevant wrapup code for the report
- **Wrapup Group**
  Relevant wrapup group for the report
- **Business Hours Only**
  This option takes only the business hours configured in OpenScape Office into account.
- **Daily report**
  This option arranges the results of report on a daily basis.

2.4 Output Values

Calculated results are presented in reports via various output values (totals, times, percentage values, etc.).

Explanation of the different output values:

- `Pickup time`
  The time the agent takes to answer a call, i.e., how long the agent phone is ringing before he or she answers.
- `Queue time`
  The amount of time a caller has been waiting to get connected to an agent.
- `Abandoned calls`
  Is the case where a caller hangs up after x seconds. The x is the Abandoned Calls Threshold setting (in seconds) in the Queue Parameters.
  Example: If the setting is 10 seconds, all calls cleared by the caller in less than 10 seconds (from arrival in the contact center) will be excluded from all Abandoned Calls Reports.
- `Missed calls`
  Refer to agents missing calls (not answering Call Center calls delivered to them).
  Example: Agent Tom is logged into the Sales queue. A call has been sent to Tom, but Tom is not at his desk because he likes talking with Katie. The Missed Call Timeout for the Sales queue is set to 20 seconds. This means Tom is given 20 seconds to answer the call. If he doesn’t answer, his agent status is set to Missed Call (i.e., he is still logged into the queue/s but he will not receive any more calls until he clicks the missed call tray pop). The fact that Tom missed this call is written in the database and is available in the Missed Call Reports.

2.5 Creating Reports

The Schedule Manager can be used to create reports using selected report templates. The scheduled generation of reports is enabled by defining schedules.

Every user can save his or her specifications for creating reports in order to use them again later.

The following output formats can be selected for the report preview and when sending reports by e-mail:

- Excel
- PDF
- Word

Note that a graphical representation of the report results is not possible in the Excel output formats.

The Report Preview displays a report in the desired output format. The report can then either be saved or opened with the appropriate application for the desired output format.

A report can also be sent as an e-mail attachment to any recipient. The subject of the e-mail always begins with the last name of the user and with the e-mail address configured in myReports. Additional text can be added to the subject if required.

2.6 Definition of a Contact Center Call

To interpret the generated reports correctly, it is important to understand the definition of a Contact Center call and the criteria for the beginning and end of a Contact Center call.

The Contact Center indicates a call as Contact Center call if the call arrives at a queue. All other calls are non-Contact-Center calls.

The lifetime of a Contact Center call comprises the interval from entering a queue until hanging up by agent or by caller.

A Contact Center call is also regarded as terminated if an agent transfers the call to another user (non-agent) or if the call is transferred to the voicemail box in a last step of a rule (Call Control Vector CCV) (For each queue, a schedule, and the rules defined in it (called a Call Control Vector or CCV), is used to specify how Contact Center calls are to be handled on certain days and at certain times.).

Call Scenarios and Call Reporting

Within the following, some call scenarios are listed to demonstrate the call counting and reporting of the Contact Center.

For all scenarios the following applies:

- A and B are configured as agents and are assigned to the same queue A.
- C is also configured as an agent but is assigned to another queue (queue C) than A and B.
- D is a normal user (non agent) within the communication system.

Scenario		Description	Description of reporting behavior
1	Incoming call to agent DID	External caller dials number of agent A directly.	Call is not counted as Contact Center call.
2	Incoming call to number of a queue	External caller dials number of queue. Call is routed to agent A. Agent A accepts call and hangs up after some time.	Call is counted as Contact Center call from entering the queue until hanging up.
3	Incoming call to agent A using number of a queue	External caller dials number of queue. Call is routed to agent A. Agent A does not answer the call. The call is routed to agent B. Agent B accepts the call and hangs up after some time.	The call is counted as a Contact Center call from entering the queue until agent B hangs up.
4	Incoming call to agent B using number of a queue	External caller dials number of queue. Call is routed to agent A. Agent A does not answer the call. The call is routed to agent B. Agent B does not answer the call. The call is routed to Voicemail. Caller hangs up after leaving a message in voicemail.	Call is counted as Contact Center call from entering the queue until it is transferred to voicemail. Voicemail recording time is not included within the Contact Center call duration.
5	Incoming call to number of a queue transferred to Voicemail	External caller dials number of queue. No agent is available. Call is held within the queue. Caller uses the callback option, leaves a message and hangs up.	Call is not counted as Contact Center call from entering the queue until caller hangs up. Time for recording the callback is included within the Contact Center call duration.

Scenario		Description	Description of reporting behavior
6	Consultation hold	A call enters queue A via the number of the queue. The phone of agent A rings. Agent A answers the call. Agent A uses consultation function to put the call on hold. Agent A retrieves call and hangs up after some time.	The call is counted as a Contact Center call. While the call is on hold, the talk time continues to count as the call still associated with agent A. Irrespective of how many times the agent may hold/unhold the call, it is still counted as one call.
7	Consultation	A call enters queue A via the number of the queue. The phone of agent A rings. Agent A answers the call. Agent A makes a consultation call to: • Agent B within the same queue • Agent C within another queue • User D (non agent) After consultation, agent A retrieves the call and hangs up after some time.	The call is counted as a Contact Center call. While the call is on hold, the talk time continues to count as the call still associated with agent A. Irrespective of how many times the agent may consult with another agent or user, it is still counted as one call and still reported as agent A's call with the whole time being his or her talk time (for that call).
8	Call transfer to agent	A call enters queue A via the number of the queue. The phone of agent A rings. Agent A answers the call. Agent A transfers the call to: • Agent B within the same queue • Agent C within another queue	The call is counted as a Contact Center call. Please note that the complete talk time is assigned to agent B or C in this example. That is, the talk time will not be split between agent A and agent B or agent C.
9	Call transfer to non-agent	A call enters queue A via the number of the queue. The phone of agent A rings. Agent A answers the call. Agent A transfers the call directly to user D (non-agent).	Call is not counted as Contact Center call. Please note that since the call is terminated with a non-agent, the call is not reported as a Contact Center call.
10	Call transfer to other queue	A call enters queue A via the number of the queue. The phone of agent A rings. Agent A answers the call. Agent A transfers the call using myAgent to queue B.	The call is counted as a Contact Center call. The reporting depends on what happens to the call in queue B. If the call is answered by an agent it is reported as such. If the call is abandoned, then it is an abandoned call.

Scenario		Description	Description of reporting behavior
11	Multiple consultations and call transfer	A call enters queue A via the number of the queue. The phone of agent A rings. Agent A answers the call. Agent A makes a consultation call to agent B within the same queue. Agent B is busy and agent A retrieves the call. Afterwards, agent A makes a consultation call to agent C. Agent C does not accept the call. Agent A retrieves the call again and initiates a third consultation to user D (non-agent). User D accepts the call, and agent A hangs up after a while. The call is now connected with user D. .	Irrespective of the number of transfer attempts, the call is still with agent A until it is successfully transferred. In this scenario, the call is not counted as a Contact Center call, since the call is terminated with a non-agent. In case the call is terminated by an agent, it would be counted as Contact Center call.
12	Call pickup	A call enters queue A via the number of the queue. The phone of agent A rings. Agent B in the same queue answers the call.	Call is counted as Contact Center call to the extent that the call is reported against agent A.
13	Call pickup by other agent	A call enters queue A via the number of the queue. The phone of agent A rings. Agent C in another queue answers the call.	Call is counted as Contact Center call to the extent that the call is reported against agent A.
14	Call pickup by non-agent	A call enters queue A via the number of the queue. The phone of agent A rings. User D (non agent) answers the call.	The call is not counted as a Contact Center call, since a non-agent picks up the call,
15	Outgoing call	Agent A initiates an outgoing call. The call is established. Agent A hangs up.	Call is not counted as Contact Center call in any reports.

Scenario		Description	Description of reporting behavior
16	Outgoing call callback	Agent A initiates an outgoing call to a callback destination using the PopUp window. The call is established. Agent A hangs up.	Call is counted as a Contact Center call in related reports.

To get clear, meaningful reports it is highly recommended for agents not to use features like

- Call pickup
- Call forwarding
- Call transfer
- Conference
- Toggle
- Call Park

It is important to note here that such features are not required for agents, since the routing is done automatically by the Contact Center, depending on the currently available resources and the programmed routing rules (Call Control Vector CCV). If the routing does not cover the requirements, the routing rules (Call Control Vector CCV) and/or agent to queue assignment should be adapted accordingly.

3 Predefined Report Templates in Detail

The following sections describe all the predefined report templates, depending on the various report groups.

The descriptions include details about

- the report parameters used,
- the output values and output format,
- the calculation rule,
- the database tables,
- the database table attributes,
- the SQL queries used,
- and possible exceptions.

3.1 Report Group Agent Activity

All predefined report templates of this report group are described below.

3.1.1 Agent Activity Logged Times

The report represents the login, logout and logged in times in the specified date interval for the selected agent.

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values	• Login time • Logout time • Logged In time • Daily total logged in time • Total logged in time
Format	• Table
Axis label	• N/A
Calculation rule	• Logged in time has event_type = 0 • Logout time has event_type = 1 • Login time: event time • Logout time: next logout event time • Logged in time: logout time – login time • Daily total logged in time: SUM(logged in time by day) • Total logged in time: SUM (logged in time)

Database tables	• tblagentactivity, tblusers, tbldepartments
Database table attributes	• tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id,user_is_agent} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available days having login activities for the selected agent in the selected date range

```
SELECT
  DISTINCT ("date"(tblagentactivity.aa_event_time)) AS
  "Date of Day"

FROM tblagentactivity, tblusers u

WHERE tblagentactivity.aa_event_time >= ? /* from time */
    AND tblagentactivity.aa_event_time <= ("date"(?) + INTERVAL
                              '24 hours') /* to date */
    AND tblagentactivity.aa_event_type = 0
    AND tblagentactivity.aa_agent_id = u.user_id
    AND u.user_login = ? /* agent login */
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT
      CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
          END
        FROM tbldepartments
        WHERE u.user_department_id = tbldepartments.department_id
     ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select agent activities (login, logout, logged in times)

```
SELECT
  "date"(sub1.aa_event_time) AS "Start Date",
  "time"(sub1.aa_event_time) AS "Start Time",
  "time"(sub2.aa_event_time) AS "End time",
    EXTRACT (
      epoch FROM (sub2.aa_event_time - sub1.aa_event_time)
            ) AS timeInSec

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
   (SELECT aa_id
      FROM tblagentactivity
      WHERE (aa_agent_id = sub1.aa_agent_id)
        AND (aa_id > sub1.aa_id)
      ORDER BY aa_event_time ASC LIMIT 1
    )
  AND sub1.aa_event_type = 0 /* 0 = logged in times */
  AND sub1.aa_event_time >= ? /* from time*/
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
             /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ?


ORDER BY "time"(sub1.aa_event_time);
```

Select total agent logged in time in seconds

```
SELECT
  SUM (
    EXTRACT (
      epoch FROM (sub2.aa_event_time - sub1.aa_event_time))
      ) AS duration

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
   (SELECT aa_id
      FROM tblagentactivity
      WHERE (aa_agent_id = sub1.aa_agent_id)
        AND (aa_id > sub1.aa_id)
      ORDER BY aa_event_time ASC LIMIT 1
    )
  AND sub1.aa_event_type = 0 /* 0 = logged in times */
  AND sub1.aa_event_time >= ? /* from time */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
             /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ? /* agent login */
```

Exception

- It is not usual, but is possible that the logout time is not the same day as the login time. In that specific case, it is possible to see the time values when the logout time is earlier then the login time, but the logged in time will be properly calculated.
  Example:
  - Login = 15:45:00
  - Logout = 10:27:00
  - Logged in time = 1 day 18:39:00
- The logged in times (login, logout, logged in time) are not represented when there is no logout time yet.

3.1.2 Agent Activity Missed Call Times

The report displays missed call times in specified date interval for selected agent.

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values	• Start time • End time • Daily total missed call time • Total missed call time
Format	• Table
Axis label	• N/A
Calculation rule	• Missed call time has event_type=6 • Start time: event time • End time: next event time • Missed call time: end time - start time • Daily total missed call time: SUM(missed call time by day) • Total missed call time: SUM(missed call time)
Database tables	• tblagentactivity, tblusers, tbldepartments
Database table attributes	• tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available days having missed calls activities for the selected agent in the selected date range

```
SELECT DISTINCT  ("date"(tblagentactivity.aa_event_time))AS "Date
of Day"

FROM tblagentactivity, tblusers u

WHERE tblagentactivity.aa_event_time >= ? /* from time */
      AND tblagentactivity.aa_event_time <= ("date"(?) + INTERVAL
                              '24 hours') /* to date */
      AND tblagentactivity.aa_event_type = 6
      AND tblagentactivity.aa_agent_id = u.user_id
      AND u.user_login = ? /* agent login */
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login, (SELECT CASE WHEN
tbldepartments.department_name = 'Unknown’
  THEN ’’ ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id )   AS
department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select agent activities (start date, start time, end time, missed call time)

```
SELECT  "date"(sub1.aa_event_time) AS "Start Date",
"time"(sub1.aa_event_time) AS "Start Time",
"time"(sub2.aa_event_time) AS "End Time", (sub2.aa_event_time -
sub1.aa_event_time) AS duration,
  EXTRACT (EPOCH FROM
  (sub2.aa_event_time - sub1.aa_event_time)) AS timeInSec

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
  (SELECT aa_id
    FROM tblagentactivity
    WHERE (aa_agent_id = sub1.aa_agent_id) AND (aa_id > sub1.aa_id)
    ORDER BY aa_event_time ASC
    LIMIT 1)
  AND sub1.aa_event_type = 6 /* 6 = Entgangener Anruf */
  AND sub1.aa_event_time >= ? /* from time */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ? /* agent login */

ORDER BY "time"(sub1.aa_event_time);
```

Select total missed call time in seconds

```
SELECT SUM (EXTRACT (EPOCH FROM
  (sub2.aa_event_time - sub1.aa_event_time))) AS duration

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
  (SELECT aa_id
    FROM tblagentactivity
    WHERE (aa_agent_id = sub1.aa_agent_id) AND (aa_id > sub1.aa_id)
    ORDER BY aa_event_time ASC
    LIMIT 1)
  AND sub1.aa_event_type = 6 /* 6 = Entgangener Anruf */
  AND sub1.aa_event_time >= ? /* from time */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ? /* agent login */
```

Exception

In the SQL queries above to convert the time values to seconds, some predefined PostgreSQL functions are used. To convert the seconds to time values in this report as well as many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.1.3 Agent Activity On Break Times

The report displays break times in the specified date interval for the selected agent.

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values	• Start time • End time • Break Name • Default Break Interval (min.) • Actual Break Time • Daily total Break Time • Total Break Time
Format	• Table
Axis label	• N/A

Calculation rule	• Break time has event_type = 2 • Start time: event time • End time: next event time • Actual break time: end time - start time • Daily total break time: SUM(actual break time by day) • Total break time: SUM(actual break time)
Database tables	• tblagentactivity, tblusers, tbldepartments, tblbreakscc
Database table attributes	• tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id} • tblbreakscc = {break_name, break_default_interval_minutes, break_id}

SQL Queries

Select all available days having break activities for the selected agent in the selected date range

```
SELECT DISTINCT   ("date"(tblagentactivity.aa_event_time)) AS
"Date of Day"

FROM tblagentactivity, tblusers u

WHERE tblagentactivity.aa_event_time >= ? /* from time */
  AND tblagentactivity.aa_event_time <= ("date"(?) + INTERVAL
                              '24 hours') /* to date */
  AND tblagentactivity.aa_event_type = 2
  AND tblagentactivity.aa_agent_id = u.user_id
  AND u.user_login = ? /* agent login */
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login, (SELECT CASE WHEN
tbldepartments.department_name='Unknown’ THEN ’’
    ELSE tbldepartments.department_name END
    FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id )
  AS department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select agent activities (start date, start time, end time, break name, default break interval in min., actual break time)

```
SELECT "date"(sub1.aa_event_time) AS "Start Date",
       "time"(sub1.aa_event_time) AS "Start Time",
       "time"(sub2.aa_event_time) AS "End Time",
  CASE WHEN sub1.aa_event_data
    NOT IN (SELECT tblbreakscc.break_id FROM tblbreakscc)
    THEN ’- - - ’
    ELSE (SELECT tblbreakscc.break_name
        FROM tblbreakscc
        WHERE tblbreakscc.break_id = sub1.aa_event_data) END
    AS break_name,
  CASE WHEN sub1.aa_event_data NOT IN
  (SELECT tblbreakscc.break_id FROM tblbreakscc)
    THEN 0
    ELSE (SELECT tblbreakscc.break_default_interval_minutes
         FROM tblbreakscc
         WHERE tblbreakscc.break_id = sub1.aa_event_data) END
    AS break_default_interval_minutes,
  (sub2.aa_event_time - sub1.aa_event_time) AS duration,
  EXTRACT (EPOCH FROM (sub2.aa_event_time - sub1.aa_event_time))
    AS timeInSec

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
  (SELECT aa_id
    FROM tblagentactivity
    WHERE (aa_agent_id = sub1.aa_agent_id) AND (aa_id > sub1.aa_id)
    ORDER BY aa_event_time ASC
    LIMIT 1)
  AND sub1.aa_event_type = 2 /* 2 = break time */
  AND sub1.aa_event_time >= ? /* from time */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                    /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ? /* agent login */

ORDER BY "time"(sub1.aa_event_time);
```

Select total break time in seconds

```
SELECT SUM (EXTRACT
  (EPOCH FROM (sub2.aa_event_time - sub1.aa_event_time)))
  AS duration

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
  (SELECT aa_id
    FROM tblagentactivity
    WHERE (aa_agent_id = sub1.aa_agent_id) AND (aa_id > sub1.aa_id)
    ORDER BY aa_event_time ASC
    LIMIT 1)
  AND sub1.aa_event_type = 2 /* 2 = break time */
  AND sub1.aa_event_time >= ? /* from time */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                    /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ? /* agent login */
```

Exception

In the SQL queries above to convert the time values to seconds, some predefined PostgreSQL functions are used. To convert the seconds to time values in this report as well as many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.1.4 Agent Activity Status (All Agents) – Daily

The report displays daily status details (logged, on break, work, missed call) by agents for one specified day.

Required input parameters	• From date (for day)
Output values (the values are grouped by agents)	• Start time • End time • Status Name • Status duration
Format	• Table
Axis label	• N/A
Calculation rule	• Office status has event_type in (0,2 ,4, 6) • Start time: event time • End time: next event time • Status duration: end time - start time
Database tables	• tblagentactivity, tblusers
Database table attributes	• tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type} • tblusers = {user_id, user_login, user_firstname, user_surname, user_is_agent}

SQL Queries

Select all available agents having activities for the selected day

```
SELECT DISTINCT u.user_id, u.user_surname, u.user_firstname
                u.user_login

FROM tblusers u

WHERE u.user_is_agent = 1
  AND u.user_id IN
   (SELECT ua.aa_agent_id FROM tblagentactivity ua
      WHERE ua.aa_event_time >= ?
      AND ua.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
    )
```

Select agent activities (agent surname, agent ID, status={logged,break, work, missed call}, start time, end time, duration)

```
SELECT u.user_surname, u.user_id,
  CASE sub1.aa_event_type
    WHEN 0 THEN 0 -- 'logged'
    WHEN 2 THEN 2 -- 'on break'
    WHEN 4 THEN 4 -- 'work'
    WHEN 6 THEN 6 -- 'missed call'
  END AS office_status,
  "time"(sub1.aa_event_time) AS "Start",
  "time"(sub2.aa_event_time) AS "End",
  (sub2.aa_event_time - sub1.aa_event_time) AS duration

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
  (SELECT aa_id
    FROM tblagentactivity
    WHERE (aa_agent_id = sub1.aa_agent_id) AND (aa_id > sub1.aa_id)
    ORDER BY aa_event_time ASC
    LIMIT 1)
  AND sub1.aa_event_type IN (0, 2, 4, 6) /* office status*/
  AND sub1.aa_event_time >= ? /* from time */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                            /* to date */
  AND sub1.aa_agent_id = u.user_id

ORDER BY sub1.aa_event_time;
```

Exception

N/A

3.1.5 Agent Activity Status (By Agent) - Daily

The report displays status details for selected agent and specified day.

Required input parameters	• From date (for day) • Agent
Output values	• Start time • End time • User Status • Duration
Format	• Table and graphics
Axis label	• Horizontal: time • Vertical: office status (logged, on break, work, missed call)

Calculation rule	• Work time has event_type = 4 • Start time: event time • End time: next event time • Duration: end time - start time
Database tables	• tblagentactivity, tblusers, tbldepartments
Database table attributes	• tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
(SELECT CASE WHEN tbldepartments.department_name = 'Unknown’
  THEN ’’ ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id
) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select agent activities (activity ID, agent ID, office status, duration) – used for the graphic

```
SELECT sub1.aa_id, sub1.aa.agent_id,
  CASE sub1.aa_event_type
    WHEN 0 THEN 'logged'
    WHEN 2 THEN 'on break'
    WHEN 4 THEN 'work time'
    WHEN 6 THEN 'missed call start'
    END AS office_status,
/* Log In */
  CASE sub1.aa_event_type
    WHEN 0 THEN sub1.aa_event_time END AS "LogInStartT",
  CASE sub1.aa_event_type
    WHEN 0 THEN sub2.aa_event_time END AS "LogOutEndT",
/* Break Start */
  CASE sub1.aa_event_type
    WHEN 2 THEN sub1.aa_event_time END AS "BreakStartStartT",
  CASE sub1.aa_event_type
    WHEN 2 THEN sub2.aa_event_time END AS "BreakEndT",
/* Work Time Start */
  CASE sub1.aa_event_type
    WHEN 4 THEN sub1.aa_event_time END AS "WTSStartT",
  CASE sub1.aa_event_type
    WHEN 4 THEN sub2.aa_event_time END AS "WTEEndT",
/* Missed Call Start */
  CASE sub1.aa_event_type
    WHEN 6 THEN sub1.aa_event_time END AS "MCSStartT",
  CASE sub1.aa_event_type
    WHEN 6 THEN sub2.aa_event_time END AS "MCEEndT",
  5 AS X,
  (sub2.aa_event_time - sub1.aa_event_time) AS duration

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
  (SELECT aa_id
    FROM tblagentactivity
    WHERE (aa_agent_id = sub1.aa_agent_id) AND (aa_id > sub1.aa_id)
    ORDER BY aa_event_time ASC
    LIMIT 1)
  AND sub1.aa_event_type IN (0,2, 4, 6)
  AND sub1.aa_event_time >= ? /* from date */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                            /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ?  /* agent login */

ORDER BY sub1.aa_event_time;
```

Select agent activities (activity ID, agent ID,office status, duration) – used for the table

```
SELECT sub1.aa_id, sub1.aa.agent_id,
  sub1.aa_event_type AS office status,
  "time"(sub1.aa_event_time) AS "Start Time",
  "time"(sub2.aa_event_time) AS "End Time",
  "date"(sub1.aa_event_time) AS "Start Date",
  (sub2.aa_event_time - sub1.aa_event_time) AS duration


FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
  (SELECT aa_id
    FROM tblagentactivity
    WHERE (aa_agent_id = sub1.aa_agent_id) AND (aa_id > sub1.aa_id)
    ORDER BY aa_event_time ASC
    LIMIT 1)
  AND sub1.aa_event_type IN (0,2, 4, 6)
  AND sub1.aa_event_time >= ? /* from date */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                             /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ?  /* agent login */

ORDER BY sub1.aa_event_time;
```

Exception

N/A

3.1.6 Agent Activity Work Times

The report displays work times in specified date interval for selected agent.

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values	• Start time • End time • Work Time • Daily total Work Time • Total work time
Format	• Table
Axis label	• N/A

Calculation rule	• Work time has event_type = 4 • Start time: event time • End time: next event time • Work time: end time - start time • Daily total work time: SUM(work time by day) • Total work time: SUM(work time)
Database tables	• tblagentactivity, tblusers, tbldepartments
Database table attributes	• tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available days having work activities for the selected agent in the selected date range

```
SELECT DISTINCT   ("date"(tblagentactivity.aa_event_time)) AS
"Date of Day"

FROM tblagentactivity, tblusers u

WHERE tblagentactivity.aa_event_time >= ? /* from time */
  AND tblagentactivity.aa_event_time <= ("date"(?) + INTERVAL
                              '24 hours') /* to date */
  AND tblagentactivity.aa_event_type = 4
  AND tblagentactivity.aa_agent_id = u.user_id
  AND u.user_login = ? /* agent login */
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login, (SELECT CASE WHEN
tbldepartments.department_name = 'Unknown’
  THEN ’’ ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id ) AS
department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select agent activities (start date, start time, end time, work time)

```
SELECT "date"(sub1.aa_event_time) AS "Start Date",
       "time"(sub1.aa_event_time) AS "Start Time",
       "time"(sub2.aa_event_time) AS "End Time",
  (sub2.aa_event_time - sub1.aa_event_time) AS duration,
  EXTRACT (EPOCH FROM (sub2.aa_event_time - sub1.aa_event_time))
    AS timeInSec

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
  (SELECT aa_id
    FROM tblagentactivity
    WHERE (aa_agent_id = sub1.aa_agent_id) AND (aa_id > sub1.aa_id)
    ORDER BY aa_event_time ASC
    LIMIT 1)
  AND sub1.aa_event_type = 4 /* 4 = work time */
  AND sub1.aa_event_time >= ? /* from time */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                             /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ? /* agent login
ORDER BY "time"(sub1.aa_event_time);
```

Select total work time in seconds

```
SELECT
  SUM (EXTRACT
    (EPOCH FROM (sub2.aa_event_time - sub1.aa_event_time)))
    AS duration

FROM tblagentactivity AS sub1,
     tblagentactivity AS sub2,
     tblusers u

WHERE sub2.aa_id =
  (SELECT aa_id
    FROM tblagentactivity
    WHERE (aa_agent_id = sub1.aa_agent_id) AND (aa_id > sub1.aa_id)
    ORDER BY aa_event_time ASC
    LIMIT 1)
  AND sub1.aa_event_type = 4 /* 4 = work time */
  AND sub1.aa_event_time >= ? /* from time */
  AND sub1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                             /* to date */
  AND sub1.aa_agent_id = u.user_id
  AND u.user_login = ? /* agent login */
```

Exception

In the SQL queries above to convert the time values to seconds, some predefined PostgreSQL functions are used. To convert the seconds to time values in this report as well as many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.2 Report Group Agents

All predefined report templates of this report group are described below.

3.2.1 Agent G.O.S.

The report shows hourly average grade of service GOS for a specified agent in a specified date range.

INFO: The report template **Agent G.O.S. – Daily** has a different graphic for each day.

Required input parameters	• From date • Until (to date) • Agent • Daily report
Output values	• N/A
Format	• Graphic
Axis label	• Horizontal: hourly intervals • Vertical: average grade of service (0-100)
Calculation rule	• N/A
Database tables	• tblcallscc, tblcalls, tbldepartments, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_agent_id, cc_gos} • tblcalls = {call_id, call_start_time, call_end_time} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
    u.user_email, u.user_login,
    (SELECT CASE WHEN tbldepartments.department_name = 'Unknown’
  THEN ’’ ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id )
    AS department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select hourly average GOS values

```
SELECT
  AVG (cc1."cc_gos"),
  EXTRACT (hour FROM c1."call_start_time") || ’:00 -’ ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ’:00’
  AS "label"

FROM tblcallscc cc1, tblcalls c1, tblusers u1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_agent_id" = u1."user_id"
  AND u1."user_login" = ? /* agent login */

GROUP BY EXTRACT (hour FROM c1."call_start_time")
ORDER BY EXTRACT (hour FROM c1."call_start_time")
```

Exception

In the SQL query above, predefined postgresql functions are used to extract the hour value from the specified date time value (call_start_time).

Example of “label” value (representing one hourly interval) : 16:00-17:00

3.2.2 Agent G.O.S. (Daily)

The report shows the hourly average grade of service GOS for the specified agent in a specified date range (there is a different graphic for each day).

Required input parameters	• From date • Until (to date) • Agent • Daily report
Output values	• N/A
Format	• Graphic
Axis label	• Horizontal: hourly intervals • Vertical: average grade of service (0-100)
Calculation rule	• N/A
Database tables	• tblcallscc, tblcalls, tbldepartments, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_agent_id, cc_gos} • tblcalls = {call_id, call_start_time, call_end_time} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
    u.user_email, u.user_login,
    (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
  THEN '' ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id )
    AS department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select all available days having calls for the selected agent in the selected date range

```
SELECT DISTINCT "date"(c1."call_start_time")

FROM tblcallscc cc1, tblcalls c1, tblusers u1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_agent_id" = u1."user_id"
  AND u1."user_login" = ? /* agent login */
```

Select hourly average GOS values

```
SELECT
  AVG (cc1."cc_gos"),
  EXTRACT (hour FROM c1."call_start_time")||':00 -' ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ':00'
  AS "label"

FROM tblcallscc cc1, tblcalls c1, tblusers u1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_agent_id" = u1."user_id"
  AND u1."user_login" = ? /* agent login */

GROUP BY EXTRACT (hour FROM c1."call_start_time")
ORDER BY EXTRACT (hour FROM c1."call_start_time")
```

Exception

In the SQL query above, predefined postgresql functions are used to extract the hour value from the specified date time value (call_start_time).

Example of “label” value (representing one hourly interval) : 16:00-17:00

3.2.3 Agent Private Calls (All Agents)

The report shows details about the agent private calls in the specified date range.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Agent • Agent extension • Department • Number of calls • Talk time • Percentage of total talk time
Format	• Table
Axis label	• N/A
Calculation rule	• Private call: {called number NOT IN available queue huntgroups} • Number of calls (by agent): COUNT (call ids for this agent) • Percentage of total talk (by agent): talk time (by agent) / total talk time (all agents) * 100
Database tables	• tblcallhistory, tblusers, tbldepartments, tblqueues
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_talk_time_seconds, ch_called_number, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_extension, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id} • tblqueues = {queue_huntgroup}

SQL Queries

Select agent private call details

```
SELECT u.user_firstname,
       u.user_surname,
       u.user_extension,
       u.user_login,
  CASE WHEN u.user_department_id IN
    (SELECT department_id FROM tbldepartments
    WHERE department_id = u.user_department_id
    ) ELSE ' - - - ' END AS department_name,
  SUM (ch.ch_talk_time_seconds) AS "Total Talk Time",
  COUNT (ch.ch_call_id) AS "NumberOfCalls"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
    AND ch.ch_start_time <= ("date" (?) + INTERVAL '24 hours')
                             /* to date */
    AND ch.ch_called_number NOT IN
     (SELECT qe.queue_huntgroup FROM tblqueues qe)
    AND ch.ch_user_id = u.user_id
    AND u.user_is_agent = 1

GROUP BY u.user_firstname, u.user_surname, u.user_extension,
         u.user_login, u.user_department_id

ORDER BY u.user_firstname, u.user_surname, u.user_login
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used:

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.2.4 Agent Private Calls (Per Agent)

The report shows detailed information about the agent private calls for the specified agent in the specified date range.

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values (the values are grouped daily)	• Call date • Start time • Calling number • Called number • Direction I/O= (inbound/outbound) • Talk time • Daily total number of calls • Daily total talk time • Total number of calls • Total talk time
Format	• Table
Axis label	• N/A
Calculation rule	• Private call: {called number NOT IN available queue huntgroups} • Outbound: ch_direction=1 • Inbound: ch_direction=0 • Daily total number of calls: COUNT(number of calls by day) • Daily total talk times: SUM(talk time by day)
Database tables	• tblcallhistory, tblusers, tbldepartments, tblqueues
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_talk_time_seconds, ch_called_number, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id} • tblqueues = {queue_huntgroup}

SQL Queries

Select agent private call details

```
SELECT "date"(ch.ch_start_time) AS "date of call",
       "time"(ch.ch_start_time) AS "time of call",
       ch.ch_called_number,
       ch.ch_calling_number,
       ch.ch_direction,
       ch.ch_talk_time_seconds

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
    AND ch.ch_start_time <= ("date" (?) + INTERVAL '24 hours')
                            /* to date */
    AND ch.ch_called_number NOT IN
     (SELECT qe.queue_huntgroup FROM tblqueues qe)
    AND ch.ch_user_id = u.user_id
    AND u.user_login = ? /* agent login */
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login, (SELECT CASE WHEN
tbldepartments.department_name = 'Unknown’
  THEN ’’ ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id ) AS
department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select grand totals (total number of calls, total talk time in seconds)

```
SELECT
  COUNT (ch.ch_called_number),
  SUM (ch.ch_talk_time_seconds)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
    AND ch.ch_start_time <= ("date" (?) + INTERVAL '24 hours')
                             /* to date */
    AND ch.ch_called_number NOT IN
     (SELECT qe.queue_huntgroup FROM tblqueues qe)
    AND ch.ch_user_id = u.user_id
    AND u.user_login = ? /* agent login */
```

Select all available days having private calls for the selected agent in the selected date range (used for grouping the information by days)

```
SELECT DISTINCT "date"(ch.ch_start_time) AS "AllDates"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
    AND ch.ch_start_time <= ("date" (?) + INTERVAL '24 hours')
                             /* to date */
    AND ch.ch_called_number NOT IN
     (SELECT qe.queue_huntgroup FROM tblqueues qe)
    AND ch.ch_user_id = u.user_id
    AND u.user_login = ? /* agent login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used:

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.2.5 All User Calls (By Agent)

The report shows detailed information about all user calls for the specified agent in the specified date range.

Required input parameters	• From date • To date (until) • Agent • Business hours only (else 24/24) • Daily report
Output values (the values are grouped daily)	• Call date • Start time • End time • Calling Number • Called number • I/C – Incoming call (yes or no) • O/G – Outgoing call (yes or no) • Int – Internal call (yes or no) • Talk time • Daily totals for: number of I/C, number of O/G, number of internal calls, talk time • Grand totals for: number of I/C, number of O/G, number of internal calls, talk time
Format	• Table
Axis label	• N/A
Calculation rule	• Daily total number of I/C: COUNT(Number of I/C by day) • Daily total number of O/G: COUNT(Number of O/G calls by day) • Daily total number of Int.: COUNT(Number of Int. by day) • Daily total talk time: SUM (Talk time by day) • Total number of I/C: COUNT(Number of I/C) • Total number of O/G: COUNT(Number of O/G) • Total number of Int.: COUNT(Number of Int.) • Total talk time: SUM(talk time) • Incoming call: ch_direction = 0 • Outgoing call: ch_direction = 1 • Internal call: ch_internal_external = 0 • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallhistory, tblusers, tbldepartments, tblswitches

Database table attributes	• tblswitches = {switch_office_start, switch_office_end} • tblcallhistory = {ch_call_id, ch_start_time, ch_talk_time_seconds, ch_called_number, ch_calling_number, ch_direction, ch_user_id, ch_internal_external} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_is_agent, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select call details

```
SELECT
  "date"(ch.ch_start_time) AS "Day_of_Call",
  "time"(ch.ch_start_time) AS "Start_Time",
  "time"(ch.ch_end_time) AS "End_Time",
  ch.ch_calling_number,
  ch.ch_called_number,
  ch.ch_talk_time_seconds AS "Talk_Time",
  ch.ch_internal_external AS "INT",
  ch.ch_direction

FROM tblcallhistory ch, tblusers u, tblswitches s

WHERE u.user_id = ch.ch_user_id
    AND ch.ch_start_time >= ? /* from time*/
    AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                             /* to date */
    AND u.user_login = ? /* agent login */
    AND (CASE
       WHEN ? = 1 THEN /* Business hours only */
         "time"(ch.ch_start_time) >= "time"(s.switch_office_start)
           AND
         "time"(ch.ch_start_time) <= "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
        "time"(ch.ch_start_time) >= '00:00:00'AND
        "time"(ch.ch_start_time) <= '23:59:59'
        END )
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login, (SELECT CASE WHEN
tbldepartments.department_name = 'Unknown’
  THEN ’’ ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id ) AS
department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select grand totals (total number of calls, total talk time in seconds, total number of Int. calls, total number of I/C calls, total number of O/G calls)

```
SELECT COUNT (ch.ch_call_id),
  SUM (ch.ch_talk_time_seconds) AS "Total_Talk_Time",
  COUNT (CASE WHEN ch.ch_internal_external = 0 THEN
        ch.ch_internal_external END) AS "INT_TOTAL",
  COUNT (CASE WHEN ch.ch_direction = 0 THEN
        ch.ch_direction END) AS "IC_TOTAL",
  COUNT (CASE WHEN ch.ch_direction = 1 THEN
        ch.ch_direction END) AS "OG_TOTAL"

FROM tblcallhistory ch, tblusers u, tblswitches s

WHERE u.user_id = ch.ch_user_id
  AND ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND u.user_login = ? /* agent login */
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
    "time"(ch.ch_start_time) >= "time"(s.switch_office_start) AND
    "time"(ch.ch_start_time) <= "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
    "time"(ch.ch_start_time) >= '00:00:00'AND
    "time"(ch.ch_start_time) <= '23:59:59'
  END )
```

Select all available days having calls for the selected agent in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT "date"(ch.ch_start_time) AS "Date_of_Call"

FROM tblcallhistory ch, tblusers u, tblswitches s

WHERE u.user_id = ch.ch_user_id
  AND ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND u.user_login = ? /* agent login */
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
    "time"(ch.ch_start_time) >= "time"(s.switch_office_start)
AND    "time"(ch.ch_start_time) <= "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
    "time"(ch.ch_start_time) >= '00:00:00' AND
    "time"(ch.ch_start_time) <= '23:59:59'
  END )
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.2.6 All User Calls (By Agent) 2

The report shows detailed information about all user calls for the specified agent in the specified date range.

INFO: For the report template **All User Calls (By Agent)**, the report parameter **Business hours only** can be additionally selected.

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values (the values are grouped daily)	• Call date • Start time • End time • Calling Number • Called number • I/C – Incoming call (yes or no) • O/G – Outgoing call (yes or no) • Int – Internal call (yes or no) • Talk time • Daily totals for: number of I/C, number of O/G, number of internal calls, talk time • Grand totals for: number of I/C, number of O/G, number of internal calls, talk time
Format	• Table
Axis label	• N/A
Calculation rule	• Daily total number of I/C: COUNT(Number of I/C by day) • Daily total number of O/G: COUNT(Number of O/G calls by day) • Daily total number of Int.: COUNT(Number of Int. by day) • Daily total talk time: SUM (Talk time by day) • Total number of I/C: COUNT(Number of I/C) • Total number of O/G: COUNT(Number of O/G) • Total number of Int.: COUNT(Number of Int.) • Total talk time: SUM(talk time) • Incoming call: ch_direction = 0 • Outgoing call: ch_direction = 1 • Internal call: ch_internal_external = 0
Database tables	• tblcallhistory, tblusers, tbldepartments

Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_talk_time_seconds, ch_called_number, ch_calling_number, ch_direction, ch_user_id, ch_internal_external} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_is_agent, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select call details

```
SELECT
  "date"(ch.ch_start_time) AS "Day_of_Call",
  "time"(ch.ch_start_time) AS "Start_Time",
  "time"(ch.ch_end_time) AS "End_Time",
  ch.ch_calling_number,
  ch.ch_called_number,
  ch.ch_talk_time_seconds AS "Talk_Time",
  ch.ch_internal_external AS "INT",
  ch.ch_direction

FROM tblcallhistory ch, tblusers u,tblswitches s

WHERE u.user_id = ch.ch_user_id
    AND ch.ch_start_time >= ? /* from time */
    AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                             /* to date */
    AND u.user_login = ? /* agent login */
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
```

Select details for the selected agent

```
SELECT
  u.user_firstname,
  u.user_surname,
  u.user_extension,
  u.user_email,
  u.user_login, (
  (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
  ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select grand totals (total number of calls, total talk time in seconds, total number of Int. calls, total number of I/C calls, total number of O/G calls)

```
SELECT
  COUNT (ch.ch_call_id),
  SUM (ch.ch_talk_time_seconds) AS "Total_Talk_Time",
  COUNT (CASE WHEN ch.ch_internal_external = 0 THEN
        ch.ch_internal_external END) AS "INT_TOTAL",
  COUNT (CASE WHEN ch.ch_direction = 0 THEN
        ch.ch_direction END) AS "IC_TOTAL",
  COUNT (CASE WHEN ch.ch_direction = 1 THEN
        ch.ch_direction END) AS "OG_TOTAL"

FROM tblcallhistory ch, tblusers u, tblswitches s

WHERE u.user_id = ch.ch_user_id
  AND ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND u.user_login = ? /* agent login */
```

Select all available days having calls for the selected agent in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT "date"(ch.ch_start_time) AS "Date_of_Call"

FROM tblcallhistory ch, tblusers u, tblswitches s

WHERE u.user_id = ch.ch_user_id
    AND ch.ch_start_time >= ? /* from time */
    AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                             /* to date */
    AND u.user_login = ? /* agent login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • sec. = ts – (d*86400) – (h*3600) – (m*60)

Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3 Report Group Call History

All predefined report templates of this report group are described below.

3.3.1 External Calls Per User

The report shows information about the user external calls for the specified user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Call date • Start time • End time • CLI • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• External call: {ch_internal_external = 1} • Length of call: end time - start time • Daily total number of calls: COUNT(number of calls by day) • Daily total length of calls: SUM (length of calls by day) • CLI: calling number, when ch_direction = 0 (incoming call) • CLI: called number, when ch_direction = 1 (outgoing call)
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_called_number, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select external call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Start_Time",
  "time"(ch.ch_end_time) AS "End_Time",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call*/
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
  CASE ch.ch_direction
    WHEN 0 THEN ch.ch_calling_number
    WHEN 1 THEN ch.ch_called_number END AS "CLI"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_internal_external = 1 /* Externer Anruf*/
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ?

ORDER BY ch.ch_start_time
```

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
        END
      FROM tbldepartments
      WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select grand totals (total number of calls, total talk time in seconds)

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_internal_external = 1 /* Externe Anrufe */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select all available days having external calls for specified user in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time*/
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date*/
  AND ch.ch_internal_external = 1 /* Externe Anrufe */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* Benutzer-Login*/
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.2 Incoming Calls (Free-Calls) - Per User

The report shows incoming free calls details for the selected user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
        END
      FROM tbldepartments
      WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having incoming free calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
```

Select incoming free call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "StartTime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call */
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_calling_number AS ch_calling number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')

ORDER BY ch.ch_start_time
```

Select incoming free calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
```

Exception

1. The report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all incoming calls will be displayed

   Example: The report named "Incoming Calls (International) Per User" will display all incoming calls (not only the incoming international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.3 Incoming Calls (International) – Per User

The report shows incoming international calls details for the selected user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
        END
      FROM tbldepartments
      WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having incoming international calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
```

Select incoming international call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "StartTime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call */
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
  ch.ch_calling_number AS ch_calling number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
ORDER BY ch.ch_start_time
```

Select incoming international calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
```

Exception

1. The report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all incoming calls will be displayed

   Example: The report named "Incoming Calls (International) Per User" will display all incoming calls (not only the incoming international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.4 Incoming Calls (Mobile/Cell) – Per User

The report shows incoming mobile/cell calls details for the selected user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
        END
      FROM tbldepartments
      WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having incoming mobile/cell calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
```

Select incoming mobile/cell call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "StartTime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call */
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
  ch.ch_calling_number AS ch_calling number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
ORDER BY ch.ch_start_time
```

Select incoming mobile/cell calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
```

Exception

1. This report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all incoming calls will be displayed

   Example: The report named "Incoming Calls (International) Per User" will display all incoming calls (not only the incoming international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.5 Incoming Calls (Other External Calls) – Per User

The report shows incoming other calls details for the specified user in the specified date range (other calls means not international, free, pay, mobile/cell and specific calls).

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
        END
      FROM tbldepartments
      WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having incoming other calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_pay') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
```

Select incoming other call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "StartTime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call*/
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
  ch.ch_calling_number AS ch_calling number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_pay') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
ORDER BY ch.ch_start_time
```

Select incoming other calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_pay') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
```

Exception

1. This report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all incoming calls will be displayed

   Example: The report named "Incoming Calls (International) Per User" will display all incoming calls (not only the incoming international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s

Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.6 Incoming Calls (Specific Calls) – Per User

The report shows incoming specific calls details for the specified user in the specified date range (incoming specific calls means incoming calls filtered by specific call number prefix).

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Calling number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
        END
      FROM tbldepartments
      WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having incoming specific calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* Benutzer-Login*/
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
```

Select incoming specific call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "StartTime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call*/
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
  ch.ch_calling_number AS ch_calling number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
ORDER BY ch.ch_start_time
```

Select incoming specific calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_calling_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
```

Exception

1. This report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all incoming calls will be displayed

   Example: The report named "Incoming Calls (International) Per User" will display all incoming calls (not only the incoming international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.7 Incoming Calls Per User

The report shows information about the incoming calls for the specified user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily per calling number)	• Calling number • Date of call • Start time • End time • Length of call • Daily total length of calls per calling number • Daily total number of calls per calling number • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Incoming call: ch_direction = 0 • Length of call: end time - start time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select incoming call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "StartTime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call*/
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
  ch.ch_calling_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */
  AND u.user_login = ? /* user login */

ORDER BY ch.ch_start_time
```

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
        END
      FROM tbldepartments
      WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select grand totals (total number of calls, total talk time in seconds)

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select all available days having incoming calls for the specified user in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select available calling numbers of the incoming calls for the specified user in the specified date range (used for grouping the information by calling numbers)

```
SELECT
  DISTINCT ch.ch_calling_number,
  ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.8 Incoming Calls Report – Group

The report shows information about all incoming calls grouped by departments.

Required input parameters	• From date • To date (until) • Daily report
Output values (the values are grouped by departments)	• Department • User • Extension • Total number of calls per user • Total ring time per user • Total talk time per user • Total number of calls, ring time and talk time per department
Format	• Table
Axis label	• N/A
Calculation rule	• Incoming call: ch_direction = 0 • Ring time: call end time - call start time - call talk time
Database tables	• tblcallhistory, tblusers, tbldepartments

Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_ ch_talk_time_seconds, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_department_id, user_extension} • tbldepartments = {department_id, department_name}

SQL Queries

Select incoming call details by user and departments

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
  CASE WHEN u.user_department_id > 0 THEN u.user_department_id
    ELSE 0 END
    AS department_id,
    SUM ((EXTRACT (EPOCH FROM
     (ch.ch_end_time - ch.ch_start_time))- ch_talk_time_seconds))
       AS totalRingTime,
    SUM (ch_talk_time_seconds) AS totalTalkTime,
    COUNT (ch_call_id) AS nbCalls

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */

GROUP BY u.user_firstname, u.user_surname, u.user_extension,
         department_id
ORDER BY u.user_firstname, u.user_surname
```

Select all available departments and the number of users per department having calls in the specified date range

```
(SELECT department_name, department_id,
  COUNT (DISTINCT u.user_id) AS total

  FROM tbldepartments, tblusers u, tblcallhistory h

  WHERE department_id = u.user_department_id
    AND u.user_id = h.ch_user_id
    AND h.ch_start_time >= ? /* from time */
    AND h.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                            /* to date */
    AND h.ch_direction = 0 /* incoming calls */

  GROUP BY department_name, department_id

  ORDER BY department_name ) /* Benutzer ohne Abteilung */
  UNION (SELECT ' - - - ', 0,
  (SELECT COUNT (DISTINCT u.user_id)
    FROM tblusers u, tblcallhistory h
    WHERE u.user_id = h.ch_user_id
      AND h.ch_start_time >= ? /* from time */
      AND h.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
      AND h.ch_direction = 0 /* incoming calls */
      AND (u.user_daperatment_id IS null
        OR u.user_department_id = 0)) AS total)
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.9 Incoming Calls Report – Group Summary

The report shows summary information about the incoming calls per departments.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Department • Total number of calls per department • Total ring time per department • Total talk time per department • Total number of calls, total ring time and total talk time (all departments)
Format	• Table
Axis label	• N/A
Calculation rule	• Incoming call: ch_direction = 0 • Ring time: call end time - call start time - call talk time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_talk_time_seconds, ch_direction, ch_user_id} • tblusers = {user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select incoming calls details (number of call, ring time, talk time) per department

```
SELECT
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name,
    SUM ((EXTRACT (EPOCH FROM
   (ch.ch_end_time - ch.ch_start_time))- ch_talk_time_seconds))
      AS totalRingTime,
  SUM (ch_talk_time_seconds) AS totalTalkTime,
  COUNT (ch_call_id) AS nbCalls

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */

GROUP BY department_name
ORDER BY department_name
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.10 Incoming Calls Report – User

The report shows information about all incoming calls for the specified user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • CLI – Calling number • Ring time • Talk Time • Daily total number of calls • Daily total ring time • Daily total talk time • Total number of calls • Total talk time
Format	• Table
Axis label	• N/A
Calculation rule	• Incoming call: ch_direction = 0 • Ring time: end time - start time - talk time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_ ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_extension} • tbldepartments = {department_id, department_name}

SQL Queries

Select incoming call details (call start time, end time, ring time, talk time, calling number)

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "StartTime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
    /* length of call*/
  (EXTRACT(EPOCH FROM
   (ch.ch_end_time - ch.ch_start_time))- ch_talk_time_seconds)
   AS ringTime,
  ch.ch_calling_number,
  ch_talk_time_seconds

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */
  AND u.user_login = ? /* user login */

ORDER BY ch.ch_start_time
```

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select grand totals (total number of calls, total talk time in seconds)

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date* /
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select all available days having incoming calls for the specified user in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.11 Incoming Calls Report – User Summary

The report shows summary information about the incoming calls per users.

Required input parameters	• From date • To date (until) • Daily report
Output values	• User first name • User surname • User extension • Total number of calls per user • Total ring time per user • Total talk time per user • Total number of calls, total ring time and total talk time (all users)
Format	• Table
Axis label	• N/A
Calculation rule	• Incoming call: ch_direction = 0 • Ring time: call end time - call start time - call talk time
Database tables	• tblcallhistory, tblusers
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_talk_time_seconds, ch_direction, ch_user_id} • tblusers = {user_id, user_firstname, user_surname, user_extension

SQL Queries

Select incoming calls details (number of call, ring time, talk time) per users

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
  SUM ((EXTRACT (EPOCH FROM
   (ch.ch_end_time - ch.ch_start_time))- ch_talk_time_seconds))
      AS totalRingTime,
  SUM (ch_talk_time_seconds) AS totalTalkTime,
  COUNT (ch_call_id) AS nbCalls

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */

GROUP BY u.user_firstname, u.user_surname, u.user_extension
ORDER BY u.user_firstname, u.user_surname
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.12 Internal Calls Per User

The report shows information about the internal calls for the specified user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Call date • Start time • End time • CLI - • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A

Calculation rule	• Internal call: {ch_internal_external = 0} • Length of call: end time - start time • Daily total number of calls: COUNT(number of calls by day) • Daily total length of calls: SUM (length of calls by day) • CLI: calling number, when ch_direction = 0 (incoming call) • CLI: called number, when ch_direction = 1 (outgoing call)
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_called_number, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select internal call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Start_Time",
  "time"(ch.ch_end_time) AS "End_Time",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
     /* length of call*/
  EXTRACT(EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
  CASE ch.ch_direction
    WHEN 0 THEN ch.ch_calling_number
    WHEN 1 THEN ch.ch_called_number END AS "CLI"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_internal_external = 0 /* Interner Anruf */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ?

ORDER BY ch.ch_start_time
```

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select grand totals (total number of calls, total talk time in seconds)

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_internal_external = 0 /* Interne Anrufe */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select all available days having internal calls for the specified user in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch,tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_internal_external = 0 /* Interne Anrufe */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.13 Missed Calls (Incoming) Per User

The report shows incoming missed calls details for the selected user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Missed Call Time • Daily total missed call time • Total missed call time • Total number of missed calls
Format	• Table
Axis label	• N/A
Calculation rule	• Missed call = {tblcallhistory.ch_talk_time_seconds = 0 , tblcallhistory.ch_user_id > 0} • Missed call time: end time - start time • Incoming call: ch_direction = 0

Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_direction, ch_talk_time_seconds, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having incoming missed calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time*/
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date*/
  AND ch.ch_talk_time_seconds = 0 /* missed call */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* Benutzer-Login*/
```

Select incoming missed call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Start_Time",
  "time"(ch.ch_end_time) AS "End_Time",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
   /* length of call until disconnected by user */
  EXTRACT (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_talk_time_seconds = 0 /* missed call*/
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */
  AND u.user_login = ? /* user login */

ORDER BY ch.ch_start_time
```

Select total number of incoming missed calls and total incoming missed calls time for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time*/
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date* /
  AND ch.ch_talk_time_seconds = 0 /* missed call */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)

Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.14 Missed Calls (Incoming) Per User 2

The report shows incoming missed calls details for the specified user in the specified date range (including calling number details).

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Calling number • Missed call time • Daily total missed call time • Total missed call time • Total number of missed calls
Format	• Table
Axis label	• N/A
Calculation rule	• Missed call = {tblcallhistory.ch_talk_time_seconds = 0 , tblcallhistory.ch_user_id > 0} • Missed call time: end time - start time • Incoming call: ch_direction = 0
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_calling_number, ch_direction, ch_talk_time_seconds, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having incoming missed calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_talk_time_seconds = 0 /* missed call*/
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select incoming missed call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Start_Time",
  "time"(ch.ch_end_time) AS "End_Time",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
   /* length of call until disconnected by user  */
  EXTRACT (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_calling_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_talk_time_seconds = 0 /* missed call*/
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 0 /* incoming calls */
  AND u.user_login = ? /* user login */

ORDER BY ch.ch_start_time
```

Select total number of incoming missed calls and total incoming missed calls time for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time*/
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date*/
  AND ch.ch_talk_time_seconds = 0 /* missed call */
  AND ch.ch_direction = 0 /* incoming calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* Benutzer-Login*/
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.15 Missed Calls (Outgoing) Per User

The report shows outgoing missed calls details for the selected user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Missed Call Time • Daily total missed call time • Total missed call time • Total number of missed calls
Format	• Table
Axis label	• N/A
Calculation rule	• Missed call = {tblcallhistory.ch_talk_time_seconds = 0 , tblcallhistory.ch_user_id > 0} • Missed call time: end time - start time • Outgoing call: ch_direction = 1

Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_direction, ch_talk_time_seconds, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having outgoing missed calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time*/
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date*/
  AND ch.ch_talk_time_seconds = 0 /* missed call */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select outgoing missed call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Start_Time",
  "time"(ch.ch_end_time) AS "End_Time",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
   /* length of call until disconnected by user  */
  EXTRACT (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_talk_time_seconds = 0 /* missed call*/
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */

ORDER BY ch.ch_start_time
```

Select total number of outgoing missed calls and total outgoing missed calls time for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_talk_time_seconds = 0 /* missed call*/
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)

Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.16 Missed Calls (Outgoing) Per User 2

The report shows outgoing missed calls details for the specified user in the specified date range (including called number details).

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Missed call time • Called number • Daily total missed call time • Total missed call time • Total number of missed calls
Format	• Table
Axis label	• N/A
Calculation rule	• Missed call = {tblcallhistory.ch_talk_time_seconds = 0 , tblcallhistory.ch_user_id > 0} • Missed call time: end time - start time • Outgoing call: ch_direction = 1
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_direction, ch_talk_time_seconds, ch_user_id, ch_called_number} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having outgoing missed calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_talk_time_seconds = 0 /* missed call*/
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select outgoing missed call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Start_Time",
  "time"(ch.ch_end_time) AS "End_Time",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
   /* length of call until disconnected by user  */
  EXTRACT(EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_called_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_talk_time_seconds = 0 /* missed call*/
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */
  AND u.user_login = ? /* user login */

ORDER BY ch.ch_start_time
```

Select total number of outgoing missed calls and total outgoing missed calls time for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_talk_time_seconds = 0 /* missed call*/
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.17 Outgoing Calls (Free Calls) – Per User

The report shows outgoing free calls details for the selected user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time • Outgoing call: ch_direction = 1

Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_called_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having outgoing free calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
```

Select outgoing free call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Starttime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call*/
  EXTRACT (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_called_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
ORDER BY ch.ch_start_time
```

Select outgoing free calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
```

Exception

1. This report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all outgoing calls will be displayed

   Example: The report named "Outgoing Calls (International) Per User" will display all outgoing calls (not only the outgoing international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.18 Outgoing Calls (International) – Per User

The report shows outgoing international calls details for the selected user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time • Outgoing call: ch_direction = 1
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_called_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having outgoing international calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
```

Select outgoing international call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Starttime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call */
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_called_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
ORDER BY ch.ch_start_time
```

Select outgoing international calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
```

Exception

1. This report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all outgoing calls will be displayed

   Example: The report named "Outgoing Calls (International) Per User" will display all outgoing calls (not only the outgoing international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.19 Outgoing Calls (Mobile/Cell) – Per User

The report shows outgoing mobile/cell calls details for the selected user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time • Outgoing call: ch_direction = 1
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_called_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having outgoing mobile/cell calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
```

Select outgoing mobile/cell call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Starttime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call */
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_called_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
ORDER BY ch.ch_start_time
```

Select outgoing mobile/cell calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
```

Exception

1. This report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all outgoing calls will be displayed

   Example: The report named "Outgoing Calls (International) Per User" will display all outgoing calls (not only the outgoing international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.20 Outgoing Calls (Other External Calls) – Per User

The report shows outgoing other calls details for the selected user in the specified date range (other calls means not international, free, pay, mobile/cell and specific calls).

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time • Outgoing call: ch_direction = 1
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_called_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having outgoing other calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_pay') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
```

Select outgoing other call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Starttime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call */
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_called_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_pay') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
ORDER BY ch.ch_start_time
```

Select outgoing other calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_international') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_mobile') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_pay') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_free') ||')')
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
```

Exception

1. This report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all outgoing calls will be displayed

   Example: The report named "Outgoing Calls (International) Per User" will display all outgoing calls (not only the outgoing international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s

Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.21 Outgoing Calls (Pay Calls) – Per User

The report shows outgoing pay calls details for the selected user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time • Outgoing call: ch_direction = 1
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_called_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having outgoing pay calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_pay') ||')')
```

Select outgoing pay call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Starttime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call*/
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_called_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_pay') ||')')
ORDER BY ch.ch_start_time
```

Select outgoing pay calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_pay') ||')')
```

Exception

1. This report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all outgoing calls will be displayed

   Example: The report named "Outgoing Calls (International) Per User" will display all outgoing calls (not only the outgoing international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.22 Outgoing Calls (Specific Calls) – Per User

The report shows outgoing specific calls details for the specified user in the specified date range (outgoing specific calls means outgoing calls filtered by specific call number prefix).

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • End time • Called Number • Length of call • Daily total number of calls • Daily total length of calls • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time • Outgoing call: ch_direction = 1
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_called_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having outgoing specific calls for the specified user in the specified date range

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
```

Select outgoing specific call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "Starttime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call*/
  EXTRACT
    (EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_called_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
ORDER BY ch.ch_start_time
```

Select outgoing specific calls grand totals (total length of calls, total number of calls) for the specified user in the specified date range

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
  AND ch.ch_called_number ~ ('^(' ||
    (SELECT cf_param_value FROM tblconfig
        WHERE cf_param_name='spcn_specific') ||')')
```

Exception

1. This report uses the call number prefix to specify the call number filter (free calls, pay calls, international calls, mobile/cell calls ...).
   The call number prefixes can be configured using myReports application (specific phone numbers).

   INFO: When the associated call number prefix is empty, all outgoing calls will be displayed

   Example: The report named "Outgoing Calls (International) Per User" will display all outgoing calls (not only the outgoing international calls) when the international call number prefix is empty.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.23 Outgoing Calls Per User

The report shows information about the outgoing calls for the specified user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily per called number)	• Called number • Date of call • Start time • End time • Length of call • Daily total length of calls per called number • Daily total number of calls per called number • Total number of calls • Total length of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Length of call: end time - start time • Outgoing call: ch_direction = 1
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_called_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select outgoing call details

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "StartTime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
  /* length of call*/
  EXTRACT(EPOCH
      FROM (ch.ch_end_time - ch.ch_start_time)) AS timeInSec,
        ch.ch_called_number

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */
  AND u.user_login = ? /* user login */

ORDER BY ch.ch_start_time
```

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
        END
      FROM tbldepartments
      WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select grand totals (total number of calls, total talk time in seconds)

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select all available days having outgoing calls for the specified user in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* Benutzer-Login*/
```

Select available called numbers of the outgoing calls for the specified user in the specified date range (used for grouping the information by called numbers)

```
SELECT
  DISTINCT ch.ch_called_number,
  ("date"(ch.ch_start_time)) AS "CallDate",

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.24 Outgoing Calls Report – Group

The report shows information about all outgoing calls grouped by departments.

Required input parameters	• From date • To date (until) • Daily report
Output values (the values are grouped by departments)	• Department • User • Extension • Total number of calls per user • Total ring time per user • Total talk time per user • Total number of calls, ring time and talk time per department
Format	• Table
Axis label	• N/A
Calculation rule	• Outgoing call: ch_direction = 1 • Ring time: call end time - call start time - call talk time
Database tables	• tblcallhistory, tblusers, tbldepartments

Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_ ch_talk_time_seconds, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_department_id, user_extension} • tbldepartments = {department_id, department_name}

SQL Queries

Select outgoing call details by user and departments

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
  CASE WHEN u.user_department_id > 0 THEN u.user_department_id
    ELSE 0 END
    AS department_id,
    SUM ((EXTRACT (EPOCH FROM
     (ch.ch_end_time - ch.ch_start_time))- ch_talk_time_seconds))
       AS totalRingTime,
    SUM (ch_talk_time_seconds) AS totalTalkTime,
    COUNT (ch_call_id) AS nbCalls

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time*/
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */

GROUP BY u.user_firstname, u.user_surname, u.user_extension,
         department_id
ORDER BY u.user_firstname, u.user_surname
```

Select all available departments and the number of users per department having calls in the specified date range

```
(SELECT department_name, department_id,
  COUNT (DISTINCT u.user_id) AS total

  FROM tbldepartments, tblusers u, tblcallhistory h

  WHERE department_id = u.user_department_id
    AND u.user_id = h.ch_user_id
    AND h.ch_start_time >= ? /* from time */
    AND h.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                            /* to date */
    AND h.ch_direction = 1 /* outgoing calls */

  GROUP BY department_name, department_id

  ORDER BY department_name ) /* Benutzer ohne Abteilung */
  UNION (SELECT ' - - - ', 0,
  (SELECT COUNT DISTINCT (u.user_id)
    FROM tblusers u, tblcallhistory h
    WHERE u.user_id = h.ch_user_id
      AND h.ch_start_time >= ? /* from time */
      AND h.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
      AND h.ch_direction = 1 /* outgoing calls */
      AND (u.user_daperatment_id IS null
        OR u.user_department_id = 0)) AS total)
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.25 Outgoing Calls Report – Group Summary

The report shows summary information about the outgoing calls per departments.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Department • Total number of calls per department • Total ring time per department • Total talk time per department • Total number of calls, total ring time and total talk time (all departments)
Format	• Table
Axis label	• N/A
Calculation rule	• Outgoing call: ch_direction = 1 • Ring time: call end time - call start time - call talk time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_talk_time_seconds, ch_direction, ch_user_id} • tblusers = {user_department_id} • tbldepartments = {department_id, department_name}

SQL Queries

Select outgoing calls details (number of call, ring time, talk time) per department

```
SELECT
    (SELECT CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
          END
      FROM tbldepartments
      WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name,
    SUM ((EXTRACT (EPOCH FROM
   (ch.ch_end_time - ch.ch_start_time))- ch_talk_time_seconds))
      AS totalRingTime,
  SUM (ch_talk_time_seconds) AS totalTalkTime,
  COUNT (ch_call_id) AS nbCalls

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */

GROUP BY department_name
ORDER BY department_name
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.26 Outgoing Calls Report - User

The report shows information about all outgoing calls for the specified user in the specified date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Start time • CLI – Called number • Ring time • Talk Time • Daily total number of calls • Daily total ring time • Daily total talk time • Total number of calls • Total talk time
Format	• Table
Axis label	• N/A
Calculation rule	• Outgoing call: ch_direction = 1 • Ring time: end time - start time - talk time
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_ ch_called_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_extension} • tbldepartments = {department_id, department_name}

SQL Queries

Select outgoing call details (call start time, end time, ring time, talk time, called number)

```
SELECT
  "date"(ch.ch_start_time) AS "CallDate",
  "time"(ch.ch_start_time) AS "StartTime",
  "time"(ch.ch_end_time) AS "EndTime",
  (ch.ch_end_time - ch.ch_start_time) AS duration,
   /* length of call*/
  (EXTRACT (EPOCH FROM
   (ch.ch_end_time - ch.ch_start_time))- ch_talk_time_seconds)
   AS ringTime,
  ch.ch_called_number,
  ch_talk_time_seconds

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */
  AND u.user_login = ? /* user login */

ORDER BY ch.ch_start_time
```

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE
    WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select grand totals (total number of calls, total talk time in seconds)

```
SELECT
  SUM (ch.ch_end_time - ch.ch_start_time) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select all available days having outgoing calls for the specified user in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT ("date"(ch.ch_start_time)) AS "CallDate"

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_direction = 1 /* outgoing calls */
  AND ch.ch_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.3.27 Outgoing Calls Report - User Summary

The report shows summary information about the outgoing calls per users.

Required input parameters	• From date • To date (until) • Daily report
Output values	• User first name • User surname • User extension • Total number of calls per user • Total ring time per user • Total talk time per user • Total number of calls, total ring time and total talk time (all users)
Format	• Table
Axis label	• N/A
Calculation rule	• Outgoing call: ch_direction = 1 • Ring time: call end time - call start time - call talk time
Database tables	• tblcallhistory, tblusers
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_end_time, ch_talk_time_seconds, ch_direction, ch_user_id} • tblusers = {user_id, user_firstname, user_surname, user_extension}

SQL Queries

Select outgoing calls details (number of call, ring time, talk time) per users

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
  SUM ((EXTRACT (EPOCH FROM
   (ch.ch_end_time - ch.ch_start_time))- ch_talk_time_seconds))
      AS totalRingTime,
  SUM (ch_talk_time_seconds) AS totalTalkTime,
  COUNT (ch_call_id) AS nbCalls

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
  AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch.ch_user_id = u.user_id
  AND ch.ch_direction = 1 /* outgoing calls */

GROUP BY u.user_firstname, u.user_surname, u.user_extension
ORDER BY u.user_firstname, u.user_surname
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.4 Report Group Calls

All predefined report templates of this report group are described below.

3.4.1 Abandoned Calls Statistics

The report represents details about the abandoned calls by queues.

Required input parameters	• From date • Until (to date) • Daily report
Output values	• Queue • Count • Percentage of abandoned calls • Max queue time (queue time - the amount of time a caller has been waiting to get connected to an agent) • Abandoned percentage of queue – number of abandoned calls per queue and percentage of all abandoned calls for that queue (per queue time: 0-30 s, 31-60 s, 61-90 s, 91-120 s, 121-300 s, 300+ s) • Totals for the columns: count, max queue time and number of calls for all columns showing abandoned calls per queue time interval • Average totals in percents for all columns showing abandoned calls per queue time interval
Format	• Table
Axis label	• N/A

Calculation rule	• Abandoned call = {talk time = 0, callback = 0, agent ID = 0} • Count – abandoned calls by queue: COUNT(abandoned calls by queue) • Percentage of abandoned calls - percentage of abandoned calls from all calls: COUNT/SUM(total number of calls) • Max queue time : MAX(queue time) • Percentage of abandoned calls of the queue 0-30 s: count number of calls: COUNT(abandoned calls by queue where 0 <= queue_time < 31) • Percentage of all abandoned calls by queue : COUNT(abandoned calls by queue where 0<= queue_time< 31) / count-abandoned calls by queue * 100 General: • Percentage of abandoned calls of the queue X-Y s: count number of calls: COUNT(abandoned calls by queue where X <= queue_time < Y) • Percentage of all abandoned calls by queue: COUNT(abandoned calls by queue where X<= queue_time< Y) / count-abandoned calls by queue * 100
Database tables	• tblcallscc, tblcalls, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_time, cc_pickup_time, cc_agent_id, cc_callback, cc_talk_time, cc_queue_id} • tblcalls = {call_id, call_start_time, call_end_time} • tblqueues = {queue_id, queue_name}

SQL Queries

Select all abandoned calls by queue and queue time interval in the selected date range

```
SELECT DISTINCT tblqueues."queue_name",
  COUNT (cc."cc_call_id") AS "count of abandoned calls",
  MAX (cc."cc_queue_time") AS "max queue time",
  MAX (cc."cc_pickup_time") AS "max pickup time",
  SUM (cc."cc_talk_time") AS "talk time",

(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE tblcalls."call_start_time" >= ? /* from time */
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                          '24 hours') /* to date */
    AND tblcallscc."cc_queue_id" = q."queue_id"
    AND tblqueues."queue_name" = q."queue_name"
    AND tblcalls."call_id" = tblcallscc."cc_call_id"
    AND tblcallscc."cc_queue_time" >= 0
    AND tblcallscc."cc_queue_time" < 31
    AND tblcallscc."cc_talk_time" = 0
    AND tblcallscc."cc_callback" = 0
    AND tblcallscc."cc_agent_id" = 0  ) AS "Count0to30",
```

```
(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE tblcalls."call_start_time" >= ? /* from time*/
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                           '24 hours') /* to date */
    AND tblcallscc."cc_queue_id" = q."queue_id"
    AND tblqueues."queue_name" = q."queue_name"
    AND tblcalls."call_id" = tblcallscc."cc_call_id"
    AND tblcallscc."cc_queue_time" > 30
    AND tblcallscc."cc_queue_time" < 61
    AND tblcallscc."cc_talk_time" = 0
    AND tblcallscc."cc_callback" = 0
    AND tblcallscc."cc_agent_id" = 0  ) AS "Count31to60",

(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE tblcalls."call_start_time" >= ? /* from time* /
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                          '24 hours') /* to date */
    AND tblcallscc."cc_queue_id" = q."queue_id"
    AND tblqueues."queue_name" = q."queue_name"
    AND tblcalls."call_id" = tblcallscc."cc_call_id"
    AND tblcallscc."cc_queue_time" > 60
    AND tblcallscc."cc_queue_time" < 91
    AND tblcallscc."cc_talk_time" = 0
    AND tblcallscc."cc_callback" = 0
    AND tblcallscc."cc_agent_id" = 0  ) AS "Count61to90",

(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE tblcalls."call_start_time" >= ? /* from time */
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                           '24 hours') /* to date */
    AND tblcallscc."cc_queue_id" = q."queue_id"
    AND tblqueues."queue_name" = q."queue_name"
    AND tblcalls."call_id" = tblcallscc."cc_call_id"
    AND tblcallscc."cc_queue_time" > 90
    AND tblcallscc."cc_queue_time" < 121
    AND tblcallscc."cc_talk_time" = 0
    AND tblcallscc."cc_callback" = 0
    AND tblcallscc."cc_agent_id" = 0  ) AS "Count91to120",

(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE tblcalls."call_start_time" >= ? /* from time */
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                           '24 hours') /* to date */
    AND tblcallscc."cc_queue_id" = q."queue_id"
    AND tblqueues."queue_name" = q."queue_name"
    AND tblcalls."call_id" = tblcallscc."cc_call_id"
    AND tblcallscc."cc_queue_time" > 120
    AND tblcallscc."cc_queue_time" < 301
    AND tblcallscc."cc_talk_time" = 0
    AND tblcallscc."cc_callback" = 0
    AND tblcallscc."cc_agent_id" = 0  ) AS "Count121to300",
```

```
(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE tblcalls."call_start_time" >= ? /* from time */
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                           '24 hours') /* to date */
    AND tblcallscc."cc_queue_id" = q."queue_id"
    AND tblqueues."queue_name" = q."queue_name"
    AND tblcalls."call_id" = tblcallscc."cc_call_id"
    AND tblcallscc."cc_queue_time" > 300
    AND tblcallscc."cc_talk_time" = 0
    AND tblcallscc."cc_callback" = 0
    AND tblcallscc."cc_agent_id" = 0  ) AS "Count300up",


(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE tblcalls."call_start_time" >= ? /* from time */
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                           '24 hours') /* to date */
    AND tblcallscc."cc_queue_id" = q."queue_id"
    AND tblqueues."queue_name" = q."queue_name"
    AND tblcalls."call_id" = tblcallscc."cc_call_id" )
      AS "Total Calls"

FROM tblcalls, tblcallscc cc, tblqueues

WHERE tblcalls."call_start_time" >= ? /* from time*/
  AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                         '24 hours') /* to date */
  AND tblcalls."call_id" = cc."cc_call_id"
  AND cc."cc_talk_time" = 0
  AND cc."cc_callback" = 0
  AND cc."cc_agent_id" = 0
  AND cc."cc_queue_id" = tblqueues."queue_id"

GROUP BY tblqueues."queue.name"

ORDER BY tblqueues."queue.name"
```

Exception

N/A

3.4.2 Abandoned Calls Statistics – Details

The report represents details about the abandoned calls.

Required input parameters	• From date • Until (to date) • Business hours only (else 24/24) • Daily report
Output values	• Call ID • Call arrived time • Queue • Queue time • Pickup time • CLI (calling number) • Customer Company • Average pickup time • Average queue time
Format	• Table
Axis label	• N/A
Calculation rule (Average : Arithmetic mean)	• Abandoned call: {talk time = 0, agent id = 0, callback = 0} • Average pickup time: SUM(pickup time) / COUNT(pickup time) • Average queue time: SUM (queue time) / COUNT(queue time) • Total average: Average for all queues (average: arithmetic mean) • Total average pickup time: SUM(average pickup time) / COUNT(average pickup time) • Total average queue time: SUM(average queue time) / COUNT(average queue time) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches, tblcustomers
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_time, cc_pickup_time, cc_agent_id, cc_callback, cc_talk_time, cc_queue_id} • tblcalls = {call_id, call_start_time, call_end_time, call_calling_number} • tblqueues = {queue_id, queue_name} • tblswitches = {switch_office_start, switch_office_end} • tblcustomers= {customer_business, customer_business2, customer_home, customer_mobile, customer_company}

SQL Queries

Select all abandoned calls in the selected date range

```
SELECT DISTINCT cc."cc_call_id",
    tblcalls."call_start_time",
    tblqueues.queue_name,
    cc."cc_queue_time",
    cc."cc_pickup_time",
    tblcalls."call_calling_number",
    (SELECT tblcustomers."customer_company" FROM tblcustomers
       WHERE tblcalls."call_calling_number" IN
        (tblcustomers."customer_business",
         tblcustomers."customer_business2",
         tblcustomers."customer_home",
         tblcustomers."customer_mobile"
      )) AS customer_company

FROM tblcallscc cc, tblcalls, tblqueues, tblswitches s

WHERE tblcalls."call_start_time" >= ? /* from time */
  AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                          '24 hours') /* to date */
  AND cc."cc_call_id" = tblcalls."call_id"
  AND cc."cc_talk_time" = 0
  AND cc."cc_agent_id" = 0
  AND cc."cc_callback" = 0
  AND cc."cc_queue_id" = tblqueues.queue_id
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
       "time"(tblcalls."call_start_time") >=
                         "time"(s.switch_office_start) AND
       "time"(tblcalls."call_start_time") <=
                           "time"(s.switch_office_end)
    WHEN ? = 1 THEN /* Not Business hours only = 24/24 */
     "time"(tblcalls."call_start_time") >= '00:00:00' AND
     "time"(tblcalls."call_start_time") <= '23:59:59
       END )
  AND tblcalls."call_calling_number" <> ''

ORDER BY tblcalls."call_start_time"
```

Exception

N/A

3.4.3 Answered Calls Alert Times

The report represents the alert times of answered calls for the selected agent in the specified date range.

Required input parameters	• From date • Until (to date) • Agent • Business hours only (else 24/24) • Daily report
Output values	• Day • Alert time (call pickup time) - daily • Percentage of total alert time - daily • Total alert time
Format	• Table and graphic
Axis label	• Horizontal: Days • Vertical: Alert time (call pickup time) in seconds
Calculation rule	• Answered calls : {talk time > 0} • Total alert time: SUM(alert time) • Percentage of total alert time: daily alert time (call pickup time) / total alert time * 100 • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblswitches, tbldepartments, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time, cc_pickup_time, cc_agent_id} • tblcalls = {call_id, call_start_time, call_end_time} • tblswitches = {switch_office_start, switch_office_end} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
       THEN '' ELSE tbldepartments.department_name
        END
        FROM tbldepartments
        WHERE u.user_department_id = tbldepartments.department_id
     ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select daily alert time (call pickup time) of answered calls

```
SELECT "date"(c.call_start_time) AS "Date of call",
      SUM (cc.cc_pickup_time) AS "Total"

FROM tblcallscc cc, tblcalls c, tblusers u, tblswitches s

WHERE c.call_start_time >= ? /* from time */
  AND c.call_start_time <= ("date"(?) + INTERVAL '24 hours')
                            /* to date */
  AND cc.cc_talk_time > 0
   AND cc.cc_call_id = c.call_id
  AND cc.cc_agent_id = u.user_id
  AND u.user_login = ? /* agent login */
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
  "time"(c.call_start_time) >= "time"(s.switch_office_start) AND
  "time"(c.call_start_time) <= "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
  "time"(c.call_start_time) >= '00:00:00'AND
  "time"(c.call_start_time) <= '23:59:59'
   END )

GROUP BY "date"(c.call_start_time)
ORDER BY "date"(c.call_start_time)
```

Exception

1. A maximum of 15 agents (vertical tubes in the tube chart) can be shown in a graphic. If there are more than 15 agents, the graphic will not be displayed because with more than 15 agents the graphic is not properly visible.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used:

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • sec. = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.4.4 Answered Calls Alert Times (All Agents)

The report represents the alert times of answered calls for all agents in the selected date range.

Required input parameters	• From date • Until (to date) • Business hours only (else 24/24) • Daily report
Output values	• Agent • Alert time (call pickup time) - by agent • Percentage of total alert time - by agent • Total alert time
Format	• Table and graphic
Axis label	• Horizontal: agents • Vertical: alert time (call pickup time) in seconds

Calculation rule	• Answered calls : {talk time > 0} • Total alert time: SUM(alert time) • Percentage of total alert time: alert time (call pickup time) by agent / total alert time * 100 • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblswitches, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time, cc_pickup_time, cc_agent_id} • tblcalls = {call_id, call_start_time, call_end_time} • tblswitches = {switch_office_start, switch_office_end} • tblusers = {user_id, user_login, user_firstname, user_surname, user_is_agent}

SQL Queries

Select daily alert time (call pickup time) of answered calls

```
SELECT SUM (cc.cc_pickup_time) AS "Total",
       u.user_firstname,
       u.user_surname,
       u.user_login

FROM tblcallscc cc, tblcalls c, tblusers u, tblswitches s

WHERE c.call_start_time >= ? /* from time */
  AND c.call_start_time <= ("date"(?) + INTERVAL '24 hours')
                            /* to date */
  AND cc.cc_talk_time > 0
   AND cc.cc_call_id = c.call_id
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
  "time"(c.call_start_time) >= "time"(s.switch_office_start) AND
  "time"(c.call_start_time) <= "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
     "time"(c.call_start_time) >= '00:00:00'AND
     "time"(c.call_start_time) <= '23:59:59'
    END )
    AND cc.cc_agent_id = u.user_id
    AND u.user_is_agent = 1

GROUP BY u.user_firstname, u.user_surname, u.user_login
```

Exception

1. A maximum of 15 agents (vertical tubes in the tube chart) can be shown in a graphic. If there are more than 15 agents, the graphic will not be displayed because with more than 15 agents the graphic is not properly visible.
2. To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used:

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • sec. = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.4.5 Answered Calls Alert Times – Details

The report represents the alert times of answered calls for the selected agent in the specified date range.

Required input parameters	• From date • Until (to date) • Agent • Business hours only (else 24/24) • Daily report
Output values	• Time of call • End of call waiting - time when call is answered • Alert time (call pickup time) • Daily total alert time • Total alert time
Format	• Table
Axis label	• N/A

Calculation rule	• Answered calls: {talk time > 0} • Daily total alert time: SUM(alert time per specific day) • Total alert time: SUM(alert time) • End of call waiting: call_start_time + call_pickup_time • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblswitches, tbldepartments, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time, cc_pickup_time, cc_agent_id} • tblcalls = {call_id, call_start_time, call_end_time} • tblswitches = {switch_office_start, switch_office_end} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available different days having answered calls for the selected agent in the selected date range

```
SELECT DISTINCT ("date"(c.call_start_time)) AS "Date of Day"

FROM tblcallscc cc, tblcalls c, tblusers u, tblswitches s

WHERE c.call_start_time >= ? /* from time */
    AND c.call_start_time <= ("date"(?) + INTERVAL '24 hours')
                             /* to date */
    AND cc.cc_talk_time > 0
    AND cc.cc_call_id = c.call_id
    AND cc.cc_agent_id = u.user_id
    AND u.user_login = ? /* agent login */
    AND (CASE WHEN ? = 1 THEN /* Business hours only */
    "time"(c.call_start_time) >= "time"(s.switch_office_start)
       AND
    "time"(c.call_start_time) <= "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(c.call_start_time) >= '00:00:00'
    AND "time"(c.call_start_time) <= '23:59:59'
    END )
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
       THEN '' ELSE tbldepartments.department_name
       END
       FROM tbldepartments
       WHERE u.user_department_id = tbldepartments.department_id
     ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select total alert time (call pickup time) of all answered calls

```
SELECT SUM (cc.cc_pickup_time) AS "Total Waiting Time"

FROM tblcallscc cc, tblcalls c, tblusers u, tblswitches s

WHERE c.call_start_time >= ? /* from time */
  AND c.call_start_time <= ("date"(?) + INTERVAL '24 hours')
                            /* to date */
  AND cc.cc_talk_time > 0
  AND cc.cc_call_id = c.call_id
  AND cc.cc_agent_id = u.user_id
  AND u.user_login = ? /* agent login */
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
  "time"(c.call_start_time) >= "time"(s.switch_office_start) AND
  "time"(c.call_start_time) <= "time"(s.switch_office_end)
   WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
  "time"(c.call_start_time) >= '00:00:00' AND
  "time"(c.call_start_time) <= '23:59:59'
  END )
```

Select alert time (call pickup time) of answered call

```
SELECT cc.cc_call_id,
      "date"(c.call_start_time) AS "Date of call",
      "time"(c.call_start_time) AS "Start time",
      "time"(c.call_start_time) + INTERVAL"(cc.cc_pickup_time ||
                   'seconds') AS "End time",
       cc.cc_pickup_time

FROM tblcallscc cc, tblcalls c, tblusers u, tblswitches s

WHERE c.call_start_time >= ? /* from time */
  AND c.call_start_time <= ("date"(?) + INTERVAL '24 hours')
                             /* to date */
  AND cc.cc_talk_time > 0
  AND cc.cc_call_id = c.call_id
  AND cc.cc_agent_id = u.user_id
  AND u.user_login = ? /* agent login */
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
  "time"(c.call_start_time) >= "time"(s.switch_office_start) AND
  "time"(c.call_start_time) <= "time"(s.switch_office_end)
  WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
  "time"(c.call_start_time) >= '00:00:00'AND
  "time"(c.call_start_time) <= '23:59:59'
  END )

ORDER BY "time"(c.call_start_time)
```

Exception

N/A

3.4.6 Answered Calls Statistics

The report represents details about the answered calls by queues.

Required input parameters	• From date • Until (to date) • From time • To time • Daily report
Output values	• Queue • Count • Percentage of answered calls • Max queue time (queue time - the amount of time a caller has been waiting to get connected to an agent) • Answered percentage of queue – number of answered calls per queue and percentage of all answered calls for that queue (per queue time : 0-30 s, 31-60 s, 61-90 s, 91-120 s, 121-300 s, 300+ s) • Totals for the columns: count, max queue time and number of calls for all columns showing answered calls per queue time interval • Average totals in percents for all columns showing answered calls per queue time interval
Format	• Table
Axis label	• N/A
Calculation rule	• Answered call = {talk time > 0} • Count – answered calls by queue : COUNT(answered calls by queue) • Percentage of answered calls - percentage of answered calls from all calls: COUNT/SUM(total number of calls) • Max queue time: MAX(queue time) • Percentage of answered calls of the queue 0-30 s: Count number of calls: COUNT(answered calls by queue where 0 <= queue_time < 31) • Percentage of all answered calls by queue: COUNT(answered calls by queue where 0<= queue_time< 31) / Count-answered calls by queue * 100 General: • Percentage of answered calls of the queue X-Y s: count number of calls: COUNT(answered calls by queue where X <= queue_time < Y) • Percentage of all answered calls by queue : COUNT(answered calls by queue where X<= queue_time < Y) / count-answered calls by queue * 100
Database tables	• tblcallscc, tblcalls, tblqueues

Database table attributes	• tblcallscc = {cc_call_id, cc_queue_time, cc_pickup_time, cc_agent_id, cc_callback, cc_talk_time, cc_queue_id} • tblcalls = {call_id, call_start_time, call_end_time} • tblqueues = {queue_id, queue_name}

SQL Queries

Select all answered calls by queue and queue time interval in the selected date range

```
SELECT DISTINCT tblqueues."queue_name",
  COUNT (cc."cc_call_id") AS "count of answered calls",
  MAX (cc."cc_queue_time") AS "max queue time",
  MAX (cc."cc_pickup_time") AS "max pickup time",
  SUM (cc."cc_talk_time") AS "talk time",

(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
          "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcallscc."cc_queue_id" = q."queue_id"
  AND tblqueues."queue_name" = q."queue_name"
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_queue_time" >= 0
  AND tblcallscc."cc_queue_time" < 31
  AND tblcallscc."cc_talk_time" > 0 ) AS "Count0to30",

(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE "date"(tblcalls."call_start_time") >= ? /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
          "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcallscc."cc_queue_id" = q."queue_id"
  AND tblqueues."queue_name" = q."queue_name"
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_queue_time" > 30
  AND tblcallscc."cc_queue_time" < 61
  AND tblcallscc."cc_talk_time" > 0 ) AS "Count31to60",
```

```
(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE "date"(tblcalls."call_start_time") >= ? /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcallscc."cc_queue_id" = q."queue_id"
  AND tblqueues."queue_name" = q."queue_name"
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_queue_time" > 60
  AND tblcallscc."cc_queue_time" < 91
  AND tblcallscc."cc_talk_time" > 0  ) AS "Count61to90",

(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE "date"(tblcalls."call_start_time") >= ? /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcallscc."cc_queue_id" = q."queue_id"
  AND tblqueues."queue_name" = q."queue_name"
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_queue_time" > 90
  AND tblcallscc."cc_queue_time" < 121
  AND tblcallscc."cc_talk_time" > 0 ) AS "Count91to120",

(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE "date"(tblcalls."call_start_time") >= ? /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcallscc."cc_queue_id" = q."queue_id"
  AND tblqueues."queue_name" = q."queue_name"
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_queue_time" > 120
  AND tblcallscc."cc_queue_time" < 301
  AND tblcallscc."cc_talk_time" > 0  ) AS "Count121to300",
```

```
(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE "date"(tblcalls."call_start_time") >= ? /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcallscc."cc_queue_id" = q."queue_id"
  AND tblqueues."queue_name" = q."queue_name"
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_queue_time" > 300
  AND tblcallscc."cc_talk_time" > 0  ) AS "Count300up",

(SELECT COUNT (tblcallscc."cc_call_id")
  FROM tblcalls, tblcallscc, tblqueues q
  WHERE "date"(tblcalls."call_start_time") >= ? /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcallscc."cc_queue_id" = q."queue_id"
  AND tblqueues."queue_name" = q."queue_name"
  AND tblcalls."call_id" = tblcallscc."cc_call_id" )
    AS "Total Calls"

FROM tblcalls, tblcallscc cc, tblqueues

WHERE "date"(tblcalls."call_start_time") >= ? /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcalls."call_id" = cc."cc_call_id"
  AND cc."cc_talk_time" > 0
  AND cc."cc_queue_id" = tblqueues."queues_id"

GROUP BY tblqueues."queue.name"

ORDER BY tblqueues."queue.name"
```

Exception

N/A

3.4.7 Answered Calls Wrap-up Information

The report displays details including wrap-up information for answered calls in the selected date range.

Required input parameters	• From date • Until (to date) • From time • To time • Business hours only (else 24/24) • Daily report
Output values (the values are grouped daily)	• Call ID • Arrived at • Queue • Agent login • Wrap-up • CLI - calling number • Daily total number of calls
Format	• Table
Axis label	• N/A
Calculation rule	• Answered calls : {talk time > 0} • Daily total number of calls: COUNT(number of calls per day) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblswitches, tblusers, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id, cc_talk_time, cc_agent_id} • tblcalls = {call_id, call_start_time, call_end_time} • tblswitches = {switch_office_start, switch_office_end} • tblusers = {user_id, user_login} • tblqueues = {queue_name, queue_id}

SQL Queries

Select all available different days having answered calls in the selected date range

```
SELECT DISTINCT "date"(tblcalls."call_start_time")
  AS date_of_call

FROM tblcallscc cc, tblcalls c, tblswitches s

WHERE
   "date"(tblcalls."call_start_time") >= "date"(?) /* from date */
  AND "date"(tblcalls."call_start_time") <=
           "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND cc."cc_call_id" = tblcalls."call_id"
  AND cc."cc_talk_time" > 0
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
          "time"(tblcalls."call_start_time") >=
                                "time"(s.switch_office_start)
       AND "time"(tblcalls."call_start_time") <=
                                "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00'
  AND "time"(tblcalls."call_start_time") <= '23:59:59'
    END )

ORDER BY "date"(tblcalls."call_start_time")
```

Select answered call details

```
SELECT cc."cc_call_id",
      "time"(tblcalls."call_start_time") AS call_start_time,
       tblqueues.queue_name,
       u."user_login",
       cc."cc_agent_id",
       tblcalls."call_calling_number",
       CASE WHEN cc."cc_call_id" IN
              (SELECT ccw.ccw_cc_id FROM tblccwrapups ccw)
         THEN GETWRAPUP(cc."cc_call_id")
         ELSE ' - - - ' END AS wrapup_description,
      "date"(tblcalls."call_start_time")

FROM tblcallscc cc, tblcalls, tblusers u, tblqueues, tblswitches s

WHERE
   "date"(tblcalls."call_start_time") >= "date"(?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND cc."cc_call_id" = tblcalls."call_id"
  AND cc."cc_agent_id" = u."user_id"
  AND cc."cc_talk_time" > 0
  AND cc."cc_queue_id" = tblqueues.queue_id
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
          "time"(tblcalls."call_start_time") >=
                                  "time"(s.switch_office_start)
       AND "time"(tblcalls."call_start_time") <=
                                  "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00'
  AND "time"(tblcalls."call_start_time") <= '23:59:59'
    END)

ORDER BY tblcalls."call_start_time"
```

Exception

For getting the wrap-up details in the SQL query above is used a function called getwrapup(CALL_ID).

The result of this function is:

- Wrap-up description 1
  Wrap-up parent 1.1
  . . .
  Wrap-up parent 1.X
  Wrap-up top parent 1

- Wrap-up description 2
  Wrap-up parent 2.1
  . . .
  Wrap-up parent 2.X
  Wrap-up top parent 2
- Wrap-up description 3 (wrap-up without parent wrap-up)

For more details please see the functions below.

Function: getwrapup(integer)

```
CREATE OR REPLACE FUNCTION getwrapup(integer) RETURNS text AS$BODY$
DECLARE
      mviews RECORD;
      res text;BEGIN
    --Wrap up Item
     res='';
     FOR mviews IN SELECT ( wcc.wrapup_description || '
     ' ||getWrapupGroups(ccw.ccw_wc_id)) as descript
      FROM tblwrapupcc wcc, tblccwrapups ccw
      where ccw.ccw_cc_id = $1
      and ccw.ccw_wc_id = wcc."wrapup_code" loop
      res = res || mviews.descript || '
     ';
     End Loop;
     return res; END
  $BODY$
  LANGUAGE 'plpgsql' VOLATILE; ALTER FUNCTION getwrapup(integer)
OWNER TO postgres;
```

Function: getwrapupgroups(bigint)

```
-- DROP FUNCTION getwrapupgroups(bigint); CREATE OR REPLACE
FUNCTION getwrapupgroups(bigint)
  RETURNS text AS $BODY$   DECLARE
  res text;
  space text;
  descript text;
  wgid bigint;
  wgidnew bigint;
  i int;
  max_s int; BEGIN
  res='';
  max_s=6;
  wgid=$1;
  select wrapup_parent_id into wgid
    from tblwrapupcc
    where wrapup_code=$1 limit 1;
  while wgid <> 0 loop
    select wg.wg_parent_id, wg."wg_caption"
      INTO wgidnew,
  descript
      from tblwrapupgroups wg
      where wg.wg_id = wgid
      limit 1;
      i=0;
      space='';
      while i<max_s loop
                i=i+1;
                space=space || ' ';
      end loop;
      res=res || space || descript || '\n' ;
      wgid=wgidnew;
      max_s=max_s+6;
  end loop;
  return res; END   $BODY$ LANGUAGE 'plpgsql' VOLATILE; ALTER
FUNCTION getwrapupgroups(bigint) OWNER TO postgres;
```

3.4.8 Call Traffic All Agents – Per Hour Daily

Count of calls (call center calls, direct calls, outbound, inbound) and talk time by agents (for all available agents having calls in the specified date range).

INFO: The report template **Call Traffic All Agents - Per Hour Daily-Details** returns the following additional output values: `Talk Time CC Calls`, `Talk Time Direct Calls`

Required input parameters	• From date • To date (until) • Daily report
Output values (the values are grouped per user/agents, per hour and daily)	• Day • User/Agent • Time – hourly interval (e.g.: 09:00-10:00) • CC calls (number of contact center calls) • Direct calls (number of direct calls) • Inbound calls (number of inbound/incoming calls) • Outbound calls (number of outbound/outgoing calls) • All calls (number of all calls = cc calls + direct calls) • Talk time - all calls (Total Talk time for the specified hourly interval) • Daily Totals and Grand Totals (CC calls, direct calls, inbound, outbound, all calls, talk time)
Format	• Table
Axis label	• N/A

Calculation rule	• One user is agent when the user_is_agent (tblusers) is set to 1 • Inbound calls: ch_direction = 0: sum (case ch_direction when 0 then 1 else 0 end) • Outbound calls: ch_direction = 1: sum (ch_direction) • Talk time: SUM(ch_talk_time_seconds) • CC calls (in tblcallhistory): if the same ch_local_id and date (ch_start_time) exists in tblcalls (call_local_id and date (call_start_time)) • Number of CC calls = sum (case when ch_local_id||'_'|| date(ch_start_time) IN (select call_local_id||'_'||date(call_start_time) from tblcalls) then 1 else 0 end) • Direct calls: if the ch_local_id and date (ch_start_time) NOT exists in tblcalls (call_local_id and date (call_start_time)) • Number of Direct Calls (NOT CC calls) = sum (case when ch_local_id||'_'|| date(ch_start_time) NOT IN (select call_local_id||'_'|| date(call_start_time) from tblcalls) then 1 else 0 end) • The BIRT function Total.sum(<row>) is used for calculating the daily totals
Database tables	• tblcallhistory, tblusers
Database table attributes	• tblcallhistory = {ch_call_id,ch_start_time, ch_talk_time_seconds, ch_direction, ch_user_id, ch_local_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_is_agent} • tblcalls = {call_start_time, call_local_id}

SQL Queries

Select call traffic per agent and per hour daily

```
SELECT "date" (t1.ch_start_time),
  user_login,
  EXTRACT (HOUR FROM t1.ch_start_time)||':00 - '||
  (EXTRACT (HOUR FROM t1.ch_start_time)+1)||':00'
  AS "label" ,
  COUNT (DISTINCT t1.id) AS "All Calls",
  SUM (CASE WHEN t1.id IN
    (SELECT call_local_id||'_'|| DATE (call_start_time)
        FROM tblcalls ) THEN 1 ELSE 0 END) AS "CC calls",
-- # NOT CC Calls = ( "# all calls" - "# cc calls" ) in the report
-- OR can be calculated here :
  SUM (CASE WHEN t1.id NOT IN
    (SELECT call_local_id||'_'|| DATE (call_start_time)
        FROM tblcalls ) THEN 1 ELSE 0 END) AS "Direct calls",
  SUM ( t1.ch_direction ) AS "Outgoing Calls",
-- # Incoming Calls = ("# all calls" - "# outgoing calls")
in the report-- OR can be calculated here :
  SUM (CASE t1.ch_direction WHEN 0 THEN 1 ELSE 0 END)
                    AS "Incoming Calls",
  SUM (t1.ch_talk_time_seconds) AS "Talk Time All Calls"
--"time" (TIMESTAMP 'epoch' + (sum (ch_talk_time_seconds))
-- * INTERVAL '1 second')

FROM tblusers,
  (SELECT DISTINCT ON (ch_local_id||'_'|| DATE (ch_start_time))
    ch_local_id||'_'|| DATE (ch_start_time) AS id,
    ch_start_time,ch_call_id,ch_local_id,ch_direction,
    ch_talk_time_seconds,ch_user_id from tblcallhistory)
    AS t1
  WHERE t1."ch_start_time" >= ? /* from time */
    AND t1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours') /
      *to date*/
    AND t1."ch_user_id" = user_id
    AND user_is_agent=1

GROUP BY user_login, EXTRACT (HOUR FROM t1."ch_start_time"),
  "date" (t1."ch_start_time")

ORDER BY EXTRACT(HOUR FROM t1."ch_start_time")
```

Select available agents (used for grouping the information by agents)

```
SELECT DISTINCT user_login, user_firstname, user_surname,
          "date"(ch_start_time)

FROM tblusers, tblcallhistory

WHERE "ch_start_time" >= ? /* from date */
      AND "ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
                     /* to date */
      AND "ch_user_id" = user_id
      AND user_is_agent= 1

ORDER BY user_login, "date"("ch.start_time")
```

Select grand totals (total number of calls, total number of call center calls, direct calls, incoming calls,outgoing calls and total talk time)

```
SELECT
  COUNT (DISTINCT t1.id) AS "All Calls",
  SUM (CASE WHEN t1.id IN (SELECT call_local_id||'_'||    DATE
(call_start_time) FROM tblcalls )
         THEN 1 ELSE 0 END) AS "CC calls",
then 1 else 0 end) as "CC calls",
-- # NOT CC Calls = ( "# all calls" - "# cc calls" ) in the report
-- OR can be calculated here :
  SUM (CASE WHEN t1.id NOT IN
    (SELECT call_local_id||'_'|| DATE (call_start_time)
    FROM tblcalls ) THEN 1 ELSE 0 END) AS "Not CC calls",
    SUM ( t1.ch_direction ) AS "Outgoing calls",
-- # Incoming Calls = ("# all calls" - "# outgoing calls")
-- in the report OR can be calculated here :
  SUM (CASE t1.ch_direction WHEN 0 THEN 1 ELSE 0 END)
                            AS "Incoming calls",
  SUM (t1.ch_talk_time_seconds) AS "TimeInSeconds"

FROM (SELECT DISTINCT ON (ch_local_id||'_'||   DATE
(ch_start_time)) ch_local_id||'_'||
  DATE (ch_start_time) AS id, ch_start_time,
    ch_call_id,ch_local_id, ch_direction, ch_talk_time_seconds,
    ch_user_id FROM tblcallhistory) AS t1
WHERE t1."ch_start_time" >= ? /* From date */
  AND t1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND t1.ch_user_id IN (SELECT user_id FROM tblusers
  WHERE user_is_agent=1)
```

Select all available days having calls in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT "date"("ch_start_time")
  SUM (ch_talk_time_seconds) AS "TalkTimeDaily"

FROM tblcallhistory

WHERE "ch_start_time" >= ? /* from date */
  AND "ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND ch_user_id IN ( SELECT user_id FROM tblusers
  WHERE user_is_agent=1)

GROUP BY "date"("ch.start_time")
ORDER BY "date"("ch.start_time")
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • sec. = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

One example in the BIRT Expression Builder:

```
ts = row["TimeInSeconds"];
day =  (ts/86400) | 0 ;
hour = ((ts-(day*86400))/3600) | 0;
min  =  ((ts-(day*86400) - (hour*3600))/60) | 0;
sec =  (ts-(day*86400) - (hour*3600) - (min*60));
if(day == 0)day = "";
else day = day+" day(s) ";
if(hour<10)hour = "0"+hour;
if(min<10)min = "0"+min;
if(min<10)min = "0"+min;
if(sec<10)sec = "0"+sec;
day+""+hour+":"+min+":"+sec
```

Output example 1 : 2 day(s) 02:15:26

Output example 2 : 10:39:07

3.4.9 Call Traffic All Agents – Per Hour Daily – Details

Count of calls (call center calls, direct calls, outbound, inbound) and talk time (for cc calls, direct calls and all calls) by agents, for all available agents having calls in the specified date range.

Required input parameters	• From date • To date (until) • Daily report
Output values (the values are grouped per user/agents, per hour and daily)	• Day • User/Agent • Time – hourly interval (e.g.: 09:00-10:00) • CC calls (number of contact center calls) • Talk time CC calls (CC calls: total talk time for the specified hourly interval) • Direct calls (number of direct calls) • Talk time direct calls (direct calls: total talk time for the specified hourly interval) • Inbound calls (number of inbound/incoming calls) • Outbound calls (number of outbound/outgoing calls) • All calls (number of all calls = cc calls + direct calls) • Talk time - all calls (total talk time for the specified hourly interval) • Daily totals and grand totals (CC calls, Direct calls, Inbound, Outbound, All calls and talk time of CC calls, direct calls and all calls)
Format	• Table
Axis label	• N/A

Calculation rule	• One user is agent when the user_is_agent (tblusers) is set to 1 • Inbound calls: ch_direction = 0: sum (case ch_direction when 0 then 1 else 0 end) • Outbound calls: ch_direction = 1: sum (ch_direction) • Total talk time: sum (ch_talk_time_seconds) • CC calls (in tblcallhistory): if the same ch_local_id and date (ch_start_time) exists in tblcalls (call_local_id and date (call_start_time)) • Number of CC calls = sum (case when ch_local_id||'_'|| date (ch_start_time) IN (select call_local_id||'_'||date(call_start_time) from tblcalls) then 1 else 0 end) • Direct calls: if ch_local_id and date(ch_start_time) NOT exists in tblcalls (call_local_id and date(call_start_time)) • Number of Direct Calls (NOT CC calls) = sum (case when ch_local_id||'_'|| date(ch_start_time) NOT IN (select call_local_id||'_'|| date(call_start_time) from tblcalls) then 1 else 0 end) • Direct calls talk time: summation of the calls talk time for call records where ch_local_id and date (ch_start_time) NOT exists in tblcalls (call_local_id and date (call_start_time)) • Direct calls talk time = sum (case when ch_local_id||'_'|| date(ch_start_time) NOT IN (select call_local_id||'_'|| date(call_start_time) from tblcalls) then ch_talk_time_seconds else 0 end) • CC calls talk time (in tblcallhistory) : summation of the calls talk time for call records, where ch_local_id und date(ch_start_time) exists in tblcalls (call_local_id and date(call_start_time)) • CC calls talk time = sum (case when ch_local_id||'_'|| date(ch_start_time) IN (select call_local_id||'_'|| date(call_start_time) from tblcalls) then ch_talk_time_seconds else 0 end) • The BIRT function Total.sum(<row>) is used for calculating the daily totals
Database tables	• tblcallhistory, tblusers
Database table attributes	• tblcallhistory = {ch_call_id,ch_start_time, ch_talk_time_seconds, ch_direction, ch_user_id, ch_local_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_is_agent} • tblcalls = {call_start_time, call_local_id}

SQL Queries

Select call traffic details grouped per agent and per hour daily

```
SELECT "date" (t1.ch_start_time),user_login,
  EXTRACT (HOUR FROM t1.ch_start_time)||':00 - '||
  (EXTRACT (HOUR FROM t1.ch_start_time)+1)||':00'
  AS "label" ,
  COUNT (DISTINCT t1.id) AS "All Calls",
  SUM (CASE WHEN t1.id IN
    (SELECT call_local_id||'_'||DATE(call_start_time)
         FROM tblcalls )THEN 1 ELSE 0 END) AS "CC calls",
-- # NOT CC Calls = ( "# all calls" - "# cc calls" ) in the report
-- OR can be calculated here :
  SUM (CASE WHEN t1.id NOT IN
     (SELECT call_local_id||'_'||DATE(call_start_time)
         FROM tblcalls )
         THEN 1 ELSE 0 END) AS "Direct calls",
  SUM (CASE WHEN t1.id IN
     (SELECT call_local_id||'_'||date(call_start_time)
         FROM tblcalls )
         THEN t1.ch_talk_time_seconds ELSE 0 END)
      AS "Talk Time CC Calls",
  SUM (CASE WHEN t1.id NOT IN
     (SELECT call_local_id||'_'||DATE(call_start_time)
         FROM tblcalls )
         THEN t1.ch_talk_time_seconds ELSE 0 END)
      AS "Talk Time Direct Calls",
  SUM ( t1.ch_direction ) AS "Outgoing Calls",
-- # Incoming Calls = ("# all calls" - "# outgoing calls")
-- in the report -- OR can be calculated here :
  SUM (CASE t1.ch_direction WHEN 0 THEN 1 ELSE 0 END)
      AS "Incoming Calls",
  SUM (t1.ch_talk_time_seconds) AS "Talk Time All Calls"
-- "time" (TIMESTAMP 'epoch' + (sum (ch_talk_time_seconds))
-- * INTERVAL '1 second')

FROM tblusers,
  (SELECT DISTINCT ON (ch_local_id||'_'|| DATE (ch_start_time))
    ch_local_id||'_'|| DATE (ch_start_time) AS id,
    ch_start_time,ch_call_id,ch_local_id,ch_direction,
    ch_talk_time_seconds,ch_user_id FROM tblcallhistory)
    AS t1
  WHERE t1."ch_start_time" >= ? /* from time */
    AND t1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours') /
      *to date*/
    AND t1."ch_user_id" = user_id
    AND user_is_agent=1

GROUP BY user_login, EXTRACT (HOUR FROM t1."ch_start_time"),
  "date" (t1."ch_start_time")

ORDER BY EXTRACT(HOUR FROM t1."ch_start_time")
```

Select available agents (used for grouping the information by agents)

```
SELECT DISTINCT user_login, user_firstname, user_surname,
          "date"(ch_start_time)

FROM tblusers, tblcallhistory

WHERE "ch_start_time" >= ? /* from date */
      AND "ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
                    /* to date */
      AND "ch_user_id" = user_id
      AND user_is_agent= 1

ORDER BY user_login, "date"("ch.start_time")
```

Select grand totals (total number of calls, total number of call center calls, direct calls, incoming calls, outgoing calls, total talk time, cc calls talk time and direct calls talk time)

```
SELECT
  COUNT (DISTINCT t1.id) AS "All Calls",
  SUM (CASE WHEN t1.id IN
     (SELECT call_local_id||'_'|| DATE (call_start_time)
          FROM tblcalls )
          THEN 1 ELSE 0 END) AS "CC calls",
-- # NOT CC Calls = ( "# all calls" - "# cc calls" ) in the report
-- OR can be calculated here :
  SUM (CASE WHEN t1.id NOT IN
     (SELECT call_local_id||'_'|| DATE (call_start_time)
          FROM tblcalls )
          THEN 1 ELSE 0 END) AS "NOT CC calls",
 SUM (CASE WHEN t1.id IN
     (SELECT call_local_id||'_'|| DATE (call_start_time)
          FROM tblcalls )
          THEN t1.ch_talk_time_seconds ELSE 0 END)
          AS "Talk Time CC Calls",
  SUM (CASE WHEN t1.id NOT IN
     (SELECT call_local_id||'_'|| DATE (call_start_time)
          FROM tblcalls )
          THEN t1.ch_talk_time_seconds ELSE 0 END)
          AS "Talk Time Direct Calls"",
  SUM (t1.ch_direction) AS "Outgoing Calls",
-- # Incoming Calls = ("# all calls" - "# outgoing calls")
--in the report OR can be calculated here :
  SUM (CASE t1.ch_direction WHEN 0 THEN 1 ELSE 0 END)
      AS "Incoming calls",
  SUM (t1.ch_talk_time_seconds) AS "TimeInSeconds"
```

```
FROM (SELECT DISTINCT ON (ch_local_id||'_'|| DATE (ch_start_time))
    ch_local_id||'_'|| DATE (ch_start_time)
  AS id, ch_start_time, ch_call_id, ch_local_id, ch_direction,
    ch_talk_time_seconds, ch_user_id
  FROM tblcallhistory) AS t1 WHERE t1."ch_start_time" >= ?
    /* from time */
  AND t1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
    /*to date*/
  AND t1.ch_user_id IN (SELECT user_id FROM tblusers
WHERE user_is_agent=1)
```

Select all available days having calls in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT "date"("ch_start_time")
  SUM (ch_talk_time_seconds) AS "TalkTimeDaily"

FROM tblcallhistory

WHERE "ch_start_time" >= ? /* from date */
  AND "ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
        /* to date */
  AND ch_user_id IN ( SELECT user_id FROM tblusers
  WHERE user_is_agent=1)

GROUP BY "date"("ch.start_time")

ORDER BY "date"("ch.start_time")
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • sec. = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

One example in the BIRT Expression Builder:

```
ts = row["TimeInSeconds"];
day =  (ts/86400) | 0 ;
hour = ((ts-(day*86400))/3600) | 0;
min  =  ((ts-(day*86400) - (hour*3600))/60) | 0;
sec =  (ts-(day*86400) - (hour*3600) - (min*60));
if(day == 0)day = "";
else day = day+" day(s) ";
if(hour<10)hour = "0"+hour;
if(min<10)min = "0"+min;
if(min<10)min = "0"+min;
if(sec<10)sec = "0"+sec;
day+""+hour+":"+min+":"+sec
```

Output example 1 : 2 day(s) 02:15:26

Output example 2 : 10:39:07

3.4.10 Call Traffic All Queues – Per Hour (Daily)

Count of calls (all calls, answered calls and abandoned calls) for all available queues having calls in the specified date range.

Required input parameters	• From date • To date (until) • Daily report
Output values (the values are grouped per queue, per hour and daily)	• Day • Queue • Time – hourly interval (e.g.: 09:00-10:00) • All calls (number of calls per hour daily) • Answered calls (number of answered calls ...) • Abandoned calls (number of abandoned calls ...) • Daily Totals per queue and Grand Totals (all calls, answered calls and abandoned calls)
Format	• Table
Axis label	• N/A
Calculation rule	• Answered calls: {cc_talk time> 0} • Abandoned calls: {cc_talk time = 0, cc_agent_id = 0, cc_callback = 0} • Number of calls: COUNT(cc_call_id) • Number of answered calls: SUM (CASE WHEN cc_talk_time >0 THEN 1 ELSE 0 END) • Number of abandoned calls: SUM (CASE WHEN (cc_talk_time = 0 AND cc_agent_id=0 AND cc_callback=0) THEN 1 ELSE 0 END) • The BIRT function Total.sum(<row>) is used for calculating the daily totals

Database tables	• tblcalls, tblcallscc, tblqueues
Database table attributes	• tblcalls = {call_id, call_start_time} • tblcallscc = {cc_call_id, cc_queue_id, cc_talk_time, cc_callback, cc_agent_id} • tblqueues = {queue_id, queue_name}

SQL Queries

Select call traffic details (number of calls, number of answered and abandoned calls) per hours daily

```
SELECT COUNT ("cc_call_id") AS "All Calls",
  EXTRACT (hour FROM "call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM "call_start_time") + 1) || ':00'
  AS "label",
  "date"("call_start_time"),
  queue_name,
  SUM (CASE WHEN cc_talk_time > 0 THEN 1 ELSE 0 END)
      AS "Answered calls",
  SUM (CASE WHEN
   (cc_talk_time = 0 AND cc_agent_id = 0 AND cc_callback = 0)
    THEN 1 ELSE 0 END) AS "Abandoned calls"

FROM tblcallscc, tblcalls, tblqueues

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id

GROUP BY EXTRACT(hour FROM "call_start_time"), queue_name,
         "date"("ch.start_time")
ORDER BY EXTRACT(hour FROM "call_start_time")
```

Select available queues (used for grouping the call details by queues)

```
SELECT DISTINCT queue_name, "date"("call_start_time")

FROM tblcallscc, tblcalls, tblqueues

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id

ORDER BY queue_name, "date"("call_start_time")
```

Select grand totals (total number of calls, total number of answered and abandoned calls) for the specified date range

```
SELECT
  COUNT ("cc_call_id") AS "All Calls",
  SUM (CASE WHEN cc_talk_time > 0 THEN 1 ELSE 0 END)
      AS "Answered calls",
  SUM (CASE WHEN
   (cc_talk_time = 0 AND cc_agent_id = 0 AND cc_callback = 0)
    THEN 1 ELSE 0 END) AS "Abandoned calls"

FROM tblcallscc, tblcalls

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
```

Select all available days having calls in the specified date range (used for grouping the call details by days)

```
SELECT DISTINCT "date"("call_start_time")

FROM tblcallscc, tblcalls

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"

ORDER BY "date"("call_start_time")
```

Exception

N/A

3.4.11 Call Traffic All Queues – Queue Time, GOS Per Hour Daily

Number of calls, maximum queue time, minimum queue time and grade of service for all available queues – having calls in the specified date range.

Required input parameters	• From date • To date (until) • Queue • Daily report
Output values (the values are grouped per queue, per hour and daily)	• Day • Queue • Time – hourly interval (e.g.: 09:00-10:00) • All calls (number of calls per hour daily) • Max Queue Time (Maximum queue time in seconds...) • Min Queue Time (Minimum queue time in seconds...) • GOS (Grade of service...) • Daily totals per queue (number of calls, average maximum queue time, average minimum queue time, average grade of service) • Grand totals all queues (number of calls, max queue time, min queue time, average GOS)
Format	• Table
Axis label	• N/A
Calculation rule	• Number of calls: COUNT(cc_call_id) • Max queue time: MAX(cc_queue_time) • Min queue time: MIN(cc_queue_time) • Average grade of service: AVG(cc_gos) • The BIRT function Total.sum(<row>) is used for calculating the daily totals • Daily average max queue time = Total.sum(dataSetRow["max"])/(row[0]+1) row[0]+1 = the number of available records
Database tables	• tblcalls, tblcallscc, tblqueues
Database table attributes	• tblcalls = {call_id, call_start_time} • tblcallscc = {cc_call_id, cc_queue_id, cc_queue_time, cc_gos} • tblqueues = {queue_id, queue_name}

SQL Queries

Select call traffic details (number of calls, max queue time, min queue time and average grade of service) per queue and per hour daily

```
SELECT COUNT ("cc_call_id") AS "All Calls",
  EXTRACT (hour FROM "call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM "call_start_time") + 1) || ':00'
  AS "label",
  "date"("call_start_time"),
  queue_name,
  MAX (cc_queue_time),
  MIN (cc_queue_time),
  AVG (cc_gos) AS "AVG GOS"

FROM tblcallscc, tblcalls, tblqueues

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id

GROUP BY EXTRACT(hour FROM "call_start_time"), queue_name,
         "date"("ch.start_time")
ORDER BY EXTRACT(hour FROM "call_start_time")
```

Select available queues (used for grouping the call details by queues)

```
SELECT DISTINCT queue_name, "date"("call_start_time")

FROM tblcallscc, tblcalls, tblqueues

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id

ORDER BY queue_name, "date"("call_start_time")
```

Select grand totals (total number of calls, max queue time, min queue time and average grade of service) for the specified date range

```
SELECT
  COUNT ("cc_call_id") AS "All Calls",
  MAX (cc_queue_time),
  MIN (cc_queue_time),
  AVG (cc_gos) AS "AVG GOS"

FROM tblcallscc, tblcalls

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
```

Select all available days having calls in the specified date range (used for grouping the call details by days)

```
SELECT DISTINCT "date"("call_start_time")

FROM tblcallscc, tblcalls

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"

ORDER BY "date"("call_start_time")
```

Exception

N/A

3.4.12 Call Traffic By Queue – Per Hour Daily – Details

Count of calls (all calls, answered calls and abandoned calls) for the selected queue and the specified date range.

Required input parameters	• From date • To date (until) • Queue • Daily report
Output values (the values are grouped per hour and daily)	• Day • Time – hourly interval (e.g.: 09:00-10:00) • All calls (number of calls per hour daily) • Answered calls (number of answered calls ...) • Abandoned calls (number of abandoned calls ...)
Format	• Table
Axis label	• N/A
Calculation rule	• Answered calls: {cc_talk time> 0} • Abandoned calls: {cc_talk time = 0, cc_agent_id = 0, cc_callback = 0} • Number of calls: COUNT(cc_call_id) • Number of answered calls: SUM (CASE WHEN cc_talk_time >0 THEN 1 ELSE 0 END) • Number of abandoned calls: SUM (CASE WHEN (cc_talk_time = 0 AND cc_agent_id=0 AND cc_callback=0) THEN 1 ELSE 0 END) • The BIRT function Total.sum(<row>) is used for calculating the daily totals
Database tables	• tblcalls, tblcallscc, tblqueues

Database table attributes	• tblcalls = {call_id, call_start_time} • tblcallscc = {cc_call_id, cc_queue_id, cc_talk_time, cc_callback, cc_agent_id} • tblqueues = {queue_id, queue_name}

SQL Queries

Select call traffic details (number of calls, number of answered and abandoned calls) per hour daily - for the selected queue and the specified date range

```
SELECT COUNT("cc_call_id") AS "All Calls",
  EXTRACT (hour FROM "call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM "call_start_time") + 1) || ':00'
  AS "label",
  "date"("call_start_time"),
  SUM (CASE WHEN cc_talk_time > 0 THEN 1 ELSE 0 END)
      AS "Answered calls",
  SUM (CASE WHEN
   (cc_talk_time = 0 AND cc_agent_id = 0 AND cc_callback = 0)
    THEN 1 ELSE 0 END) AS "Abandoned calls"

FROM tblcallscc, tblcalls, tblqueues

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id
  AND queue_name = ? /* queue name */

GROUP BY EXTRACT(hour FROM "call_start_time"),
         "date"("ch.start_time")
ORDER BY EXTRACT(hour FROM "call_start_time")
```

Select available queues (used for selecting the queue)

```
SELECT tblqueues."queue_name"
FROM tblqueues
ORDER BY tblqueues."queue_name"
```

Select grand totals (total number of calls, total number of answered and abandoned calls) - for the selected queue and the specified date range

```
SELECT
  COUNT ("cc_call_id") AS "All Calls",
  SUM (CASE WHEN cc_talk_time > 0 THEN 1 ELSE 0 END)
      AS "Answered calls",
  SUM (CASE WHEN
   (cc_talk_time = 0 AND cc_agent_id = 0 AND cc_callback = 0)
    THEN 1 ELSE 0 END) AS "Abandoned calls"

FROM tblcallscc, tblcalls, tblqueues

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id
  AND queue_name = ? /* queue name */
```

Select all available days having calls in the specified date range for the selected queue (used for grouping the call details daily)

```
SELECT DISTINCT "date"("call_start_time")

FROM tblcallscc, tblcalls, tblqueues

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id
  AND queue_name = ? /* queue name */

ORDER BY "date"("call_start_time")
```

Exception

N/A

3.4.13 Call Traffic One Agent – Per Hour Daily

Count of calls (call center calls, direct calls, outbound, inbound) and talk time for selected agent and the specified date range.

INFO: The report template **Call Traffic One Agent-Per Hour Daily-Details** returns the following additional output values: `Talk Time CC Calls`, `Talk Time Direct Calls`

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values (the values are grouped per hour and daily)	• Specified agent details (first name, surname, email and department) • Time – hourly interval (e.g.: 09:00-10:00) • CC calls (number of contact center calls) • Direct calls (number of direct calls) • Inbound calls (number of inbound/incoming calls) • Outbound calls (number of outbound/outgoing calls) • All calls (number of all calls = cc calls + direct calls) • Talk times (total talk time for the specified hourly interval) • Daily totals and grand totals (CC calls, direct calls, inbound, outbound, all calls, talk time)
Format	• Table
Axis label	• N/A
Calculation rule	• Inbound calls: ch_direction = 0: SUM (CASE ch_direction WHEN 0 THEN 1 ELSE 0 END) • Outbound calls: ch_direction = 1: SUM (ch_direction) • Talk time: SUM(ch_talk_time_seconds) • CC calls (in tblcallhistory): if the same ch_local_id and date (ch_start_time) exists in tblcalls (call_local_id and date (call_start_time)) • Number of CC calls = sum (case when ch_local_id||'_'||date(ch_start_time) IN (select call_local_id||'_'||date(call_start_time) from tblcalls) then 1 else 0 end) • Direct calls: if ch_local_id and date (ch_start_time) NOT exists in tblcalls (call_local_id and date (call_start_time)) • Number of Direct Calls (NOT CC calls) = sum (case when ch_local_id||'_'|| date (ch_start_time) NOT IN (select call_local_id||'_'|| date (call_start_time) from tblcalls) then 1 else 0 end) • The BIRT function Total.sum(<row>) is used for calculating the daily totals

Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_talk_time_seconds, ch_direction, ch_user_id, ch_local_id} • tblcalls = {call_start_time, call_local_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_name, department_id}

SQL Queries

Select call traffic details grouped per hour daily, for specified agent in the selected date range

```
SELECT "date"(t1.ch_start_time),
  EXTRACT (HOUR FROM t1.ch_start_time)||':00 - '||
  (EXTRACT (HOUR FROM t1.ch_start_time)+1)||':00’
  AS "label",
  COUNT (DISTINCT t1.id) AS "All Calls",
  SUM (CASE WHEN t1.id IN
     (SELECT call_local_id||'_'|| DATE (call_start_time)
         FROM tblcalls) THEN 1 ELSE 0 END) AS "CC calls",
-- # NOT CC Calls = ( "# all calls" - "# cc calls" ) in the report
-- OR can be calculated here :
  SUM (CASE WHEN t1.id NOT IN
     (SELECT call_local_id||'_'|| DATE (call_start_time)
         FROM tblcalls) THEN 1 ELSE 0 END) AS "Direct calls",
  SUM (t1.ch_direction) AS "Outgoing Calls",
-- # Incoming Calls = ("# all calls" - "# outgoing calls")
-- in the report OR can be calculated here :
  SUM (CASE t1.ch_direction WHEN 0 THEN 1 ELSE 0 END)
                     AS "Incoming Calls",
  SUM (t1.ch_talk_time_seconds) AS "Talk Time All Calls"
--"time" (TIMESTAMP 'epoch' + (sum (ch_talk_time_seconds)) *
-- INTERVAL '1 second'))

FROM tblusers,
  (SELECT DISTINCT ON (ch_local_id||'_'|| DATE (ch_start_time))
    ch_local_id||'_'|| DATE (ch_start_time) AS id,
    ch_start_time,ch_call_id,ch_local_id,ch_direction,
    ch_talk_time_seconds,ch_user_id FROM tblcallhistory)
    AS t1
  WHERE t1."ch_start_time" >= ? /* from time */
    AND t1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours') /
      *to date*/
    AND t1."ch_user_id" = user_id
    AND user_login = ? /*User login*/

GROUP BY EXTRACT (hour FROM t1."ch_start_time"),
         "date"(t1."ch_start_time")
ORDER BY EXTRACT(hour FROM t1."ch_start_time")
```

Select available agents (used for selecting the agent)

```
SELECT user_login, user_surname, user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY user_firstname, user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT
      CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE  tbldepartments.department_name
          END
        FROM tbldepartments
        WHERE u.user_department_id = tbldepartments.department_id
     ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select grand totals (total number of calls, total number of call center calls, direct calls, incoming calls, outgoing calls and total talk time) for the selected agent in the specified date range

```
SELECT
  COUNT (DISTINCT t1.id) AS "All Calls",
  SUM (CASE WHEN t1.id IN
    (SELECT call_local_id||'_'|| DATE (call_start_time)
      FROM tblcalls)
         THEN 1 ELSE 0 END) AS "CC calls",
-- # NOT CC Calls = ( "# all calls" - "# cc calls")
-- in the report OR can be calculated here :
  SUM (CASE WHEN t1.id NOT IN (SELECT call_local_id||'_'||
    DATE (call_start_time) FROM tblcalls)
         THEN 1 ELSE 0 END) AS "NOT CC calls",
    SUM (t1.ch_direction) AS "Outgoing Calls",
-- # Incoming Calls = ("# all calls" - "# outgoing calls")
-- in the report OR can be calculated here :
  SUM (CASE t1.ch_direction WHEN 0 THEN 1 ELSE 0 END)
           AS "Incoming calls",
  SUM (t1.ch_talk_time_seconds) AS "TimeInSeconds"

FROM tblusers,
  (SELECT DISTINCT ON (ch_local_id||'_'|| DATE (ch_start_time))
    ch_local_id||'_'|| DATE (ch_start_time) AS id,
    ch_start_time,ch_call_id,ch_local_id,ch_direction,
    ch_talk_time_seconds,ch_user_id FROM tblcallhistory)
    AS t1
  WHERE t1."ch_start_time" >= ? /* from time */
    AND t1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours') /
      *to date*/
    AND t1."ch_user_id" = user_id
    AND user_login = ? /*User login*/
```

Select all available days having calls for the selected agent in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT "date"("ch_start_time")
  SUM (ch_talk_time_seconds) AS "TalkTimeDaily"

FROM tblusers, tblcallhistory

WHERE "ch_start_time" >= ? /* from date */
  AND "ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND user_id = ch_user_id
  AND user_login = ? /* user login */

GROUP BY "date"("ch.start_time")
ORDER BY "date"("ch.start_time")
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

One example in the BIRT Expression Builder:

```
ts = row["TimeInSeconds"];
day =  (ts/86400) | 0 ;
hour = ((ts-(day*86400))/3600) | 0;
min  =  ((ts-(day*86400) - (hour*3600))/60) | 0;
sec =  (ts-(day*86400) - (hour*3600) - (min*60));
if(day == 0)day = "";
else day = day+" day(s) ";
if(hour<10)hour = "0"+hour;
if(min<10)min = "0"+min;
if(min<10)min = "0"+min;
if(sec<10)sec = "0"+sec;
day+""+hour+":"+min+":"+sec
```

Output example 1 : 2 day(s) 02:15:26

Output example 2 : 10:39:07

3.4.14 Call Traffic One Agent – Per Hour Daily – Details

Count of calls (all calls, call center calls, direct calls, outbound, inbound) and talk time (cc calls talk time, direct calls talk time and total talk time-for all calls) for selected agent and the specified date range.

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values (the values are grouped per hour and daily)	• Specified agent details (first name, surname, email and department) • Time – hourly interval (e.g.: 09:00-10:00) • CC calls (number of contact center calls) • Talk time CC calls (CC calls: total talk time for the specified hourly interval) • Direct calls (number of direct calls) • Talk time direct calls (direct calls: total talk time for the specified hourly interval) • Inbound calls (number of inbound/incoming calls) • Outbound calls (number of outbound/outgoing calls) • All calls (number of all calls = cc calls + direct calls) • Talk time - all calls (total talk time for the specified hourly interval) • Daily totals and grand totals (CC calls, Direct calls, Inbound, Outbound, All calls and talk time of CC calls, direct calls and all calls)
Format	• Table
Axis label	• N/A

Calculation rule	• Inbound calls: ch_direction = 0: SUM (CASE ch_direction WHEN 0 THEN 1 ELSE 0 END) • Outbound calls: ch_direction = 1: SUM (ch_direction) • Talk time: SUM(ch_talk_time_seconds) • CC calls (in tblcallhistory): if the same ch_local_id and date (ch_start_time) exists in tblcalls (call_local_id and date (call_start_time)) • Number of CC calls = sum (case when ch_local_id||'_'|| date (ch_start_time) IN (select call_local_id||'_'|| date (call_start_time) from tblcalls) then 1 else 0 end) • Direct calls :if ch_local_id and date (ch_start_time) NOT exists in tblcalls (call_local_id and date (call_start_time)) • Number of Direct Calls (NOT CC calls) = sum (case when ch_local_id||'_'|| date (ch_start_time) NOT IN (select call_local_id||'_'|| date (call_start_time) from tblcalls) then 1 else 0 end) • Direct calls talk time: summation of the calls talk time for call records, where ch_local_id and date (ch_start_time) NOT exists in tblcalls (call_local_id and date (call_start_time)) • Direct calls talk time = sum (case when ch_local_id||'_'|| date (ch_start_time) NOT IN (select call_local_id||'_'|| date (call_start_time) from tblcalls) then ch_talk_time_seconds else 0 end) • CC calls talk time (in tblcallhistory) : summation of the calls talk time for call records, where ch_local_id und date (ch_start_time) exists in tblcalls (call_local_id and date (call_start_time)) • CC calls talk time = sum (case when ch_local_id||'_'|| date (ch_start_time) IN (select call_local_id||'_'|| date (call_start_time) from tblcalls) then ch_talk_time_seconds else 0 end) • The BIRT function Total.sum(<row>) is used for calculating the daily totals
Database tables	• tblcallhistory, tblusers, tbldepartments
Database table attributes	• tblcallhistory = {ch_call_id,ch_start_time, ch_talk_time_seconds, ch_direction, ch_user_id, ch_local_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tblcalls = {call_start_time, call_local_id} • tbldepartments = {department_name, department_id}

SQL Queries

Select call traffic details grouped per hour and daily, for the specified agent in the selected date range.

```
SELECT "date" (t1.ch_start_time),
  EXTRACT (hour FROM t1.ch_start_time)||':00 - '||
  (EXTRACT (hour FROM t1.ch_start_time)+1)||':00'
  AS "label",
  COUNT (DISTINCT t1.id) AS "All Calls",
  SUM (CASE WHEN t1.id IN (select call_local_id||'_'||
     DATE (call_start_time) FROM tblcalls)
         THEN 1 ELSE 0 END) AS "CC calls",
-- # NOT CC Calls = ( "# all calls" - "# cc calls" )
-- in the report OR can be calculated here :
  SUM (CASE WHEN t1.id NOT IN
     (SELECT call_local_id||'_'||
     DATE (call_start_time) FROM tblcalls)
         THEN 1 ELSE 0 END) AS "Direct calls",
  SUM (CASE WHEN t1.id IN (select call_local_id||'_'||
     DATE (call_start_time) FROM tblcalls)
         THEN t1.ch_talk_time_seconds ELSE 0 END)
      AS "Talk Time CC Calls",
  SUM (CASE WHEN t1.id NOT IN
     (SELECT call_local_id||'_'||
     DATE (call_start_time) FROM tblcalls)
         THEN t1.ch_talk_time_seconds ELSE 0 END)
      AS "Talk Time Direct Calls",
  SUM (t1.ch_direction) AS "Outgoing Calls",
-- # Incoming Calls = ("# all calls" - "# outgoing calls")
-- in the report OR can be calculated here :
  SUM (CASE t1.ch_direction WHEN 0 THEN 1 ELSE 0 END)
      AS "Incoming Calls",
  SUM (t1.ch_talk_time_seconds) AS "Talk Time All Calls"
--"time" (TIMESTAMP 'epoch' + (sum (ch_talk_time_seconds))
-- * INTERVAL '1 second')

FROM tblusers,
  (SELECT DISTINCT ON (ch_local_id||'_'|| DATE (ch_start_time))
    ch_local_id||'_'|| DATE (ch_start_time) AS id,
    ch_start_time,ch_call_id,ch_local_id,ch_direction,
    ch_talk_time_seconds,ch_user_id FROM tblcallhistory)
    AS t1
  WHERE t1."ch_start_time" >= ? /* from time */
    AND t1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours') /
      *to date*/
    AND t1."ch_user_id" = user_id
    AND user_login = ? /*User login*/

GROUP BY EXTRACT (HOUR FROM t1."ch_start_time"),
  "date" (t1."ch_start_time")

ORDER BY EXTRACT(HOUR FROM t1."ch_start_time")
```

Select available agents (used for selecting the agent)

```
SELECT user_login, user_surname, user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY user_firstname, user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
  THEN '' ELSE  tbldepartments.department_name
  END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id
     ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select grand totals (total number of calls, total number of call center calls, direct calls, incoming calls, outgoing calls and total talk time) for the selected agent in the specified date range

```
SELECT
  COUNT (DISTINCT t1.id) AS "All Calls",
  SUM (CASE WHEN t1.id IN (select call_local_id||'_'||
     DATE (call_start_time) FROM tblcalls)
         THEN 1 ELSE 0 END) AS "CC calls",
-- # NOT CC Calls = ( "# all calls" - "# cc calls" )
-- in the report OR can be calculated here :
  SUM (CASE WHEN t1.id NOT IN (select call_local_id||'_'||
     DATE (call_start_time) FROM tblcalls)
         THEN 1 ELSE 0 END) AS "NOT CC calls",
  SUM (CASE WHEN t1.id IN (select call_local_id||'_'||
     DATE (call_start_time) FROM tblcalls)
         THEN t1.ch_talk_time_seconds ELSE 0 END)
      AS "Talk Time CC Calls",
  SUM (CASE WHEN t1.id NOT IN (select call_local_id||'_'||
     DATE (call_start_time) FROM tblcalls)
         THEN t1.ch_talk_time_seconds ELSE 0 END)
      AS "Talk Time Direct Calls", ,
  SUM (t1.ch_direction) AS "Outgoing calls",
-- # Incoming Calls = ("# all calls" - "# outgoing calls")
-- in the report OR can be calculated here :
  SUM (CASE t1.ch_direction WHEN 0 THEN 1 ELSE 0 END)
      AS "Incoming Calls",
  SUM (t1.ch_talk_time_seconds) AS "TimeInSeconds"

FROM tblusers (SELECT DISTINCT ON (ch_local_id||'_'||
    DATE (ch_start_time)) ch_local_id||'_'||
    DATE (ch_start_time)
  AS id, ch_start_time, ch_call_id, ch_local_id, ch_direction,
    ch_talk_time_seconds, ch_user_id
  FROM tblcallhistory) AS t1 WHERE t1."ch_start_time" >= ?
    /* from time */
  AND t1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
    /*to date*/
  AND t1."ch_user_id" = user_id
  AND user_login = ? /*User login*/
```

Select all available days having calls in the specified date range (used for grouping the information by days)

```
SELECT
  DISTINCT "date"("ch_start_time"),
  SUM (ch_talk_time_seconds) AS "TalkTimeDaily"

FROM tblcallhistory

WHERE "ch_start_time" >= ? /* from date */
  AND "ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
  AND user_id = ch_user_id
  AND user_login = ? /* user login */

GROUP BY "date"("ch.start_time")
ORDER BY "date"("ch.start_time")
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used.

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

One example in the BIRT Expression Builder:

```
ts = row["TimeInSeconds"];
day =  (ts/86400) | 0 ;
hour = ((ts-(day*86400))/3600) | 0;
min  =  ((ts-(day*86400) - (hour*3600))/60) | 0;
sec =  (ts-(day*86400) - (hour*3600) - (min*60));
if(day == 0)day = "";
else day = day+" day(s) ";
if(hour<10)hour = "0"+hour;
if(min<10)min = "0"+min;
if(min<10)min = "0"+min;
if(sec<10)sec = "0"+sec;
day+""+hour+":"+min+":"+sec
```

Output example 1 : 2 day(s) 02:15:26

Output example 2 : 10:39:07

3.4.15 Call Traffic One Queue – Queue Time, GOS Per Hour Daily

Number of calls, maximum queue time, minimum queue time and grade of service for the selected queue and the specified date range.

Required input parameters	• From date • To date (until) • Queue • Daily report
Output values (the values are grouped per hour and daily)	• Day • Time – hourly interval (e.g.: 09:00-10:00) • All calls (number of calls per hour daily) • Max Queue Time (Maximum queue time in seconds...) • Min Queue Time (Minimum queue time in seconds...) • GOS (Grade of service...) • Daily totals (number of calls, average maximum queue time, average minimum queue time, average grade of service) • Grand totals (number of calls, max queue time, min queue time, average GOS)
Format	• Table
Axis label	• N/A

Calculation rule	• Number of calls: COUNT(cc_call_id) • Max queue time: MAX(cc_queue_time) • Min queue time: MIN(cc_queue_time) • Average grade of service: AVG(cc_gos) • The BIRT function Total.sum(<row>) is used for calculating the daily totals • Daily average max queue time = Total.sum(dataSetRow["max"])/(row[0]+1) row[0]+1 = the number of available records
Database tables	• tblcalls, tblcallscc, tblqueues
Database table attributes	• tblcalls = {call_id, call_start_time} • tblcallscc = {cc_call_id, cc_queue_id, cc_queue_time, cc_gos} • tblqueues = {queue_id, queue_name}

SQL Queries

Select call traffic details (number of calls, max queue time, min queue time and average grade of service) per hours daily - for the selected queue and the specified date range

```
SELECT COUNT ("cc_call_id") AS "All Calls",
  EXTRACT (hour FROM "call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM "call_start_time") + 1) || ':00'
  AS "label",
  "date"("call_start_time"),
  MAX (cc_queue_time),
  MIN (cc_queue_time),
  AVG (cc_gos) AS "AVG GOS"

FROM tblcallscc, tblcalls, tblqueues

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id
  AND queue_name = ? /* queue name */

GROUP BY EXTRACT(hour FROM "call_start_time"), queue_name,
         "date"("ch.start_time")
ORDER BY EXTRACT(hour FROM "call_start_time")
```

Select available queues (used for grouping the call details by queues)

```
SELECT tblqueues."queue_name"
FROM tblqueues
ORDER BY tblqueues."queue_name"
```

Select grand totals (total number of calls, max queue time, min queue time and average grade of service) for the selected queue and the specified date range

```
SELECT
  COUNT ("cc_call_id") AS "All Calls",
  MAX (cc_queue_time),
  MIN (cc_queue_time),
  AVG (cc_gos) AS "AVG GOS"

FROM tblcallscc, tblcalls

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id
  AND queue_name = ? /* queue name */
```

Select all available days having calls for the selected queue in the specified date range (used for grouping the call details by days)

```
SELECT DISTINCT "date"("call_start_time")

FROM tblcallscc, tblcalls

WHERE "call_start_time" >= ? /* from time */
  AND "call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
  AND "cc_call_id" = "call_id"
  AND "cc_queue_id" = queue_id
  AND queue_name = ? /* queue name */

ORDER BY "date"("call_start_time")
```

Exception

N/A

3.4.16 Callback Calls

The report displays callback details for all calls in the specified date/time range.

Required input parameters	• From date • To date (until) • From time • To time • Business hours only (else 24/24) • Daily report
Output values (the values are grouped by queues and daily)	• Call date • Queue name • Time of call • Call ID • CLI – calling number • Agent • Callback number • Daily total number of callback calls by queue • Daily total number of callback calls (all queues) • Total number of callback calls
Format	• Table
Axis label	• N/A
Calculation rule	• Callback call= {tblcallscc.cc_callback = 1} • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id, cc_callback, cc_agent_id} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_id, queue_name} • tblswitches = {switch_office_start, switch_office_end}

SQL Queries

Select call details (call ID, call date, call time, calling number, queue name, queue ID, callback number and agent)

```
SELECT tblcallscc."cc_call_id",
       "date"(tblcalls."call_start_time"),
       "time"(tblcalls."call_start_time"),
       tblcalls."call_calling_number",
       q."queue_name",
       q."queue_id",
       tblcallscc.cc_callback_number,
      (SELECT u.user_firstname || ' ' || u.user_surname
        FROM tblusers u
         WHERE u.user_id = tblcallscc."cc_agent_id")
         AS "Agent"

FROM tblcallscc, tblcalls, tblqueues q, tblswitches s

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcallscc."cc_queue_id" = q."queue_id"
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_callback" = 1
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
         "time"(tblcalls."call_start_time" >=
                                "time"(s.switch_office_start)
     AND
        "time"(tblcalls."call_start_time") <=
                                 "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00' AND
      "time"(tblcalls."call_start_time") <= '23:59:59'
    END )

ORDER BY (tblcalls."call_start_time")
```

Select all available days having calls in the selected date/time range and the daily total number of calls

```
SELECT DISTINCT "date"(tblcalls."call_start_time") AS "AllDates",
       COUNT (tblcallscc."cc_call_id")

FROM tblcallscc, tblcalls, tblswitches s

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
          "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_callback" = 1
  AND ( CASE WHEN ? = 1 THEN /* Business hours only */
         "time"(tblcalls."call_start_time" >=
                                  "time"(s.switch_office_start)
     AND
        "time"(tblcalls."call_start_time") <=
                                   "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(c.call_start_time) >= '00:00:00' AND
      "time"(tblcalls."call_start_time") <= '23:59:59'
    END )

GROUP BY "date"(tblcalls."call_start_time")
ORDER BY "date"(tblcalls."call_start_time")
```

Select all available queues (used for grouping by queue) in the specified date/time range

```
SELECT DISTINCT q.queue_id, q.queue_name,
               "date"(tblcalls."call_start_time")

FROM tblcallscc, tblcalls, tblqueues q, tblswitches s

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_queue_id" = q.queue_id
  AND tblcallscc."cc_callback" = 1
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
        "time"(tblcalls."call_start_time" >=
                                  "time"(s.switch_office_start)
     AND
        "time"(tblcalls."call_start_time") <=
                                   "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00' AND
      "time"(tblcalls."call_start_time") <= '23:59:59'
    END )

ORDER BY "date"(tblcalls."call_start_time")
```

Select total number of callback calls in the specified date/time range

```
SELECT COUNT (tblcallscc."cc_call_id")

FROM tblcallscc, tblcalls, tblswitches s

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_callback" = 1
  AND ( CASE WHEN ? = 1 THEN /* Business hours only */
         "time"(tblcalls."call_start_time" >=
                                  "time"(s.switch_office_start)
     AND
        "time"(tblcalls."call_start_time") <=
                                   "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00' AND
      "time"(tblcalls."call_start_time") <= '23:59:59'
    END )
```

Exception

N/A

3.4.17 Calls List Agent

Calls list for selected agent in specified date/time range.

Required input parameters	• From date • To date (until) • From time • To time • Agent • Business hours only (else 24/24) • Daily report
Output values	• Start time • End time • Queue name • Queue time (queue time - the amount of time a caller has been waiting to get connected to an agent) • Talk time • CLI – Calling number • Grade of Service • Total number of calls • Total queue time • Total talk time
Format	• Table
Axis label	• N/A
Calculation rule	• Total number of calls: COUNT(number of calls) • Total talk time: SUM(talk time) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches, tbldepartments, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_time, cc_agent_id, cc_talk, cc_queue_id, cc_gos} • tblcalls = {call_id, call_start_time, call_end_time, call_calling_number} • tblqueues = {queue_id, queue_name} • tblswitches = {switch_office_start, switch_office_end} • tbldepartments = {department_name, department_id} • tblusers = {u.user_firstname, u.user_surname, u.user_extension, u.user_email, u.user_login, user_id}

SQL Queries

Select call list details for the selected agent in the specified date range

```
SELECT tblcalls."call_start_time",
       tblcalls."call_end_time",
       tblcalls."call_calling_number",
       tblcallscc."cc_call_id",
       tblcallscc."cc_queue_time",
       tblcallscc."cc_talk_time",
       tblcallscc."cc_gos",
       tblqueues."queue_name"

FROM tblcallscc, tblcalls, tblqueues, tblswitches s

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_agent_id" = ? */ agent id */
  AND tblcallscc."cc_queue_id" = tblqueues."queue_id"
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
          "time"(tblcalls."call_start_time") >=
                                  "time"(s.switch_office_start)
       AND
          "time"(tblcalls."call_start_time") <=
                                  "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00' AND
      "time"(tblcalls."call_start_time") <= '23:59:59'
    END )
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login, (
  SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
  THEN '' ELSE tbldepartments.department_name
  END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id )
  AS department_name

FROM tblusers u
```

```
WHERE u.user_login = ? /* agent login */
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • sec. = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.4.18 Calls List Queue

Calls list for selected queue in specified date range.

Required input parameters	• From date • To date (until) • Queue • Business hours only (else 24/24) • Daily report
Output values	• Start time • End time • Agent • Queue time (queue time - the amount of time a caller has been waiting to get connected to an agent) • Talk time • CLI – Calling number • Grade of Service • Total number of calls • Total queue time • Total talk time

Format	• Table
Axis label	• N/A
Calculation rule	• Total number of calls: COUNT(number of calls) • Total talk time: SUM(talk time) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_time, cc_agent_id, cc_talk_time, cc_queue_id, cc_gos} • tblcalls = {call_id, call_start_time, call_end_time, call_calling_number} • tblqueues = {queue_id, queue_name} • tblswitches = {switch_office_start, switch_office_end} • tblusers = {u.user_firstname, u.user_surname, u.user_login, user_id}

SQL Queries

Select call list details for the selected agent in the specified date range

```
SELECT tblcalls."call_start_time",
       tblcalls."call_end_time",
       tblcalls."call_calling_number",
       tblcallscc."cc_call_id",
       tblcallscc."cc_queue_time",
       tblcallscc."cc_talk_time",
       tblcallscc."cc_gos",
       tblusers."user_firstname",
       tblusers."user_surname",
       tblusers."user_login"

FROM tblcallscc, tblcalls, tblusers, tblqueues, tblswitches s

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND tblcalls."call_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_agent_id" = tblusers."user_id"
  AND tblcallscc."cc_queue_id" = tblqueues."queue_id"
  AND tblqueues."queue_name" = ? /* queue name */
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
         "time"(tblcalls."call_start_time") >=
                                "time"(s.switch_office_start)
       AND
          "time"(tblcalls."call_start_time") <=
                                 "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00' AND
      "time"(tblcalls."call_start_time") <= '23:59:59'
    END )
```

Select all available queues (used for selecting the queue)

```
SELECT tblqueues."queue_name"
FROM tblqueues
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • sec. = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.4.19 Contact Center (Per Agents) – Chart

Number of calls (total number of calls, answered and missed calls) by agents for the specified date range.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Total number of calls • Total number of Answered Calls • Total number of missed calls
Format	• Graphics and Grids
Axis label	• Horizontal: Agents • Vertical: Number of calls

Calculation rule	• Missed call has event_type = 6 in tblagentactivity • Total number of calls: COUNT(Number of calls) • Total number of answered calls: COUNT(number of calls | talk time > 0 s) • Total number of missed calls: COUNT(call ID | aa_agent_id > 0, aa_call_id > 0, aa_event_type = 6)
Database tables	• tblcallscc, tblcalls, tblagentactivity, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_agent_id, cc_talk_time} • tblcalls = {call_id, call_start_time} • tblusers = {u.user_firstname, u.user_surname, u.user_login, user_id} • tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type}

SQL Queries

Select number of calls (total number of calls, answered and missed calls) by agents for the specified date range

```
SELECT u.user_firstname,
       u.user_surname,
       u.user_id,
       u.user_login,
       COUNT (tblcalls.call_id) AS "All calls",
       (SELECT COUNT (aa.aa_call_id)
          FROM tblagentactivity aa
          WHERE aa.aa_event_time >= ? /* from time */
          AND aa.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                                  /* to date */
          AND aa.aa_agent_id > 0
          AND aa.aa_call_id > 0
          AND aa.aa_agent_id = u.user_id
          AND aa.aa_event_type = 6
        )  AS "Missed Calls",
       (SELECT COUNT (c.call_id)
          FROM tblcalls c, tblcallscc cc
          WHERE c."call_start_time" >= ? /* from time */
          AND c."call_start_time" <= ("date"(?) + INTERVAL
                                '24 hours') /* to date */
          AND c.call_id = cc.cc_call_id
          AND cc.cc_agent_id = u.user_id
          AND cc."cc_talk time" > 0
        )  AS "Answered Calls"

FROM tblcalls, tblcallscc, tblusers u

WHERE tblcalls."call_start_time" >= "date" (?) /* from date */
  AND tblcalls."call_start_time" <= "date"(?) /* to date */
  AND tblcalls.call_id = tblcallscc.cc_call_id
  AND tblcallscc.cc_agent_id = u.user_id

GROUP BY u.user_firstname, u.user_surname, u.user_id, u.user_login
```

Exception

A maximum of 15 agents (vertical tubes) can be shown in a graphic. If there are more than 15 agents, the graphic will not be displayed because with more than 15 agents the graphic is not properly visible.

3.4.20 Contact Center (Per Agents) – List

Number of calls (total number of calls, answered and missed calls), percents of calls, average queue time and talk time by agents for the specified date range.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Agent • Number of calls by agent (All calls) (Nc) • Percentage of total number of calls by agent • Number of answered calls by agent (Na) • Percentage of total number of answered calls • Number of missed calls by agent (Nm) • Percentage of total number of missed calls • Average queue time in seconds by agent • Average talk time in seconds by agent • Total number of calls (Ntc) • Total number of answered calls (Nta) • Total number of missed calls (Ntm) • Total average queue time in seconds (all agents) • Total average talk time in seconds (all agents)
Format	• Table
Axis label	• N/A
Calculation rule	• Missed calls: {aa_agent_id>0, aa_call_id>0, aa_event_type=6} • Answered calls : {talk time > 0 seconds} • Total number of calls: SUM(number of calls by agents) • Total number of answered calls: SUM(answered calls by agents) • Total number of missed calls: SUM(missed calls by agents) • Percentage of total number of calls by agent : Nc / Ntc * 100 (number of calls by agent / total number of calls * 100) • Percentage of total number of answered calls: Na / Nta * 100 (number of answered calls by agent / total number of answered calls * 100) • Percentage of total number of missed calls: Nm / Ntm * 100 (number of missed calls by agent / total number of missed calls * 100)

Database tables	• tblcallscc, tblcalls, tblagentactivity, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_agent_id, cc_talk_time, cc_queue_time} • tblcalls = {call_id, call_start_time} • tblusers = {user_firstname, user_surname, user_login, user_id} • tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type}

SQL Queries

Select number of calls (total number of calls, answered and missed calls), average queue and talk time by agents for the specified date range

```
SELECT u.user_firstname,
       u.user_surname,
       u.user_id,
       u.user_login,
       AVG (tblcallscc.cc_queue_time),
       AVG (tblcallscc.cc_talk_time),
       COUNT (tblcalls.call_id) AS "All calls",
       (SELECT  COUNT (aa.aa_call_id)
          FROM tblagentactivity aa
          WHERE aa.aa_event_time >= ? /* from time */
          AND aa.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                                    /* to date */
          AND aa.aa_agent_id > 0
          AND aa.aa_call_id > 0
          AND aa.aa_agent_id = u.user_id
          AND aa.aa_event_type = 6
        )  AS "Missed Calls",
       (SELECT  COUNT (c.call_id)
          FROM tblcalls c, tblcallscc cc
          WHERE c."call_start_time" >= ? /* from time */
          AND c."call_start_time" <= ("date"(?) + INTERVAL
                                 '24 hours') /* to date */
          AND c.call_id = cc.cc_call_id
          AND cc.cc_agent_id = u.user_id
          AND cc."cc_talk time" > 0
        )  AS "Answered Calls"

FROM tblcalls, tblcallscc, tblusers u

WHERE tblcalls."call_start_time" >= "date" (?) /* from time */
  AND tblcalls."call_start_time" <= "date"(?) /* to date */
  AND tblcalls.call_id = tblcallscc.cc_call_id
  AND tblcallscc.cc_agent_id = u.user_id

GROUP BY u.user_firstname, u.user_surname, u.user_id, u.user_login
```

Exception

N/A

3.4.21 Contact Center (Per Queues) – Chart

Number of calls (total number of calls, answered and missed calls) by queues for the specified date range.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Total number of calls • Total number of Answered Calls • Total number of missed calls
Format	• Graphics and Grids
Axis label	• Horizontal: Queues • Vertical: Number of calls
Calculation rule	• Missed call has event_type = 6 in tblagentactivity • Total number of calls: COUNT(Number of calls) • Total number of answered calls: COUNT(number of calls | talk time > 0 s) • Total number of missed calls: COUNT(call ID | aa_agent_id > 0, aa_call_id > 0, aa_event_type = 6)
Database tables	• tblcallscc, tblcalls, tblagentactivity, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id, cc_talk_time} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_name, queue_id} • tblagentactivity = {aa_id, aa_queue_id, aa_event_time, aa_event_type}

SQL Queries

Select number of calls (total number of calls, answered and missed calls) by queues for the specified date range

```
SELECT q.queue_name,
       q.queue_id,
       COUNT (tblcalls.call_id) AS "All calls",
       (SELECT COUNT (aa.aa_call_id)
          FROM tblagentactivity aa
          WHERE aa.aa_event_time >= ? /* from time */
          AND aa.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                                   /* to date */
          AND aa.aa_agent_id > 0
          AND aa.aa_call_id > 0
          AND aa.aa_queue_id = q.queue_id
          AND aa.aa_event_type = 6
        )  AS "Missed Calls",
       (SELECT COUNT (c.call_id)
          FROM tblcalls c, tblcallscc cc
          WHERE c."call_start_time" >= ? /* from time */
          AND c."call_start_time" <= ("date"(?) + INTERVAL
                                 '24 hours') /* to date */
          AND c.call_id = cc.cc_call_id
          AND cc.cc_queue_id = q.queue_id
          AND cc."cc_talk time" > 0
        )  AS "Answered Calls"

FROM tblcalls, tblcallscc, tblqueues q

WHERE tblcalls."call_start_time" >= "date" (?) /* from date */
  AND tblcalls."call_start_time" <= "date"(?) /* to date */
  AND tblcallscc.cc_queue_id = q.queue_id
  AND tblcalls.call_id = tblcallscc.cc_call_id

GROUP BY q.queue_name, q.queue_id
```

Exception

The maximum of queues (vertical tubes) shown in a graphic is 15. If there are more than 15 agents, the graphic will not be displayed because with more than 15 agents the graphic is not properly visible.

3.4.22 Contact Center (Per Queues) – List

Number of calls (total number of calls, answered and missed calls), percents of calls, average queue time and talk time by queues for the specified date range.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Queue • Number of calls by queue (All calls) (Nc) • Percentage of total number of calls by queue • Number of answered calls by queue (Na) • Percentage of total number of answered calls • Number of missed calls by queue (Nm) • Percentage of total number of missed calls • Average queue time in seconds by queue • Average talk time in seconds by queue • Total number of calls (Ntc) • Total number of answered calls (Nta) • Total number of missed calls (Ntm) • Total average queue time in seconds (all queues) • Total average talk time in seconds (all queues)
Format	• Table
Axis label	• N/A
Calculation rule	• Missed calls: {aa_agent_id>0, aa_call_id>0, aa_event_type=6} • Answered calls : {talk time > 0 seconds} • Total number of calls: SUM(number of calls by agents) • Total number of answered calls: SUM(answered calls by agents) • Total number of missed calls: SUM(missed calls by agents) • Percentage of total number of calls by agent : Nc / Ntc * 100 (number of calls by agent / total number of calls * 100) • Percentage of total number of answered calls: Na / Nta * 100 (number of answered calls by agent / total number of answered calls * 100) • Percentage of total number of missed calls: Nm / Ntm * 100 (number of missed calls by agent / total number of missed calls * 100)
Database tables	• tblcallscc, tblcalls, tblagentactivity, tblqueues

Database table attributes	• tblcallscc = {cc_call_id, cc_agent_id, cc_talk_time, cc_queue_time} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_name, queue_id} • tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type}

SQL Queries

Select number of calls (total number of calls, answered and missed calls), average queue and talk time by queues for the specified date range

```
SELECT q.queue_name,
       q.queue_id,
       AVG (tblcallscc.cc_queue_time),
       AVG (tblcallscc.cc_talk_time),
       COUNT (tblcalls.call_id) AS "All calls",
       (SELECT COUNT (aa.aa_call_id)
          FROM tblagentactivity aa
          WHERE aa.aa_event_time >= ? /* from time */
          AND aa.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                                   /* to date */
          AND aa.aa_agent_id > 0
          AND aa.aa_call_id > 0
          AND aa.aa_queue_id = q.queue_id
          AND aa.aa_event_type = 6
        )  AS "Missed Calls",
       (SELECT  COUNT (c.call_id)
          FROM tblcalls c, tblcallscc cc
          WHERE c."call_start_time" >= ? /* from time */
          AND c."call_start_time" <= ("date"(?) + INTERVAL
                                 '24 hours') /* to date */
          AND c.call_id = cc.cc_call_id
          AND cc.cc_queue_id = q.queue_id
          AND cc."cc_talk time" > 0
        )  AS "Answered Calls"

FROM tblcalls, tblcallscc, tblqueues q

WHERE tblcalls."call_start_time" >= "date" (?) /* from time */
  AND tblcalls."call_start_time" <= "date"(?) /* to date */
  AND tblcallscc.cc_queue_id = q.queue_id
  AND tblcalls.call_id = tblcallscc.cc_call_id

GROUP BY q.queue_name, q.queue_id
```

Exception

N/A

3.4.23 Contact Center Calls

The report displays call details (missed, answered and abandoned calls) in the selected date/time range.

Required input parameters	• From date • To date (until) • From time • To time • Business hours only (else 24/24) • Daily report
Output values	Missed calls • Call ID • Arrived at • Agent • Queue • Missed call time (seconds) • CLI - calling number • Average missed call time (seconds) • Total number of missed calls Abandoned calls • Call ID • Arrived at • Queue • Queue time (seconds) • CLI - calling number • Average queue time (seconds) • Total number of abandoned calls Answered calls • Call ID • Arrived at • Queue • Agent • Queue time (seconds) • Talk time (seconds) • Pickup time (seconds) • CLI - calling number • Average queue time (seconds) • Average talk time (seconds) • Average pickup time (seconds) • Total number of answered calls
Format	• Table
Axis label	• N/A

Calculation rule	• Average X time (seconds): SUM(X time (seconds)) / COUNT(X time (seconds)) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches, tblusers, tblagentactivity
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_name, queue_id} • tblswitches = {switch_office_start, switch_office_end} • tblusers = {user_id, user_login} • tblagentactivity = {aa_id, aa_agent_id, aa_event_time, aa_event_type, aa_associated_data}

SQL Queries

Select missed call details for the selected date/time range

```
SELECT
  aa."aa_call_id",
  aa.aa_event_time,
  tblqueues.queue_name,
  aa."aa_agent_id",
  tblcalls."call_calling_number",
  EXTRACT (EPOCH FROM
   ((SELECT tblagentactivity.aa_event_time
     FROM tblagentactivity
     WHERE tblagentactivity.aa_event_type = 7
     AND tblagentactivity.aa_id = aa.aa_associated_data)
    - aa.aa_event_time ))
   AS timeInSec,
  u.user_login

FROM tblcalls, tblqueues, tblswitches s, tblagentactivity aa,
     tblusers u

WHERE
  "date"(aa.aa_event_time) >= "date" (?) /* from time */
  AND "date"(aa.aa_event_time) <= "date"(?) /* to date */
  AND "time"(aa.aa_event_time) >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(aa.aa_event_time) <=
          "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND aa."aa_call_id" = tblcalls."call_id"
  AND aa."aa_queue_id" = tblqueues.queue_id
  AND aa.aa_event_type = 6 /* missed call event */
  AND aa.agent_id = u.user_id
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
       "time"(aa.aa_event_time) >= "time"(s.switch_office_start)
        AND
      "time"(aa.aa_event_time) <= "time"(s.switch_office_end)
```

```
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(aa.aa_event_time) >= '00:00:00' AND
      "time"(aa.aa_event_time) <= '23:59:59'
    END )

ORDER BY aa.aa_event_time
```

Select abandoned call details for the selected date/time range

```
SELECT cc."cc_call_id",
    tblcalls."call_start_time",
    tblqueues.queue_name,
    cc."cc_agent_id",
    cc."cc_queue_time",
    cc."cc_talk_time",
    cc."cc_pickup_time",
    tblcalls."call_calling_number"

FROM tblcallscc cc, tblcalls, tblqueues, tblswitches

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
          "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND cc."cc_call_id" = tblcalls."call_id"
  AND cc."cc_talk_time" = 0
  AND cc."cc_agent_id" = 0
  AND cc."cc_callback" = 0
  AND cc."cc_queue_id" = tblqueues.queue_id
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
       "time"(tblcalls."call_start_time") >=
                       "time"(s.switch_office_start) AND
       "time"(tblcalls."call_start_time") <=
                        "time"(s.switch_office_end
      WHEN ? = 1 THEN /* Not Business hours only = 24/24 */
       "time"(tblcalls."call_start_time") >= '00:00:00' AND
       "time"(tblcalls."call_start_time") <= '23:59:59
         END )

ORDER BY tblcalls."call_start_time"
```

Select answered call details for the selected date/time range (the answered calls are grouped daily in the report)

```
SELECT cc."cc_call_id",
    "time"(tblcalls."call_start_time") AS call_start_time,
    tblqueues.queue_name,
    u."user_login",
    cc."cc_agent_id",
    cc."cc_queue_time",
    cc."cc_talk_time",
    cc."cc_pickup_time",
    tblcalls."call_calling_number",
    "date"(tblcalls."call_start_time")

FROM tblcallscc cc, tblcalls, tblusers u, tblqueues, tblswitches

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND cc."cc_call_id" = tblcalls."call_id"
  AND cc."cc_agent_id" = u."user_id"
  AND cc."cc_talk_time" > 0
  AND cc."cc_queue_id" = tblqueues.queue_id
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
       "time"(tblcalls."call_start_time") >=
                       "time"(s.switch_office_start) AND
       "time"(tblcalls."call_start_time") <=
                        "time"(s.switch_office_end
      WHEN ? = 1 THEN /* Not Business hours only = 24/24 */
       "time"(tblcalls."call_start_time") >= '00:00:00' AND
       "time"(tblcalls."call_start_time") <= '23:59:59
         END )

ORDER BY tblcalls."call_start_time"
```

Select all available days having answered calls in the selected date/time range (used for grouping the answered calls daily).

```
SELECT DISTINCT "date"(tblcalls."call_start_time")
             AS date_of_call

FROM tblcallscc cc, tblcalls, tblswitches s

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
          "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND cc."cc_call_id" = tblcalls."call_id"
  AND cc."cc_talk_time" > 0
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
       "time"(tblcalls."call_start_time") >=
                        "time"(s.switch_office_start) AND
       "time"(tblcalls."call_start_time") <=
                         "time"(s.switch_office_end
      WHEN ? = 1 THEN /* Not Business hours only = 24/24 */
       "time"(tblcalls."call_start_time") >= '00:00:00' AND
       "time"(tblcalls."call_start_time") <= '23:59:59
         END )

ORDER BY "date"(tblcalls."call_start_time")
```

Exception

Caused by the different tables and the complexity of this report when this report is empty the label “There is no data to report” is missed.

3.4.24 Contact Center Summary

Number of calls, average queue time, talk time and pickup time by queues for the specified date/time range.

Required input parameters	• From date • To date (until) • From time • To time • Business hours only (else 24/24) • Daily report
Output values	• Queue • Total number of calls per queue • Average pickup time (seconds) • Average talk time (seconds) • Average queue time (seconds) • Total number of calls • Total average pickup time, queue time and talk time
Format	• Table
Axis label	• N/A
Calculation rule	• Total number of calls: SUM(total number of calls per queue) • Average X time (seconds) : SUM(X time (seconds)) / COUNT(X time (seconds)) • Business hours only : switch_office_start <= call_start_time <= switch_office_end • 24/24 : 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches
Database table attributes	• tblcallscc = {cc_call_id, cc_agent_id, cc_talk_time, cc_queue_time, cc_pickup_time } • tblcalls = {call_id, call_start_time} • tblqueues = {queue_id, queue_name} • tblswitches = {switch_office_start, switch_office_end}

SQL Queries

Select number of calls, average queue time, talk time and pickup time by queues for the specified date/time range

```
SELECT tblqueues."queue_name",
  COUNT (cc."cc_call_id") AS "Total of calls",
  AVG (cc."cc_queue_time") AS "Queue time",
  AVG (cc."cc_pickup_time") AS "Pickup time",
  AVG (cc."cc_talk_time") AS "Talk time"

FROM tblcallscc cc, tblqueues, tblcalls, tblswitches s

WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
          "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblcalls."call_id" = cc."cc_call_id"
  AND cc."cc_queue_id" = tblqueues."queue_id"
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
       "time"(tblcalls."call_start_time") >=
                        "time"(s.switch_office_start) AND
       "time"(tblcalls."call_start_time") <=
                         "time"(s.switch_office_end)
      WHEN ? = 1 THEN /* Not Business hours only = 24/24 */
       "time"(tblcalls."call_start_time") >= '00:00:00' AND
       "time"(tblcalls."call_start_time") <= '23:59:59
         END )

GROUP BY tblqueues."queue_name"
```

Exception

N/A

3.4.25 Contact Center Summary 2

Number of calls, average queue time, talk time and pickup time, number of callback calls and queue time by queues for the specified date range.

Required input parameters	• From date • To date (until) • Business hours only (else 24/24) • Daily report
Output values	• Queue • Total number of calls per queue • Average pickup time (seconds) • Average talk time (seconds) • Average queue time (seconds) • Callback calls per queue • Queue time • Total number of calls • Total average pickup time, queue time and talk time • Total number of callback calls • Total queue time
Format	• Table
Axis label	• N/A
Calculation rule	• Callback call: {cc_callback = 1} • Total number of calls: SUM(total number of calls per queue) • Total number of callback calls: SUM(callback calls per queue) • Total queue time: SUM(queue time queue) • Average X time (seconds) : SUM(X time (seconds)) / COUNT(X time (seconds)) • Business hours only : switch_office_start <= call_start_time <= switch_office_end • 24/24 : 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches
Database table attributes	• tblcallscc = {cc_call_id, cc_agent_id, cc_talk_time, cc_queue_time, cc_pickup_time, cc_callback, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_id, queue_name} • tblswitches = {switch_office_start, switch_office_end}

SQL Queries

Select contact center summary details by queues for the specified date range

```
SELECT tblqueues."queue_name",
  COUNT (cc."cc_call_id") AS "Total of calls",
  AVG (cc."cc_queue_time") AS "Queue time",
  AVG (cc."cc_pickup_time") AS "Pickup time",
  AVG (cc."cc_talk_time") AS "Talk time",
  SUM (cc."cc_callback") AS "Callback calls",
  SUM (cc."cc_queue_time") AS "Total queue time"

FROM tblcallscc cc, tblqueues, tblcalls, tblswitches s

WHERE tblcalls."call_start_time" >= ? /* from time */
  AND tblcalls."call_start_time" <= " + INTERVAL '24 hours')
                                  /* to date */
  AND tblcalls."call_id" = cc."cc_call_id"
  AND cc."cc_queue_id" = tblqueues."queue_id"
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
       "time"(tblcalls."call_start_time") >=
                        "time"(s.switch_office_start) AND
       "time"(tblcalls."call_start_time") <=
                         "time"(s.switch_office_end)
      WHEN ? = 1 THEN /* Not Business hours only = 24/24 */
       "time"(tblcalls."call_start_time") >= '00:00:00' AND
       "time"(tblcalls."call_start_time") <= '23:59:59
         END )

GROUP BY tblqueues."queue_name"
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.4.26 Contact Center Summary – Answered Calls

Number of calls, average queue time, talk time and pickup time, number of callback calls and queue time of answered calls by queues for the specified date range.

Required input parameters	• From date • To date (until) • Business hours only (else 24/24) • Daily report
Output values	• Queue • Total number of calls per queue • Average pickup time (seconds) • Average talk time (seconds) • Average queue time (seconds) • Callback calls per queue • Queue time • Total number of calls • Total average pickup time, queue time and talk time • Total number of callback calls • Total queue time
Format	• Table
Axis label	• N/A
Calculation rule	• Answered call: {cc_talk_time > 0} • Callback call: {cc_callback = 1} • Total number of calls: SUM(total number of calls per queue) • Total number of callback calls: SUM(callback calls per queue) • Total queue time: SUM(queue time queue) • Average X time (seconds) : SUM(X time (seconds)) / COUNT(X time (seconds)) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches
Database table attributes	• tblcallscc = {cc_call_id, cc_agent_id, cc_talk_time, cc_queue_time, cc_pickup_time, cc_callback} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_id, queue_name} • tblswitches = {switch_office_start, switch_office_end}

SQL Queries

Select contact center summary details of answered calls for the specified date range.

```
SELECT tblqueues."queue_name",
  COUNT (cc."cc_call_id") AS "Total of calls",
  AVG (cc."cc_queue_time") AS "Queue time",
  AVG (cc."cc_pickup_time") AS "Pickup time",
  AVG (cc."cc_talk_time") AS "Talk time",
  SUM (cc."cc_callback") AS "Callback calls",
  SUM (cc."cc_queue_time") AS "Total queue time"

FROM tblcallscc cc, tblqueues, tblcalls, tblswitches s

WHERE tblcalls."call_start_time" >= ? /* from time */
  AND tblcalls."call_start_time" <= " + INTERVAL '24 hours')
                                  /* to date */
  AND tblcalls."call_id" = cc."cc_call_id"
  AND cc."cc_queue_id" = tblqueues."queue_id"
  AND cc."cc_talk_time" > 0
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
       "time"(tblcalls."call_start_time") >=
                        "time"(s.switch_office_start) AND
       "time"(tblcalls."call_start_time") <=
                         "time"(s.switch_office_end)
      WHEN ? = 1 THEN /* Not Business hours only = 24/24 */
       "time"(tblcalls."call_start_time") >= '00:00:00' AND
       "time"(tblcalls."call_start_time") <= '23:59:59
      END )

GROUP BY tblqueues."queue_name"
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.4.27 Contact Center Summary – Details

Number of calls (total number of calls, answered and abandoned calls), average queue time, talk time and pickup time by queues for the specified date/time range.

Required input parameters	• From date • To date (until) • From time • To time • Daily report
Output values	• Queue • Total number of calls per queue • Call ID • Average pickup time (seconds) • Average talk time (seconds) • Average queue time (seconds) • Callback calls • Answered calls • Abandoned calls • Total number of calls • Total average pickup time, queue time and talk time • Total number of answered calls • Total number of abandoned calls
Format	• Table
Axis label	• N/A
Calculation rule	• Answered calls: {cc_talk time> 0} • Abandoned calls: {cc_talk time = 0, cc_agent_id = 0, cc_callback = 0} • Total number of calls: SUM(total number of calls per queue) • Average X time (seconds) : SUM(X time (seconds)) / COUNT(X time (seconds))
Database tables	• tblcallscc, tblcalls, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_agent_id, cc_talk_time, cc_queue_time, cc_pickup_time, cc_callback, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_id, queue_name}

SQL Queries

Select contact center summary details by queues for the specified date/time range

```
SELECT tblqueues."queue_name",
  COUNT (cc."cc_call_id") AS "Total of calls",
  AVG (cc."cc_queue_time") AS "Queue time",
  AVG (cc."cc_pickup_time") AS "Pickup time",
  AVG (cc."cc_talk_time") AS "Talk time",
  SUM (cc."cc_callback") AS "Callback calls",
  (SELECT COUNT (tblcallscc."cc_call_id")
     FROM tblcalls, tblcallscc, tblqueues q
     WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
     AND
  "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
    AND "time"(tblcalls."call_start_time") >=
               "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
    AND "time"(tblcalls."call_start_time") <=
              "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
    AND tblcallscc."cc_queue_id" = q."queue_id"
    AND tblqueues."queue_name" = q."queue_name"
    AND tblcalls."call_id" = tblcallscc."cc_call_id"
    AND tblcallscc."cc_talk_time" > 0
  ) AS "Answered calls",
  (SELECT COUNT (tblcallscc."cc_call_id")
     FROM tblcalls, tblcallscc, tblqueues q
     WHERE
  "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
     AND
  "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
    AND "time"(tblcalls."call_start_time") >=
               "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
    AND "time"(tblcalls."call_start_time") <=
              "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
    AND tblcallscc."cc_queue_id" = q."queue_id"
    AND tblqueues."queue_name" = q."queue_name"
    AND tblcalls."call_id" = tblcallscc."cc_call_id"
    AND tblcallscc."cc_talk_time" = 0
    AND tblcallscc."cc_callback" = 0
    AND tblcallscc."cc_agent_id" = 0
  ) AS "Abandonded calls"


FROM tblcallscc cc, tblqueues, tblcalls c

WHERE
  "date"(c."call_start_time") >= "date" (?) /* from time */
  AND "date"(c."call_start_time") <= "date"(?) /* to date */
  AND "time"(c."call_start_time") >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(c."call_start_time") <=
          "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND c."call_id" = cc."cc_call_id"
  AND c."cc_queue_id" = tblqueues."queue_id"
```

```
GROUP BY tblqueues."queue_name"
```

Exception

N/A

3.4.28 Missed Calls Report

The report displays details for missed calls in the specified date/time range.

Required input parameters	• From date • To date (until) • From time • To time • Business hours only (else 24/24) • Daily report
Output values (the values are grouped by queues and daily)	• Call date • Queue name • Time of call • Call ID • CLI – calling number • Customer name • Customer company • Daily total number of missed calls by queue • Daily total number of missed calls (all queues) • Total number of missed calls
Format	• Table
Axis label	• N/A
Calculation rule	• Missed call = {tblagentactivity.aa_event_type = 6} • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches, tblagentactivity, tblcustomers
Database table attributes	• tblagentactivity = {aa_call_id, aa_queue_id, aa_event_type, aa_event_time} • tblcalls = {call_id,call_calling_number} • tblqueues = {queue_id, queue_name} • tblswitches = {switch_office_start, switch_office_end} • tblcustomers = {customer_firstname, customer_surname, customer_company, customer_business, customer_business2, customer_home, customer_mobile}

SQL Queries

Select calls details (call ID, call date, call time, calling number, queue name, customer name and company)

```
SELECT
  aa."aa_call_id",
  "date"(aa."aa_event_time"),
  "time"(aa."aa_event_time"),
  tblcalls."call_calling_number",
  tblqueues."queue_name",
  aa."aa_event_time",
    (SELECT tblcustomers."customer_company"
      FROM tblcustomers
      WHERE tblcalls."call_calling_number"
      IN
       (tblcustomers."customer_business",
        tblcustomers."customer_business2",
        tblcustomers."customer_home" ,
        tblcustomers."customer_mobile" )
      LIMIT 1 )
      AS "customer_company",
    (SELECT tblcustomers."customer_firstname" ||
      tblcustomers."customer_surname"
      FROM tblcustomers
      WHERE tblcalls."call_calling_number"
      IN
       (tblcustomers."customer_business",
        tblcustomers."customer_business2",
        tblcustomers."customer_home" ,
        tblcustomers."customer_mobile" )
      LIMIT 1 )
      AS "customer"

FROM tblcalls, tblqueues, tblswitches s, tblagentactivity aa

WHERE aa."aa_call_id" = tblcalls."call_id"
  AND aa."aa_queue_id" = tblqueues."queue_id"
  AND "date"(aa."aa_event_time") >= "date"(?) /* from date */
  AND "date"(aa."aa_event_time") <= "date"(?) /* to date */
  AND "time"(aa."aa_event_time") >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(aa."aa_event_time") <=
           "time"(to_timestamp(?,'HH:MI:SS'))/* to time */
  AND aa."aa_event_type" = 6 /* Missed call event */
  AND (CASE
    WHEN ? = 1 THEN /* Business hours only */
    "time"(aa."aa_event_time") >= "time"(s.switch_office_start)
  AND
     "time"(aa."aa_event_time") <= "time"(s.switch_office_end)
      WHEN ? != 1 THEN /* 24/24 */
        "time"(aa."aa_event_time") >= '00:00:00' AND
        "time"(aa."aa_event_time") <= '23:59:59'
    END )

ORDER BY (aa."aa_event_time")
```

Select all available days having missed calls in the specified date/time range and the daily total number of missed calls

```
SELECT
  DISTINCT "date"(aa."aa_event_time") AS "AllDates",
  COUNT (aa."aa_call_id")

FROM tblqueues, tblswitches s, tblagentactivity aa

WHERE aa."aa_call_id" = tblcalls."call_id"
  AND aa."aa_queue_id" = tblqueues."queue_id"
  AND "date"(aa."aa_event_time") >= "date"(?) /* from date */
  AND "date"(aa."aa_event_time") <= "date"(?) /* to date */
  AND "time"(aa."aa_event_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(aa."aa_event_time") <=
            "time"(to_timestamp(?,'HH:MI:SS'))/* to time */
  AND aa."aa_event_type" = 6 /* Missed call event */
  AND (CASE
    WHEN ? = 1 THEN /* Business hours only */
    "time"(aa."aa_event_time") >= "time"(s.switch_office_start)
  AND
     "time"(aa."aa_event_time") <= "time"(s.switch_office_end)
      WHEN ? != 1 THEN /* 24/24 */
        "time"(aa."aa_event_time") >= '00:00:00' AND
        "time"(aa."aa_event_time") <= '23:59:59'
    END )

GROUP BY "date"(aa."aa_event_time")

ORDER BY "date"(aa."aa_event_time")
```

Select all available queues (used for daily grouping by queue) per days in the specified date/time range

```
SELECT
  DISTINCT tblqueues."queue_name",
  "date"(aa."aa_event_time")

FROM tblqueues, tblswitches s, tblagentactivity aa

WHERE aa."aa_call_id" = tblcalls."call_id"
  AND aa."aa_queue_id" = tblqueues."queue_id"
  AND "date"(aa."aa_event_time") >= "date"(?) /* from time */
  AND "date"(aa."aa_event_time") <= "date"(?) /* to date */
  AND "time"(aa."aa_event_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(aa."aa_event_time") <=
            "time"(to_timestamp(?,'HH:MI:SS'))/* to time */
  AND aa."aa_event_type" = 6 /* missed call event */
  AND (CASE
    WHEN ? = 1 THEN /* Business hours only */
    "time"(aa."aa_event_time") >= "time"(s.switch_office_start)
  AND
     "time"(aa."aa_event_time") <= "time"(s.switch_office_end)
      WHEN ? != 1 THEN /* 24/24 */
        "time"(aa."aa_event_time") >= '00:00:00' AND
        "time"(aa."aa_event_time") <= '23:59:59'
    END )

ORDER BY "date"(aa."aa_event_time")
```

Select total number of missed calls in the specified date/time range

```
SELECT COUNT (aa."aa_call_id")

FROM  tblswitches s, tblagentactivity aa

WHERE "date"(aa."aa_event_time") >= "date"(?) /* from time */
  AND "date"(aa."aa_event_time") <= "date"(?) /* to date */
  AND "time"(aa."aa_event_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(aa."aa_event_time") <=
            "time"(to_timestamp(?,'HH:MI:SS'))/* to time */
  AND aa."aa_event_type" = 6 /* missed call event */
  AND (CASE
    WHEN ? = 1 THEN /* Business hours only */
    "time"(aa."aa_event_time") >= "time"(s.switch_office_start)
  AND
     "time"(aa."aa_event_time") <= "time"(s.switch_office_end)
      WHEN ? != 1 THEN /* 24/24 */
        "time"(aa."aa_event_time") >= '00:00:00' AND
        "time"(aa."aa_event_time") <= '23:59:59'
    END )
```

Exception

N/A

3.4.29 Missed Calls Summary (Per Agent)

The report displays missed calls summary details (number of calls and percent of all missed calls) per agent for calls in the specified date/time range.

Required input parameters	• From date • To date (until) • Daily report
Output values (the values are grouped by queues and daily)	• Agent • Number of missed calls (per agent) • Percentage of all missed calls • Total number of missed calls (all agents)
Format	• Table
Axis label	• N/A
Calculation rule	• Missed call = { tblagentactivity.aa_event_type = 6 } (also in that case the aa_call_id is greater then 0)
Database tables	• tblusers, tblagentactivity
Database table attributes	• tblagentactivity = { aa_call_id, aa_agent_id, aa_event_type, aa_event_type } • tblusers = {user_firstname,user_surname,user_id, user_login}

SQL Queries

Select count of missed calls per agent and agent details : first name, last name and login

```
SELECT COUNT (aa.aa_call_id) AS "Missed Calls",
  u.user_id,
  u.user_firstname,
  u.user_surname,
  u.user_login

FROM tblagentactivity aa, tblusers u

WHERE aa.aa_event_time >= ? /*from date*/
  AND aa.aa_event_time <= ("date"(?) + interval '24 hours')
    /*to date*/
  AND aa.aa_agent_id > 0
  AND aa.aa_call_id > 0
  AND aa.aa_agent_id = u.user_id
  AND aa.aa_event_type=6 /*Missed call event*/

GROUP BY u.user_id,u.user_firstname,u.user_surname,u.user_login

ORDER BY u.user_firstname,u.user_surname,u.user_login
```

Exception

One call can be missed more then once by one or other agent(s)!

3.4.30 Missed Calls Summary (Per Queue)

The report displays missed calls summary details (number of calls and percent of all missed calls) per agent for calls in the specified date/time range; these details are grouped per queue.

Required input parameters	• From date • To date (until) • Daily report
Output values (the values are grouped by queues and daily)	• Queue • Agent • Number of missed calls (per agent) • Percentage of all missed calls • Total number of missed calls per queue • Grand total of missed calls (all queues)
Format	• Table
Axis label	• N/A
Calculation rule	• Missed call = { tblagentactivity.aa_event_type = 6 } (also in that case the aa_call_id is greater then 0)
Database tables	• tblusers, tblagentactivity, tblqueues
Database table attributes	• tblagentactivity = { aa_call_id, aa_agent_id, aa_event_type, aa_event_type, aa_queue_id} • tblusers = {user_firstname, user_surname,user_id, user_login} • tblqueues = {queue_id, queue_name}

SQL Queries

Select count of missed calls per agent and agent details : first name, last name and login (the queue id is used for grouping the output per queue)

```
SELECT COUNT (aa.aa_call_id) AS "Missed Calls",
  u.user_id,
  u.user_firstname,
  u.user_surname,
  u.user_login
  aa.aa_queue_id

FROM tblagentactivity aa, tblusers u

WHERE aa.aa_event_time >= ? /*from date*/
  AND aa.aa_event_time <= ("date"(?) + interval '24 hours')
    /*to date*/
  AND aa.aa_agent_id > 0
  AND aa.aa_call_id > 0
  AND aa.aa_agent_id = u.user_id
  AND aa.aa_event_type=6 /*Missed call event*/

GROUP BY u.user_id,u.user_firstname,u.user_surname,u.user_login,
  aa.aa_queue_id

ORDER BY u.user_firstname,u.user_surname,u.user_login
```

Select all queues with missed calls in the specified date range (used for grouping the information per queue)

```
SELECT DISTINCT (queue_id), queue_name

FROM tblagentactivity aa, tblqueues

WHERE aa.aa_event_time >= ? /*from date*/
  AND aa.aa_event_time <= ("date"(?) + interval '24 hours')
    /*to date*/
  AND aa.aa_agent_id > 0
  AND aa.aa_call_id > 0
  AND aa.aa_queue_id = queue_id
  AND aa.aa_event_type=6 /*Missed call event*/

ORDER BY queue_name
```

Select total number of missed calls (all queues)

```
SELECT COUNT (aa.aa_call_id)

FROM tblagentactivity aa

WHERE aa.aa_event_time >= ? /*from date*/
  AND aa.aa_event_time <= ("date"(?) + interval '24 hours')
    /*to date*/
  AND aa.aa_agent_id > 0
  AND aa.aa_call_id > 0
  AND aa.aa_event_type=6 /*Missed call event*/
  AND aa.aa_queue_id IN (SELECT queue_id FROM tblqueues)
```

Exception

One call can be missed more then once by one or other agent(s)!

3.5 Report Group - Other

All predefined report templates of this report group are described below.

3.5.1 Calls History Per User

The report displays call history information for the specified user in the selected date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Call date • Start time • Calling number • Called number • Direction I/O= (inbound/outbound) • Talk time • Daily total number of calls • Daily total talk time • Total number of calls • Total talk time
Format	• Table
Axis label	• N/A
Calculation rule	• Outbound: ch_direction = 1 • Inbound: ch_direction = 0 • Daily total number of calls: COUNT(number of calls by day) • Daily total talk times: SUM(talk time by day)
Database tables	• tblcallhistory, tblusers, tbldepartments, tblqueues
Database table attributes	• tblcallhistory = {ch_call_id, ch_start_time, ch_talk_time_seconds, ch_called_number, ch_calling_number, ch_direction, ch_user_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_extension} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_id, u.user_surname, u.user_firstname, u.user_login
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login, (SELECT CASE WHEN
tbldepartments.department_name = 'Unknown’
  THEN ’’ ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id ) AS
department_name

FROM tblusers u

WHERE u.user_id = ? /* user login */
```

Select all available days having call history information for the specified user in the selected date range

```
SELECT
  DISTINCT ("date"(tblcallhistory.ch_start_time)) AS "Date of day"

FROM tblcallhistory

WHERE tblcallhistory.ch_start_time >= ? /* from time */
    AND tblcallhistory.ch_start_time <= ("date" (?) + INTERVAL
                           '24 hours') /* to date */
    AND tblcallhistory."ch_user_id" = ? /* user login */
```

Select call history details

```
SELECT tblcallhistory.ch_user_id,
  tblcallhistory.ch_calling_number,
  tblcallhistory.ch_called_number,
  tblcallhistory.ch_direction,
  "date"(tblcallhistory.ch_start_time) AS "Date of day",
  "time"(tblcallhistory.ch_start_time) AS "Start time",
  "time"(tblcallhistory.ch_end_time) AS "End time",
  tblcallhistory.ch_talk_time_seconds

FROM tblcallhistory

WHERE tblcallhistory.ch_start_time >= ? /* from time */
    AND tblcallhistory.ch_start_time <= ("date" (?) + INTERVAL
                           '24 hours') /* to date */
    AND tblcallhistory."ch_user_id" = ? /* user login */

ORDER BY "time"(tblcallhistory.ch_end_time)
```

Select call history grand totals (total talk time, total number of calls) for the specified user in the selected date range

```
SELECT SUM (ch.ch_talk_time_seconds) AS totalduration,
  COUNT (ch.ch_call_id)

FROM tblcallhistory ch, tblusers u

WHERE ch.ch_start_time >= ? /* from time */
    AND ch.ch_start_time <= ("date" (?) + INTERVAL '24 hours')
                            /* to date */
    AND ch.ch_user_id = u.user_id
    AND u.user_id = ? /* user login */
```

Exception

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used:

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • sec. = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.5.2 Default Break Information

The report displays the default break information (break name and default break interval in minutes).

Required input parameters	• N/A
Output values	• N/A
Format	• Graphic

Axis label	• Horizontal: break default interval (minutes) • Vertical: break name
Calculation rule	• N/A
Database tables	• tblbreakscc
Database table attributes	• tblbreakscc = {break_name, break_default_interval_minutes}

SQL Queries

Select break information

```
SELECT (break_name, break_default_interval_minutes

FROM tblbreakscc

ORDER BY break_name
```

Exception

N/A

3.5.3 External Directory User Details

The report displays information about the user external directory (user company, first name, surname, business phone 1, business phone 2, home phone and mobile phone).

Required input parameters	• N/A
Output values	• User Company • First name • Surname • Business phone 1 • Business phone 2 • Home phone • Mobile Phone
Format	• Table
Axis label	• N/A
Calculation rule	• N/A
Database tables	• tblcustomers
Database table attributes	• tblcustomers = {customer_firstname, customer_surname, customer_company, customer_business, customer_business2, customer_home, customer_mobile}

SQL Queries

Select external directory

```
SELECT
  tblcustomers.customer_firstname,
  tblcustomers.customer_surname,
  TRIM (tblcustomers.customer_company) AS "Company Name",
  tblcustomers.customer_business,
  tblcustomers.customer_business2,
  tblcustomers.customer_home,
  tblcustomers.customer_mobile

FROM tblcustomers

ORDER BY
  tblcustomers.customer_company, tblcustomers.customer_firstname
```

Exception

N/A

3.5.4 Fax Journal – Received Faxes (By User)

The report shows details of the received faxes for a specified user in the selected date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Time • Fax Group • Contact (last name, first name) • Company • CLI – fax calling number • Fax Status • Fax Pages • Total number of daily received faxes and fax pages • Total number of received faxes • Total number of received fax pages
Format	• Table
Axis label	• N/A
Calculation rule	• Fax status: 0 - new, 1 - read, 2 - deleted
Database tables	• tblfaxes, tblfaxgroups, tbldepartments, tblusers, tblfaxgroupsusers, tblcontactcache

Database table attributes	• tblcontactcache = {cc_contact_firstname, cc_contact_surname, cc_contact_company, cc_entry_id} • tblfaxgroupsusers = {fgu_fg_id, fgu_user_id} • tblfaxgroups = {fg_name, fg_id} • tblfaxes = {fax_id, fax_date, fax_calling_number, fax_status, fax_pages, fax_fg_id, fax_contact_cache_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT
  u.user_firstname,
  u.user_surname,
  u.user_extension,
  u.user_email,
  u.user_login,
  (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select received faxes details for the selected user in the specified date range

```
SELECT "date" (tblfaxes.fax_date),
  "time" (tblfaxes.fax_date),
  tblfaxes.fax_calling_number,
  tblfaxes.fax_status,
  tblfaxes.fax_pages,
  tblfaxgroups.fg_name,
  tblcontactcache.cc_contact_firstname,
  tblcontactcache.cc_contact_surname,
  tblcontactcache.cc_contact_company

FROM tblfaxes

LEFT OUTER JOIN tblfaxgroups ON
    tblfaxes.fax_fg_id = tblfaxgroups.fg_id
  LEFT OUTER JOIN tblcontactcache ON tblfaxes.fax_contact_cache_id
= tblcontactcache.cc_entry_id

WHERE tblfaxes.fax_date >= ? /* from time */
  AND tblfaxes.fax_date" <= " + INTERVAL '24 hours') /* to date */
  AND tblfaxes.fax_fg_id IN
    (SELECT fgu_fg_id FROM tblfaxgroupsusers
       WHERE fgu_user_id IN
        (SELECT user_id FROM tblusers
           WHERE user_login = ? )) /* user login */

ORDER BY tblfaxes.fax_date
```

Select all available days having received faxes for the selected user in the specified date range

```
SELECT DISTINCT ("date"(tblfaxes.fax_date))

FROM tblfaxes

WHERE tblfaxes.fax_date >= ? /* from time */
  AND tblfaxes.fax_date" <= " + INTERVAL '24 hours') /* to date */
  AND tblfaxes.fax_fg_id IN
    (SELECT fgu_fg_id FROM tblfaxgroupsusers
       WHERE fgu_user_id IN
        (SELECT user_id FROM tblusers
           WHERE user_login = ? )) /* user login */

GROUP BY "date"(tblfaxes.fax_date)
```

Select total number of received faxes and fax pages for the selected user in the specified date range

```
SELECT
  COUNT (tblfaxes.fax_id) AS "Number of Faxes",
  SUM (tblfaxes.fax_pages) AS "Total number of fax pages"

FROM tblfaxes

WHERE tblfaxes.fax_date >= ? /* from time */
  AND tblfaxes.fax_date" <= " + INTERVAL '24 hours') /* to date */
  AND tblfaxes.fax_fg_id IN
    (SELECT fgu_fg_id FROM tblfaxgroupsusers
       WHERE fgu_user_id IN
        (SELECT user_id FROM tblusers
           WHERE user_login = ? )) /* user login */
```

Exception

N/A

3.5.5 Fax Journal – Sent Faxes (By User)

The report shows details of the sent faxes for a specified user in the selected date range.

Required input parameters	• From date • To date (until) • User • Daily report
Output values (the values are grouped daily)	• Time • Fax Group • Contact (last name, first name) • Company • Destination • Status • Pages • Total number of daily sent faxes and fax pages • Total number of sent faxes • Total number of sent fax pages
Format	• Table
Axis label	• N/A

Calculation rule	tblspooling.sp_spool_recipient_class: • 0 = A contact item can therefore be found by tblusers.user_id • 1 = A contact item can therefore be found by tblpersonaldirectory.contact_id • 2 = A contact item can therefore be found by tblcustomers.customer_id Spool status: • 0 = unknown • 1 = accepted • 2 = preprocessing • 3 = preprocessed • 4 = queued • 5 = transport setup • 6 = processing • 7 = failed • 8 = completed
Database tables	• tblspooling, tblfaxgroups, tbldepartments, tblusers, tblfaxgroupsusers, tblpersonaldirectory, tblcustomers, tblspooldocuments
Database table attributes	• tblpersonaldirectory = {contact_firstname, contact_surname, contact_company, contact_id} • tblcustomers = {customer_firstname, customer_surname, customer_company, customer_id} • tblfaxgroupsusers = {fgu_fg_id, fgu_user_id} • tblfaxgroups = {fg_name, fg_id} • tblspooling = {sp_job_id, sp_spooled_at, sp_spool_recipient_class, sp_spool_recipient_id, sp_spool_recipient_address, sp_spool_state, sp_spool_recipient_class, sp_spool_group_id sp_spool_id, sp_spool_owner_id } • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_name, department_id} • tblspooldocuments = {sd_spool_id, sd_pages}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT
  u.user_firstname,
  u.user_surname,
  u.user_extension,
  u.user_email,
  u.user_login,
  (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
  THEN '' ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id
  ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select total number of sent faxes and fax pages for the selected user in the specified date range

```
SELECT
  COUNT (tblspooling.sp_job_id) AS "Number of Faxes",
  SUM (tblspooldocuments.sd_pages) AS "Total number of fax pages"

FROM tblspooling

LEFT OUTER JOIN tblspooldocuments ON
  tblspooling.sp_spool_id = tblspooldocuments.sd_spool_id

WHERE
   to_timestamp(sp_spooled_at, 'DDMMyyyyHHMI') >= ? /* from time */
  AND to_timestamp(sp_spooled_at, 'DDMMyyyyHHMI') <= ("date"(?)
                            + INTERVAL '24 hours') /* to date */
  AND (tblspooling.sp_spool_owner_id =
        (SELECT user_id FROM tblusers
           WHERE user_login = ?
         ) OR tblspooling.sp_spool_group_id IN /* user login */
           (SELECT fgu_fg_id FROM tblfaxgroupsusers
             WHERE fgu_user_id =
              (SELECT user_id FROM tblusers
                WHERE user_login = ? ))) /* user login */
```

Select sent faxes details for the selected user in the specified date range

```
SELECT tblspooling.sp_job_id,
  "date" (to_timestamp(sp_spooled_at, 'DDMMyyyyHHMI')) AS "date",
  "time" (to_timestamp(sp_spooled_at, 'DDMMyyyyHHMI')) AS "time",
  tblfaxgroups.fg_name AS "group name",
  tblspooling.sp_spool_recipient_class,
  tblspooling.sp_spool_recipient_id,
  tblspooling.sp_spool_recipient_address AS "destination",
  tblspooling.sp_spool_state AS "status",
  tblspooldocuments.sd_pages AS "pages",
  CASE tblspooling.sp_spool_recipient_class
   WHEN 0 THEN /* users */
   (SELECT user_surname ||' '|| user_firstname FROM tblusers
       WHERE user_id = tblspooling.sp_spool_recipient_id)
   WHEN 1 THEN /* personal */
   (SELECT contact_surname ||' '|| contact_firstname || ' - '
           || contact_company FROM tblpersonaldirectory
       WHERE contact_id = tblspooling.sp_spool_recipient_id)
   WHEN 2 THEN /* customer */
   (SELECT customer_surname ||' '|| customer_firstname || ' - '
           || customer_company FROM tblcustomers
       WHERE customer_id = tblspooling.sp_spool_recipient_id)
  ELSE '-' /* unknown */
  END AS "contact"


FROM tblspooling

LEFT OUTER JOIN tblfaxgroups ON
    tblspooling.sp_spool_group_id = tblfaxgroups.fg_id  LEFT OUTER
JOIN tblspooldocuments ON
    tblspooling.sp_spool_id = tblspooldocuments.sd_spool_id

WHERE
   to_timestamp(sp_spooled_at, 'DDMMyyyyHHMI') >= ? /* from time */
  AND to_timestamp(sp_spooled_at, 'DDMMyyyyHHMI') <= ("date"(?)
                            + INTERVAL '24 hours') /* to date */
  AND (tblspooling.sp_spool_owner_id =
        (SELECT user_id FROM tblusers
           WHERE user_login = ?
         ) OR tblspooling.sp_spool_group_id IN /* user login */
           (SELECT fgu_fg_id FROM tblfaxgroupsusers
             WHERE fgu_user_id =
              (SELECT user_id FROM tblusers
                WHERE user_login = ? ))) /* user login */
```

Select all available days having sent faxes for selected user in the specified date range

```
SELECT DISTINCT
  ("date"(to_timestamp(sp_spooled_at,'DDMMyyyyHHMI'))) AS "date"

FROM tblspooling

WHERE
   to_timestamp(sp_spooled_at, 'DDMMyyyyHHMI') >= ? /* from time */
  AND to_timestamp(sp_spooled_at, 'DDMMyyyyHHMI') <= ("date"(?)
                            + INTERVAL '24 hours') /* to date */
  AND (tblspooling.sp_spool_owner_id =
        (SELECT user_id FROM tblusers
           WHERE user_login = ?
         ) OR tblspooling.sp_spool_group_id IN /* user login */
           (SELECT fgu_fg_id FROM tblfaxgroupsusers
             WHERE fgu_user_id =
              (SELECT user_id FROM tblusers
                WHERE user_login = ? ))) /* user login */
```

Exception

N/A

3.5.6 Fax Transmission Report

The report shows fax details including the fax itself.

Required input parameters	• From date • Until (to date) • Business hours only (else 24/24) • Daily report
Output values	• Fax ID
Format	• Text (fax details) + Fax (embedded picture)
Axis label	• N/A
Calculation rule	Recipient class (sp_spool_recipient_class): • 0 = user • 1 = personal directory • 2 = customers
Database tables	• tblspooling, tblusers, tblpersonaldirectory, tblcustomers, tblfaxgroups, tblspooldocuments

Database table attributes	• tblspooling = {sp_spool_id, sp_spooled_at, sp_spool_owner_id, sp_spool_group_id, s[_spool_recipient_class, sp_spool_recipient_address, sp_spool_state, sp_spool_tx_duration, sp_spool_failure_code} • tblusers = {user_id, user_firstname, user_surname} • tblpersonaldirectory = {contact_surname, contact_firstname, contact_company, contact_id} • tblcustomers = {customer_firstname, customer_surname, customer_company, customer_id} • tblfaxgroups = {fg_name, fg_id} • tblspooldocuments = {sd_pages, sd_filename, sd_spool_id} • tblfaxgroupsusers = {fgu_user_id}

SQL Queries

Select all sent faxes IDs (used for selecting the report fax ID)

```
SELECT tblspooling.sp_spool_id
FROM tblspooling
ORDER BY tblspooling.sp_spool_id
```

Select fax details (user, fax group, contact, destination, date, pages, status, duration, failure code)

```
SELECT tblspooling.sp_spool_id,
  (to_timestamp(sp_spooled_at, 'DDMMyyyyHHMI')) AS "date",
  tblfaxgroups.fg_name AS "group name",
  tblspooling.sp_spool_owner_id AS "owner_id",
  CASE WHEN (tblspooling.sp_spool_owner_id
    IN (SELECT user_id FROM tblusers))
  THEN (SELECT user_firstname ||' '|| user_surname
    FROM tblusers
    WHERE user_id = tblspooling.sp_spool_owner_id)
  WHEN (tblspooling.sp_spool_group_id
    IN (SELECT fgu_user_id FROM tblfaxgroupsusers))
  THEN (SELECT user_firstname ||' '|| user_surname
    FROM tblusers
    WHERE user_id = tblspooling.sp_spool_owner_id)
  ELSE '-'
  END AS "user",
tblspooling.sp_spool_recipient_class,
--tblspooling.sp_spool_recipient_id,
getcallnumber (tblspooling.sp_spool_recipient_address)
  AS "destination",
tblspooling.sp_spool_state AS "status",
tblspooldocuments.sd_pages AS "pages",
  CASE tblspooling.sp_spool_recipient_class
  WHEN 0 THEN
--User
  (SELECT user_surname ||' '|| user_firstname
    FROM tblusers
    WHERE user_id = "int4"(tblspooling.sp_spool_recipient_id))
```

```
  WHEN 1 THEN
--Personal
  (SELECT contact_surname ||' '|| contact_firstname
    ||' '|| contact_company
    FROM tblpersonaldirectory
    WHERE contact_id = "int4"(tblspooling.sp_spool_recipient_id))
  WHEN 2 THEN
--Customer
  (SELECT customer_surname ||' '|| customer_firstname
    ||' '|| customer_company
    FROM tblcustomers
    WHERE customer_id = "int4"(tblspooling.sp_spool_recipient_id))
  ELSE
--Unknown
  '-'
  END AS "contact",
  tblspooldocuments.sd_filename,
  tblspooling.sp_spool_tx_duration,
  tblspooling.sp_spool_failure_code
  FROM tblspooling
  LEFT OUTER JOIN tblfaxgroups ON tblspooling.sp_spool_group_id =
    tblfaxgroups.fg_id
  LEFT OUTER JOIN tblspooldocuments ON tblspooling.sp_spool_id =
    tblspooldocuments.sd_spool_id
  WHERE tblspooling.sp_spool_id = ? /* Fax ID*/
```

Exception

N/A

3.5.7 Incoming Calls Report – Hourly

The report shows the number of incoming calls per hour and daily.

Required input parameters	• From date • Until (to date) • Daily report
Output values	• Number of abandoned calls per hour 0-24 • Total number of calls per day (horizontally) • Total number of calls per hour (for all available days - vertically)
Format	• Table
Axis label	• N/A
Calculation rule	• N/A
Database tables	• tblcallhistory
Database table attributes	• tblcallhistory = {ch_start_time, ch_direction}

SQL Queries

Count of incoming calls per hour (0-24) and per day (date) for the specified date range

```
SELECT DATE,
  SUM (CASE HOUR WHEN 0 THEN nb_calls ELSE 0 END) AS "0-1",
  SUM (CASE HOUR WHEN 1 THEN nb_calls ELSE 0 END) AS "1-2",
  SUM (CASE HOUR WHEN 2 THEN nb_calls ELSE 0 END) AS "2-3",
  SUM (CASE HOUR WHEN 3 THEN nb_calls ELSE 0 END) AS "3-4",
  SUM (CASE HOUR WHEN 4 THEN nb_calls ELSE 0 END) AS "4-5",
  SUM (CASE HOUR WHEN 5 THEN nb_calls ELSE 0 END) AS "5-6",
  SUM (CASE HOUR WHEN 6 THEN nb_calls ELSE 0 END) AS "6-7",
  SUM (CASE HOUR WHEN 7 THEN nb_calls ELSE 0 END) AS "7-8",
  SUM (CASE HOUR WHEN 8 THEN nb_calls ELSE 0 END) AS "8-9",
  SUM (CASE HOUR WHEN 9 THEN nb_calls ELSE 0 END) AS "9-10",
  SUM (CASE HOUR WHEN 10 THEN nb_calls ELSE 0 END) AS "10-11",
  SUM (CASE HOUR WHEN 11 THEN nb_calls ELSE 0 END) AS "11-12",
  SUM (CASE HOUR WHEN 12 THEN nb_calls ELSE 0 END) AS "12-13",
  SUM (CASE HOUR WHEN 13 THEN nb_calls ELSE 0 END) AS "13-14",
  SUM (CASE HOUR WHEN 14 THEN nb_calls ELSE 0 END) AS "14-15",
  SUM (CASE HOUR WHEN 15 THEN nb_calls ELSE 0 END) AS "15-16",
  SUM (CASE HOUR WHEN 16 THEN nb_calls ELSE 0 END) AS "16-17",
  SUM (CASE HOUR WHEN 17 THEN nb_calls ELSE 0 END) AS "17-18",
  SUM (CASE HOUR WHEN 18 THEN nb_calls ELSE 0 END) AS "18-19",
  SUM (CASE HOUR WHEN 19 THEN nb_calls ELSE 0 END) AS "19-20",
  SUM (CASE HOUR WHEN 20 THEN nb_calls ELSE 0 END) AS "20-21",
  SUM (CASE HOUR WHEN 21 THEN nb_calls ELSE 0 END) AS "21-22",
  SUM (CASE HOUR WHEN 22 THEN nb_calls ELSE 0 END) AS "22-23",
  SUM (CASE HOUR WHEN 23 THEN nb_calls ELSE 0 END) AS "23-0",
FROM (
  SELECT "date" (ch.ch_start_time) AS date,
    EXTRACT (HOUR FROM ch.ch_start_time) AS hour,
    COUNT (*) AS nb_calls,
  FROM tblcallhistory ch
  WHERE ch.ch_start_time >= ? /* From date */
    AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
      /* To date */
    AND ch.ch_direction = 0
  GROUP BY "date" (ch.ch_start_time), EXTRACT
    (HOUR FROM ch.ch_start_time)
  ORDER BY "date" (ch.ch_start_time), EXTRACT
    (HOUR FROM ch.ch_start_time)) AS weekly_data
GROUP BY DATE
ORDER BY DATE
```

Exception

N/A

3.5.8 Incoming Calls Report – Hourly Per Weekday

The report shows the number of incoming calls per hour and weekday.

Required input parameters	• From date • Until (to date) • Daily report
Output values	• Number of abandoned calls per hour 0-24 • Total number of calls per weekday (horizontally) • Total number of calls per hour (for all available days - vertically)
Format	• Table
Axis label	• N/A
Calculation rule	• N/A
Database tables	• tblcallhistory
Database table attributes	• tblcallhistory = {ch_start_time, ch_direction}

SQL Queries

Count of incoming calls per hour (0-24) and per day of the week for the specified date range

```
SELECT DATE,
  SUM (CASE HOUR WHEN 0 THEN nb_calls ELSE 0 END) AS "0-1",
  SUM (CASE HOUR WHEN 1 THEN nb_calls ELSE 0 END) AS "1-2",
  SUM (CASE HOUR WHEN 2 THEN nb_calls ELSE 0 END) AS "2-3",
  SUM (CASE HOUR WHEN 3 THEN nb_calls ELSE 0 END) AS "3-4",
  SUM (CASE HOUR WHEN 4 THEN nb_calls ELSE 0 END) AS "4-5",
  SUM (CASE HOUR WHEN 5 THEN nb_calls ELSE 0 END) AS "5-6",
  SUM (CASE HOUR WHEN 6 THEN nb_calls ELSE 0 END) AS "6-7",
  SUM (CASE HOUR WHEN 7 THEN nb_calls ELSE 0 END) AS "7-8",
  SUM (CASE HOUR WHEN 8 THEN nb_calls ELSE 0 END) AS "8-9",
  SUM (CASE HOUR WHEN 9 THEN nb_calls ELSE 0 END) AS "9-10",
  SUM (CASE HOUR WHEN 10 THEN nb_calls ELSE 0 END) AS "10-11",
  SUM (CASE HOUR WHEN 11 THEN nb_calls ELSE 0 END) AS "11-12",
  SUM (CASE HOUR WHEN 12 THEN nb_calls ELSE 0 END) AS "12-13",
  SUM (CASE HOUR WHEN 13 THEN nb_calls ELSE 0 END) AS "13-14",
  SUM (CASE HOUR WHEN 14 THEN nb_calls ELSE 0 END) AS "14-15",
  SUM (CASE HOUR WHEN 15 THEN nb_calls ELSE 0 END) AS "15-16",
  SUM (CASE HOUR WHEN 16 THEN nb_calls ELSE 0 END) AS "16-17",
  SUM (CASE HOUR WHEN 17 THEN nb_calls ELSE 0 END) AS "17-18",
  SUM (CASE HOUR WHEN 18 THEN nb_calls ELSE 0 END) AS "18-19",
  SUM (CASE HOUR WHEN 19 THEN nb_calls ELSE 0 END) AS "19-20",
  SUM (CASE HOUR WHEN 20 THEN nb_calls ELSE 0 END) AS "20-21",
  SUM (CASE HOUR WHEN 21 THEN nb_calls ELSE 0 END) AS "21-22",
  SUM (CASE HOUR WHEN 22 THEN nb_calls ELSE 0 END) AS "22-23",
  SUM (CASE HOUR WHEN 23 THEN nb_calls ELSE 0 END) AS "23-0",
```

```
FROM (
  SELECT
    EXTRACT (DOW FROM ch.ch_start_time) AS day_of_week,
    EXTRACT (HOUR FROM ch.ch_start_time) AS hour,
    COUNT (*) AS nb_calls,
  FROM tblcallhistory ch
  WHERE ch.ch_start_time >= ? /* From date */
    AND ch.ch_start_time <= ("date"(?) + INTERVAL '24 hours')
      /* To date */
    AND ch.ch_direction = 0
  GROUP BY EXTRACT (DOW FROM ch.ch_start_time), EXTRACT
    (HOUR FROM ch.ch_start_time)
  ORDER BY EXTRACT (DOW FROM ch.ch_start_time), EXTRACT
    (HOUR FROM ch.ch_start_time)) AS weekly_data
GROUP BY day_of_week
ORDER BY day_of_week
```

Exception

N/A

3.5.9 Internal Directory User Details

The report shows information about the user internal directory.

Required input parameters	• N/A
Output values	• User • Email • External 1 • External 2 • Mobile phone • Home phone • Fax
Format	• Table
Axis label	• N/A
Calculation rule	• User: first name, surname (if the user first name and surname are empty the user login is used)
Database tables	• tblusers
Database table attributes	• tblusers = {user_firstname, user_surname, user_email, user_contact_external, user_contact_external2, user_contact_mobile, user_contact_home, user_contact_fax, user_login}

SQL Queries

Select internal directory

```
SELECT
  u.user_firstname,
  u.user_surname,
  u.user_email,
  u.user_contact_external,
  u.user_contact_external2,
  u.user_contact_mobile,
  u.user_contact_home,
  u.user_contact_fax,
  u.user_login

FROM tblusers u

ORDER BY u.user_surname, u.user_firstname, u.user_login
```

Exception

N/A

3.5.10 Voicemail Center (All Users)

The report shows voicemail details in the specified date range.

Required input parameters	• From date • Until (to date) • Daily report
Output values (the values are grouped daily)	• User (shows the user first name and surname, when both are empty, the user login is used) • Call start time • Office Status (Office, Meeting, Sick, Break, Gone out, Holiday, Lunch, Home, DND) • Calling number • Priority (Normal, Urgent, Private) • Duration • Total number of daily voicemail messages • Total number of voicemail messages
Format	• Table
Axis label	• N/A

Calculation rule	Voicemail priority • 0 - urgent • 1 - private • 2 - normal Status: • 0 - office • 1 - meeting • 2 - sick • 3 - break • 4 - gone out • 5 - vacation (holiday) • 6 - lunch • 7 - home • 8 - DND (do not disturb)
Database tables	• tblusers, tblvoicemail
Database table attributes	• tblvoicemail = {vm_date, vm_user_id, vm_status, vm_calling_number, vm_priority, vm_duration} • tblusers = {user_id, user_login, user_firstname, user_surname}

SQL Queries

Select all available different days having voicemails for the selected user in the specified date range

```
SELECT DISTINCT ("date"(tblvoicemail.vm_date)) AS "Date of Day"

FROM tblvoicemail

WHERE tblvoicemail.vm_date >= ) /* from time */
  AND tblvoicemail.vm_date <= ("date"(?) + INTERVAL '24 hours')
                                 /* to date */
```

Select voicemails details (user detail, voicemail date, status, calling number, priority and duration) in the specified date range

```
SELECT u.user_firstname,
       u.user_surname,
       "date"(tblvoicemail.vm_date) AS "Date of Day",
       "time"(tblvoicemail.vm_date) AS "Start Time",
       tblvoicemail.vm_status,
       tblvoicemail.vm_calling_number,
       tblvoicemail.vm_priority,
       tblvoicemail.vm_duration

FROM tblvoicemail, tblusers u

WHERE tblvoicemail.vm_date >= ) /* from time */
  AND tblvoicemail.vm_date <= ("date"(?) + INTERVAL '24 hours')
                                 /* to date */
  AND tblvoicemail.vm_user_id = u.user_id

ORDER BY tblvoicemail.vm_date;
```

Exception

N/A

3.5.11 Voicemail Center (By User)

The report shows voicemail details for the selected user in the specified date range.

Required input parameters	• From date • Until (to date) • User • Daily report
Output values (the values are grouped daily)	• Call start time • Office Status (Office, Meeting, Sick, Break, Gone out, Holiday, Lunch, Home, DND) • Calling number • Priority (Normal, Urgent, Private) • Duration • Total number of daily voicemail messages • Total number of voicemail messages
Format	• Table
Axis label	• N/A

Calculation rule	Voicemail priority • 0 - urgent • 1 - private • 2 - normal Status: • 0 - office • 1 - meeting • 2 - sick • 3 - break • 4 - gone out • 5 - vacation (holiday) • 6 - lunch • 7 - home • 8 - DND (do not disturb)
Database tables	• tblvoicemail, tblusers, tbldepartments
Database table attributes	• tblvoicemail = {vm_date, vm_user_id, vm_status, vm_calling_number, vm_priority, vm_duration} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available different days having voicemails for the selected user in the specified date range

```
SELECT DISTINCT ("date"(tblvoicemail.vm_date)) AS "Date of Day"

FROM tblvoicemail, tblusers u

WHERE tblvoicemail.vm_date >= ) /* from time */
  AND tblvoicemail.vm_date <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND tblvoicemail.vm_user_id = u.user_id
  AND u.user_login = ? /* user login */
```

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_email,
       u.user_login,
  (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select voicemails details for the selected user in the specified date range

```
SELECT u.user_firstname,
       u.user_surname,
       "date"(tblvoicemail.vm_date) AS "Date of Day",
       "time"(tblvoicemail.vm_date) AS "Start Time",
       tblvoicemail.vm_status,
       tblvoicemail.vm_calling_number,
       tblvoicemail.vm_priority,
       tblvoicemail.vm_duration

FROM tblvoicemail, tblusers u

WHERE tblvoicemail.vm_date >= ) /* from time */
  AND tblvoicemail.vm_date <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND tblvoicemail.vm_user_id = u.user_id
  AND u.user_login = ? /* user login */

ORDER BY tblvoicemail.vm_date;
```

Exception

N/A

3.6 Report Group - Performance

All predefined report templates of this report group are described below.

3.6.1 Abandoned Calls Per Hour

Hourly representation of all abandoned calls in the specified date range.

Required input parameters	• From date • Until (to date) • Business hours only (else 24/24) • Daily report
Output values	• Number of abandoned calls • Percentage of abandoned calls • Total number of abandoned calls
Format	• Table and graphic
Axis label	• Horizontal: hourly intervals • Vertical: number of abandoned calls
Calculation rule	• Abandoned call = {talk time = 0, callback = 0, agent ID = 0} • Number of abandoned calls (per hour): COUNT(abandoned call for specific hourly interval)) • Percentage of all abandoned calls (per hour): (number of abandoned calls (per hour) / total number of abandoned calls) * 100 • Total number of abandoned calls: COUNT(abandoned call) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls,tblswitches
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time, cc_callback, cc_agent_id} • tblcalls = {call_id, call_start_time} • tblswitches = {switch_office_start, switch_office_end}

SQL Queries

Select all abandoned calls in the selected date range

```
SELECT COUNT (c1."call_id") AS "count",
  EXTRACT (hour FROM c1."call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ':00'
  AS "label"

FROM tblcallscc cc1, tblcalls c1, tblswitches s

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_talk_time" = 0
  AND cc1."cc_callback" = 0
  AND cc1."cc_agent_id" = 0
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
    "time"(c1."call_start_time") >= "time"(s.switch_office_start)
      AND
    "time"(c1."call_start_time") <= "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(c1."call_start_time") >= '00:00:00'AND
      "time"(c1."call_start_time") <= '23:59:59'
    END )

GROUP BY
    EXTRACT (hour FROM c1."call_start_time")

ORDER BY
    EXTRACT (hour FROM c1."call_start_time")
```

Exception

N/A

3.6.2 Agent Calls Percentage

The report displays information about the percentage and number of calls received by agents.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Agent • Number of calls by agent • Percentage of total number of calls (all agents) • Percentage of total talk time (all agents) • Total number of calls for all agents

Format	• Table and graphics
Axis label	• Horizontal: Agents • Vertical: Number of calls
Calculation rule	• Number of calls (by agent): COUNT(number of calls by agent) • Percentage of total number of calls (all agents): (number of calls (by agent) / total number of calls) * 100 • Percentage of total talk times (all agents): (talk time (by agent) / total talk time) * 100 • Total number of calls: COUNT(number of calls)
Database tables	• tblcallscc, tblcalls, tblusers
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time} • tblcalls = {call_id, call_start_time} • tblusers = {user_id, user_firstname, user_surname, user_is_agent}

SQL Queries

Select report details (agent first-name, surname and login, number of calls and total talk time) in the specified date range

```
SELECT u."user_firstname", u."user_surname", u.user_login,
  COUNT (cc."cc_call_id") AS "NumberOfCalls",
  SUM (cc."cc_talk_time") AS "TotalTalkTime"

FROM tblcallscc cc, tblcalls, tblusers u

WHERE cc."cc_agent_id" = u."user_id"
  AND u.user_is_agent = 1    AND cc."cc_call_id" =
tblca1ls."call_id"
  AND tblcalls."call_start_time" >= ? /* from time */
  AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                            '24 hours')/* to date */

GROUP BY u."user_firstname", u."user_surname", u.user_login
```

Exception

N/A

3.6.3 Agent Performance Details

The report shows agent performance details for the specified agent in a specified date/time range.

Required input parameters	• From date • To date (until) • From time • To time • Agent • Daily report
Output values (the values are grouped by queue and by day)	• Queue name • Start time • Pickup Time • Talk time • Grade of Service • Daily total number of calls, pickup time, talk time per queue • Daily average grade of service per queue • Total number of calls • Total average pickup time, talk time and grade of service
Format	• Table
Axis label	• N/A
Calculation rule	• Daily total number of calls (per queue): COUNT(number of calls per queue and day) • Daily total pickup time: SUM(pickup time per queue and day) • Daily total talk time: SUM(talk time per queue and day) • Total average of calls: COUNT(number of calls) • Total average pickup time: AVG(pickup time) • Total average talk time: AVG(talk time) • Total average GOS: AVG(GOS)
Database tables	• tblcalls, tblcallscc, tblusers, tbldepartments, tblqueues
Database table attributes	• tblcalls = {call_id, call_start_time, call_end_time} • tblcallscc = {cc_call_id, cc_agent_id, cc_gos, cc_talk_time, cc_pickup_time} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_department_id, user_is_agent} • tbldepartments = {department_name, department_id} • tblqueues = {queue_id, queue_name}

SQL Queries

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login, (SELECT CASE WHEN
tbldepartments.department_name = 'Unknown’
  THEN ’’ ELSE tbldepartments.department_name END
  FROM tbldepartments
  WHERE u.user_department_id = tbldepartments.department_id ) AS
department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Select all available days having calls for the selected agent in the selected date/time range

```
SELECT DISTINCT "date"(tblcalls."call_start_time") AS "SDate"

FROM tblcallscc cc, tblcalls, tblusers u, tblqueues q

WHERE cc."cc_agent_id" = u."user_id"
    AND cc."cc_queue_id" = q."queue.id"
    AND cc."cc_call_id" = tblcalls."call_id"
    AND "date"(tblcalls."call_start_time") >=
                                     "date"(?) /* from date */
    AND "date"(tblcalls."call_start_time") <=
                                     "date"(?) /* to date */
    AND "time"(tblcalls."call_start_time") >=
              "time"(to_timestamp(?,’HH:MI:SS’)) /* from time */
    AND "time"(tblcalls."call_start_time") <=
              "time"(to_timestamp(?,’HH:MI:SS’)) /* to time */
    AND u.user_login = ? /* agent login */
```

Select agent performance details (start date, start time, queue name, call id, pickup time, talk time, GOS)

```
SELECT "date"(tblcalls."call_start_time") AS "sdate1",
  "time"(tblcalls."call_start_time") AS "stime1",
  q."queue_name",
  cc."cc_call_id" AS "CallID",
  cc."cc_pickup_time" AS "Pickup Time",
  cc."cc_talk_time" AS "Talk Time",
  cc."cc_gos" AS "GOS"

FROM tblcallscc cc, tblcalls, tblusers u, tblqueue q

WHERE cc."cc_agent_id" = u."user_id"
    AND cc."cc_queue_id" = q."queue.id"
    AND cc."cc_call_id" = tblcalls."call_id"
    AND "date"(tblcalls."call_start_time") >=
                                    "date"(?) /* from date */
    AND "date"(tblcalls."call_start_time") <=
                                    "date"(?) /* to date */
    AND "time"(tblcalls."call_start_time") >=
              "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
    AND "time"(tblcalls."call_start_time") <=
              "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
    AND u.user_login = ? /* agent login */
ORDER BY tblcalls."call_start_time"
```

Select grand totals (total number of calls, average pickup time, average talk time and average GOS

```
SELECT
  COUNT (cc."cc_call_id") AS "NumberOfCalls",
  AVG(cc."cc_pickup_time") AS "Avg Pickup Time",
  AVG(cc."cc_talk_time") AS "Avg Talk Time",
  AVG(cc."cc_gos") AS "Avg GOS"

FROM tblcallscc cc, tblcalls, tblusers u, tblqueue q

WHERE cc."cc_agent_id" = u."user_id"
    AND cc."cc_queue_id" = q."queue.id"
    AND cc."cc_call_id" = tblcalls."call_id"
    AND "date"(tblcalls."call_start_time") >=
                                    "date"(?) /* from time */
    AND "date"(tblcalls."call_start_time") <=
                                    "date"(?) /* to date */
    AND "time"(tblcalls."call_start_time") >=
              "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
    AND "time"(tblcalls."call_start_time") <=
              "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
    AND u.user_login = ? /* agent login */
```

Exception

Since postgresql 8.3, the predefined function to_timestamp is required to compare the time values.

To convert the time calculated in seconds in the SQL queries above to time values in this report, but also in many others, the following calculation rules are used:

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.6.4 Answered Calls Per Hour

Hourly representation of all answered calls in the specified date range.

Required input parameters	• From date • Until (to date) • Business hours only (else 24/24) • Daily report
Output values	• Number of answered calls • Percentage of answered calls • Total number of calls
Format	• Table and graphic
Axis label	• Horizontal: hourly intervals • Vertical: number of answered calls

Calculation rule	• Answered call = {talk time > 0} • Number of answered calls (per hour) : COUNT(answered call for specific hourly interval)) • Percentage of all answered calls (per hour) : (number of answered calls (per hour) / total number of calls) * 100 • Total number of calls : COUNT(answered call) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls,tblswitches
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time} • tblcalls = {call_id, call_start_time} • tblswitches = {switch_office_start, switch_office_end}

SQL Queries

Select all answered calls in the selected date range

```
SELECT COUNT (c1."call_id") AS "count",
  EXTRACT (hour FROM c1."call_start_time")|| ':00 - ' ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ':00'
  AS "label"

FROM tblcallscc cc1, tblcalls c1, tblswitches s

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_talk_time" > 0
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
    "time"(c1."call_start_time") >= "time"(s.switch_office_start)
       AND
    "time"(c1."call_start_time") <= "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(c1."call_start_time") >= '00:00:00'<

AND
      "time"(c1."call_start_time") <= '23:59:59'
    END )

GROUP BY EXTRACT (hour FROM c1."call_start_time")

ORDER BY EXTRACT (hour FROM c1."call_start_time")
```

Exception

N/A

3.6.5 Call Traffic By Queue Per Hour

Hourly representation of the number of calls for specified queue in the selected date range.

INFO: The report template **Call Traffic By Queue Per Hour (Daily)** has a different table and graphic for each day.

Required input parameters	• From date • To date (until) • Queue • Daily report
Output values	• Time • Number of calls • Total number of calls • Percentage of total number of calls
Format	• Table and graphic
Axis label	• Horizontal: hourly intervals • Vertical: number of calls
Calculation rule	• Number of calls: COUNT(tblcallscc.cc_call_id) • Total number of calls : SUM(number of calls for specified hourly interval) • Percentage of total number of calls: number of calls / total number of calls * 100
Database tables	• tblcallscc, tblcalls, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_id, queue_name}

SQL Queries

Select number of calls for the selected queue in the specified date range

```
SELECT
  COUNT (cc1."cc_call_id"),
  EXTRACT (hour FROM c1."call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ':00'
  AS "label"

FROM tblcallscc cc1, tblcalls c1, tblqueues q1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
   AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_queue_id" = q1."queue_id"
  AND q1."queue_name" = ? /* queue name */

GROUP BY EXTRACT (hour FROM c1."call_start_time")
ORDER BY EXTRACT (hour FROM c1."call_start_time")
```

Select all available queues (used for selecting the queue)

```
SELECT tblqueues."queue_name"
FROM tblqueues
ORDER BY tblqueues."queue_name"
```

Exception

N/A

3.6.6 Call Traffic By Queue Per Hour (Daily)

Hourly representation of the number of calls for specified queue in the selected date range. There is a different table and graphic for each day.

Required input parameters	• From date • To date (until) • Queue • Daily report
Output values	• Time • Number of calls • Total number of calls • Percentage of total number of calls
Format	• Table and graphic
Axis label	• Horizontal: hourly intervals • Vertical: number of calls

Calculation rule	• Number of calls: COUNT(tblcallscc.cc_call_id) • Total number of calls : SUM(number of calls for specified hourly interval) • Percentage of total number of calls: number of calls / total number of calls * 100
Database tables	• tblcallscc, tblcalls, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_id, queue_name}

SQL Queries

Select number of calls for the selected queue in the specified date range

```
SELECT
  COUNT (cc1."cc_call_id"),
  EXTRACT (hour FROM c1."call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ':00'
  AS "label"

FROM tblcallscc cc1, tblcalls c1, tblqueues q1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_queue_id" = q1."queue_id"
  AND q1."queue_name" = ? /* queue name */

GROUP BY EXTRACT (hour FROM c1."call_start_time")
ORDER BY EXTRACT (hour FROM c1."call_start_time")
```

Select all available days having calls for the selected queue in the specified date range

```
SELECT DISTINCT "date"(c1."call_start_time")

FROM tblcallscc cc1, tblcalls c1 , tblqueues q1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_queue_id" = q1."queue_id"
  AND q1."queue_name" = ? /* queue name */
```

Select all available queues (used for selecting the queue)

```
SELECT tblqueues."queue_name"
FROM tblqueues
ORDER BY tblqueues."queue_name"
```

Exception

N/A

3.6.7 Contact Center Traffic Per Hour

Hourly representation of the number of calls in the selected date range.

INFO: The report template **Contact Center Traffic Per Hour (Daily)** has a different table and graphic for each day.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Time of call • Number of calls • Total number of calls • Percentage of total number of calls
Format	• Table and graphic
Axis label	• Horizontal: hourly intervals • Vertical: number of calls
Calculation rule	• Number of calls: COUNT(tblcallscc.cc_call_id) • Total number of calls : SUM(number of calls for specified hourly interval) • Percentage of total number of calls: number of calls / total number of calls * 100
Database tables	• tblcallscc, tblcalls
Database table attributes	• tblcallscc = {cc_call_id} • tblcalls = {call_id, call_start_time}

SQL Queries

Select hourly number of calls in the specified date range

```
SELECT COUNT (cc1."cc_call_id"),
  EXTRACT (hour FROM c1."call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ':00''
  AS "label"

FROM tblcallscc cc1, tblcalls c1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"

GROUP BY EXTRACT (hour FROM c1."call_start_time")
ORDER BY EXTRACT (hour FROM c1."call_start_time")
```

Exception

N/A

3.6.8 Contact Center Traffic Per Hour (Daily)

Hourly representation of the number of calls in the selected date range. There is a different table and graphic for each day.

Required input parameters	• From date • To date (until) • Daily report
Output values	• Time (hourly interval) • Number of calls • Total number of calls • Percentage of total number of calls
Format	• Table and graphic
Axis label	• Horizontal: hourly intervals • Vertical: number of calls
Calculation rule	• Number of calls: COUNT(tblcallscc.cc_call_id) • Total number of calls : SUM(number of calls for specified hourly interval) • Percentage of total number of calls: number of calls / total number of calls * 100
Database tables	• tblcallscc, tblcalls
Database table attributes	• tblcallscc = {cc_call_id } • tblcalls = {call_id, call_start_time}

SQL Queries

Select number of calls for the selected queue in the specified date range

```
SELECT "date"(c1."call_start_time"),
  COUNT (cc1."cc_call_id"),
  EXTRACT (hour FROM c1."call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ':00'
  AS "label"

FROM tblcallscc cc1, tblcalls c1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"

GROUP BY EXTRACT (hour FROM c1."call_start_time"),
    "date"(c1."call_start_time")

ORDER BY EXTRACT (hour FROM c1."call_start_time")
```

Select all available days having calls for the selected queue in the specified date range

```
SELECT DISTINCT "date"(c1."call_start_time")

FROM tblcallscc cc1, tblcalls c1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                             /* to date */
  AND cc1."cc_call_id" = c1."call_id"
```

Exception

N/A

3.6.9 Missed Calls Per Hour

Hourly representation of all missed calls in the specified date range.

Required input parameters	• From date • Until (to date) • Business hours only (else 24/24) • Daily report
Output values	• Number of missed calls • Percentage of all missed calls • Total number of missed calls
Format	• Table and graphic

Axis label	• Horizontal: hourly intervals • Vertical: number of missed calls
Calculation rule	• Missed call = {aa_event_type = 6} • Number of missed calls (per hour): COUNT(missed call for specific hourly interval)) • Percentage of all missed calls (per hour) : (number of missed calls (per hour) / total number of missed calls) * 100 • Total number of missed calls : COUNT(missed call) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblswitches, tblagentactivity
Database table attributes	• tblagentactivity = {aa_call_id, aa_queue_id, aa_event_time, aa_event_type} • tblswitches = {switch_office_start, switch_office_end}

SQL Queries

Select all missed calls in the selected date range

```
SELECT  COUNT (aa."aa_call_id") AS "count",
  EXTRACT (hour FROM aa.aa_event_time) || ':00 - ' ||
  (EXTRACT (hour FROM aa.aa_event_time) + 1) || ':00' AS "label"

FROM tblagentactivity aa, tblswitches s

WHERE aa.aa_event_time >= ? /* from time */
  AND aa.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
  AND aa.aa_event_type = 6
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
     "time"(aa.aa_event_time) >= "time"(s.switch_office_start)
       AND
    "time"(aa.aa_event_time) <= "time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(aa.aa_event_time) >= '00:00:00'AND
      "time"(aa.aa_event_time) <= '23:59:59'
    END )

GROUP BY EXTRACT (hour FROM aa.aa_event_time)

ORDER BY EXTRACT (hour FROM aa.aa_event_time)
```

Exception

N/A

3.6.10 Summary of Details per Agent

This report contains a summary of the details (duration of agent activities, percentage of work, break and absence times during the logged in time, percentage of logged in time during business hours, calls, talk times) for a specific agent in the specified date range.

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values	• Duration of agent activities: logged in, work, break and absence times • Percentage of work, break and absence times during the logged in time • Percentage of logged in time during business hours • (***) All calls, outgoing calls, incoming calls, direct calls, CC Calls (Contact Center calls), CC callback calls, answered CC calls, answered CC overflow calls, answered CC callback calls, missed CC calls, missed CC overflow calls, missed CC callback calls • Internal calls, external calls and calls during business hours for all (***) columns • Total talk time, average talk time and average number of calls per hour during business hours for all (***) columns, except for missed calls
Format	• Table
Axis label	• N/A

Calculation rule	• Internal calls: ch_internal_external = 0 • External calls: ch_internal_external = 1 • Incoming calls: ch_direction = 0 • Outgoing calls: ch_direction = 1 • Direct calls: ch_cc_call_id = 0 • CC calls (Contact Center calls): ch_cc_call_id > 0 • CC callback calls: ch_cc_call_id IN tblcallscc.cc_call_id and cc_callback = 1 • CC overflow calls: cch_cc_call_id IN tblcallscc.cc_call_id and cc_agent_overflow = 1 • Missed calls: ch_cc_call_id IN tblagentactivity.aa_call_id and aa_event_type = 6 All available combinations in the report template are formed by mixing the above calculation rules. Examples: • Internal CC calls = CC calls and internal Call = h_cc_call_id > 0 and ch_internal_external = 0 • Missed overflow calls = Missed calls and CC overflow calls aa_event_type: • Logged in time has event_type = 0 • Break time has event_type = 2 • Work time has event_type = 4 • Missed call time has event_type = 6
Database tables	• tblcallscc, tblcallhistory, tblusers, tblagentactivity, tblswitches
Database table attributes	• tblcallscc = {cc_call_id, cc_callback, cc_agent_overflow} • tblswitches = {switch_office_start, switch_office_end} • tblcallshistory = {ch_call_id, ch_cc_call_id, ch_talk_time_seconds, ch_user_id } • tblagentactivity = {aa_call_id, aa_event_type, aa_event_duration}

SQL Queries

Select duration of agent activities (logged in time, break time, work time, missed call time)

```
SELECT
--Logged--
  SUM (logged.aa_event_duration) AS "logged_time"
    EXTRACT (
      (epoch FROM SUM (logged.aa_event_duration))
            ) AS "logged_time_sec"

--Work--
  SUM (work1.aa_event_duration) AS "work_time"
    EXTRACT (
      (epoch FROM SUM (work1.aa_event_duration))
            ) AS "work_time_sec"

--Break--
  SUM (break1.aa_event_duration) AS "break_time"
    EXTRACT (
      (epoch FROM SUM (break1.aa_event_duration))
            ) AS "break_time_sec"

--Missed--
  SUM (missed1.aa_event_duration) AS "missed_time"
    EXTRACT (
      (epoch FROM SUM (missed1.aa_event_duration))
            ) AS "missed_time_sec"

FROM tblagentactivity a1

LEFT JOIN tblagentactivity logged ON (a1.aa_id=logged.aa_id
  AND a1.aa_event_type = 0)
LEFT JOIN tblagentactivity work1 ON (a1.aa_id=work1.aa_id
  AND a1.aa_event_type = 4)
LEFT JOIN tblagentactivity break1 ON (a1.aa_id=break1.aa_id
  AND a1.aa_event_type = 2)
LEFT JOIN tblagentactivity missed1 ON (a1.aa_id=missed1.aa_id
  AND a1.aa_event_type = 6), tblusers u

WHERE a1.aa_event_type in (0,2,4,6)
    AND a1.aa_event_time >= ? /*from date*/
    AND a1.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
              /*to date*/
    AND a1.aa_agent_id = u.user_id
    AND u.user_login = ? /*agent login*/
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
    (SELECT
      CASE
        WHEN tbldepartments.department_name = 'Unknown'
          THEN '' ELSE tbldepartments.department_name
          END
        FROM tbldepartments
        WHERE u.user_department_id = tbldepartments.department_id
     ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /*agent login*/
```

Select total logged in time and business hours for the selected agent in the specified date range (used for calculating the relationship (percentage) between the total logged in time and business hours)

```
SELECT
--Logged--
  EXTRACT (
      (epoch FROM SUM (logged.aa_event_duration))
            ) AS "logged_time_sec"

--Business Time--
  EXTRACT (
      (epoch FROM SUM (business.aa_event_duration))
            ) AS "business_time_sec"

FROM tblagentactivity logged

LEFT JOIN tblagentactivity business
  ON (logged.aa_id=business.aa_id
  AND logged.aa_event_type = 0)
  AND logged.aa_event_time
  IN (SELECT a2.aa_event_time
  FROM tblagentactivity a2, tblswitches s
  WHERE "time"(a2.aa_event_time) >= "time"(s.switch_office_start)
  AND "time"(a2.aa_event_time) <= "time"(s.switch_office_end))),
      tblusers u
```

```
WHERE logged.aa_event_type = 0
  AND logged.aa_event_time >= ? /*from date*/
  AND logged.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
             /*to date*/
  AND logged.aa_agent_id = u.user_id
  AND u.user_login = ? /*agent login*/
```

Select summary of details for the selected agent

The PostgreSQL functions SUM, COUNT and AVG are used.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  COUNT (inc.ch_call_id) AS "inc_calls",
  COUNT (outg.ch_call_id) AS "outg_calls",
  COUNT (direct.ch_call_id) AS "direct_calls",
  COUNT (cc_calls.ch_call_id) AS "cc_calls",
  COUNT (cc_callback.ch_call_id) AS "cc_callback_calls",

SUM (CASE WHEN ch1.ch_internal_external = 0 THEN 1 ELSE 0 END)
  AS "all_internal_calls",
SUM (CASE WHEN ch1.ch_internal_external = 1 THEN 1 ELSE 0 END)
  AS "all_external_calls",
SUM (CASE WHEN ("time"(ch1."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "all_business_calls",

SUM (CASE WHEN inc.ch_internal_external = 0 THEN 1 ELSE 0 END)
  AS "inc_internal_calls",
SUM (CASE WHEN inc.ch_internal_external = 1 THEN 1 ELSE 0 END)
  AS "inc_external_calls",
SUM (CASE WHEN ("time"(inc."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(inc."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "inc_business_calls",

SUM (CASE WHEN outg.ch_internal_external = 0 THEN 1 ELSE 0 END)
  AS "outg_internal_calls",
SUM (CASE WHEN outg.ch_internal_external = 1 THEN 1 ELSE 0 END)
  AS "outg_external_calls",
SUM (CASE WHEN ("time"(outg."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(outg."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "outg_business_calls",
```

```
SUM (CASE WHEN direct.ch_internal_external = 0 THEN 1 ELSE 0 END)
  AS "direct_internal_calls",
SUM (CASE WHEN direct.ch_internal_external = 1 THEN 1 ELSE 0 END)
  AS "direct_external_calls",
SUM (CASE WHEN ("time"(direct."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(direct."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "direct_business_calls",

SUM (CASE WHEN cc_calls.ch_internal_external = 0 THEN 1 ELSE 0 END)
  AS "cc_internal_calls",
SUM (CASE WHEN cc_calls.ch_internal_external = 1 THEN 1 ELSE 0 END)
  AS "cc_external_calls",
SUM (CASE WHEN ("time"(cc_calls."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(cc_calls."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "cc_business_calls",

SUM (CASE WHEN cc_callback.ch_internal_external = 0
   THEN 1 ELSE 0 END) AS "cc_callback_internal_calls",
SUM (CASE WHEN cc_callback.ch_internal_external = 1
   THEN 1 ELSE 0 END) AS "cc_callback_external_calls",
SUM (CASE WHEN ("time"(cc_callback."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(cc_callback."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "cc_callback_business_calls",

SUM (CASE WHEN cc_answered.ch_internal_external = 0
   THEN 1 ELSE 0 END) AS "cc_answered_internal_calls",
SUM (CASE WHEN cc_answered.ch_internal_external = 1
   THEN 1 ELSE 0 END) AS "cc_answered_external_calls",
SUM (CASE WHEN ("time"(cc_answered."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(cc_answered."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "cc_answered_business_calls",

SUM (CASE WHEN ans_overflow.ch_internal_external = 0
   THEN 1 ELSE 0 END) AS "ans_overflow_internal_calls",
SUM (CASE WHEN ans_overflow.ch_internal_external = 1
   THEN 1 ELSE 0 END) AS "ans_overflow_external_calls",
SUM (CASE WHEN ("time"(ans_overflow."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(ans_overflow."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "ans_overflow_business_calls",
```

```
SUM (CASE WHEN ans_callback.ch_internal_external = 0
   THEN 1 ELSE 0 END) AS "ans_callback_internal_calls",
SUM (CASE WHEN ans_callback.ch_internal_external = 1
   THEN 1 ELSE 0 END) AS "ans_callback_external_calls",
SUM (CASE WHEN ("time"(ans_callback."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(ans_callback."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "ans_callback_business_calls",

SUM (CASE WHEN cc_missed.ch_internal_external= 0
   THEN 1 ELSE 0 END) AS "cc_missed_internal_calls",
SUM (CASE WHEN cc_missed.ch_internal_external = 1
   THEN 1 ELSE 0 END) AS "cc_missed_external_calls",
SUM (CASE WHEN ("time"(cc_missed."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(cc_missed."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "cc_missed_business_calls",

SUM (CASE WHEN missed_overflow.ch_internal_external = 0
   THEN 1 ELSE 0 END) AS "missed_overflow_internal_calls",
SUM (CASE WHEN missed_overflow.ch_internal_external = 1
   THEN 1 ELSE 0 END) AS "missed_overflow_external_calls",
SUM (CASE WHEN ("time"(missed_overflow."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(missed_overflow."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "missed_overflow_business_calls",

SUM (CASE WHEN missed_callback.ch_internal_external = 0
   THEN 1 ELSE 0 END) AS "missed_callback_internal_calls",
SUM (CASE WHEN missed_callback.ch_internal_external = 1
   THEN 1 ELSE 0 END) AS "missed_callback_external_calls",
SUM (CASE WHEN ("time"(missed_callback."ch_start_time") >=
      "time" (s.switch_office_start)
  AND "time"(missed_callback."ch_start_time") <=
      "time"(s.switch_office_end)) THEN 1 ELSE 0 END) AS
    "missed_callback_business_calls",

AVG (direct.ch_talk_time_seconds) AS "avg_talk_time_direct",
AVG (cc_calls.ch_talk_time_seconds) AS "avg_talk_time_cc_calls",
AVG (cc_callback.ch_talk_time_seconds)
   AS "avg_talk_time_cc_callback",

SUM (direct.ch_talk_time_seconds) AS "total_talk_time_direct",
SUM (cc_calls.ch_talk_time_seconds) AS "total_talk_time_cc_calls",
SUM (cc_callback.ch_talk_time_seconds)
   AS "total_talk_time_cc_callback",
```

```
--Answered--
COUNT (cc_answered.ch_call_id) AS "answered_calls",
COUNT (ans_overflow.ch_call_id) AS "ans_overflow_calls",
COUNT (ans_callback.ch_call_id) AS "ans_callback_calls",

AVG (cc_answered.ch_talk_time_seconds) AS "avg_talk_time_ans",
AVG (ans_overflow.ch_talk_time_seconds)
   AS "avg_talk_time_ans_overflow",
AVG (ans_callback.ch_talk_time_seconds)
   AS "avg_talk_time_ans_callback",

SUM (cc_answered.ch_talk_time_seconds) AS "total_talk_time_ans",
SUM (ans_overflow.ch_talk_time_seconds)
   AS "total_talk_time_ans_overflow",
SUM (ans_callback.ch_talk_time_seconds)
   AS "total_talk_time_ans_callback",


--Missed--
COUNT (cc_missed.ch_call_id) AS "missed_calls",
COUNT (missed_overflow.ch_call_id) AS "missed_overflow_calls",
COUNT (missed_callback.ch_call_id) AS "missed_callback_calls",

AVG (ch1.ch_talk_time_seconds) AS "avg_talk_time_all",
AVG (inc.ch_talk_time_seconds) AS "avg_talk_time_inc",
AVG (outg.ch_talk_time_seconds) AS "avg_talk_time_outg",

SUM (ch1.ch_talk_time_seconds) AS "total_talk_time_all",
SUM (inc.ch_talk_time_seconds) AS "total_talk_time_inc",
SUM (outg.ch_talk_time_seconds) AS "total_talk_time_outg",

FROM tblcallhistory ch1

LEFT JOIN tblcallhistory inc
  ON (inc.ch_call_id = ch1.ch_call_id
  AND ch1.ch_direction = 0)
LEFT JOIN tblcallhistory outg
  ON (outg.ch_call_id = ch1.ch_call_id
  AND ch1.ch_direction = 1)
LEFT JOIN tblcallhistory direct
  ON (direct.ch_call_id = ch1.ch_call_id
  AND ch1.ch_cc_call_id = 0)
LEFT JOIN tblcallhistory cc_calls
  ON (cc_calls.ch_call_id = ch1.ch_call_id
  AND ch1.ch_cc_call_id > 0)
LEFT JOIN tblcallhistory cc_callback
  ON (cc_callback.ch_call_id = cc_calls.ch_call_id
  AND cc_calls.ch_cc_call_id
  IN (SELECT c1.cc_call_id
  FROM tblcallscc c1
  WHERE c1.cc_callback = 1))
```

```
--CC Answered--
LEFT JOIN tblcallhistory cc_answered
  ON (cc_answered.ch_call_id = cc_calls.ch_call_id
  AND cc_calls.ch_talk_time_seconds > 0)
LEFT JOIN tblcallhistory ans_overflow
  ON (ans_overflow.ch_call_id = cc_answered.ch_call_id
  AND cc_answered.ch_cc_call_id
  IN (SELECT c1.cc_call_id
  FROM tblcallscc c1
  WHERE c1.cc_agent_overflow = 1))
LEFT JOIN tblcallhistory ans_callback
  ON (ans_callback.ch_call_id = cc_answered.ch_call_id
  AND cc_answered.ch_cc_call_id
  IN (SELECT c1.cc_call_id
  FROM tblcallscc c1
  WHERE c1.cc_callback = 1))


--CC Missed--
LEFT JOIN tblcallhistory cc_missed
  ON (cc_missed.ch_call_id = cc_calls.ch_call_id
  AND cc_calls.ch_cc_call_id
  IN (SELECT a1.aa_call_id
  FROM tblagentactivity a1
  WHERE a1.aa_event_type = 6))
LEFT JOIN tblcallhistory missed_overflow
  ON (missed_overflow.ch_call_id = cc_missed.ch_call_id
  AND cc_missed.ch_cc_call_id
  IN (SELECT c1.cc_call_id
  FROM tblcallscc c1
  WHERE c1.cc_agent_overflow = 1))
LEFT JOIN tblcallhistory missed_callback
  ON (missed_callback.ch_call_id = cc_missed.ch_call_id
  AND cc_missed.ch_cc_call_id
  IN (SELECT c1.cc_call_id
  FROM tblcallscc c1
  WHERE c1.cc_callback = 1))


,tblusers u,tblswitches s

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
              /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*agent login*/
```

Select grand total of calls per hour during business hours (used for calculating the average number of calls per hour during business hours)

Business hours without calls are not considered in the calculation.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  "date" (ch1."ch_start_time")
    EXTRACT (
      HOUR FROM ch1."ch_start_time")

FROM tblcallhistory ch1, tblusers u, tblswitches s,

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
              /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*agent login*/
  AND "time"(ch1."ch_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(ch1."ch_start_time"),
    EXTRACT (
      HOUR FROM ch1."ch_start_time"
```

Select grand total of Contact Center calls per hour during business hours (used for calculating the average number of calls per hour during business hours)

Business hours without Contact Center calls are not considered in the calculation.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  "date" (ch1."ch_start_time")
    EXTRACT (
      HOUR FROM ch1."ch_start_time")

FROM tblcallhistory ch1, tblusers u, tblswitches s,

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
              /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*agent login*/
  AND ch1.ch_cc_call_id > 0 /*CC Calls*/
  AND "time"(ch1."ch_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(ch1."ch_start_time"),
    EXTRACT (
      HOUR FROM ch1."ch_start_time"
```

Select grand total of answered calls per hour during business hours (used for calculating the average number of calls per hour during business hours)

Business hours without answered calls are not considered in the calculation.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  "date" (ch1."ch_start_time")
    EXTRACT (
      HOUR FROM ch1."ch_start_time")

FROM tblcallhistory ch1, tblusers u, tblswitches s,

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
             /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*agent login*/
  AND ch1.ch_cc_call_id > 0 /*CC Calls*/
  AND ch1.ch_talk_time_seconds > 0 /*Answered Calls*/
  AND "time"(ch1."ch_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(ch1."ch_start_time"),
    EXTRACT (
      HOUR FROM ch1."ch_start_time"
```

Select grand total of Contact Center callback calls per hour during business hours (used for calculating the average number of calls per hour during business hours)

Business hours without Contact Center callback calls are not considered in the calculation.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  "date" (ch1."ch_start_time")
    EXTRACT (
      HOUR FROM ch1."ch_start_time")

FROM tblcallhistory ch1, tblusers u, tblswitches s,

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
             /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*agent login*/
  AND ch1.ch_cc_call_id > 0 /*CC Calls*/
  AND ch1.ch_talk_time_seconds > 0 /*Answered Calls*/
  AND ch1.ch_cc_call_id
  IN (SELECT c1.cc_call_id
     FROM tblcallscc c1
     WHERE c1.cc_callback = 1) /*Callback Calls*/
  AND "time"(ch1."ch_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(ch1."ch_start_time"),
    EXTRACT (
      HOUR FROM ch1."ch_start_time"
```

Select grand total of Contact Center overflow calls per hour during business hours (used for calculating the average number of calls per hour during business hours)

Business hours without Contact Center overflow calls are not considered in the calculation.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  "date" (ch1."ch_start_time")
    EXTRACT (
      HOUR FROM ch1."ch_start_time")

FROM tblcallhistory ch1, tblusers u, tblswitches s,

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
             /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*agent login*/
  AND ch1.ch_cc_call_id > 0 /*CC Calls*/
  AND ch1.ch_talk_time_seconds > 0 /*Answered Calls*/
  AND ch1.ch_cc_call_id
  IN (SELECT c1.cc_call_id
     FROM tblcallscc c1
     WHERE c1.cc_agent_overflow = 1) /*Overflow Calls*/
  AND "time"(ch1."ch_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(ch1."ch_start_time"),
    EXTRACT (
      HOUR FROM ch1."ch_start_time"
```

Select grand total of Contact Center callback calls per hour during business hours (used for calculating the average number of calls per hour during business hours)

Business hours without Contact Center callback calls are not considered in the calculation.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  "date" (ch1."ch_start_time")
    EXTRACT (
      HOUR FROM ch1."ch_start_time")

FROM tblcallhistory ch1, tblusers u, tblswitches s,

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
             /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*agent login*/
  AND ch1.ch_cc_call_id > 0 /*CC Calls*/
  AND ch1.ch_cc_call_id
  IN (SELECT c1.cc_call_id
     FROM tblcallscc c1
     WHERE c1.cc_callback = 1) /*Callback Calls*/
  AND "time"(ch1."ch_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(ch1."ch_start_time"),
    EXTRACT (
      HOUR FROM ch1."ch_start_time"
```

Select grand total of direct calls per hour during business hours (used for calculating the average number of calls per hour during business hours)

Business hours without direct calls are not considered in the calculation.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  "date" (ch1."ch_start_time")
    EXTRACT (
      HOUR FROM ch1."ch_start_time")

FROM tblcallhistory ch1, tblusers u, tblswitches s,

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
             /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*agent login*/
  AND ch1.ch_cc_call_id = 0 /*Direct Calls*/
  AND "time"(ch1."ch_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(ch1."ch_start_time"),
    EXTRACT (
      HOUR FROM ch1."ch_start_time"
```

Select grand total of incoming calls per hour during business hours (used for calculating the average number of calls per hour during business hours)

Business hours without incoming calls are not considered in the calculation.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  "date" (ch1."ch_start_time")
    EXTRACT (
      HOUR FROM ch1."ch_start_time")

FROM tblcallhistory ch1, tblusers u, tblswitches s,

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
             /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*From Agent*/
  AND ch1.ch_direction = 0 /*Incoming*/
  AND "time"(ch1."ch_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(ch1."ch_start_time"),
    EXTRACT (
      HOUR FROM ch1."ch_start_time"
```

Select grand total of outgoing calls per hour during business hours (used for calculating the average number of calls per hour during business hours)

Business hours without outgoing calls are not considered in the calculation.

```
SELECT
  COUNT (ch1.ch_call_id) AS "all_calls",
  "date" (ch1."ch_start_time")
    EXTRACT (
      HOUR FROM ch1."ch_start_time")

FROM tblcallhistory ch1, tblusers u, tblswitches s,

WHERE ch1."ch_start_time" >= ? /*from date*/
  AND ch1."ch_start_time" <= ("date"(?) + INTERVAL '24 hours')
              /*to date*/
  AND ch1.ch_user_id = u.user_id
  AND u.user_login = ? /*From Agent*/
  AND ch1.ch_direction = 1 /*Outgoing*/
  AND "time"(ch1."ch_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(ch1."ch_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(ch1."ch_start_time"),
    EXTRACT (
      HOUR FROM ch1."ch_start_time"
```

Exception

N/A

3.6.11 Summary of Details per Queue

This report contains a summary of the details (calls, call- and wait times, details for answered and abandoned calls, percentage of total number of all answered and abandoned calls) for a specific queue in the specified date range.

Required input parameters	• From date • To date (until) • Queue • Daily report
Output values	• All calls, internal calls, external calls, callback calls, answered calls, calls answered during business hours, calls answered outside business hours, calls answered by primary agent / overflow agents, abandoned calls, etc. • Number of calls, total talk time, average talk time, average queue time, max. queue time for all of the above columns • Details for answered and abandoned calls with respect to queue time: up to 30 seconds, less than 3 seconds, between 3 and 20 seconds, between 20 and 30 seconds • Percentage of all answered calls for internal calls, external calls and callback calls during business hours and outside business hours • Percentage of all abandoned calls for internal calls and external calls and with respect to the queue time • Other values: date, business hours, number of calls and Grade of Service (GOS) during business hours
Format	• Table
Axis label	• N/A
Calculation rule	• Internal calls: The call no. (incoming and outgoing calls) can have a maximum of five digits. • External calls: The call no. (incoming and outgoing calls) can have a maximum of six digits • Callback calls: cc_callback = 1 • Calls for overflow agents: cc_agent_overflow = 1 • Calls for primary agent: cc_agent_overflow = 0 • Answered calls: cc_talk_time > 0 • Abandoned calls: cc_talk_time = 0, cc_callback = 0, cc_agent_id = 0
Database tables	• tblcalls, tblcallscc, tblcallhistory, tblqueues, tblagentqueues, tblswitches

Database table attributes	• tblcalls = {call_id, call_start_time, call_end_time, call_calling_number, call_called_number} • tblcallscc = {cc_call_id, cc_callback,cc_gos, cc_talk_time, cc_queue_time, cc_agent_id} • tblqueues = {queue_id, queue_name, queue_type} • tblswitches = {switch_office_start, switch_office_end} • tblagentqueues = {aq_queue_id, aq_user_id, aq_agent_type}

SQL Queries

Select other values: date, business hours, number of calls and Grade of Service (GOS) during business hours)

```
--Maximum number of calls per hour in business time--
SELECT
  COUNT (c."cc_call_id") AS "nb_calls",
  AVG (c.cc_gos) AS "gos",
     EXTRACT (
      HOUR FROM c0."call_start_time")||':00 - '||(
     EXTRACT (
      HOUR FROM c0."call_start_time")+1)||':00' AS "label",
        "date" (c0."call_start_time") AS "dat")

FROM tblcallscc c, tblcalls c0, tblqueues, tblswitches s

WHERE c0."call_start_time >= ? /*from date*/
  AND c0."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
              /*to date*/
  AND c."cc_call_id" = c0."call_id"
  AND c."cc_queue_id" = queue_id
  AND queue_name = ? /*queue name*/
  AND "time"(c0."call_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(c0."call_start_time") <=
      "time"(s.switch_office_end)
  GROUP BY "date"(c0."call_start_time"),
    EXTRACT (
      HOUR FROM c0."call_start_time"
       ORDER BY "nb_calls" DESC LIMIT 1
```

Select summary of details for the selected queue

The PostgreSQL functions SUM, COUNT and AVG are used.

```
SELECT
  COUNT (c."cc_call_id") AS "all_calls",
  SUM (c.cc_talk_time) AS "total_talk_time",
  AVG (c.cc_talk_time) AS "avg_talk_time",
  AVG (c.cc_queue_time) AS "avg_gueue_time",
  MAX (c.cc_queue_time) AS "max_queue_time",
```

```
--Abandoned--
  COUNT (aband."cc_call_id") AS "abandoned_calls",
  AVG (aband.cc_queue_time) AS "abandoned_avg_gueue_time",
  MAX (aband.cc_queue_time) AS "abandoned_max_queue_time",


--Abandoned < 3sec--
  COUNT (aband3."cc_call_id") AS "abandoned_calls_3",
  AVG (aband3.cc_queue_time)
    AS "abandoned0_3_avg_gueue_time",
  MAX (aband3.cc_queue_time)
    AS "abandoned0_3_max_queue_time",


--Abandoned 3-20 sec--
  COUNT (aband3_20."cc_call_id") AS "abandoned_calls_3_20",
  AVG (aband3_20.cc_queue_time)
    AS "abandoned3_20_avg_gueue_time",
  MAX (aband3_20.cc_queue_time)
    AS "abandoned3_20_max_queue_time",


--Abandoned 20-30 sec--
  COUNT (aband20_30."cc_call_id") AS "abandoned_calls_20_30",
  AVG (aband20_30.cc_queue_time)
    AS "abandoned20_30_avg_gueue_time",
  MAX (aband20_30.cc_queue_time)
    AS "abandoned20_30_max_queue_time",


--Abandoned > 30 sec--
  COUNT (aband30up."cc_call_id") AS "abandoned_calls_30up",
  AVG (aband30up.cc_queue_time)
    AS "abandoned30up_avg_gueue_time",
  MAX (aband30up.cc_queue_time)
    AS "abandoned30up_max_queue_time",


--Answered--
  COUNT (answ."cc_call_id") AS "answered_calls",
  SUM (answ.cc_talk_time) AS "answ_total_talk_time",
  AVG (answ.cc_talk_time) AS "answ_avg_talk_time",
  AVG (answ.cc_queue_time) AS "answ_avg_gueue_time",
  MAX (answ.cc_queue_time) AS "answ_max_queue_time",
```

```
--Callback--
  COUNT (callback."cc_call_id") AS "callback_calls",
  SUM (CASE WHEN callback.cc_talk_time > 0
    AND callback.cc_agent_id > 0 THEN 1 ELSE 0 END)
    AS "callback_calls_answered",
   SUM (CASE WHEN callback.cc_talk_time > 0
    AND callback.cc_agent_id > 0
      AND callback.cc_queue_time >= 30 THEN 1 ELSE 0 END)
    AS "callback_calls_answered_after30",


   SUM (callback.cc_talk_time) AS "callback_total_talk_time",
   AVG (callback.cc_talk_time) AS "callback_avg_talk_time",
  AVG (callback.cc_queue_time) AS "callback_avg_gueue_time",
  MAX (callback.cc_queue_time) AS "callback_max_queue_time",


--Answered in business time--
  COUNT (answbuss."cc_call_id") AS "answered_calls_business",
  SUM (answbuss.cc_talk_time) AS "answbuss_total_talk_time",
  AVG (answbuss.cc_talk_time) AS "answbuss_avg_talk_time",
  AVG (answbuss.cc_queue_time) AS "answbuss_avg_gueue_time",
  MAX (answbuss.cc_queue_time) AS "answbuss_max_queue_time",


--Answered in business time when queue_time >= 30--
  COUNT (answbuss30."cc_call_id")
    AS "answered_calls_business_30",
  SUM (answbuss30.cc_talk_time)
    AS "answbuss30_total_talk_time",
  AVG (answbuss30.cc_talk_time) AS "answbuss30_avg_talk_time",
  AVG (answbuss30.cc_queue_time) AS "answbuss30_avg_gueue_time",
  MAX (answbuss30.cc_queue_time) AS "answbuss30_max_queue_time",


--Answered business time out--
  COUNT (answbussout."cc_call_id")
    AS "answered_calls_business_out",
  SUM (answbussout.cc_talk_time)
    AS "answbussout_total_talk_time",
  AVG (answbussout.cc_talk_time)
    AS "answbussout_avg_talk_time",
  AVG (answbussout.cc_queue_time)
    AS "answbussout_avg_gueue_time",
  MAX (answbussout.cc_queue_time)
    AS "answbussout_max_queue_time",
```

```
--Answered business out when queue time >= 30--
  COUNT (answbuss30out."cc_call_id")
    AS "answered_calls_business_out_30",
  SUM (answbuss30out.cc_talk_time)
    AS "answbuss30out_total_talk_time",
  AVG (answbuss30out.cc_talk_time)
    AS "answbuss30out_avg_talk_time",
  AVG (answbuss30out.cc_queue_time)
    AS "answbuss30out_avg_gueue_time",
  MAX (answbuss30out.cc_queue_time)
    AS "answbuss30out_max_queue_time",


--Internal calls--
  COUNT (internal_calls."cc_call_id") AS "internal_calls",
  SUM (CASE WHEN internal_calls.cc_talk_time > 0
    AND internal_calls.cc_agent_id > 0 THEN 1 ELSE 0 END)
    AS "internal_calls_answered",
   SUM (CASE WHEN internal_calls.cc_talk_time = 0
    AND internal_calls.cc_agent_id = 0
      AND internal_calls.cc_callback=0 THEN 1 ELSE 0 END)
    AS "internal_calls_abandoned",
   SUM (CASE WHEN internal_calls.cc_talk_time > 0
    AND internal_calls.cc_agent_id > 0
      AND internal_calls.cc_queue_time >= 30 THEN 1 ELSE 0 END)
    AS "internal_calls_answered_after30",
   SUM (internal_calls.cc_talk_time)
    AS "internal_calls_total_talk_time",
   AVG (internal_calls.cc_talk_time)
    AS "internal_calls_avg_talk_time",
  AVG (internal_calls.cc_queue_time)
    AS "internal_calls_avg_gueue_time",
  MAX (internal_calls.cc_queue_time)
    AS "internal_calls_max_queue_time",


--External calls--
  COUNT (ext_calls."cc_call_id") AS "ext_calls",
  SUM (CASE WHEN ext_calls.cc_talk_time > 0
    AND ext_calls.cc_agent_id > 0 THEN 1 ELSE 0 END)
    AS "ext_calls_answered",
   SUM (CASE WHEN ext_calls.cc_talk_time = 0
    AND ext_calls.cc_agent_id = 0
      AND ext_calls.cc_callback=0 THEN 1 ELSE 0 END)
    AS "ext_calls_abandoned",
   SUM (CASE WHEN ext_calls.cc_talk_time > 0
    AND ext_calls.cc_agent_id > 0
      AND ext_calls.cc_queue_time >= 30 THEN 1 ELSE 0 END)
    AS "ext_calls_answered_after30",
   SUM (ext_calls.cc_talk_time)
    AS "ext_calls_total_talk_time",
   AVG (ext_calls.cc_talk_time)
    AS "ext_calls_avg_talk_time",
  AVG (ext_calls.cc_queue_time)
    AS "ext_calls_avg_gueue_time",
```

```
  MAX (ext_calls.cc_queue_time)
    AS "ext_calls_max_queue_time",


--Contact resolution by primary agent--
  COUNT (prim."cc_call_id") AS "prim_calls",
  SUM (prim.cc_talk_time) AS "prim_total_talk_time",
  AVG (prim.cc_talk_time) AS "prim_avg_talk_time",
  AVG (prim.cc_queue_time) AS "prim_avg_gueue_time",
  MAX (prim.cc_queue_time) AS "prim_max_queue_time",


--Contact resolution by overflow agent--
  COUNT (over."cc_call_id") AS "over_calls",
  SUM (over.cc_talk_time) AS "over_total_talk_time",
  AVG (over.cc_talk_time) AS "over_avg_talk_time",
  AVG (over.cc_queue_time) AS "over_avg_gueue_time",
  MAX (over.cc_queue_time) AS "over_max_queue_time",

FROM tblcalls c0, tblqueues, tblcallscc c

LEFT JOIN tblcallscc  ext_calls
  ON (ext_calls.cc_call_id = c.cc_call_id
  AND ext_calls.cc_call_id
  IN (SELECT c1.call_id
  FROM tblcalls c1
  WHERE (c1.call_calling_number IS NOT NULL
  AND char_length(c1.call_calling_number) > 5)
  OR (c1.call_called_number IS NOT NULL
  AND char_length(c1.call_called_number) > 5)))

LEFT JOIN tblcallscc internal_calls
  ON (internal_calls.cc_call_id = c.cc_call_id
  AND internal_calls.cc_call_id
  NOT IN (SELECT c1.call_id
  FROM tblcalls c1
  WHERE (c1.call_calling_number IS NOT NULL
  AND char_length(c1.call_calling_number) > 5)
  OR (c1.call_called_number IS NOT NULL
  AND char_length(c1.call_called_number) > 5)))

LEFT JOIN tblcallscc aband
  ON (aband.cc_call_id = c.cc_call_id
  AND aband.cc_talk_time = 0
  AND aband.cc_agent_id = 0
   AND aband.cc_callback = 0)

LEFT JOIN tblcallscc aband3
  ON (aband3.cc_call_id = aband.cc_call_id
  AND aband3.cc_queue_time < 3)

LEFT JOIN tblcallscc aband3_20
  ON (aband3_20.cc_call_id = aband.cc_call_id
  AND aband3_20.cc_queue_time >= 3
  AND aband3_20.cc_queue_time <= 20)
```

```
LEFT JOIN tblcallscc aband20_30
  ON (aband20_30.cc_call_id = aband.cc_call_id
  AND aband20_30.cc_queue_time > 20
  AND aband20_30.cc_queue_time <= 30)

LEFT JOIN tblcallscc aband30up
  ON (aband30up.cc_call_id = aband.cc_call_id
  AND aband30up.cc_queue_time >= 30)

LEFT JOIN tblcallscc answ
  ON (answ.cc_call_id = c.cc_call_id
  AND answ.cc_talk_time > 0
  AND answ.cc_agent_id > 0)

LEFT JOIN tblcallscc callback
  ON (callback.cc_call_id = c.cc_call_id
  AND callback.cc_callback = 1)

LEFT JOIN tblcallscc  answbuss
  ON (answbuss.cc_call_id = answ.cc_call_id
  AND answbuss.cc_call_id
  IN (SELECT c1.call_id
  FROM tblcalls c1, tblswitches s
  WHERE "time"(c1."call_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(c1."call_start_time") <=
      "time"(s.switch_office_end)))

LEFT JOIN tblcallscc answbuss30
  ON (answbuss30.cc_call_id = answbuss.cc_call_id
  AND answbuss30.cc_queue_time >= 30)

LEFT JOIN tblcallscc answbussout
  ON (answbussout.cc_call_id = answ.cc_call_id
  AND answbussout.cc_call_id
  NOT IN (SELECT c1.call_id
  FROM tblcalls c1, tblswitches s
  WHERE "time"(c1."call_start_time") >=
      "time"(s.switch_office_start)
  AND "time"(c1."call_start_time") <=
      "time"(s.switch_office_end)))

LEFT JOIN tblcallscc answbuss30out
  ON (answbuss30out.cc_call_id = answbussout.cc_call_id
  AND answbuss30out.cc_queue_time >= 30)

LEFT JOIN tblcallscc prim
  ON (prim.cc_call_id = answ.cc_call_id
  AND prim.cc_agent_overflow = 0)

LEFT JOIN tblcallscc over
  ON (over.cc_call_id = answ.cc_call_id
  AND over.cc_agent_overflow = 1)
```

```
WHERE c0."call_start_time" >= ? /*from date*/
  AND c0."call_start_time" <= ("date"(?) +
    INTERVAL '24 hours') /*to date*/
  AND c."cc_call_id" = c0."call_id"
  AND c."cc_queue_id" = queue_id
  AND queue_name = ?
```

Exception

N/A

3.7 Report Group - Queues

All predefined report templates of this report group are described below.

3.7.1 Agent Calls Queue Specific

The report displays information about the percentage and number of calls received by agents for selected queue in specified date range.

Required input parameters	• From date • To date (until) • Queue name • Daily report
Output values	• Agent • Percentage of calls received by agents (by Queue) • Number of calls by agent • Percentage of total number of calls (all agents) • Percentage of total talk time (all agents) • Total number of calls for all agents
Format	• Table and graphics
Axis label	• Horizontal: Agents • Vertical: Number of calls
Calculation rule	• Number of calls (by agent): COUNT(number of calls by agent) • Percentage of total number of calls (all agents) : (number of calls (by agent) / total number of calls) * 100 • Percentage of total talk times (all agents) : (talk time (by agent) / total talk time) * 100 • Total number of calls: COUNT(number of calls)
Database tables	• tblcallscc, tblcalls, tblusers, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time} • tblcalls = {call_id, call_start_time} • tblusers = {user_id, user_firstname, user_surname, user_login, user_is_agent} • tblqueues = {queue_name, queue_id}

SQL Queries

Select all available queues (used for selecting the queue)

```
SELECT tblqueues."queue_name"
FROM tblqueues
GROUP BY tblqueues."queue_name"
```

Select report details (agent first-name, surname and login, number of calls and total talk time) for selected queue in the specified date range

```
SELECT u."user_firstname", u."user_surname", u.user_login,
  COUNT (cc."cc_call_id") AS "NumberOfCalls",
  SUM (cc."cc_talk_time") AS "TotalTalkTime"

FROM tblcallscc cc, tblcalls, tblusers u, tblqueues
WHERE cc."cc_agent_id" = u."user_id"
    AND u.user_is_agent = 1
    AND cc."cc_call_id" = tblca1ls."call_id"
    AND tblcalls."call_start_time" >= ? /* from time */
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                            '24 hours')/* to date */
    AND cc."cc_queue_id" = tblqueues.queue_id
    AND tblqueues.queue_name = ? /* queue name */

GROUP BY u."user_firstname", u."user_surname", u.user_login
```

Exception

The user_login is used in the case when the user_firstname and user_surname are not specified.

When a report shows information about contact center agent(s) the user_is_agent value must be 1.

3.7.2 Agent Properties

The report displays the agent properties for all available agents.

Required input parameters	• N/A
Output values (the values are grouped by agents)	• Agent • Queue • Agent type (primary or overflow) • Callback calls (yes or no) • Start calls overflow • Start seconds overflow - seconds of call in queue before it is delivered to overflow agent • Work time - in seconds • Grand totals for start call overflow, start seconds overflow and work time

Format	• Table
Axis label	• N/A
Calculation rule	• Total start call overflow (by agent): SUM(start call overflow by agent) • Total start seconds overflow (by agent): SUM(start seconds overflow by agent) • Total work time: SUM(work time by agent)
Database tables	• tblagentqueues, tblusers, tblqueues
Database table attributes	• tblagentqueues = {aq_user_id, aq_queue_id, aq_agent_type, aq_callback, aq_start_calls, aq_start_seconds, aq_worktime} • tblusers = {user_id, user_login, user_firstname, user_surname, user_extension, user_is_agent} • tblqueues = {queue_id, queue_name}

SQL Queries

Select agent properties

```
SELECT aq_user_id,
       aq_queue_id,
       queue_name,
       aq_agent_type,
       aq_callback,
       aq_start_calls,
       aq_start_seconds,
       aq_worktime

FROM tblagentqueues, tblqueues

WHERE queue_id = aq_queue_id

ORDER BY aq_user_id
```

Select agents

```
SELECT aq_user_id,
       user_firstname,
       user_surname,
       user_extension,
       user_login

FROM tblagentqueues, tblusers

WHERE aq_user_id = user_id
  AND user_is_agent = 1

ORDER BY user_firstname, user_surname, user_login
```

Exception

N/A

3.7.3 Agent Queue Load

The report shows queue load information for the specified agent in the specified date range.

Required input parameters	• From date • To date (until) • Agent • Daily report
Output values	• Queue • Number of calls (by queue) • Percentage of total number of calls • Total number of calls
Format	• Table and graphics
Axis label	• Horizontal: Queues • Vertical: Number of calls
Calculation rule	• Number of calls: COUNT(number of calls by queue) • Total number of calls : SUM(number of calls) • Percentage of total number of calls: (number of calls (by queue) / total number of calls) * 100
Database tables	• tblcalls, tblcallscc, tblusers, tbldepartments, tblqueues
Database table attributes	• tblcalls = {call_id, call_start_time} • tblcallscc = {cc_call_id, cc_queue_id, cc_agent_id} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_is_agent, user_department_id} • tbldepartments = {department_name, department_id} • tblqueues = {queue_name, queue_id}

SQL Queries

Select agent queue load (number of calls per queue, queue name)

```
SELECT COUNT (tblcalls."call_id"),
       tblqueues."queue_name"

FROM tblcallscc, tblqueues, tblcalls, tblusers u

WHERE tblcalls."call_start_time" >= ? /* from time */
  AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                          '24 hours') /* to date */
  AND tblcallscc."cc_call_id" = tblca1ls."call_id"
  AND tblcallscc."cc_queue_id" = tblqueues."queue_id"
  AND tblcallscc."cc_agent_id" = u.user_id
  AND u.user_login = ? /* agent login */

GROUP BY tblqueues."queue_name"
ORDER BY tblqueues."queue_name"
```

Select all available agents (used for selecting the agent)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
WHERE u.user_is_agent = 1
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected agent

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login, (SELECT CASE WHEN
tbldepartments.department_name = 'Unknown'
  THEN '' ELSE tbldepartments.department_name END
  FROM tbldepartments,
  WHERE u.user_department_id = tbldepartments.department_id )
    AS department_name

FROM tblusers u

WHERE u.user_login = ? /* agent login */
```

Exception

The maximum of queues (vertical tubes) shown in a graphic is 15. If there are more than 15 agents, the graphic will not be displayed because with more than 15 agents the graphic is not properly visible.

3.7.4 Avg. G.O.S Per Queue

Hourly representation of the average grade of service GOS for the specified queue in the selected date range.

INFO: The report template **Avg. G.O.S. Per Queue (Daily)** has a different graphic for each day.

Required input parameters	• From date • Until (to date) • Queue • Daily report
Output values	• N/A
Format	• Graphic
Axis label	• Horizontal: hourly intervals • Vertical: average grade of service (0-100)
Calculation rule	• Average GOS (per hour) : AVG(tblcallscc.cc_gos for specific hourly interval) (average: arithmetic mean)
Database tables	• tblcallscc, tblcalls, tblqueues

Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id, cc_gos} • tblcalls = {call_id, call_start_time) • tblqueues = {queue_name, queue_id}

SQL Queries

Select average grade of service for the selected queue in the selected date range

```
SELECT AVG (cc1."cc_gos"),
  EXTRACT (hour FROM c1."call_start_time") || ':00 - ' ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ':00'
  AS "label"

FROM tblcallscc cc1, tblcalls c1, tblqueues q1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_queue_id" = q1."queue_id"
  AND q1."queue_name" = ? /* queue name */

GROUP BY EXTRACT (hour FROM c1."call_start_time")
ORDER BY EXTRACT (hour FROM c1."call_start_time")
```

Select all available queues (used for selecting the queue)

```
SELECT tblqueues."queue_name"
FROM tblqueues
ORDER BY tblqueues."queue_name"
```

Exception

N/A

3.7.5 Avg. G.O.S. Per Queue (Daily)

Hourly representation of the average grade of service GOS for specified queue in the selected date range (there is a different graphic for each day).

Required input parameters	• From date • Until (to date) • Queue • Daily report
Output values	• N/A
Format	• Graphic
Axis label	• Horizontal: hourly intervals • Vertical: average grade of service (0-100)

Calculation rule	• Average GOS (per hour) : AVG(tblcallscc.cc_gos for specific hourly interval) (average: arithmetic mean)
Database tables	• tblcallscc, tblcalls, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id, cc_gos} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_name, queue_id}

SQL Queries

Select average grade of service for the selected queue in the selected range

```
SELECT AVG (cc1."cc_gos"),
  EXTRACT (hour FROM c1."call_start_time")|| ':00 - ' ||
  (EXTRACT (hour FROM c1."call_start_time") + 1) || ':00'
  AS "label"

FROM tblcallscc cc1, tblcalls c1, tblqueues q1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_queue_id" = q1."queue_id"
  AND q1."queue_name" = ? /* queue name */

GROUP BY EXTRACT (hour FROM c1."call_start_time")
ORDER BY EXTRACT (hour FROM c1."call_start_time")
```

Select all available queues (used for selecting the queue)

```
SELECT tblqueues."queue_name"
FROM tblqueues
ORDER BY tblqueus."queue_name"
```

Select all available days having calls for the selected queue in the specified date range

```
SELECT DISTINCT "date"(c1."call_start_time")

FROM tblcallscc cc1, tblcalls c1, tblqueues q1

WHERE c1."call_start_time" >= ? /* from time */
  AND c1."call_start_time" <= ("date"(?) + INTERVAL '24 hours')
                              /* to date */
  AND cc1."cc_call_id" = c1."call_id"
  AND cc1."cc_queue_id" = q1."queue_id"
  AND q1."queue_name" = ? /* queue name */
```

Exception

In the SQL query above, predefined postgresql functions are used to extract the hour value from the specified date time value (call_start_time).

Example of “label” value (representing one hourly interval) : 16:00-17:00

3.7.6 Missed Calls Per Queue

Missed calls grouped by queues for call in the specified date range.

Required input parameters	• From date • Until (to date) • Business hours only (else 24/24) • Daily report
Output values	• Queue name • Number of missed calls (per queue) • Percentage of total number of missed calls • Total number of missed calls
Format	• Table and graphic (pie chart)
Axis label	• Horizontal: queue name • Vertical: number of missed calls
Calculation rule	• Missed call = {aa_event_type = 6} • Total number of missed calls : SUM(number of missed calls(per queue)) • Percentage of total missed calls : (number of missed calls (per queue) / total number of missed calls) * 100 • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblqueues, tblswitches, tblagentactivity
Database table attributes	• tblqueues = {queue_name, queue_id} • tblagentactivity = {aa_call_id, aa_queue_id, aa_event_time, aa_event_type} • tblswitches = {switch_office_start, switch_office_end}

SQL Queries

Select all missed calls in the selected date range

```
SELECT tblqueues.queue_name,
  COUNT (aa."aa_call_id") AS "Number of calls"

FROM tblqueues, tblagentactivity aa, tblswitches s

WHERE aa.aa_event_time >= ? /* from time */
  AND aa.aa_event_time <= ("date"(?) + INTERVAL '24 hours')
                            /* to date */
  AND aa."aa_queue_id" = tblqueues."queue_id"
  AND aa.aa_event_type = 6
  AND ( CASE WHEN ? = 1 THEN /* Business hours only */
      "time"(aa.aa_event_time) >= "time"(s.switch_office_start)
    AND "time"(aa.aa_event_time) <= "time"(s.switch_office_end)
      WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(aa.aa_event_time) >= '00:00:00'
    AND "time"(aa.aa_event_time) <= '23:59:59'
  END )

GROUP BY tblqueues."queue_name"
```

Exception

N/A

3.7.7 Queue Summary Details

Queue summary details for selected queue and specified date range.

Required input parameters	• From date • To date (until) • Queue • Daily report
Output values	• Answered calls • Abandoned calls • Other calls • Maximum queue time for answered calls • Minimum queue time for answered calls • Average queue time for answered calls • Maximum queue time for abandoned calls • Minimum queue time for abandoned calls • Average queue time for abandoned calls • Maximum talk time for answered calls • Minimum talk time for answered calls • Average talk time for answered calls • Total number of calls for all agents • Average grade of service for selected queue
Format	• Grid and graphics
Axis label	• Horizontal: Number of calls • Vertical: Call types (answered/abandoned calls)
Calculation rule	• Predefined postgresql function are used (SUM, COUNT, MIN, MAX, AVG) • Answered call: {talk time > 0} • Abandoned call: {talk time = 0, callback = 0, agent ID = 0}
Database tables	• tblcallscc, tblcalls, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id, cc_gos, cc_talk_time, cc_callback, cc_agent_id} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_name, queue_id}

SQL Queries

Select all available queues (used for selecting the queue)

```
SELECT tblqueues.queue_name
FROM tblqueues
```

Select queue summary details (grade of service, total number of calls, abandoned calls and answered calls) for the selected date range

```
SELECT
  COUNT (cc."cc_call_id") AS "Count of calls",
  AVG (cc."cc_gos") AS "Avg GOS",
  (SELECT
    COUNT (cc2."cc_call_id")
    FROM tblcallscc cc2, tblcalls c2, tblqueues q2
    WHERE c2."call_start_time" >= ? /* from time */
      AND c2."call_start_time" <= ("date"(?) + INTERVAL
                        '24 hours') /* to date */
      AND cc2."cc_call_id" = c2."call_id"
      AND cc2."cc_talk_time" > 0
      AND cc2."cc_queue_id" = q2."queue_id"
      AND q2."queue_name" = ? /* queue name */
  ) AS "answered",
  (SELECT
    COUNT (cc3."cc_call_id")
    FROM tblcallscc cc3, tblcalls c3, tblqueues q3
    WHERE c3."call_start_time" >= ? /* from time */
      AND c3."call_start_time" <= ("date"(?) + INTERVAL
                        '24 hours') /* to date */
      AND cc3."cc_call_id" = c3."call_id"
      AND cc3."cc_talk_time" = 0
      AND cc3."cc_callback" = 0
      AND cc3."cc_agent_id" = 0
      AND cc3."cc_queue_id" = q3."queue_id"
      AND q3."queue_name" = ? /* queue name */
  ) AS "abandoned",
   'Answered Abandoned' AS "X"

FROM tblcallscc cc, tblcalls, tblqueues

WHERE tblcalls."call_start_time" >= ? /* from time */
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                            '24 hours')/* to date */
    AND cc."cc_call_id" = tblcalls."call_id"
    AND cc."cc_queue_id" = tblqueues."queue_id"
    AND tblqueues."queue_name" = ? /* queue name */
```

Select answered call details for the selected queue in selected date range

```
SELECT
  COUNT (cc."cc_call_id") AS "Count of answered calls",
  MAX (cc."cc_queue_time") AS "Max Queue Time",
  MIN (cc."cc_queue_time") AS "Min Queue Time",
  AVG (cc."cc_queue_time") AS "Avg Queue Time",
  MAX (cc."cc_talk_time") AS "Max Talk Time",
  MIN (cc."cc_talk_time") AS "Min Talk Time",
  AVG (cc."cc_talk_time") AS "Avg Talk Time"

FROM tblcallscc cc, tblcalls, tblqueues

WHERE tblcalls."call_start_time" >= ? /* from time */
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                            '24 hours')/* to date */
    AND cc."cc_call_id" = tblcalls."call_id"
    AND cc."cc_queue_id" = tblqueues."queue_id"
    AND cc."cc_talk_time" > 0
    AND tblqueues."queue_name" = ? /* queue name */
```

Select abandoned call details for the selected queue in the selected date range

```
SELECT
  COUNT (cc."cc_call_id") AS "Count of abandoned calls",
  MAX (cc."cc_queue_time") AS "Max Queue Time",
  MIN (cc."cc_queue_time") AS "Min Queue Time",
  AVG (cc."cc_queue_time") AS "Avg Queue Time"

FROM tblcallscc cc, tblcalls, tblqueues

WHERE tblcalls."call_start_time" >= ? /* from time */
    AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                            '24 hours')/* to date */
    AND cc."cc_call_id" = tblcalls."call_id"
    AND cc."cc_queue_id" = tblqueues."queue_id"
    AND cc."cc_talk_time" = 0
    AND cc."cc_callback" = 0
    AND cc."cc_agent_id" = 0
    AND tblqueues."queue_name" = ? /* queue name */
```

Exception

N/A

3.7.8 Queue Traffic Comparison

Queue traffic comparison by numbers of calls for selected date/time range.

Required input parameters	• From date • To date (until) • From time • To time • Business hours only (else 24/24) • Daily report
Output values	• Queue name • Number of calls (by queue) • Percentage of total number of calls • Total number of calls
Format	• Table and graphics (pie chart)
Axis label	• N/A
Calculation rule	• Total number of calls : SUM(Number of calls (per queue)) • Percentage of total number of calls: (number of calls (by queue) / total number of calls) * 100 • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblqueues, tblswitches
Database table attributes	• tblcallscc = {cc_call_id, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblqueues = {queue_name, queue_id} • tblswitches = {switch_office_start, switch_office_end}

SQL Queries

Select the number of calls per queues for the selected date/time range

```
SELECT qq.queue_name
   COUNT (tblcalls."call_id") AS "NumberOfCalls"

FROM tblcallscc cc, tblcalls, tblqueues qq, tblswitches s

WHERE cc."cc_queue_id" = qq."queue_id"
  AND cc."cc_call_id" = tblcalls."call_id"
  AND
   "date"(tblcalls."call_start_time") >= "date" (?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
           "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
           "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
"time"(tblcalls."call_start_time" >= "time"(s.switch_office_start)
    AND  "time"(tblcalls."call_start_time") <=
"time"(s.switch_office_end)
       WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00'
    AND
      "time"(tblcalls."call_start_time") <= '23:59:59'
  END )

GROUP BY qq.queue_name
```

Exception

N/A

3.8 Report Group - User Presence Status

All predefined report templates of this report group are described below.

3.8.1 User Presence Status (All Users) – Daily

The report shows the user presence status details for the “daily” office statuses : meeting, break, lunch, gone out, DND. The report data is selected for a specified day (date) and grouped by users.

Required input parameters	• From date (for day)
Output values (the values are grouped by users)	• Start time • End time • Status (meeting, break, lunch, gone out, DND - do not disturb) • Duration
Format	• Table
Axis label	• N/A
Calculation rule	Status: • 1 - meeting • 3 - break • 4 - gone out • 6 - lunch • 8 - DND • Start time: event time • End time: next event time • Status duration: end time - start time
Database tables	• tblusers, tbluseractivity, tbldepartments
Database table attributes	• tbluseractivity = {ua_id, ua_user_id, ua_time, ua_office_status} • tblusers = {user_id, user_login, user_firstname, user_surname}

SQL Queries

Select all available users having office statuses for the specified day

```
SELECT DISTINCT u.user_id, u.user_surname, u.user_firstname,
                u.user_login
FROM tblusers u

WHERE u.user_id IN
  (SELECT sub1.ua_user_id
    FROM tbluseractivity AS sub1, tbluseractivity AS sub2
    WHERE sub2.ua_id =
     (SELECT ua_id
       FROM tbluseractivity
       WHERE (ua_user_id = sub1.ua_user_id)
    AND (ua_id > sub1.ua_id)
      ORDER BY ua_time ASC
      LIMIT 1)
    AND sub1.ua_time >= ? /* from date: specified day */
    AND sub1.ua_office_status IN (1, 3, 4, 6, 8)
    AND sub1.ua_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date: specified day */
  )

ORDER BY u.user_firstname, u.user_surname
```

Select office status details (user id, status id, start time, end time and status duration) for the selected user and the specified day

```
SELECT u.user_surname, u.user_id,
  CASE sub1.ua_office_status -- WHEN 0 THEN 'office'
  WHEN 1 THEN 1 /* meeting */  -- WHEN 2 THEN 'sick'
  WHEN 3 THEN 3 /* break */
  WHEN 4 THEN 4 /* gone out */  -- WHEN 5 THEN 'holiday'
  WHEN 6 THEN 6 /* lunch */  -- WHEN 7 THEN 'home'
  WHEN 8 THEN 8 /* do not disturb */
  END AS office_status,
  "time"(sub1.ua_time) AS "Start",
  "time"(sub2.ua_time) AS "End",
  (sub2.ua_time - sub1.ua_time) AS duration

FROM tbluseractivity AS sub1,
     tbluseractivity AS sub2,
     tblusers u
```

```
WHERE sub2.ua_id =
     (SELECT ua_id
       FROM tbluseractivity
       WHERE (ua_user_id = sub1.ua_user_id)
    AND (ua_id > sub1.ua_id)
      ORDER BY ua_time ASC
      LIMIT 1)
    AND sub1.ua_time >= ? /* from date: specified day */
    AND sub1.ua_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date: specified day */
    AND sub1.ua_user_id = u.user_id
    AND sub1.ua_office_status IN (1, 3, 4, 6, 8)

ORDER BY sub1.ua_time
```

Exception

N/A

3.8.2 User Presence Status (All Users)

The report shows the user presence status details for the two “longest” statuses: sick and holiday. The duration of these statuses in most of the cases will be in days unlike the duration of the others office statuses (meeting, break, gone out, lunch and DND) usually measured in minutes and hours.

Required input parameters	• From date • Until (to date) • Daily report
Output values (the values are grouped by users)	• User (shows the user first name and surname, when both are empty, the user login is used) • Start time • End time • Status (sick or holiday) • Duration • Total duration time for all users
Format	• Table
Axis label	• N/A
Calculation rule	Status • 2 - sick • 5 - vacation (holiday) • Start time: event time • End time: next event time • Status duration: end time - start time
Database tables	• tblusers, tbluseractivity

Database table attributes	• tbluseractivity = {ua_id, ua_user_id, ua_time, ua_office_status} • tblusers = {user_id, user_login, user_firstname, user_surname}

SQL Queries

Select all available users having present statuses (sick and holiday) for the specified date range

```
SELECT DISTINCT u.user_id, u.user_surname, u.user_firstname,
                u.user_login

FROM tblusers u

WHERE u.user_id IN
  (SELECT sub1.ua_user_id
    FROM tbluseractivity AS sub1, tbluseractivity AS sub2
    WHERE sub2.ua_id =
     (SELECT ua_id
       FROM tbluseractivity
       WHERE (ua_user_id = sub1.ua_user_id)
    AND (ua_id > sub1.ua_id)
      ORDER BY ua_time ASC
      LIMIT 1)
    AND sub1.ua_time >= ? /* from time */
    AND sub1.ua_office_status IN (2, 5)
    AND sub1.ua_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
ORDER BY u.user_firstname, u.user_surname, u.user_login
```

Select status (sick and holidays) details (user, date/ time, status, duration) grouped by users in the specified date range

```
SELECT u.user_surname, u.user_firstname, u.user_login,
  CASE sub1.ua_office_status
  WHEN 2 THEN 'sick'
  WHEN 5 THEN 'holiday'
  END AS office_status,
  (sub2.ua_time - sub1.ua_time) AS duration,
  CASE sub1.ua_office_status
  WHEN 2 THEN sub1.ua_time END AS SStartT,
  CASE sub1.ua_office_status
  WHEN 2 THEN sub2.ua_time END AS SEndT,
  CASE sub1.ua_office_status
  WHEN 5 THEN sub1.ua_time END AS HStartT,
  CASE sub1.ua_office_status
  WHEN 5 THEN sub2.ua_time END AS HEndT

FROM tbluseractivity AS sub1,
     tbluseractivity AS sub2,
     tblusers u
```

```
WHERE sub2.ua_id =
     (SELECT ua_id
       FROM tbluseractivity
       WHERE (ua_user_id = sub1.ua_user_id)
    AND (ua_id > sub1.ua_id)
      ORDER BY ua_time ASC
      LIMIT 1)
    AND sub1.ua_time >= ? /* from time */
    AND sub1.ua_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
    AND sub1.ua_user_id = u.user_id
    AND sub1.ua_office_status IN (2,5)

ORDER BY sub1.ua_user_id, u.user_surname, sub1.ua_office_status,
         sub1.ua_time, sub2.ua_time
```

Select total status (sick and holiday) duration time (for all users) in the specified date range

```
SELECT SUM (EXTRACT
  (EPOCH FROM (sub2.ua_time - sub1.ua_time))) AS duration

FROM tbluseractivity AS sub1,
     tbluseractivity AS sub2

WHERE sub2.ua_id =
  (SELECT ua_id
    FROM tbluseractivity
    WHERE (ua_user_id = sub1.ua_user_id)
    AND (ua_id > sub1.ua_id)
    ORDER BY ua_time ASC
      LIMIT 1)
    AND sub1.ua_time >= ? /* from time */
    AND sub1.ua_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
    AND sub1.ua_office_status IN (2,5)
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)

Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.8.3 User Presence Status (By User) – Daily

The report shows the user presence status details for the “daily” office statuses : meeting, break, lunch, gone out, DND. The report data is selected for a specified user and the specified day (date).

Required input parameters	• From date (for day) • User
Output values	• Start time • End time • Status (meeting, break, lunch, gone out, DND - do not disturb) • Duration
Format	• Table and graphic
Axis label	• Horizontal: office status duration • Vertical: office statutes (meeting, break, lunch, gone out, DND)
Calculation rule	Status: • 1 - meeting • 3 - break • 4 - gone out • 6 - lunch • 8 - DND • Start time: event time • End time: next event time • Status duration: end time - start time
Database tables	• tblusers, tbluseractivity, tbldepartments
Database table attributes	• tbluseractivity = {ua_id, ua_user_id, ua_time, ua_office_status} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_extension, user_department_id} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select office status details (user id, status id, start time, end time and status duration) for the selected user and the specified day

```
SELECT sub1.ua_id, sub1.ua_user_id,
  CASE sub1.ua_office_status
  WHEN 1 THEN 1 /* meeting */  -- WHEN 2 THEN 'sick'
  WHEN 3 THEN 3 /* break */
  WHEN 4 THEN 4 /* gone out */  -- WHEN 5 THEN 'holiday'
  WHEN 6 THEN 6 /* lunch */  -- WHEN 7 THEN 'home'
  WHEN 8 THEN 8 /* do not disturb */
  END AS office_status,
  sub1.ua_time AS "Start",
  sub2.ua_time AS "End",
  "time"(sub1.ua_time) AS "Start Time",
  "time"(sub2.ua_time) AS "End time",
  "date"(sub1.ua_time) AS "Start Date",
  (sub2.ua_time - sub1.ua_time) AS duration

FROM tbluseractivity AS sub1,
     tbluseractivity AS sub2,
     tblusers u
```

```
WHERE sub2.ua_id =
     (SELECT ua_id
       FROM tbluseractivity
       WHERE (ua_user_id = sub1.ua_user_id)
    AND (ua_id > sub1.ua_id)
      ORDER BY ua_time ASC
      LIMIT 1)
    AND sub1.ua_time >= ? /* from time */
    AND sub1.ua_time <= ("date"(?) + INTERVAL '24 hours')
                           /* to date */
    AND sub1.ua_user_id = u.user_id
    AND u.user_login = ? /* user login */
    AND sub1.ua_office_status IN (1, 3, 4, 6, 8)

ORDER BY sub1.ua_time;
```

Select office status details for the selected user and the specified day (used for the graphic)

```
SELECT sub1.ua_id, sub1.ua_user_id,
  CASE sub1.ua_office_status -- when 0 THEN 'office'
  WHEN 1 THEN 'meeting'  -- when 2 THEN 'sick'
  WHEN 3 THEN 'break'
  WHEN 4 THEN 'gone out'  -- when 5 THEN 'holiday'
  WHEN 6 THEN 6 'lunch'  -- when 7 THEN 'home'
  WHEN 8 THEN 8 'do not disturb'
  END AS office_status, -- meeting
  CASE sub1.ua_office_status
    WHEN 1 THEN sub1.ua_time END AS "MeetingStartT",
  CASE sub1.ua_office_status
    WHEN 1 THEN sub2.ua_time END AS "MeetingEndT",  -- break
  CASE sub1.ua_office_status
    WHEN 3 THEN sub1.ua_time END AS "BreakStartT",
  CASE sub1.ua_office_status
    WHEN 3 THEN sub2.ua_time END AS "BreakEndT",  -- gone out
  CASE sub1.ua_office_status
    WHEN 4 THEN sub1.ua_time END AS "OutStartT",
  CASE sub1.ua_office_status
    WHEN 4 THEN sub2.ua_time END AS "OutEndT",  -- lunch
  CASE sub1.ua_office_status
    WHEN 6 THEN sub1.ua_time END AS "LunchStartT",
  CASE sub1.ua_office_status
    WHEN 6 THEN sub2.ua_time END AS "LunchEndT",  -- not disturb
  CASE sub1.ua_office_status
    WHEN 8 THEN sub1.ua_time END AS "NotDisturbStartT",
  CASE sub1.ua_office_status
    WHEN 8 THEN sub2.ua_time END AS "NotDisturbEndT",
  5 AS X

FROM tbluseractivity AS sub1,
     tbluseractivity AS sub2,
     tblusers u
```

```
WHERE sub2.ua_id =
     (SELECT ua_id
       FROM tbluseractivity
       WHERE (ua_user_id = sub1.ua_user_id)
    AND (ua_id > sub1.ua_id)
      ORDER BY ua_time ASC
      LIMIT 1)
    AND sub1.ua_time >= ? /* from time */
    AND sub1.ua_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
    AND sub1.ua_user_id = u.user_id
    AND u.user_login = ? /* user login */
    AND sub1.ua_office_status IN (1, 3, 4, 6, 8)

ORDER BY sub1.ua_time;
```

Exception

N/A

3.8.4 User Presence Status (By User)

The report shows the user presence status details for the two “longest” statuses: sick and holiday. The report data is selected for a specified user in the specified date range. The duration of these statuses in most of the cases will be in days unlike the duration of the others office statuses (meeting, break, gone out, lunch and DND) usually measured in minutes and hours.

Required input parameters	• From date • Until (to date) • User • Daily report
Output values	• Start time • End time • Status (sick or holiday) • Duration • Total duration time for the selected users
Format	• Table
Axis label	• N/A
Calculation rule	Status: • 2 - sick • 5 - vacation (holiday) • Start time = event time • End time: next event time • Status duration: end time - start time
Database tables	• tblusers, tbluseractivity, tbldepartments

Database table attributes	• tbluseractivity = {ua_id, ua_user_id, ua_time, ua_office_status} • tblusers = {user_id, user_login, user_firstname, user_surname, user_email, user_extension, user_department_id} • tbldepartments = {department_name, department_id}

SQL Queries

Select all available users (used for selecting the user)

```
SELECT u.user_login, u.user_surname, u.user_firstname
FROM tblusers u
ORDER BY u.user_firstname, u.user_surname
```

Select details for the selected user

```
SELECT u.user_firstname, u.user_surname, u.user_extension,
       u.user_email, u.user_login,
  (SELECT CASE WHEN tbldepartments.department_name = 'Unknown'
    THEN '' ELSE tbldepartments.department_name
    END
    FROM tbldepartments
    WHERE u.user_department_id = tbldepartments.department_id
    ) AS department_name

FROM tblusers u

WHERE u.user_login = ? /* user login */
```

Select status (sick and holidays) details (date/ time, status, duration) for the selected user in the specified date range

```
SELECT sub1.ua_id, sub1.ua_user_id,
  CASE sub1.ua_office_status
  WHEN 2 THEN 2 /* sick */
  WHEN 5 THEN 5 /* holiday */
  END AS office_status,
  sub1.ua_time AS "Start",
  sub2.ua_time AS "End",
  (sub2.ua_time - sub1.ua_time) AS duration

FROM tbluseractivity AS sub1,
     tbluseractivity AS sub2,
     tblusers u
```

```
WHERE sub2.ua_id =
     (SELECT ua_id
       FROM tbluseractivity
       WHERE (ua_user_id = sub1.ua_user_id)
    AND (ua_id > sub1.ua_id)
      ORDER BY ua_time ASC
      LIMIT 1)
    AND sub1.ua_time >= ? /* from time */
    AND sub1.ua_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
    AND sub1.ua_user_id = u.user_id
    AND u.user_id = ? /* user login */
    AND sub1.ua_office_status IN (2,5)

ORDER BY sub1.ua_time;
```

Select total status (sick and holiday) duration time (for all users) for the selected user in the specified date range

```
SELECT SUM (EXTRACT
  (EPOCH FROM (sub2.ua_time - sub1.ua_time))) AS duration

FROM tbluseractivity AS sub1,
     tbluseractivity AS sub2,
     tblusers u

WHERE sub2.ua_id =
  (SELECT ua_id
    FROM tbluseractivity
    WHERE (ua_user_id = sub1.ua_user_id)
    AND (ua_id > sub1.ua_id)
    ORDER BY ua_time ASC
      LIMIT 1)
    AND sub1.ua_time >= ? /* from time */
    AND sub1.ua_time <= ("date"(?) + INTERVAL '24 hours')
                          /* to date */
    AND sub1.ua_user_id = u.user_id
    AND u.user_id = ? /* user login */
    AND sub1.ua_office_status IN (2,5)
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)

Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.9 Report Group - Wrap-up Codes

All predefined report templates of this report group are described below.

3.9.1 Wrap-up Code Usage All Queues

The report shows wrap-up code usage details in the specified date/time range.

Required input parameters	• From date • Until (to date) • From time • To time • Business hours only (else 24/24) • Daily report
Output values	• Wrap-up description • Count (number of calls) - per wrap-up • Percentage of total number of calls • Average Talk Time -per wrap-up • Talk time- per wrap-up • Average queue time - per wrap-up (queue time - the amount of time a caller has been waiting to get connected to an agent) • Total for all previous values (average talk time, talk time, average queue time)
Format	• Table and graphic (pie chart displaying number of calls and percentage of total number of calls per wrap-ups)
Axis label	• Vertical: number of calls • Horizontal: wrap-up description

Calculation rule	• Percentage of total number of calls: COUNT(number of calls (by wrap-up)) / total number of calls * 100 • Average talk time (per wrap-up): AVG(talk time) (average: arithmetic mean) • Average queue time (per wrap-up): AVG(queue time) (average: arithmetic mean) • Total number of calls : COUNT(number of calls) • Total average talk time (per wrap-up): average talk time for all wrap-ups: SUM(average talk time for all wrap-up) / COUNT(average talk time for all wrap-up) (average: arithmetic mean) • Total average queue time (per wrap-up): average queue time for all wrap-ups: SUM(average queue time for all wrap-up) / COUNT(average queue time for all wrap-up) (average: arithmetic mean) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblswitches, tblwrapupscc, tblccwrapups
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time, cc_queue_time, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblswitches = {switch_office_start, switch_office_end} • tblwrapupscc = {wrapup_code, wrapup_queue_id, wrapup_description} • tblccwrapups = {ccw_wc_id, ccw_cc_id}

SQL Queries

Select wrap-up details (total number of calls and talk time, average talk time and queue time) for all queues in the specified date/time range

```
SELECT COUNT
  (DISTINCT tblccwrapups."ccw_cc_id") AS "Number of calls",
  tblwrapupcc."wrapup_description",
  AVG (tblcallscc."cc_talk_time") AS "Avg Talk Time",
  SUM (tblcallscc."cc_talk_time") AS "Total Talk Time",
  AVG (tblcallscc."cc_queue_time") AS "Avg Queue Time"

FROM tblcallscc, tblcalls, tblwrapupcc, tblccwrapups, tblswitches s

WHERE
    "date"(tblcalls."call_start_time") >= "date"(?) /* from time */
  AND "date"(tblcalls."call_start_time") <= "date"(?) /* to date */
  AND "time"(tblcalls."call_start_time") >=
          "time"(to_timestamp(?,'HH:MI:SS')) /* from time */
  AND "time"(tblcalls."call_start_time") <=
          "time"(to_timestamp(?,'HH:MI:SS')) /* to time */
  AND tblccwrapups."ccw_wc_id" = tblwrapupcc."wrapup_code"
  AND tblccwrapups."ccw_cc_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_call_id" = tblcalls."call_id"
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
         "time"(tblcalls."call_start_time") >=
                                "time"(s.switch_office_start)
      AND "time"(tblcalls."call_start_time") <=
                                "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00'
      AND "time"(tblcalls."call_start_time") <= '23:59:59'
    END
GROUP BY tblwrapupcc."wrapup_description"
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)

Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.9.2 Wrap-up Code Usage Per Group

The report shows wrap-up details for the selected wrap-up group in the specified date range.

Required input parameters	• From date • Until (to date) • Wrap-up group • Business hours only (else 24/24) • Daily report
Output values	• Wrap-up description • Count (number of calls) - per wrap-up • Percentage of total number of calls • Average Talk Time -per wrap-up • Talk time- per wrap-up • Average queue time - per wrap-up (queue time - the amount of time a caller has been waiting to get connected to an agent) • Total for all previous values (average talk time, talk time, average queue time)
Format	• Table and graphic (pie chart displaying number of calls and percentage of total number of calls per wrap-ups)
Axis label	• Vertical: number of calls • Horizontal: wrap-up description

Calculation rule	• Percentage of total number of calls: COUNT(number of calls (by wrap-up)) / total number of calls * 100 • Average talk time (per wrap-up): AVG(talk time) (average: arithmetic mean) • Average queue time (per wrap-up): AVG(queue time) (average: arithmetic mean) • Total number of calls : COUNT(number of calls) • Total average talk time (per wrap-up): average talk time for all wrap-ups: SUM(average talk time for all wrap-up) / COUNT(average talk time for all wrap-up) (average: arithmetic mean) • Total average queue time (per wrap-up): average queue time for all wrap-ups: SUM(average queue time for all wrap-up) / COUNT(average queue time for all wrap-up) (average: arithmetic mean) • Business hours only : switch_office_start <= call_start_time <= switch_office_end • 24/24 : 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblswitches, tblwrapupscc, tblccwrapups, tblwrapupgroups
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time, cc_queue_time, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblswitches = {switch_office_start, switch_office_end} • tblwrapupscc = {wrapup_code, wrapup_queue_id, wrapup_description} • tblccwrapups = {ccw_wc_id, ccw_cc_id} • tblccwrapupgroups = {wg_caption, wg_id, wg_parent_id}

SQL Queries

Select all available wrap-up groups (used for selecting the wrap-up group)

```
SELECT w.wg_caption,
  w.wg_id,
  w.wg_parent_id,
  CASE WHEN w.wg_parent_id > 0 THEN
  (SELECT w1.wg_caption FROM tblwrapupgroups w1
    WHERE wg_id = w.wg_parent_id )
  END AS parent_group

FROM tblwrapupgroups w

ORDER BY w.wg_caption
```

Select details for selected wrap-up group

```
SELECT w.wg_caption,
  w.wg_id,
  w.wg_parent_id,
  CASE WHEN w.wg_parent_id > 0 THEN
  (SELECT w1.wg_caption FROM tblwrapupgroups w1
    WHERE wg_id = w.wg_parent_id )
  END AS parent_group

FROM tblwrapupgroups w

WHERE w.wg_id = ? /* wrap-up group */
```

Select wrap-up details (total number of calls and talk time, average talk time and queue time) for the specified wrap-up group in the specified date range

```
SELECT COUNT (tblcallscc."cc_call_id") AS "Number of calls",
  AVG(tblcallscc."cc_talk_time") AS "Avg Talk Time",
  SUM(tblcallscc."cc_talk_time") AS "Total Talk Time",
  AVG(tblcallscc."cc_queue_time") AS "Avg Queue Time",
  tblwrapupcc.wrapup_description

FROM tblcallscc, tblcalls, tblwrapupcc, tblccwrapups,
     tblwrapupgroups, tblswitches s

WHERE tblcalls."call_start_time" >= ? /* from time */
  AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                          '24 hours') /* to date */
  AND tblcallscc."cc_call_id" = tblcalls."call_id"
  AND tblccwrapups."ccw_cc_id" = tblcallscc."cc_call_id"
  AND tblwrapupcc.wrapup_code = tblccwrapups."ccw_wc_id"
  AND tblwrapupcc.wrapup_parent_id = tblwrapupgroups.wg_id
  AND tblwrapupgroups.wg_id = ? /* wrap-up group */
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
          "time"(tblcalls."call_start_time") >=
                                "time"(s.switch_office_start)
      AND "time"(tblcalls."call_start_time") <=
                                "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00'
      AND "time"(tblcalls."call_start_time") <= '23:59:59'
    END

GROUP BY tblwrapupcc.wrapup_description
ORDER BY tblwrapupcc.wrapup_description
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.9.3 Wrap-up Code Usage Per Queue

The report shows wrap-up details for the specified queue.

Required input parameters	• From date • Until (to date) • Queue • Business hours only (else 24/24) • Daily report
Output values	• Wrap-up description • Count (number of calls) - per wrap-up • Percentage of total number of calls • Average Talk Time -per wrap-up • Talk time- per wrap-up • Average queue time - per wrap-up (queue time - the amount of time a caller has been waiting to get connected to an agent) • Total for all previous values (average talk time, talk time, average queue time)
Format	• Table and graphic (pie chart displaying number of calls and percentage of total number of calls per wrap-ups)

Axis label	• Vertical: number of calls • Horizontal: wrap-up description
Calculation rule	• Percentage of total number of calls: COUNT(number of calls (by wrap-up)) / total number of calls * 100 • Average talk time (per wrap-up): AVG(talk time) (average: arithmetic mean) • Average queue time (per wrap-up): AVG(queue time) (average: arithmetic mean) • Total number of calls: COUNT(number of calls) • Total average talk time (per wrap-up): average talk time for all wrap-ups: SUM(average talk time for all wrap-up) / COUNT(average talk time for all wrap-up) (average: arithmetic mean) • Total average queue time (per wrap-up): average queue time for all wrap-ups: SUM(average queue time for all wrap-up) / COUNT(average queue time for all wrap-up) (average: arithmetic mean) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblswitches, tblwrapupscc, tblccwrapups, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time, cc_queue_time, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblswitches = {switch_office_start, switch_office_end} • tblwrapupscc = {wrapup_code, wrapup_queue_id, wrapup_description} • tblccwrapups = {ccw_wc_id, ccw_cc_id} • tblqueues = {queue_name, queue_id}

SQL Queries

Select all available queues (used for selecting the queue)

```
SELECT tblqueues.queue_name
FROM tblqueues
```

Select wrap-up details (total number of calls and talk time, average talk time and queue time) for specified queue in the specified date range

```
SELECT COUNT
  (DISTINCT tblcallscc."cc_call_id") AS "Number of calls",
  tblwrapupcc."wrapup_description",
  AVG (tblcallscc."cc_talk_time") AS "Avg Talk Time",
  SUM (tblcallscc."cc_talk_time") AS "Total Talk Time",
  AVG (tblcallscc."cc_queue_time") AS "Avg Queue Time"

FROM tblcallscc, tblcalls, tblqueues, tblwrapupcc, tblccwrapups,
     tblswitches s

WHERE tblcalls."call_start_time" >= ? /* from time */
  AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                          '24 hours') /* to date */
  AND tblccwrapups."ccw_wc_id" = tblwrapupcc."wrapup_code"
  AND tblccwrapups."ccw_cc_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_call_id" = tblcalls."call_id"
  AND tblcallscc."cc_queue_id" = tblqueues."queue_id"
  AND tblqueues."queue_name" = ? /* queue name */
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
          "time"(tblcalls."call_start_time") >=
                                "time"(s.switch_office_start)
      AND "time"(tblcalls."call_start_time") <=
                                "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00'
      AND "time"(tblcalls."call_start_time") <= '23:59:59'
    END
GROUP BY tblwrapupcc."wrapup_description"
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)

Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

3.9.4 Wrap-up Code Usage Per Wrap-up

The report shows wrap-up details for the specified wrap-up.

Required input parameters	• From date • Until (to date) • Wrap-up description • Business hours only (else 24/24) • Daily report
Output values	• Queue name • Count (number of calls) - per queue • Percentage of total number of calls • Average Talk Time -per queue • Talk time- per queue • Average queue time - per queue • Total for all previous values
Format	• Table and graphic (pie chart displaying number of calls and percentage of total number of calls per queues)
Axis label	• Vertical: number of calls • Horizontal: wrap-up description

Calculation rule	• Percentage of total number of calls: COUNT(number of calls (per queue)) / total number of calls * 100 • Average talk time (per queue): AVG(talk time) (average: arithmetic mean) • Average queue time (per queue): AVG(queue time) (average: arithmetic mean) • Total number of calls: COUNT(number of calls) • Total average talk time: average talk time for all queues: SUM(average talk time for all queues) / COUNT(average talk time for all queues) (average: arithmetic mean) • Total average queue time: average queue time for all queues: SUM(average queue time for all queues) / COUNT(average queue time for all queues) (average: arithmetic mean) • Business hours only: switch_office_start <= call_start_time <= switch_office_end • 24/24: 00:00:00 <= call_start_time <= 23:59:59
Database tables	• tblcallscc, tblcalls, tblswitches, tblwrapupscc, tblccwrapups, tblqueues
Database table attributes	• tblcallscc = {cc_call_id, cc_talk_time, cc_queue_time, cc_queue_id} • tblcalls = {call_id, call_start_time} • tblswitches = {switch_office_start, switch_office_end} • tblwrapupscc = {wrapup_code, wrapup_queue_id, wrapup_description} • tblccwrapups = {ccw_wc_id, ccw_cc_id} • tblqueues = {queue_name, queue_id}

SQL Queries

Select all queues and wrap-up codes (used for selecting the wrap-up)

```
SELECT
  (SELECT queue_name FROM tblqueues q
    WHERE w.wrapup_queue_id = q.queue_id )
  || '->' || w.wrapup_description || ' - ' || w.wrapup_code
  AS wrapup_description,
  w.wrapup_code

FROM tblwrapupcc w

ORDER BY 1
```

Select description for specified wrap-up

```
SELECT
  (SELECT queue_name FROM tblqueues q
    WHERE w.wrapup_queue_id = q.queue_id )
  || '->' || w.wrapup_description || ' - ' || w.wrapup_code
  AS wrapup_description

FROM tblwrapupcc w

WHERE w.wrapup_code = ? /* selected wrap-up code */

ORDER BY 1
```

Select wrap-up details (total number of calls and talk time, average talk time and queue time) for the specified wrap-up in the specified date range

```
SELECT COUNT (tblcallscc."cc_call_id") AS "Number of calls",
  q.queue_name,
  AVG (tblcallscc."cc_talk_time") AS "Avg Talk Time",
  SUM (tblcallscc."cc_talk_time") AS "Total Talk Time",
  AVG (tblcallscc."cc_queue_time") AS "Avg Queue Time"

FROM tblcallscc, tblcalls, tblwrapupcc, tblccwrapups,
     tblswitches s, tblqueues q

WHERE tblcalls."call_start_time" >= ? /* from time */
  AND tblcalls."call_start_time" <= ("date"(?) + INTERVAL
                          '24 hours') /* to date */
  AND tblccwrapups."ccw_wc_id" = tblwrapupcc."wrapup_code"
  AND tblccwrapups."ccw_cc_id" = tblcallscc."cc_call_id"
  AND tblcallscc."cc_call_id" = tblcalls."call_id"
  AND tblcallscc."cc_queue_id" = q.queue_id
  AND tblwrapupcc."wrapup_code" = ?
  AND (CASE WHEN ? = 1 THEN /* Business hours only */
          "time"(tblcalls."call_start_time") >=
                                "time"(s.switch_office_start)
      AND "time"(tblcalls."call_start_time") <=
                                "time"(s.switch_office_end)
    WHEN ? != 1 THEN /* Not Business hours only = 24/24 */
      "time"(tblcalls."call_start_time") >= '00:00:00'
      AND "time"(tblcalls."call_start_time") <= '23:59:59'
    END

GROUP BY q.queue.name
```

Exception

To convert the seconds to time values in this report, but also in many others, the following calculation rules are used :

Parameter	• ts – time in seconds
Problem	• Convert s to d h:m:s
Solution	• d = ts/86400 (86400 seconds in 1 day) d is the number of entire days in seconds • h = (ts – (d*86400))/3600 (3600 seconds in 1 hour) • m = (ts – (d*86400) – (h*3600))/60 (60 seconds in 1 minute) • s = ts – (d*86400) – (h*3600) – (m*60)
Output	• d h:m:s – d – days in ts – h – left hours in ts (after calculation of days) – m – left minutes in ts (after calculation of days and hours) – s – left seconds in ts (after calculation of days, hours and minutes) Depending of the report and of the specified time values, "d" is sometimes not calculated.

4 myReports User Roles

Access to the functions of myReports is controlled via user roles.

Your current user role is set when you log into myReports.

- Logging in as a myReports user:
  - Login Name: This is usually your station number
  - Password: The default password is `1234`.
- Logging in as a myReports administrator:
  - Login Name: The login name is `Administrator`.
  - Password (Administrator Password): The default password is `reports`.

The differences between the roles are summarized in the following table.

myReports: Activity	User Role	
	myReports Users	myReports Administrator
Reports		
Preview report	X	X
Send report immediately by e-mail	X	X
Add report template	X	X
Delete report template	X	X
Start Report Designer		X
Define new report template		X
Update predefined report templates		X
Schedules		
Add a schedule	X	X
Display details of a schedule	X	X
Edit schedule	X	X
Delete schedule	X	X

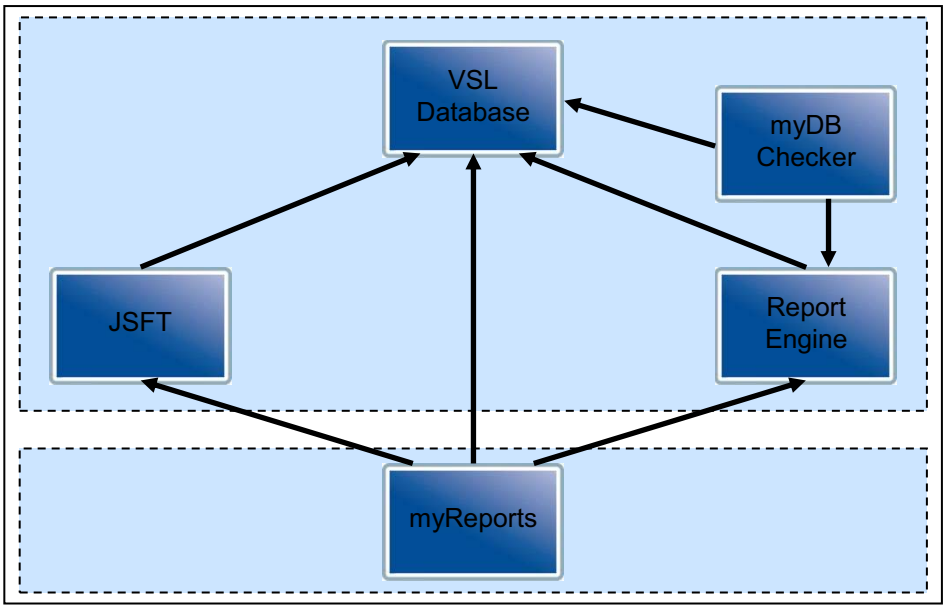
myReports: Activity	User Role	
	myReports Users	myReports Administrator
Configuration		
Change language of user interface	X	X
Change color of user interface	X	X
Configure e-mail template	X[1]	X
Change server address	X	X
Change administrator password		X
Configure e-mail account to send reports by e-mail		X
Configure phone number prefixes		X
Select language		X[2]
Set up default language		X[2]

1 To configure the e-mail template, you will need to enter the administrator password.
2 In order to configure languages and set the default language, you will need to log in as a myReports administrator with a special password.

5 myReports Software Architecture

This section contains information on the architecture and the main components of the myReports software.

Figure: myReports Software Architecture

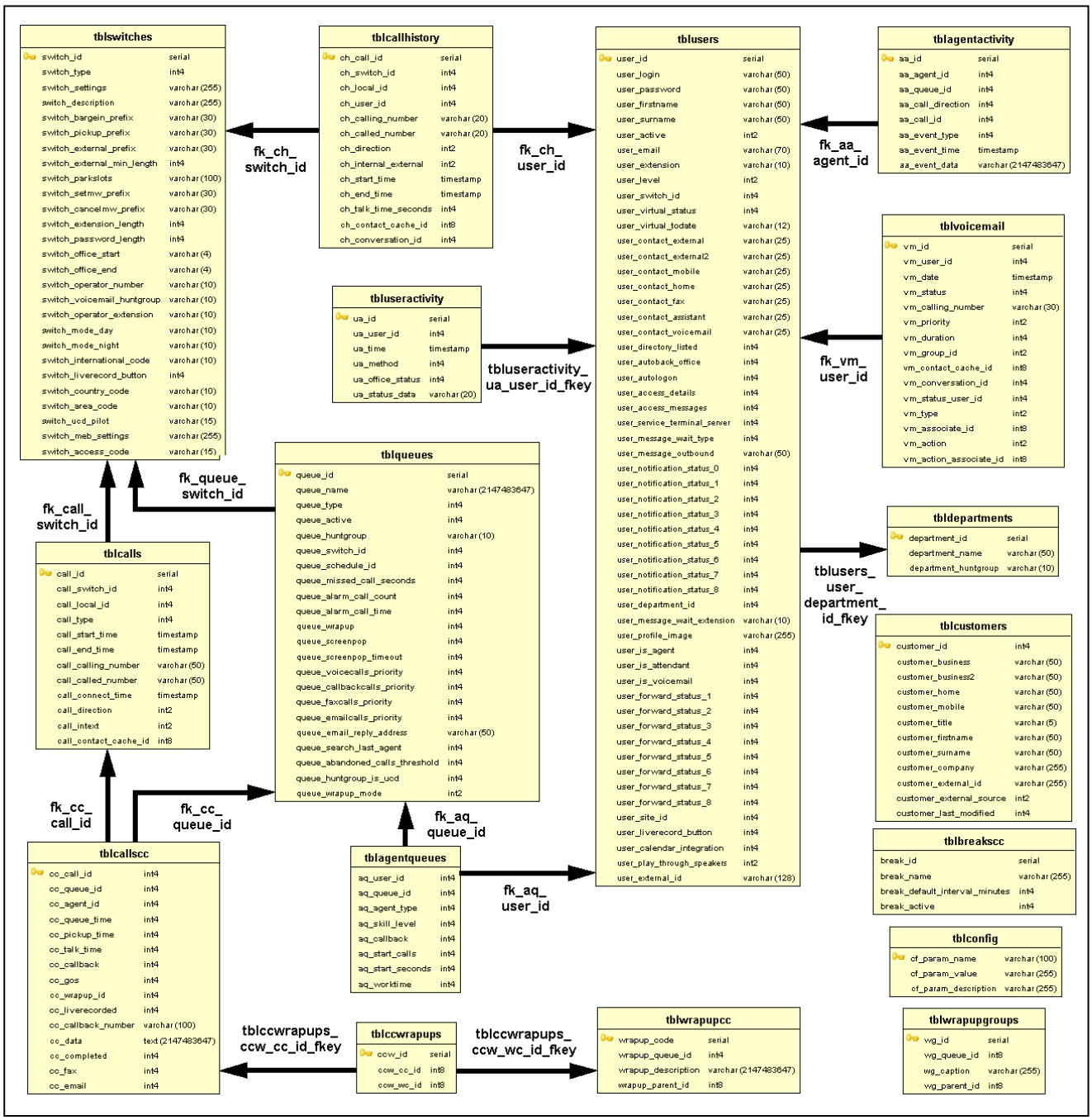


Main components of the myReports software:

- myDBChecker
  Server application (Java) to check report schedules. If a schedule is found, the report is sent to the printer by the application.
- JSFT
  Server application responsible for connecting the myReports client as well as uploading/downloading reports, language resource files and template messages.
- Report Engine
  The Business Intelligence & Reporting Tool (BIRT) is the platform used by myReports.
- myReports
  Client application for creating custom reports, schedules and for running ad-hoc reports.

6 myReports Data Relationship Model

The model shows all database tables and their relationships.



Index

A

abandoned calls per hour 291
abandoned calls statistics 177
    details 181
agent activity
    logged times 55
    missed call times 58
    on break times 61
    status all agents, daily 65
    status by agent, daily 66
    work times 69
agent calls percentage 292
agent calls, queue specific 330
agent performance details 294
agent private calls
    all agents 77
    per agent 79
agent properties 331
agent queue load 333
all user calls by agent 82, 86
answered calls
    per hour 297
    statistics 191
    wrap-up information 195
answered calls alert times
    all agents 185
    by agent 183
    by agent, details 187

B

BIRT RCP Designer 48

C

call traffic
    all agents, per hour daily 200
    all agents, per hour daily (details) 205
    all queues, per hour daily 210
    all queues, queue time and GOS per hour daily 213
    by queue, per hour 299
    by queue, per hour daily 300
    by queue, per hour daily (details) 215
    one agent, per hour daily 218
    one agent, per hour daily (details) 223
    one queue, queue time and GOS per hour daily 229
callback calls 232
calls history per user 269
calls list
    agent 236
    queue 238
contact center
    call definition 50
    call scenarios 51
    calls 248
    calls per agents, chart 240
    calls per agents, list 242
    calls per queues, chart 244
    calls per queues, list 246
    forbidden features 54
contact center summary 253, 255
    answered calls 257
    details 259
contact center traffic
    per hour 302
    per hour, daily 303
creating reports 50

D

data relationship model 372
default break information 271
details per agent, summary 306
details per queue, summary 322
display conventions 6
documentation for the Contact Center 7

E

external calls per user 90
external directory user details 272

F

fax journal
    received faxes by user 273
    sent faxes by user 276
fax transmission report 280

G

grade of service
    per agent 73
    per agent (daily) 75
    per queue 334
    per queue (daily) 335

I
incoming calls
- free calls per user 93
- international calls per user 97
- mobile/cell calls per user 101
- other external calls per user 105
- per user 113
- specific calls per user 109

incoming calls report
- group 116
- group summary 119
- hourly 282
- hourly per weekday 284
- user 121
- user summary 124

internal calls per user 125
internal directory user details 285

M
missed calls 261
- incoming, per user 128, 131
- outgoing, per user 134, 137
- per hour 304
- per queue 337
- summary per agent 265
- summary per queue 266

myReports administrator 369
myReports software architecture 371
myReports users 369

O
outgoing calls
- free calls per user 140
- international calls per user 144
- mobile/cell calls per user 148
- other external calls per user 152
- pay calls per user 156
- per user 164
- specific calls per user 160

outgoing calls report
- group 167
- group summary 170
- user 172
- user summary 175

output formats 50
output values 49

Q
queue summary details 339
queue traffic comparison 342

R
Report Designer 48
report group
- agent activity 9
- agents 10
- call history 12
- calls 19
- other 35
- performance 38
- queues 42
- user presence status 45
- wrap-up codes 46

report parameters 48
report templates 8, 55

S
Schedule Manager 50

U
user presence status
- all users 346
- all users (daily) 344
- by user 352
- by user (daily) 349

user roles 369

V
voicemail center
- all users 286
- by user 288

W
wrap-up code usage
- all queues 356
- per group 359
- per queue 362
- per wrap-up 365