This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Workforce Management ETL Database Reference

5/3/2026

Table of Contents

Getting started

Database tables

Queries and purging

Search the table of all articles in this guide, listed in alphabetical order, to find the article you need.

Related documentation:

-

Preface

Contents

- Administrator

Find out who the intended audience is for this guide before using and preparing the ETL database schema.

Related documentation:

-

The Workforce Management (WFM) Extract, Transform, and Load (ETL) database schema enables Genesys Interactive Insights and other third-party reporting applications to easily create reports that incorporate Genesys WFM data. Once configured, this functionality can obtain Schedule, Adherence, and Performance information from WFM and store it into a documented relational database schema.

The ETL schema can co-exist with the main operational WFM database, be a standalone database, or part of any other database. WFM provides the SQL script to create the database schema, but does not specify which physical tablespace, user, or database on which to create it. The script is included in WFM Database Utility (DBU) IP, but is not executed automatically by the DBU.

Intended audience

This reference guide is intended for:

- Reporting and business analysts who want to leverage the data that is contained in Genesys WFM, Data Mart, Info Mart and other third party applications to produce reports for business users.
- IT administrators who want to gain an understanding of the components that enable WFM.

It is assumed that the reader understands of the following:

- Relational database concepts.
- Structured Query Language (SQL) for querying and mining data.
- Genesys WFM configuration and its data sources.
- Data warehouse concepts—including working with star schema, dimensions, aggregates, and measures.
- Extraction, transformation, and loading (ETL) concepts.

Abbreviation of database terms

This reference uses abbreviations throughout all topics to provide detailed information about and within the tables, including a concise listing of primary and foreign keys, default field values, and mandatory fields for each table. The field and index abbreviations for database terms are described here:

Field characterizations	Index characterizations
P—Primary key	**C**—Cluster
M—Mandatory field	**U**—Unique
F—Foreign key	
DV—Default value	

Overview

Contents

- Administrator

Get a summary of the contents of the database tables in the WFM ETL Database schema.

Related documentation:

-

The Workforce Management (WFM) ETL Database schema contains Dimension, Fact, Service and Control, and Referred Info Mart tables.

The Dimension tables correspond to the WFM Organization, Configuration, and Policy objects, and provide sorting, grouping, and filtering capabilities for reports. The Fact tables contain Adherence, Performance, and Schedule information and can be sorted, grouped, and filtered by dimensions.

The ETL database script includes database-stored procedures that enable automated purging of the WFM ETL database.

This reference contains detailed descriptions of these tables (summarized in this topic), and examples of queries that can be run on the data.

Dimension tables

The WFM ETL process uses the following three types of Dimension tables:

1. **General** tables include:
   - `WFM_BU`—Business unit descriptive information
   - `WFM_SITE`—Site descriptive information
   - `WFM_TEAM`—Team descriptive information
   - `WFM_AGENT`—Agent descriptive information
   - `WFM_ACTIVITY_TYPE`—Activity types
   - `WFM_ACTIVITY`—Activity descriptive information
2. **Schedule** tables include:
   - `WFM_SSG_TYPE`—Schedule State Group types
   - `WFM_SSG`—Schedule State Group descriptive information
   - `WFM_STATE_TYPE`—Schedule state types
   - `WFM_STATE`—Schedule state descriptive information
3. **Performance** tables include:
   - `WFM_PERF_ITEM`—Performance statistics

See the Dimension tables in detail.

Fact tables

The Fact tables provide the following data:

- Agent/team/site adherence totals aggregates for the calendar day and for the 15-minute interval
- Agent/team/site schedule totals aggregates for the schedule day and for the 15-minute interval
- Agent schedule states
- Schedule state and Schedule State Group (SSG) duration aggregates for the 15-minute interval
- Numerous activity (single-site, multi-site, Activity Group) and site performance statistics aggregates for the calendar day and for the 15-minute interval

The WFM ETL process uses the following three types of Fact tables:

1. **Adherence** tables inlcude:
    - `WFM_ADH_AGENT_DAY`—Aggregate of the agent adherence information for 24-hour days
    - `WFM_ADH_AGENT_TIMESTEP`—Aggregate of the agent adherence for 15-minute intervals
2. **Schedule** tables include:
    - `WFM_SCH_AGENT_DAY`—Agent schedule day information
    - `WFM_SCH_AGENT_TIMESTEP`—Aggregate of agent’s schedule totals for 15-minute intervals
    - `WFM_SCH_AGENT_STATE`—Agent Schedule state information
    - `WFM_SCH_AGENT_STATE_TIMESTEP`—Aggregate of schedule state duration for 15-minute intervals
3. **Performance** tables include:
    - `WFM_PERF_ITEM_DAY`—Performance statistics in 24-hour day granularity
    - `WFM_PERF_ITEM_TIMESTEP`—Performance statistics in 15-minute granularity

See the Fact tables in detail.

Service and Control tables

The WFM ETL process uses the following Service and Control tables:

- `WM_DB_VERSION`—Internal version table
- `CTL_ETL_HISTORY`—Parallels the `CTL_ETL_HISTORY` table in the Genesys Info Mart database
- `CTL_AUDIT_LOG`—Parallels the `CTL_AUDIT_LOG` table in the Genesys Info Mart database

See the Service and Control tables in detail.

Referred Genesys Info Mart tables

The WFM ETL process uses the following six referred Info Mart tables:

1. `CTL_AUDIT_LOG`
2. `DATE_TIME`
3. `TIME_ZONE`
4. `GIDB_GC_TENANT`
5. `GIDB_GC_SWITCH`
6. `GIDB_GC_AGENT`

Query examples

The WFM ETL process uses the following three categories of ETL query examples:

1. Adherence queries
2. Schedule queries
3. Performance statistics queries

See the Query examples in detail.

Dimension tables

Contents

- Administrator

Use these tables to define descriptive information for WFM objects in the WFM ETL database schema.

Related documentation:

-

For a description of the abbreviations used in these tables, see Abbreviation of database terms.

When to use IANA time zones

The IANA_TIMEZONE column in the WFM_BU and WFM_SITE tables supports Internet Assigned Numbers Authority (IANA) time zones in WFM business units and sites.

In certain configurations, WFM switches from using proprietary time zones (synchronized with Genesys Configuration Server) to standard IANA time zones (see Internet Assigned Numbers Authority (IANA)). Also, the ETL column TIME_ZONE_KEY associated with BUs and sites becomes NULL as this key represents Configuration Server’s time zone assignment, which WFM no longer uses.

If you are using WFM's classic Forecast UI, you can disable IANA time zones in WFM and restore the previous time zone functionality. However, if you are using WFM's latest Forecast UI, it requires IANA time zones.

> Tip
>
> Some configuration is necessary to support this feature. Contact your Genesys Professional Services representative to implement this feature

WFM_BU

This table contains business unit descriptive information.

Column	Data type	P	M	F	DV
WFM_BU_KEY	int	✔	✔		
WFM_BU_NAME	varchar(255)		✔		
WFM_TIMESTAMP	numeric(19)		✔		
TIME_ZONE_KEY	int			✔	
IANA_TIMEZONE	varchar(255)				

Column	Data type	P	M	F	DV
ACTIVE_FLAG	int		✔		
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_BU_KEY**—The primary key for this table.
- **WFM_BU_NAME**—The name of Business Unit (BU).
- **WFM_TIMESTAMP**—An internal timestamp value.
- **TIME_ZONE_KEY**—The surrogate key used to join the TIME_ZONE dimension to the fact tables. It specifies the time zone of the Business Unit.
- **IANA_TIMEZONE**—The name of the standard IANA time zone. It specifies the time zone of the Business Unit. See more information about IANA time zones in When to use IANA time zones.
- **ACTIVE_FLAG**—Indicates whether the Business Unit is currently active and the corresponding record exists in operational WFM Database: 0 = No, 1 = Yes. See more about ACTIVE_FLAG in Purging terminated agents.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_SITE

This table contains site descriptive information.

Column	Data type	P	M	F	DV
WFM_SITE_KEY	int		✔		
WFM_BU_KEY	int		✔	✔	
WFM_SITE_NAME	varchar(255)		✔		
WFM_TIMESTAMP	numeric(19)		✔		
SWITCH_KEY	int			✔	
TIME_ZONE_KEY	int			✔	
IANA_TIMEZONE	varchar(255)				

Column	Data type	P	M	F	DV
ACTIVE_FLAG	int		✔		
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_SITE_KEY**—The primary key for this table.
- **WFM_BU_KEY**—The surrogate key used to join the WFM_BU dimension to the fact tables. It specifies the Business Unit of the Site.
- **WFM_SITE_NAME**—The name of the Site.
- **WFM_TIMESTAMP**—An internal timestamp value.
- **SWITCH_KEY**—The surrogate key used to join the GIDB_GC_SWITCH dimension to the fact tables. It specifies the switch associated with the Site.
- **TIME_ZONE_KEY**—The surrogate key used to join the TIME_ZONE dimension to the fact tables. It specifies the time zone of the Site.
- **IANA_TIMEZONE**—The name of the standard IANA time zone. It specifies the time zone of the Business Unit. See more information about IANA time zones in When to use IANA time zones.
- **ACTIVE_FLAG**—Indicates whether the Site is currently active and the corresponding record exists in operational WFM Database: 0 = No, 1 = Yes. See more about ACTIVE_FLAG in Purging terminated agents.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data updates. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_TEAM

This table contains team descriptive information.

Column	Data type	P	M	F	DV
WFM_TEAM_KEY	int	✔	✔		
WFM_TEAM_NAME	varchar(255)		✔		
WFM_SITE_KEY	int		✔	✔	

Column	Data type	P	M	F	DV
WFM_TIMESTAMP	numeric(19)		✔		
ACTIVE_FLAG	int		✔		
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_TEAM_KEY**—The primary key for this table.
- **WFM_TEAM_NAME**—The name of the Team.
- **WFM_SITE_KEY**—The surrogate key used to join the WFM_SITE dimension to the fact tables. It specifies the Site of the Team.
- **WFM_TIMESTAMP**—An internal timestamp value.
- **ACTIVE_FLAG**—Indicates whether the Team is currently active and the corresponding record exists in operational WFM Database: 0 = No, 1 = Yes. See more about ACTIVE_FLAG in Purging terminated agents.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_AGENT

This table contains agent descriptive information.

Column	Data type	P	M	F	DV
WFM_AGENT_KEY	int	✔	✔		
WFM_BU_KEY	int			✔	
WFM_SITE_KEY	int			✔	
WFM_TEAM_KEY	int			✔	
EMPLOYEE_ID	varchar(64)		✔		
FIRST_NAME	varchar(64)		✔		
LAST_NAME	varchar(64)		✔		

Column	Data type	P	M	F	DV
HIRE_DATE	date		✔		
TERMINATION_DATE	date				
WFM_TIMESTAMP	numeric(19)		✔		
WM_HOURLY_WAGE	int				
WM_SENIORITY	int				
AGENT_KEY	int			✔	
ACTIVE_FLAG	int		✔		
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_AGENT_KEY**—The primary key for this table.
- **WFM_BU_KEY**—The surrogate key used to join the WFM_BU dimension to the fact tables. It specifies the the Agent's business unit.
- **WFM_SITE_KEY**—The surrogate key used to join the WFM_SITE dimension to the fact tables. It specifies the Agent's site.
- **WFM_TEAM_KEY**—The surrogate key used to join the WFM_TEAM dimension to the fact tables. It specifies the Agent's team. It is NULL if Agent does not belong to any team.
- **EMPLOYEE_ID**—The Agent's employee ID.
- **FIRST_NAME**—The Agent's first name.
- **LAST_NAME**—The Agent's last name.
- **HIRE_DATE**—The Agent's hire date.
- **TERMINATION_DATE**—The Agent's termination date.
- **WFM_TIMESTAMP**—An internal timestamp value.
- **WM_HOURLY_WAGE**—The agent's hourly wage.
- **WM_SENIORITY**—The user-defined rank field.
- **AGENT_KEY**—The surrogate key used to join the GIDB_GC_AGENT dimension to the fact tables.
- **ACTIVE_FLAG**—Indicates whether the agent is currently active and the corresponding record exists in operational WFM Database: 0 = No, 1 = Yes. See more about ACTIVE_FLAG in Purging terminated agents.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key

specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).

- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_ACTIVITY_TYPE

This table contains activity types and descriptions.

Column	Data type	P	M	F	DV
WFM_ACTIVITY_TYPE_KEY	int	✔	✔		
WFM_ACTIVITY_TYPE_NAME	varchar(64)		✔		

Description of Columns

- **WFM_ACTIVITY_TYPE_KEY**—The Activity type ID.
- **WFM_ACTIVITY_TYPE_NAME**—The Activity type name. The table below contains valid values.

ID	Name
0	'Immediate'
2	'Fixed Staffing'
4	'Deferred'
10	'Activity Group'

WFM_ACTIVITY

This table contains activity descriptive information.

Column	Data type	P	M	F	DV
WFM_ACTIVITY_KEY	int	✔	✔		
WFM_BU_KEY	int		✔	✔	
WFM_SITE_KEY	int			✔	
WFM_MSA_KEY	int			✔	
WFM_ACTIVITY_NAME	varchar(255)		✔		
WFM_ACTIVITY_SHORT_NAME	varchar(6)		✔		
WFM_ACTIVITY_TYPE_KEY	int		✔	✔	
WFM_TIMESTAMP	numeric(19)		✔		
ACTIVE_FLAG	int		✔		
TENANT_KEY	int			✔	

Column	Data type	P	M	F	DV
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_ACTIVITY_KEY**—The primary key for this table.
- **WFM_BU_KEY**—The surrogate key used to join the WFM_BU dimension to the fact tables. It specifies the Business Unit of the Activity.
- **WFM_SITE_KEY**—The surrogate key used to join the WFM_SITE dimension to the fact tables. It specifies the Site of the Activity. It is NULL if Activity is Multi-Site Activity (MSA) or Activity Group (AG).
- **WFM_MSA_KEY**—The surrogate key used to join the parent Multi-Site Activity to the child Activity. It is NULL for Multi-Site Activity and Activity Group or if Activity does not belong to any Multi-Site Activity.
- **WFM_ACTIVITY_NAME**—The name of the Activity.
- **WFM_ACTIVITY_SHORT_NAME**—The short name of the Activity.
- **WFM_ACTIVITY_TYPE_KEY**—The surrogate key used to join the WFM_ACTIVITY_TYPE dimension. It specifies the type of the Activity.
- **WFM_TIMESTAMP**—An internal timestamp value.
- **ACTIVE_FLAG**—Indicates whether the Activity is currently active and the corresponding record exists in operational WFM Database: 0 = No, 1 = Yes. See more about ACTIVE_FLAG in Purging terminated agents.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_SSG_TYPE

This table contains schedule state group types and descriptions.

Column	Data type	P	M	F	DV
WFM_SSG_TYPE_KEY	int	✔	✔		
WFM_SSG_TYPE_NAME	varchar(64)		✔		

Description of Columns

- **WFM_SSG_TYPE_KEY**—The Schedule State Group type ID.
- **WFM_SSG_TYPE_NAME**—The Schedule State Group type name. The table below contains valid values.

ID	Name
1	'Working Overhead'
2	'Non-Working Overhead'
3	'Actual Work'

WFM_SSG

This table contains schedule state group descriptive information.

Column	Data type	P	M	F	DV
WFM_SSG_KEY	int	✔	✔		
WFM_SITE_KEY	int		✔	✔	
WFM_SSG_NAME	varchar(255)		✔		
WFM_SSG_TYPE_KEY	int		✔	✔	
WFM_SSG_WEIGHT	int		✔		
WFM_TIMESTAMP	numeric(19)		✔		
ACTIVE_FLAG	int		✔		
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)				
PURGE_FLAG	int				

Description of Columns

- **WFM_SSG_KEY**—The primary key for this table.
- **WFM_SITE_KEY**—*The surrogate key used to join the WFM_SITE dimension to the fact tables. It specifies the site of the Schedule State Group (SSG).
- **WFM_SSG_NAME**—The name of the Schedule State Group.
- **WFM_SSG_TYPE_KEY**—The surrogate key used to join the WFM_SSG_TYPE dimension. It specifies the type of the Schedule State Group.
- **WFM_SSG_WEIGHT**—The superficial weight value of Schedule State Group used for grouping.
- **WFM_TIMESTAMP**—An internal timestamp value.
- **ACTIVE_FLAG**—Indicates whether the Schedule State Group is currently active and the corresponding record exists in operational WFM Database: `0` = No, 1 = Yes. See more about ACTIVE_FLAG in Purging

terminated agents.

- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, enterprise application integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_STATE_TYPE

This table contains schedule state types and descriptions.

Column	Data type	P	M	F	DV
WFM_STATE_TYPE_KEY	int	✔	✔		
WFM_STATE_TYPE_NAME	varchar(64)		✔		

Description of Columns

- **WFM_STATE_TYPE_KEY**—The State type ID.
- **WFM_STATE_TYPE_NAME**—The State type name. The table below contains valid values.

ID	Name
0	'None'
1	'Day Off'
2	'Time Off'
3	'Exception'
4	'Break'
5	'Meal'
6	'Activity'
7	'Activity Set'
8	'Shift'
9	'Marked Time'

WFM_STATE

This table contains schedule state descriptive information.

Column	Data type	P	M	F	DV
WFM_STATE_KEY	int	✔	✔		
WFM_SITE_KEY	int		✔	✔	
WFM_SSG_KEY	int			✔	
WFM_STATE_TYPE_KEY	int		✔	✔	
WFM_STATE_ID	int		✔		
WFM_STATE_NAME	varchar(255)		✔		
WFM_STATE_SHORT_NAME	varchar(6)		✔		
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric (19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_STATE_KEY**—The primary key for this table.
- **WFM_SITE_KEY**—The surrogate key used to join the WFM_SITE dimension to the fact tables. It specifies the Site of the Schedule State.
- **WFM_SSG_KEY**—The surrogate key used to join the WFM_SSG dimension to the fact tables. It specifies the SSG of the Schedule State.
- **WFM_STATE_TYPE_KEY**—The surrogate key used to join the WFM_STATE_TYPE dimension. It specifies the type of the Schedule State.
- **WFM_STATE_ID**—The ID of Schedule State corresponding to the type of Schedule State. The ID is unique within the context of Schedule State type.
- **WFM_STATE_NAME**—The name of the Schedule State.
- **WFM_STATE_SHORT_NAME**—The short name of the Schedule State.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_PERF_ITEM

This table contains performance items (statistics) and descriptions.

Column	Data type	P	M	F	DV
WFM_PERF_ITEM_KEY	int	✔	✔		
WFM_PERF_ITEM_CODE	varchar(64)		✔		
WFM_PERF_ITEM_DESCRIPTION	varchar(256)		✔		

Description of Columns

- **WFM_PERF_ITEM_KEY**—The Performance item ID.
- **WFM_PERF_ITEM_CODE**—The code of Performance item (statistic). The table below contains WFM Performance statistics.
- **WFM_PERF_ITEM_DESCRIPTION**—The description of the Performance item.

Performance Statistics

ID	Code	Description
12	FRC_CALC_STAFFING	Total Calculated Staffing Difference (between Optimal number of agents for Forecast workload and Calculated Staffing)
16	FRC_REQ_STAFFING	Total Required Staffing
38	FRC_CALC_SERVICE_PCT	Weighted average of (Achieved) Calculated Service Level % (weighted on Forecast Interaction Volume)
15	FRC_REQ_SERVICE_PCT	Weighted average of (Achieved) Required Service Level % (weighted on Forecast Interaction Volume) for Activity of type Deferred
40	FRC_CALC_ASA	Weighted average of (Achieved) Calculated Average Speed of Answer (weighted on Forecast Interaction Volume)
14	FRC_REQ_ASA	Weighted average of Required Average Speed of Answer (weighted on Forecast Interaction Volume)
39	FRC_CALC_ABANDONED_IV_PCT	Weighted average of (Achieved) Calculated Abandoned Interaction Volume % (weighted on Forecast Interaction Volume)
18	FRC_REQ_ABANDONED_IV_PCT	Weighted average of Required Abandoned Interaction Volume % (weighted on Forecast Interaction Volume)
41	FRC_CALC_MAX_OCCUPANCY_PCT	Weighted average of (Achieved)

ID	Code	Description
		Calculated Maximum Occupancy % (weighted on Forecast Interaction Volume)
77	FRC_REQ_MAX_OCCUPANCY_PCT	Weighted average of Required Maximum Occupancy % (weighted on Forecast Interaction Volume)
10	FRC_IV	Total of Forecast Interaction Volume
50	FRC_CALC_FTE	Total of Calculated Full-time Equivalent
51	FRC_REQ_FTE	Total of Required Full-time Equivalent
56	FRC_CALC_MAN_HOURS	Total of Calculated Man Hours
57	FRC_REQ_MAN_HOURS	Total of Required Man Hours
21	SCH_COVERAGE	Total of Scheduled Coverage
24	SCH_SERVICE_PCT	Weighted average of Scheduled Service Level % (weighted on Forecast Interaction Volume)
19	SCH_ASA	Weighted average of Scheduled Average Speed of Answer (weighted on Forecast Interaction Volume)
22	SCH_ABANDONED_IV_PCT	Weighted average of Scheduled Abandoned Interaction Volume % (weighted on Forecast Interaction Volume)
23	SCH_MAX_OCCUPANCY_PCT	Weighted average of Scheduled Maximum Occupancy % (weighted on Forecast Interaction Volume)
49	SCH_FTE	Total of Scheduled Full-time Equivalent
55	SCH_MAN_HOURS	Total of Scheduled Man Hours
6	ACT_STAFFING	Total of Actual Staffing Difference (between Optimal number of agents for Actual workload and Scheduled Coverage)
59	ACT_COVERAGE	Total of Actual Coverage (agent minutes divided by timestep)
3	ACT_SERVICE_PCT	Weighted average of Actual Service Level % (weighted on Actual Distributed Interaction Volume) for Activity of type Deferred
5	ACT_ASA	Weighted average of Actual Average Speed of Answer

ID	Code	Description
		(weighted on Actual Interaction Volume)
4	ACT_ABANDONED_IV_PCT	Total of Actual Abandoned Interaction Volume %
1	ACT_IV	Total of Actual Interaction Volume
62	ACT_ABANDONED_IV	Total of Actual Abandoned Interaction Volume %
8	ACT_DISTRIBUTED_IV	Total of Actual Distributed Interaction Volume
9	ACT_HANDLED_IV	Total of Actual Handled Interaction Volume
60	ACT_FTE	Total of Actual Full-time Equivalent
61	ACT_MAN_HOURS	Total of Actual Man Hours
20	SCH_HEADCOUNT	Total of Scheduled Headcount
2	ACT_AHT	Weighted average of Actual Handle Time (weighted on Actual Handled Interaction Volume)
78	ACT_SIMPLE_AHT	Simple average of Actual Handle Time
11	FRC_AHT	Weighted average of Forecast Handle Time (weighted on Forecast Interaction Volume)
58	FRC_SIMPLE_AHT	Simple average of Forecast Average Handle Time
70	SCH_AHT	Weighted average of Scheduled Average Handle Time (weighted on Forecast Interaction Volume)
69	SCH_IV	Total of Scheduled Interaction Volume

Fact tables

Contents

Use these tables to define aggregated agent statistical data the WFM ETL database schema.

Related documentation:

-

This topic describes the Fact tables in the Workforce Management (WFM) ETL Database schema. To view the details in each table click the table name in the first column. For example, WFM_ADH_AGENT_DAY or WFM_ADH_AGENT_TIMESTEP.

For a description of the abbreviations used in these tables, see Abbreviation of database terms.

WFM_ADH_AGENT_DAY

This table contains a 24-hour day aggregate of agent adherence information.

Column	Data type	P	M	F	DV
WFM_ADH_AGENT_DAY_KEY	numeric(19)	✔	✔		
WFM_AGENT_KEY	int		✔	✔	
WFM_SITE_KEY	int		✔	✔	
WFM_TEAM_KEY	int			✔	
WFM_DATE	date		✔		
WFM_NON_ADHERENCE_DURATION	int		✔		
WFM_OUT_SCH_NON_ADH_DURATION	int		✔		
WFM_SCHEDULE_DURATION	int		✔		
WFM_ACTUAL_WORK_DURATION	int		✔		
WFM_ADHERENCE_PERC	float		✔		
WFM_CONFORMANCE_PERC	float		✔		
WFM_TIMESTAMP	numeric(19)		✔		
DATE_TIME_DAY_KEY	int		✔	✔	
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_ADH_AGENT_DAY_KEY**—The primary key for this table.

- **WFM_AGENT_KEY**—The surrogate key used to join the WFM_AGENT dimension to the fact tables. It specifies the Agent of the Agent Adherence Day.
- **WFM_SITE_KEY**—The surrogate key used to join the WFM_SITE dimension to the fact tables. It specifies the Site of the Agent Adherence Day.
- **WFM_TEAM_KEY**—The surrogate key used to join the WFM_TEAM dimension to the fact tables. It specifies the historical Team of the Agent at the time of adherence date specified in WM_DATE column. It is NULL if the Agent was not under any team at that time.
- **WFM_DATE**—The date of the Agent Adherence Day in the Agent's Site time zone.
- **WFM_NON_ADHERENCE_DURATION**—The Agent’s total non-adherence time in seconds for the day.
- **WFM_OUT_SCH_NON_ADH_DURATION**—The Agent’s total out of schedule non-adherence time in seconds for the day.
- **WFM_SCHEDULE_DURATION**—The Agent’s total schedule time plus Agent’s total out of schedule non-adherence time for the day in seconds.
- **WFM_ACTUAL_WORK_DURATION**—The Agent’s total work time (in seconds) for the day. The work time is counted as the time in any real-time state that is not mapped to the Schedule State Group containing the "No Activity" fixed state.
- **WFM_ADHERENCE_PERC**—The Agent’s adherence percentage for the day. The adherence percentage is calculated using the following formula:
  WFM_ADHERENCE_PERC = 100.0 - (100.0 * WFM_NON_ADHERENCE_DURATION) WFM_SCHEDULE_DURATION
- **WFM_CONFORMANCE_PERC**—The Agent’s conformance percentage for the day, calculated by using the following formula:
  WFM_CONFORMANCE_PERC = (100.0 * WFM_ACTUAL_WORK_DURATION) WFM_SCHEDULE_DURATION
- **WFM_TIMESTAMP**—An internal timestamp value.
- **DATE_TIME_DAY_KEY**—Identifies the start of a day interval in which the fact began and is equal to the UTC-equivalent time value, at which the day interval started. The value is the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time). Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert day interval start to an appropriate time zone.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the Fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_ADH_AGENT_TIMESTEP

This table contains a 24-hour day aggregate of agent adherence information.

Column	Data type	P	M	F	DV
WFM_ADH_AGENT_DAY_KEY	numeric(19)	✔	✔	✔	

WFM_TIME_STEP	datetime	✔	✔		
WFM_NON_ADHERENCE_DURATION	int		✔		
WFM_OUT_SCH_NON_ADH_DURATION	int		✔		
WFM_SCHEDULE_DURATION	int		✔		
WFM_ACTUAL_WORK_DURATION	int		✔		
DATE_TIME_KEY	int		✔	✔	
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_ADH_AGENT_DAY_KEY**—The surrogate key used to join parent WFM_ADH_AGENT_DAY record containing the Agent, Site and Team, as well as the corresponding calendar day information.
- **WFM_TIME_STEP**—The start date/time of 15-minute interval in the Agent's Site time zone.
- **WFM_NON_ADHERENCE_DURATION**—The Agent's total non-adherence time in seconds for the 15-minute interval.
- **WFM_OUT_SCH_NON_ADH_DURATION**—The Agent's total out of schedule non-adherence time in seconds for the 15-minute interval.
- **WFM_SCHEDULE_DURATION**—The Agent's total schedule time in seconds for the 15-minute interval.
- **WFM_ACTUAL_WORK_DURATION**—The Agent’s total work time (in seconds) for the day. The work time is counted as the time in any real-time state that is not mapped to the Schedule State Group containing the "No Activity" fixed state.
- **DATE_TIME_KEY**—Identifies the start of a 15-minute interval, in which the fact began and is equal to the UTC-equivalent time, at which the interval started. The value is the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time). Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert interval start to an appropriate time zone.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the Fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_SCH_AGENT_DAY

This table contains the Agent’s shift-day schedule information. The shift-day starts anywhere within the corresponding 24-hour calendar day, but it can end on the next calendar day, if the scheduled shift is an overnight shift.

Column	Data type	P	M	F	DV
WFM_SCH_AGENT_DAY_KEY	numeric(19)	✔	✔		
WFM_AGENT_KEY	int		✔	✔	
WFM_SITE_KEY	int		✔	✔	
WFM_TEAM_KEY	int			✔	
WFM_DATE	date		✔		
WFM_DAY_START	datetime		✔		
WFM_DAY_END	datetime		✔		
WFM_STATE_KEY	numeric(19)		✔	✔	
WFM_FULL_DAY	int		✔		
WFM_SCHEDULE_DURATION	float		✔		
WFM_WORK_DURATION	float		✔		
WFM_PAID_DURATION	float		✔		
WFM_OVERTIME_DURATION	float		✔		
WFM_TIMESTAMP	numeric(19)		✔		
START_DATE_TIME_KEY	int		✔	✔	
END_DATE_TIME_KEY	int		✔	✔	
START_TS	int		✔		
END_TS	int		✔		
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_SCH_AGENT_DAY_KEY**—The primary key for this table.
- **WFM_AGENT_KEY**—The surrogate key used to join the WFM_AGENT dimension to the fact tables. It specifies the Agent of the schedule day.
- **WFM_SITE_KEY**—The surrogate key used to join the WFM_SITE dimension to the fact tables. It specifies the Site of the schedule day.
- **WFM_TEAM_KEY**—The surrogate key used to join the WFM_TEAM dimension to the fact tables. It specifies the historical Team of the Agent at the time of schedule date specified in WM_DATE column. It is NULL if the Agent was not in any team at that time.
- **WFM_DATE**—The date of Agent Adherence Day in the Agent's Site time zone.
- **WFM_DAY_START**—The start date/time of schedule day in the Agent's Site time zone. It is a start time of the first (the earliest) schedule state within the schedule day.

- **WFM_DAY_END**—The end date/time of schedule day in the Agent's Site time zone. It is a end time of the last (the latest) schedule state within the schedule day.
- **WFM_STATE_KEY**—The surrogate key used to join the WFM_STATE dimension to the Fact tables. It specifies the full-day schedule state corresponding to the schedule day.
- **WFM_FULL_DAY**—Indicates whether the schedule is full-day or not: 0 = No, 1 = Yes. The full-day schedule day is one that has no specific start/end time defined (for example, Day-Off).
- **WFM_SCHEDULE_DURATION**—The total schedule time, in minutes, for the schedule day.
- **WFM_WORK_DURATION**—The total scheduled work on activities time, in minutes, for the schedule day.
- **WFM_PAID_DURATION**—The total scheduled paid time, in minutes, for the schedule day.
- **WFM_OVERTIME_DURATION**—The total scheduled overtime, in minutes, for the schedule day.
- **WFM_TIMESTAMP**—An internal timestamp value.
- **START_DATE_TIME_KEY**—Identifies the start of a 15-minute interval, in which the fact began. Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert the START_TS timestamp to an appropriate time zone.
- **END_DATE_TIME_KEY**—Identifies the start of a 15-minute interval, in which the fact ended. Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert the END_TS timestamp to an appropriate time zone.
- **START_TS**—The date and time, at which the fact began, as a Coordinated Universal Time (UTC) value—the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time).
- **END_TS**—The date and time, at which the fact ended, as a Coordinated Universal Time (UTC) value—the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time).
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the Fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_SCH_AGENT_TIMESTEP

This table contains a 15-minute interval aggregate of agent’s schedule information.

Column	Data type	P	M	F	DV
WFM_SCH_AGENT_DAY_KEY	numeric(19)	✔	✔	✔	
WFM_TIME_STEP	datetime	✔	✔		
WFM_SCHEDULE_DURATION	float		✔		
WFM_WORK_DURATION	float		✔		

WFM_PAID_DURATION	float		✔		
WFM_OVERTIME_DURATION	float		✔		
DATE_TIME_KEY	int		✔	✔	
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_SCH_AGENT_DAY_KEY**—The surrogate key used to join parent WFM_SCH_AGENT_DAY record containing the Agent, Site and Team, as well as corresponding schedule day information.
- **WFM_TIME_STEP**—The start date/time of the 15-minute interval in the Agent's Site time zone.
- **WFM_SCHEDULE_DURATION**—The total schedule time, in minutes, for the 15-minute interval.
- **WFM_WORK_DURATION**—The total scheduled work on activities time, in minutes, for the 15-minute interval.
- **WFM_PAID_DURATION**—The total scheduled paid time, in minutes, for the 15-minute interval.
- **WFM_OVERTIME_DURATION**—The total scheduled overtime, in minutes, for the 15-minute interval.
- **DATE_TIME_KEY**—Identifies the start of a 15-minute interval, in which the fact began and is equal to the UTC-equivalent time, at which the interval started. The value is the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time). Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert interval start to an appropriate time zone.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the Fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_SCH_AGENT_STATE

This table contains agent’s schedule state information.

Column	Data type	P	M	F	DV
WFM_SCH_AGENT_DAY_KEY	numeric(19)	✔	✔	✔	
WFM_STATE_KEY	numeric(19)	✔	✔	✔	
WFM_STATE_START	datetime	✔	✔		

WFM_STATE_END	datetime		✔		
WFM_STATE_DURATION	float		✔		
WFM_PAID_DURATION	float		✔		
WFM_FULL_DAY	int		✔		
START_DATE_TIME_KEY	int		✔	✔	
END_DATE_TIME_KEY	int		✔	✔	
START_TS	int		✔		
END_TS	int		✔		
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_SCH_AGENT_DAY_KEY**—The surrogate key used to join the parent WFM_SCH_AGENT_DAY record containing the Agent, Site and Team, as well as corresponding schedule day information.
- **WFM_STATE_KEY**—The surrogate key used to join the WFM_STATE dimension to the Fact tables. It specifies the schedule state of the agent schedule state.
- **WFM_STATE_START**—The start date/time of the Agent schedule state in the Agent's Site time zone.
- **WFM_STATE_END**—The end date/time of the Agent schedule state in the Agent's Site time zone.
- **WFM_FULL_DAY**—Indicates whether the Agent schedule state is full-day or not: 0 = No, 1 = Yes. The full-day schedule state is one that has no specific start/end time defined (for example, Day-Off).
- **WFM_STATE_DURATION**—The schedule state duration in minutes.
- **WFM_PAID_DURATION**—The schedule state paid duration in minutes.
- **START_DATE_TIME_KEY**—Identifies the start of a 15-minute interval, in which the fact began. Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert the START_TS timestamp to an appropriate time zone.
- **END_DATE_TIME_KEY**—Identifies the start of a 15-minute interval, in which the fact ended. Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert the END_TS timestamp to an appropriate time zone.
- **START_TS**—The date and time, at which the fact began, as a Coordinated Universal Time (UTC) value—the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time).
- **END_TS**—The date and time, at which the fact ended, as a Coordinated Universal Time (UTC) value—the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time).
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the Fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).

- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_SCH_AGENT_STATE_TIMESTEP

This table contains a 15-minute interval aggregate of schedule state duration information.

Column	Data type	P	M	F	DV
WFM_SCH_AGENT_DAY_KEY	numeric(19)	✔	✔	✔	
WFM_STATE_KEY	numeric(19)	✔	✔	✔	
WFM_TIME_STEP	datetime	✔	✔		
WFM_STATE_DURATION	float		✔		
DATE_TIME_KEY	int		✔	✔	
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_SCH_AGENT_DAY_KEY**—The surrogate key used to join the parent WFM_SCH_AGENT_DAY record containing Agent, Site and Team, as well as corresponding schedule day information.
- **WFM_STATE_KEY**—The surrogate key used to join the WFM_STATE dimension to the Fact tables. It specifies the schedule state of the 15-minute interval aggregate.
- **WFM_TIME_STEP**—The start date/time of the 15-minute interval in the Agent's Site time zone.
- **WFM_STATE_DURATION**—The total schedule state time in minutes for the 15-minute interval.
- **DATE_TIME_KEY**—Identifies the start of a 15-minute interval, in which the fact began and is equal to the UTC-equivalent time, at which the interval started. The value is the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time). Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert interval start to an appropriate time zone.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the Fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_PERF_ITEM_DAY

This table contains a 24-hour calendar day aggregate of the activity and/or the site performance statistics.

Column	Data type	P	M	F	DV
WFM_PERF_ITEM_DAY_KEY	numeric(19)	✔	✔		
WFM_ACTIVITY_KEY	int			✔	
WFM_SITE_KEY	int			✔	
WFM_DATE	date		✔		
WFM_PERF_ITEM_KEY	int		✔	✔	
WFM_PERF_ITEM_VALUE	float				
WFM_TIMESTAMP	numeric(19)		✔		
DATE_TIME_DAY_KEY	int		✔	✔	
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_PERF_ITEM_DAY_KEY**—The primary key for this table
- **WFM_ACTIVITY_KEY**—The surrogate key used to join the WFM_ACTIVITY dimension to the Fact tables. It specifies the Activity (Single-Site or Multi-Site or Activity Group) of the performance statistic aggregate for the day. It is NULL for the Site statistic aggregate.
- **WFM_SITE_KEY**—The surrogate key used to join the WFM_SITE dimension to the fact tables. It specifies the Site of the performance statistic aggregate for the day. It is NULL for the Activity statistic aggregate.
- **WFM_DATE**—The date of performance statistic day aggregate in time zone of the Activity or the Site. Single-Site Activity uses the Site time zone, while Multi-Site Activity and Activity Group use the Business Unit time zone.
- **WFM_PERF_ITEM_KEY**—The surrogate key used to join the WFM_PERF_ITEM dimension to the Fact tables. It specifies the performance statistic type of the day aggregate. See the list of available statistics in the description of the WFM_PERF_ITEM dimension.
- **WFM_PERF_ITEM_VALUE**—The value of the Activity or Site performance statistic aggregate for the day.
- **WFM_TIMESTAMP**—An internal timestamp value.
- **DATE_TIME_DAY_KEY**—Identifies the start of a day interval, in which the fact began and is equal to the UTC-equivalent time value, at which the day interval started. The value is the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time). Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert day interval start to an appropriate time zone.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the Fact tables.

- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).
- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

WFM_PERF_ITEM_TIMESTEP

This table contains a 15-minute interval aggregate of the activity and/or the site performance statistics.

Column	Data type	P	M	F	DV
WFM_PERF_ITEM_DAY_KEY	numeric(19)	✔	✔	✔	
WFM_TIME_STEP	datetime	✔	✔		
WFM_PERF_ITEM_VALUE	float				
DATE_TIME_KEY	int		✔	✔	
TENANT_KEY	int			✔	
CREATE_AUDIT_KEY	numeric(19)			✔	
UPDATE_AUDIT_KEY	numeric(19)			✔	
PURGE_FLAG	int				

Description of Columns

- **WFM_PERF_ITEM_DAY_KEY**—The surrogate key used to join the parent WFM_PERF_ITEM_DAY record containing the Activity or Site Performance Statistic, as well as corresponding calendar day information.
- **WFM_TIME_STEP**—The start date/time of 15-minute interval in time zone of the Activity or Site. Single-Site Activity uses the Site time zone, while Multi-Site Activity and Activity Group use the Business Unit time zone.
- **WFM_PERF_ITEM_VALUE**—The value of the Activity or the Site performance statistic aggregate for the 15-minute interval.
- **DATE_TIME_KEY**—Identifies the start of a 15-minute interval, in which the fact began and is equal to the UTC-equivalent time, at which the interval started. The value is the number of seconds that have elapsed since midnight on January 1, 1970, not counting leap seconds (also known as UNIX time). Use this value as a key to join the Fact tables to any configured DATE_TIME dimension to group the facts that are related to the same interval and/or convert interval start to an appropriate time zone.
- **TENANT_KEY**—The surrogate key used to join the TENANT dimension to the Fact tables.
- **CREATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value is useful for aggregation, Enterprise Application Integration (EAI), and ETL tools (that is, applications that need to identify newly added data).
- **UPDATE_AUDIT_KEY**—The surrogate key used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data updates. This value is useful for aggregation, Enterprise Application

Integration (EAI), and ETL tools (that is, applications that need to identify recently modified data).

- **PURGE_FLAG**—This field is reserved (used internally by the purging script to mark records for purging).

Service and Control tables

Contents

- Administrator

Find information about the Service and Control tables used in the Workforce Management (WFM) ETL Database schema.

Related documentation:

-

Only the Service and Control tables in the Workforce Management (WFM) ETL Database schema that are relevant for customer use are described here. Other Service and Control tables (for example, WM_DB_VERSION) are internal tables and not relevant for customer use.

CTL_ETL_HISTORY

The CTL_ETL_HISTORY table in the WFM ETL Database schema parallels the CTL_ETL_HISTORY table in the Genesys Info Mart database, to indicate the status of ETL processing. A row is added to this table after each job completes.

The ETL cycle is broken down into many small tasks for different types of data and date periods, and the CTL_ETL_HISTORY table is a useful indicator of the status of ETL processing. Failure of even a single small task will result in a value of FAILED in the STATUS field for the whole ETL job. Therefore, a value of FAILED does not necessarily indicate that there is a major ETL problem requiring immediate attention. Genesys recommends that you start monitoring ETL processing more closely if the STATUS field in the table shows a persistently recurring value of FAILED.

CTL_AUDIT_LOG

The CTL_AUDIT_LOG table in the WFM ETL Database schema parallels the CTL_AUDIT_LOG table in the Genesys Info Mart database, to allow facts and dimensions to be described by data lineage attributes.

More information

For more information about the CTL_ETL_HISTORY and CTL_AUDIT_LOG tables in the Info Mart database, see the *Genesys Info Mart Physical Data Model for a Microsoft SQL Server Database*.

Query examples

Contents

- Administrator

Use these examples to construct three different categories of database queries for the WFM ETL database schema.

Related documentation:

-

Adherence queries

Agent adherence totals (daily granularity)

```
SELECT
WFM_SITE.WFM_SITE_NAME,
WFM_TEAM.WFM_TEAM_NAME,
WFM_AGENT.FIRST_NAME,
WFM_AGENT.LAST_NAME,
WFM_ADH_AGENT_DAY.WFM_DATE,
WFM_ADH_AGENT_DAY.WFM_SCHEDULE_DURATION,
WFM_ADH_AGENT_DAY.WFM_NON_ADHERENCE_DURATION,
WFM_ADH_AGENT_DAY.WFM_OUT_SCH_NON_ADH_DURATION,
WFM_ADH_AGENT_DAY.WFM_ADHERENCE_PERC
FROM WFM_ADH_AGENT_DAY
JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_ADH_AGENT_DAY.WFM_SITE_KEY)
JOIN WFM_AGENT ON (WFM_AGENT.WFM_AGENT_KEY = WFM_ADH_AGENT_DAY.WFM_AGENT_KEY)
LEFT JOIN WFM_TEAM ON (WFM_TEAM.WFM_TEAM_KEY = WFM_AGENT.WFM_TEAM_KEY)
WHERE
WFM_ADH_AGENT_DAY.WFM_DATE >= ? AND WFM_ADH_AGENT_DAY.WFM_DATE
ORDER BY
WFM_SITE.WFM_SITE_NAME,
WFM_TEAM.WFM_TEAM_NAME,
WFM_ADH_AGENT_DAY.WFM_DATE,
WFM_AGENT.FIRST_NAME,
WFM_AGENT.LAST_NAME
```

Team adherence totals (daily granularity)

```
SELECT
WFM_BU.WFM_BU_NAME,
WFM_SITE.WFM_SITE_NAME,
WFM_ADH_AGENT_DAY.WFM_DATE,
WFM_TEAM.WFM_TEAM_NAME,
SUM(WFM_ADH_AGENT_DAY.WFM_SCHEDULE_DURATION),
SUM(WFM_ADH_AGENT_DAY.WFM_NON_ADHERENCE_DURATION),
SUM(WFM_ADH_AGENT_DAY.WFM_OUT_SCH_NON_ADH_DURATION)
FROM WFM_ADH_AGENT_DAY
JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_ADH_AGENT_DAY.WFM_SITE_KEY)
JOIN WFM_BU ON (WFM_BU.WFM_BU_KEY = WFM_SITE.WFM_BU_KEY)
LEFT JOIN WFM_TEAM ON (WFM_TEAM.WFM_TEAM_KEY = WFM_ADH_AGENT_DAY.WFM_TEAM_KEY)
WHERE
WFM_ADH_AGENT_DAY.WFM_DATE >= ? AND WFM_ADH_AGENT_DAY.WFM_DATE
GROUP BY
WFM_BU.WFM_BU_NAME,
WFM_SITE.WFM_SITE_NAME,
WFM_ADH_AGENT_DAY.WFM_DATE,
WFM_TEAM.WFM_TEAM_NAME
HAVING SUM(WFM_ADH_AGENT_DAY.WFM_SCHEDULE_DURATION) > 0
ORDER BY
WFM_BU.WFM_BU_NAME,
WFM_SITE.WFM_SITE_NAME,

WFM_ADH_AGENT_DAY.WFM_DATE,

WFM_TEAM.WFM_TEAM_NAME
```

Schedule queries

Schedule states

```
SELECT

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.EMPLOYEE_ID,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_DAY.WFM_DATE,

WFM_STATE.WFM_STATE_NAME,

WFM_SSG.WFM_SSG_NAME,

WFM_SCH_AGENT_STATE.WFM_FULL_DAY,

WFM_SCH_AGENT_STATE.WFM_STATE_START,

WFM_SCH_AGENT_STATE.WFM_STATE_END,

WFM_SCH_AGENT_STATE.WFM_STATE_DURATION,

WFM_SCH_AGENT_STATE.WFM_PAID_DURATION

FROM WFM_SCH_AGENT_STATE

JOIN WFM_SCH_AGENT_DAY '''ON''' (WFM_SCH_AGENT_DAY.WFM_SCH_AGENT_DAY_KEY =
WFM_SCH_AGENT_STATE.WFM_SCH_AGENT_DAY_KEY)

JOIN WFM_AGENT ON (WFM_AGENT.AGENT_KEY = WFM_SCH_AGENT_DAY.WFM_AGENT_KEY)

JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_SCH_AGENT_DAY.WFM_SITE_KEY)

LEFT JOIN WFM_TEAM ON (WFM_TEAM.WFM_TEAM_KEY = WFM_AGENT.WFM_TEAM_KEY)

JOIN WFM_STATE ON (WFM_STATE.WFM_STATE_KEY = WFM_SCH_AGENT_STATE.WFM_STATE_KEY)

JOIN WFM_STATE_TYPE ON (WFM_STATE_TYPE.WFM_STATE_TYPE_KEY =
WFM_STATE.WFM_STATE_TYPE_KEY)
LEFT JOIN WFM_SSG ON (WFM_SSG.WFM_SSG_KEY = WFM_STATE.WFM_SSG_KEY)
WHERE
WFM_STATE_TYPE.WFM_STATE_TYPE_NAME NOT IN ('Shift', 'Activity Set', 'Marked Time') AND
WFM_SCH_AGENT_STATE.WFM_STATE_END >= ?
AND WFM_SCH_AGENT_STATE.WFM_STATE_START < ?
ORDER BY
WFM_SITE.WFM_SITE_NAME,
WFM_TEAM.WFM_TEAM_NAME,
WFM_AGENT.EMPLOYEE_ID,
WFM_SCH_AGENT_DAY.WFM_DATE,
WFM_SCH_AGENT_STATE.WFM_STATE_START,
WFM_STATE.WFM_STATE_KEY
```

Agent schedule state totals

```
SELECT
WFM_SITE.WFM_SITE_NAME,
WFM_TEAM.WFM_TEAM_NAME,
WFM_AGENT.FIRST_NAME,
WFM_AGENT.LAST_NAME,
WFM_SCH_AGENT_DAY.WFM_DATE,
SUM(WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_DURATION)
FROM WFM_SCH_AGENT_STATE_TIMESTEP
JOIN WFM_SCH_AGENT_DAY ON (WFM_SCH_AGENT_DAY.WFM_SCH_AGENT_DAY_KEY =
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_SCH_AGENT_DAY_KEY)
JOIN WFM_AGENT ON (WFM_AGENT.AGENT_KEY = WFM_SCH_AGENT_DAY.WFM_AGENT_KEY)
JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_SCH_AGENT_DAY.WFM_SITE_KEY)
LEFT JOIN WFM_TEAM ON (WFM_TEAM.WFM_TEAM_KEY = WFM_AGENT.WFM_TEAM_KEY)
JOIN WFM_STATE ON (WFM_STATE.WFM_STATE_KEY =
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_KEY)

JOIN WFM_SSG ON (WFM_SSG.WFM_SSG_KEY = WFM_STATE.WFM_SSG_KEY)

WHERE

WFM_SCH_AGENT_DAY.WFM_DATE >= ? AND WFM_SCH_AGENT_DAY.WFM_DATE <= ?

GROUP BY

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_DAY.WFM_DATE

ORDER BY

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_DAY.WFM_DATE
```

Team schedule state totals

```
SELECT

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_SCH_AGENT_DAY.WFM_DATE,

SUM(WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_DURATION)

FROM WFM_SCH_AGENT_STATE_TIMESTEP

JOIN WFM_SCH_AGENT_DAY ON (WFM_SCH_AGENT_DAY.WFM_SCH_AGENT_DAY_KEY =
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_SCH_AGENT_DAY_KEY)

JOIN WFM_AGENT ON (WFM_AGENT.AGENT_KEY = WFM_SCH_AGENT_DAY.WFM_AGENT_KEY)

JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_SCH_AGENT_DAY.WFM_SITE_KEY)
JOIN WFM_BU ON (WFM_BU.WFM_BU_KEY = WFM_SITE.WFM_BU_KEY)
LEFT JOIN WFM_TEAM ON (WFM_TEAM.WFM_TEAM_KEY = WFM_SCH_AGENT_DAY.WFM_TEAM_KEY)
JOIN WFM_STATE ON (WFM_STATE.WFM_STATE_KEY =
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_KEY)
JOIN WFM_SSG ON (WFM_SSG.WFM_SSG_KEY = WFM_STATE.WFM_SSG_KEY)
WHERE
WFM_SCH_AGENT_DAY.WFM_DATE >= ? AND WFM_SCH_AGENT_DAY.WFM_DATE <= ?
GROUP BY
WFM_BU.WFM_BU_NAME,
WFM_SITE.WFM_SITE_NAME,
WFM_TEAM.WFM_TEAM_NAME,
WFM_SCH_AGENT_DAY.WFM_DATE
ORDER BY
WFM_BU.WFM_BU_NAME,
WFM_SITE.WFM_SITE_NAME,
WFM_TEAM.WFM_TEAM_NAME,
WFM_SCH_AGENT_DAY.WFM_DATE
```

Schedule marked time report

```
SELECT
WFM_SITE.WFM_SITE_NAME,
WFM_TEAM.WFM_TEAM_NAME,
WFM_AGENT.FIRST_NAME,
WFM_AGENT.LAST_NAME,
WFM_SCH_AGENT_DAY.WFM_DATE,
WFM_STATE.WFM_STATE_NAME,
WFM_SCH_AGENT_STATE.WFM_STATE_START,
WFM_SCH_AGENT_STATE.WFM_STATE_END,
WFM_SCH_AGENT_STATE.WFM_STATE_DURATION,
WFM_SCH_AGENT_STATE.WFM_PAID_DURATION

FROM WFM_SCH_AGENT_STATE

JOIN WFM_SCH_AGENT_DAY '''ON''' (WFM_SCH_AGENT_DAY.WFM_SCH_AGENT_DAY_KEY =
WFM_SCH_AGENT_STATE.WFM_SCH_AGENT_DAY_KEY)

JOIN WFM_AGENT ON (WFM_AGENT.AGENT_KEY = WFM_SCH_AGENT_DAY.WFM_AGENT_KEY)

JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_SCH_AGENT_DAY.WFM_SITE_KEY)

LEFT JOIN WFM_TEAM ON (WFM_TEAM.WFM_TEAM_KEY = WFM_AGENT.WFM_TEAM_KEY)

JOIN WFM_STATE ON (WFM_STATE.WFM_STATE_KEY = WFM_SCH_AGENT_STATE.WFM_STATE_KEY)

JOIN WFM_STATE_TYPE ON (WFM_STATE_TYPE.WFM_STATE_TYPE_KEY =
WFM_STATE.WFM_STATE_TYPE_KEY)

WHERE

WFM_STATE_TYPE.WFM_STATE_TYPE_NAME IN ('Marked Time') AND

WFM_SCH_AGENT_STATE.WFM_STATE_END >= ? AND WFM_SCH_AGENT_STATE.WFM_STATE_START
< ?

ORDER BY

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_DAY.WFM_DATE,

WFM_SCH_AGENT_STATE.WFM_STATE_START,

WFM_STATE.WFM_STATE_KEY
```

Schedule marked time totals (daily granularity)

```
SELECT

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_DAY.WFM_DATE,
SUM(WFM_SCH_AGENT_STATE.WFM_STATE_DURATION),

SUM(WFM_SCH_AGENT_STATE.WFM_PAID_DURATION)

FROM WFM_SCH_AGENT_STATE

JOIN WFM_SCH_AGENT_DAY ON (WFM_SCH_AGENT_DAY.WFM_SCH_AGENT_DAY_KEY = WFM_SCH_AGENT_STATE.WFM_SCH_AGENT_DAY_KEY)

JOIN WFM_AGENT ON (WFM_AGENT.AGENT_KEY = WFM_SCH_AGENT_DAY.WFM_AGENT_KEY)

JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_SCH_AGENT_DAY.WFM_SITE_KEY)

LEFT JOIN WFM_TEAM ON (WFM_TEAM.WFM_TEAM_KEY = WFM_AGENT.WFM_TEAM_KEY)

JOIN WFM_STATE ON (WFM_STATE.WFM_STATE_KEY = WFM_SCH_AGENT_STATE.WFM_STATE_KEY)

JOIN WFM_STATE_TYPE ON (WFM_STATE_TYPE.WFM_STATE_TYPE_KEY = WFM_STATE.WFM_STATE_TYPE_KEY)

WHERE

WFM_STATE_TYPE.WFM_STATE_TYPE_NAME IN ('Marked Time') AND

WFM_SCH_AGENT_STATE.WFM_STATE_END >= ? AND WFM_SCH_AGENT_STATE.WFM_STATE_START < ?

GROUP BY

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_DAY.WFM_DATE

ORDER BY

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_DAY.WFM_DATE
```

Schedule marked time totals (timestep granularity)

```
SELECT
```

```
WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP,

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_DURATION

FROM WFM_SCH_AGENT_STATE_TIMESTEP

JOIN WFM_SCH_AGENT_DAY ON (WFM_SCH_AGENT_DAY.WFM_SCH_AGENT_DAY_KEY =
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_SCH_AGENT_DAY_KEY)

JOIN WFM_AGENT ON (WFM_AGENT.AGENT_KEY = WFM_SCH_AGENT_DAY.WFM_AGENT_KEY)

JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_SCH_AGENT_DAY.WFM_SITE_KEY)

LEFT JOIN WFM_TEAM ON (WFM_TEAM.WFM_TEAM_KEY = WFM_AGENT.WFM_TEAM_KEY)

JOIN WFM_STATE ON (WFM_STATE.WFM_STATE_KEY =
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_KEY)

JOIN WFM_STATE_TYPE ON (WFM_STATE_TYPE.WFM_STATE_TYPE_KEY =
WFM_STATE.WFM_STATE_TYPE_KEY)

WHERE

WFM_STATE_TYPE.WFM_STATE_TYPE_NAME IN ('Marked Time') AND

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP >= ? AND
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP < ?

ORDER BY

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP
```

Weekly schedule report

```
SELECT

WFM_SITE.WFM_SITE_NAME,
WFM_TEAM.WFM_TEAM_NAME,
WFM_AGENT.EMPLOYEE_ID,
WFM_AGENT.FIRST_NAME,
WFM_AGENT.LAST_NAME,
WFM_SCH_AGENT_DAY.WFM_DATE,
WFM_STATE.WFM_STATE_NAME,
WFM_SCH_AGENT_DAY.WFM_FULL_DAY,
WFM_SCH_AGENT_DAY.WFM_DAY_START,
WFM_SCH_AGENT_DAY.WFM_DAY_END,
SUM(WFM_SCH_AGENT_DAY.WFM_SCHEDULE_DURATION) AS SCHEDULE_DURATION,
SUM(WFM_SCH_AGENT_DAY.WFM_PAID_DURATION) AS PAID_DURATION,
SUM(WFM_SCH_AGENT_DAY.WFM_WORK_DURATION) AS WORK_DURATION,
SUM(WFM_SCH_AGENT_DAY.WFM_OVERTIME_DURATION) AS OVERTIME_DURATION
FROM WFM_SCH_AGENT_DAY
JOIN WFM_STATE ON (WFM_STATE.WFM_STATE_KEY = WFM_SCH_AGENT_DAY.WFM_STATE_KEY)
JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_SCH_AGENT_DAY.WFM_SITE_KEY)
JOIN WFM_AGENT ON (WFM_AGENT.WFM_AGENT_KEY = WFM_SCH_AGENT_DAY.WFM_AGENT_KEY)
LEFT JOIN WFM_TEAM ON (WFM_TEAM.WFM_TEAM_KEY = WFM_AGENT.WFM_TEAM_KEY)
WHERE
WFM_SCH_AGENT_DAY.WFM_DATE >= ? AND WFM_SCH_AGENT_DAY.WFM_DATE <= ?
GROUP BY
WFM_SITE.WFM_SITE_NAME,
WFM_TEAM.WFM_TEAM_NAME,
WFM_AGENT.EMPLOYEE_ID,
WFM_AGENT.FIRST_NAME,
WFM_AGENT.LAST_NAME,
WFM_SCH_AGENT_DAY.WFM_DATE,
WFM_STATE.WFM_STATE_NAME,
WFM_SCH_AGENT_DAY.WFM_DAY_START,

WFM_SCH_AGENT_DAY.WFM_DAY_END,

WFM_SCH_AGENT_DAY.WFM_FULL_DAY

ORDER BY

WFM_SITE.WFM_SITE_NAME,

WFM_TEAM.WFM_TEAM_NAME,

WFM_AGENT.EMPLOYEE_ID,

WFM_AGENT.FIRST_NAME,

WFM_AGENT.LAST_NAME,

WFM_SCH_AGENT_DAY.WFM_DATE
```

Schedule state group totals

```
SELECT

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP,

WFM_SSG.WFM_SSG_NAME,

SUM(WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_DURATION) / 15 AS WFM_SSG_TOTAL,

WFM_SSG.WFM_SSG_WEIGHT

FROM WFM_SCH_AGENT_STATE_TIMESTEP

JOIN WFM_SCH_AGENT_DAY v (WFM_SCH_AGENT_DAY.WFM_SCH_AGENT_DAY_KEY = WFM_SCH_AGENT_STATE_TIMESTEP.WFM_SCH_AGENT_DAY_KEY)

JOIN WFM_AGENT ON (WFM_AGENT.AGENT_KEY = WFM_SCH_AGENT_DAY.WFM_AGENT_KEY)

JOIN WFM_SITE ON (WM_SITE.WFM_SITE_KEY = WFM_SCH_AGENT_DAY.WFM_SITE_KEY)

JOIN WFM_BU ON (WFM_BU.WFM_BU_KEY = WFM_SITE.WFM_BU_KEY)

JOIN WFM_STATE ON (WFM_STATE.WFM_STATE_KEY = WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_KEY)

JOIN WFM_SSG ON (WFM_SSG.WFM_SSG_KEY = WFM_STATE.WFM_SSG_KEY)

WHERE
```

```
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP >= '11/14/2013' AND
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP < '11/15/2013'

AND WFM_SITE.WFM_SITE_NAME = 'Sched Pot 4'

GROUP BY

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP,

WFM_SSG.WFM_SSG_NAME,

WFM_SSG.WFM_SSG_WEIGHT

ORDER BY

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP,

WFM_SSG.WFM_SSG_WEIGHT
```

Activity schedule coverage

```
SELECT

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP,

SUM(WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_DURATION) / 15 AS WFM_ACTIVITY_COVERAGE

FROM WFM_SCH_AGENT_STATE_TIMESTEP

JOIN WFM_SCH_AGENT_DAY '''ON''' (WFM_SCH_AGENT_DAY.WFM_SCH_AGENT_DAY_KEY =
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_SCH_AGENT_DAY_KEY)

JOIN WFM_AGENT ON (WFM_AGENT.AGENT_KEY = WFM_SCH_AGENT_DAY.WFM_AGENT_KEY)

JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_SCH_AGENT_DAY.WFM_SITE_KEY)

JOIN WFM_BU ON (WFM_BU.WFM_BU_KEY = WFM_SITE.WFM_BU_KEY)

JOIN WFM_STATE ON (WFM_STATE.WFM_STATE_KEY =
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_STATE_KEY)

JOIN WFM_STATE_TYPE ON (WFM_STATE_TYPE.WFM_STATE_TYPE_KEY =
WFM_STATE.WFM_STATE_TYPE_KEY)
```

```
JOIN WFM_ACTIVITY ON (WFM_ACTIVITY.WFM_ACTIVITY_KEY = WFM_STATE.WFM_STATE_ID AND
WFM_STATE_TYPE.WFM_STATE_TYPE_NAME = 'Activity')

WHERE

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP >= '11/14/2013' AND
WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP < '11/15/2013'

AND WFM_SITE.WFM_SITE_NAME = 'Sched Pot 4'

AND WFM_ACTIVITY.WFM_ACTIVITY_NAME = 'Broadband Priority Care'

GROUP BY

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP

ORDER BY WFM_SCH_AGENT_STATE_TIMESTEP.WFM_TIME_STEP, WFM_ACTIVITY.WFM_ACTIVITY_NAME
```

Performance statistics queries

Schedule daily summary for activity

```
SELECT

WFM_SITE.WFM_SITE_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,

SUM(WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_VALUE)

FROM WFM_PERF_ITEM_DAY

JOIN WFM_ACTIVITY ON (WFM_ACTIVITY.WFM_ACTIVITY_KEY =
WFM_PERF_ITEM_DAY.WFM_ACTIVITY_KEY)

JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_ACTIVITY.WFM_SITE_KEY)

JOIN WFM_PERF_ITEM ON (WFM_PERF_ITEM.WFM_PERF_ITEM_KEY =
WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_KEY)

WHERE
```

```
WFM_PERF_ITEM_DAY.WFM_DATE >= ? AND WFM_PERF_ITEM_DAY.WFM_DATE <= ? AND
WFM_PERF_ITEM.WFM_PERF_ITEM_CODE IN ('SCH_HEADCOUNT', 'SCH_SERVICE_PCT',
'FRC_CALC_SERVICE_PCT', 'SCH_IV', 'FRC_IV', 'SCH_AHT', 'FRC_AHT', 'FRC_CALC_MAN_HOURS',
'FRC_REQ_MAN_HOURS', 'SCH_MAN_HOURS', 'SCH_ASA', 'FRC_CALC_ASA',
'SCH_MAX_OCCUPANCY_PCT', 'FRC_CALC_MAX_OCCUPANCY_PCT')

GROUP BY

WFM_SITE.WFM_SITE_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

ORDER BY

WFM_SITE.WFM_SITE_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE
```

Schedule daily summary for multi-site activity

```
SELECT

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,

SUM(WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_VALUE)

FROM WFM_PERF_ITEM_DAY

JOIN WFM_ACTIVITY ON (WFM_ACTIVITY.WFM_ACTIVITY_KEY =
WFM_PERF_ITEM_DAY.WFM_ACTIVITY_KEY AND WFM_ACTIVITY.WFM_SITE_KEY IS NULL AND
WFM_ACTIVITY.WFM_ACTIVITY_TYPE_KEY 10)

JOIN WFM_BU ON (WFM_BU.WFM_BU_KEY = WFM_ACTIVITY.WFM_BU_KEY)

JOIN WFM_PERF_ITEM ON (WFM_PERF_ITEM.WFM_PERF_ITEM_KEY =
WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_KEY)

WHERE

WFM_PERF_ITEM_DAY.WFM_DATE >= ? AND WFM_PERF_ITEM_DAY.WFM_DATE <= ? AND
```

```
WFM_PERF_ITEM.WFM_PERF_ITEM_CODE IN ('SCH_HEADCOUNT', 'SCH_SERVICE_PCT',
'FRC_CALC_SERVICE_PCT', 'SCH_IV', 'FRC_IV', 'SCH_AHT', 'FRC_AHT', 'FRC_CALC_FTE',
'FRC_REQ_FTE', 'SCH_FTE', 'SCH_ASA', 'FRC_CALC_ASA', 'SCH_MAX_OCCUPANCY_PCT',
'FRC_CALC_MAX_OCCUPANCY_PCT')

GROUP BY

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

ORDER BY

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE
```

Schedule daily summary for activity group

```
SELECT

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,

SUM(WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_VALUE)

FROM WFM_PERF_ITEM_DAY

JOIN WFM_ACTIVITY ON (WFM_ACTIVITY.WFM_ACTIVITY_KEY =
WFM_PERF_ITEM_DAY.WFM_ACTIVITY_KEY AND WFM_ACTIVITY.WFM_SITE_KEY IS NULL AND
WFM_ACTIVITY.WFM_ACTIVITY_TYPE_KEY = 10)

JOIN WFM_BU ON (WFM_BU.WFM_BU_KEY = WFM_ACTIVITY.WFM_BU_KEY)

JOIN WFM_PERF_ITEM ON (WFM_PERF_ITEM.WFM_PERF_ITEM_KEY =
WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_KEY)

WHERE

WFM_PERF_ITEM_DAY.WFM_DATE >= ? AND WFM_PERF_ITEM_DAY.WFM_DATE <= ? AND
```

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE IN ('SCH_HEADCOUNT', 'SCH_SERVICE_PCT', 'FRC_CALC_SERVICE_PCT', 'SCH_IV', 'FRC_IV', 'SCH_AHT', 'FRC_AHT', 'FRC_CALC_MAN_HOURS', 'FRC_REQ_MAN_HOURS', 'SCH_MAN_HOURS', 'SCH_ASA', 'FRC_CALC_ASA', 'SCH_MAX_OCCUPANCY_PCT', 'FRC_CALC_MAX_OCCUPANCY_PCT')

GROUP BY

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

ORDER BY

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

Schedule daily summary for site

SELECT

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,

SUM(WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_VALUE)

FROM WFM_PERF_ITEM_DAY

JOIN WFM_SITE **ON** (WFM_SITE.WFM_SITE_KEY = WFM_PERF_ITEM_DAY.WFM_SITE_KEY)

JOIN WFM_BU **ON** (WFM_BU.WFM_BU_KEY = WFM_SITE.WFM_BU_KEY)

JOIN WFM_PERF_ITEM **ON** (WFM_PERF_ITEM.WFM_PERF_ITEM_KEY = WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_KEY)

WHERE

WFM_PERF_ITEM_DAY.WFM_DATE >= ? AND WFM_PERF_ITEM_DAY.WFM_DATE <= ? AND

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE IN ('SCH_HEADCOUNT', 'SCH_SERVICE_PCT', 'FRC_CALC_SERVICE_PCT', 'SCH_IV', 'FRC_IV', 'SCH_AHT', 'FRC_AHT', 'FRC_CALC_FTE', 'FRC_REQ_FTE',

```
'SCH_FTE', 'SCH_ASA', 'FRC_CALC_ASA', 'SCH_MAX_OCCUPANCY_PCT',
'FRC_CALC_MAX_OCCUPANCY_PCT')

GROUP BY

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

ORDER BY

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_PERF_ITEM_DAY.WFM_DATE,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE
```

Schedule intra-day summary for activity

```
SELECT

WFM_SITE.WFM_SITE_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,

SUM(WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_VALUE)

FROM WFM_PERF_ITEM_TIMESTEP

JOIN WFM_PERF_ITEM_DAY ON (WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_DAY_KEY =
WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_DAY_KEY)

JOIN WFM_ACTIVITY ON (WFM_ACTIVITY.WFM_ACTIVITY_KEY =
WFM_PERF_ITEM_DAY.WFM_ACTIVITY_KEY)

JOIN WFM_SITE ON (WFM_SITE.WFM_SITE_KEY = WFM_ACTIVITY.WFM_SITE_KEY)

JOIN WFM_PERF_ITEM ON (WFM_PERF_ITEM.WFM_PERF_ITEM_KEY =
WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_KEY)

WHERE

WFM_PERF_ITEM_DAY.WFM_DATE = ? AND
```

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE IN ('SCH_HEADCOUNT', 'SCH_SERVICE_PCT', 'FRC_CALC_SERVICE_PCT', 'SCH_IV', 'FRC_IV', 'SCH_AHT', 'FRC_AHT', 'FRC_CALC_STAFFING', 'FRC_REQ_STAFFING', 'SCH_COVERAGE', 'SCH_ASA', 'FRC_CALC_ASA', 'SCH_MAX_OCCUPANCY_PCT', 'FRC_CALC_MAX_OCCUPANCY_PCT')

GROUP BY

WFM_SITE.WFM_SITE_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

ORDER BY

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_SITE.WFM_SITE_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

Schedule intra-day summary for multi-site activity

SELECT

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,

SUM(WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_VALUE)

FROM WFM_PERF_ITEM_TIMESTEP

JOIN WFM_PERF_ITEM_DAY **ON** (WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_DAY_KEY = WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_DAY_KEY)

JOIN WFM_ACTIVITY **ON** (WFM_ACTIVITY.WFM_ACTIVITY_KEY = WFM_PERF_ITEM_DAY.WFM_ACTIVITY_KEY AND WFM_ACTIVITY.WFM_SITE_KEY IS NULL AND WFM_ACTIVITY.WFM_ACTIVITY_TYPE_KEY 10)

JOIN WFM_BU **ON** (WFM_BU.WFM_BU_KEY = WFM_ACTIVITY.WFM_BU_KEY)

JOIN WFM_PERF_ITEM **ON** (WFM_PERF_ITEM.WFM_PERF_ITEM_KEY = WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_KEY)

WHERE

```
WFM_PERF_ITEM_DAY.WFM_DATE = ? AND

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE IN ('SCH_HEADCOUNT', 'SCH_SERVICE_PCT',
'FRC_CALC_SERVICE_PCT', 'SCH_IV', 'FRC_IV', 'SCH_AHT', 'FRC_AHT', 'FRC_CALC_STAFFING',
'FRC_REQ_STAFFING', 'SCH_COVERAGE', 'SCH_ASA', 'FRC_CALC_ASA', 'SCH_MAX_OCCUPANCY_PCT',
'FRC_CALC_MAX_OCCUPANCY_PCT')

GROUP BY

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

ORDER BY

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE
```

Schedule intra-day summary for activity group

```
SELECT

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,

SUM(WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_VALUE)

FROM WFM_PERF_ITEM_TIMESTEP

JOIN WFM_PERF_ITEM_DAY ON (WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_DAY_KEY =
WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_DAY_KEY)

JOIN WFM_ACTIVITY ON (WFM_ACTIVITY.WFM_ACTIVITY_KEY =
WFM_PERF_ITEM_DAY.WFM_ACTIVITY_KEY AND .WFM_SITE_KEY IS NULL AND
WFM_ACTIVITY.WFM_ACTIVITY_TYPE_KEY = 10)

JOIN WFM_BU ON (WFM_BU.WFM_BU_KEY = WFM_ACTIVITY.WFM_BU_KEY)

JOIN WFM_PERF_ITEM ON (WFM_PERF_ITEM.WFM_PERF_ITEM_KEY =
WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_KEY)
```

```
WHERE

WFM_PERF_ITEM_DAY.WFM_DATE = ? AND

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE IN ('SCH_HEADCOUNT', 'SCH_SERVICE_PCT',
'FRC_CALC_SERVICE_PCT', 'SCH_IV', 'FRC_IV', 'SCH_AHT', 'FRC_AHT', 'FRC_CALC_STAFFING',
'FRC_REQ_STAFFING', 'SCH_COVERAGE', 'SCH_ASA', 'FRC_CALC_ASA', 'SCH_MAX_OCCUPANCY_PCT',
'FRC_CALC_MAX_OCCUPANCY_PCT')

GROUP BY

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

ORDER BY

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_BU.WFM_BU_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE
```

Schedule intra-day summary for site

```
SELECT

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,

SUM(WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_VALUE)

FROM WFM_PERF_ITEM_TIMESTEP

JOIN WFM_PERF_ITEM_DAY '''ON''' (WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_DAY_KEY =
WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_DAY_KEY)

JOIN WFM_SITE '''ON''' (WFM_SITE.WFM_SITE_KEY = WFM_PERF_ITEM_DAY.WFM_SITE_KEY)

JOIN WFM_BU '''ON''' (WFM_BU.WFM_BU_KEY = WFM_SITE.WFM_BU_KEY)

JOIN WFM_PERF_ITEM '''ON''' (WFM_PERF_ITEM.WFM_PERF_ITEM_KEY =
WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_KEY)
```

WHERE

WFM_PERF_ITEM_DAY.WFM_DATE = ? AND

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE IN ('SCH_HEADCOUNT', 'SCH_SERVICE_PCT', 'FRC_CALC_SERVICE_PCT', 'SCH_IV', 'FRC_IV', 'SCH_AHT', 'FRC_AHT', 'FRC_CALC_STAFFING', 'FRC_REQ_STAFFING', 'SCH_COVERAGE', 'SCH_ASA', 'FRC_CALC_ASA', 'SCH_MAX_OCCUPANCY_PCT', 'FRC_CALC_MAX_OCCUPANCY_PCT')

GROUP BY

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

ORDER BY

WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,

WFM_BU.WFM_BU_NAME,

WFM_SITE.WFM_SITE_NAME,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE

Contact center performance report for activity

SELECT

WFM_TIME_STEP,

WFM_SITE.WFM_SITE_NAME,

WFM_ACTIVITY.WFM_ACTIVITY_NAME,

WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,

SUM(WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_VALUE)

FROM WFM_PERF_ITEM_TIMESTEP

JOIN WFM_PERF_ITEM_DAY **ON** (WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_DAY_KEY = WFM_PERF_ITEM_TIMESTEP.WFM_PERF_ITEM_DAY_KEY)

JOIN WFM_ACTIVITY **ON** (WFM_ACTIVITY.WFM_ACTIVITY_KEY = WFM_PERF_ITEM_DAY.WFM_ACTIVITY_KEY)

JOIN WFM_SITE **ON** (WFM_SITE.WFM_SITE_KEY = WFM_ACTIVITY.WFM_SITE_KEY)

JOIN WFM_PERF_ITEM '''ON''' (WFM_PERF_ITEM.WFM_PERF_ITEM_KEY =

```
WFM_PERF_ITEM_DAY.WFM_PERF_ITEM_KEY)
WHERE
WFM_TIME_STEP >= ? AND WFM_TIME_STEP < ? AND
WFM_PERF_ITEM.WFM_PERF_ITEM_CODE IN ('ACT_IV', 'ACT_ABANDONED_IV_PCT', 'ACT_AHT',
'ACT_ASA', 'ACT_IV', 'ACT_SERVICE_PCT')
GROUP BY
WFM_SITE.WFM_SITE_NAME,
WFM_ACTIVITY.WFM_ACTIVITY_NAME,
WFM_PERF_ITEM.WFM_PERF_ITEM_CODE,
WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP
ORDER BY
WFM_SITE.WFM_SITE_NAME,
WFM_ACTIVITY.WFM_ACTIVITY_NAME,
WFM_PERF_ITEM_TIMESTEP.WFM_TIME_STEP,
WFM_PERF_ITEM.WFM_PERF_ITEM_CODE
```

Purge procedures and parameters

Contents

- Administrator

Learn how to safely and efficiently purge data in the WFM ETL database.

Related documentation:

-

The WFM ETL database script includes the following database-stored procedures used to enable purging of the ETL database:

Procedure	Description
WFM_PURGE_PROC	The main procedure, used to call all the procedures below, passing a date that is calculated by subtracting a specified number of days from the current date. This procedure is designed to be executed regularly (daily or more often) as a scheduled task. Use the following parameters with this procedure: • `@DAYS_BACK_FOR_DAY`—Purges daily granularity data that is older than the specified number of days from the current day. For example, if this parameter value is`10`, WFM purges data that is older than 10 days prior to today (data for the 10 days prior to today, plus today's data is retained). • `@DAYS_BACK_FOR_TIMESTEP`—Purges 15-minute granularity data that is older than the specified number of days from the current day. See the example above. • `@MAX_PURGE_CHUNK`—Specifies the maximum number of records to purge in one pass.
WFM_PERF_PURGE_PROC	Purges performance statistics data older than a given date. Use the following parameters with this procedure: • `@PURGE_DATE`—Specifies the purge date. Data older than this date is purged. • `@MAX_PURGE_CHUNK`—Specifies the maximum number of records to purge in one pass.
WFM_SCH_PURGE_PROC	Purges schedule data older than a given date.

	Use the following parameters with this procedure: • @PURGE_DATE—Specifies the purge date. Data older than this date is purged. • @MAX_PURGE_CHUNK—Specifies the maximum number of records to purge in one pass.
WFM_ADH_PURGE_PROC	Purges adherence data older than a given date. Use the following parameters with this procedure: • @PURGE_DATE—Specifies the purge date. Data older than this date is purged. • @MAX_PURGE_CHUNK—Specifies the maximum number of records to purge in one pass.
WFM_CFG_PURGE_PROC	Purges configuration data (uses no parameters).

Specifying the number of records to delete

When using the @MAX_PURGE_CHUNK parameter in the procedures described in the table above, be sure to carefully weight it to prevent overtaxing the database server if there are already millions of records that need purging.

Genesys recommends that you limit the number of records to be deleted in one pass to avoid large transactions and allow the database to purge gradually, during subsequent executions of scheduled purging tasks.

Purging terminated agents

When purging terminated agents, the ACTIVE_FLAG setting plays an important role. ACTIVE_FLAG is used in several Dimension tables in WFM ETL database. It's setting indicates whether or not the corresponding record exists in the operational WFM Database.

When you delete agent records from the Genesys Configuration Database, the termination date is set for agents in the operational WFM Database. However, the record is retained so that schedules and other agent information contain all of the related details.

The TERMINATION_DATE is used to indicate when an agent has been deactivated. After you apply the procedure to purge terminated agents, WFM purges all of the agent records with termination dates earlier than the selected date from WFM database and then, sets the ACTIVE_FLAG to 0 to indicate that the record is now retained only in the WFM ETL database.

Frequently asked questions

> ### Tip
> The information presented here is provided to help you understand how the WFM ETL database purges data. However, if issues arise that require troubleshooting, ask your Genesys Professional Services representative for assistance.

The information in these FAQs might be useful when you are purging the database:

Q: Can you suppress or deactivate the data production for the `TIMESTEP` tables?
A: No, only the complete subsystem (Perf, Sch, Adh) can be disabled.

Q: When the DaysBack option is set to a value of `30`, can you purge the `TIMESTEP` tables to a retention time lower than 30 days—using an extreme example, `TRUNCATE TABLE` every night? In this case, would ETL try to recreate the last 30 days of data?
A: Yes, you can purge `TIMESTEP` tables to a retention time lower than the one specified for `DAY` tables. No, ETL would not try to recreate those tables unless there was a change in the main WFM database for those days and data was synchronized back to ETL. When data is synchronized for some date(s), both `TIMESTEP` and `DAY` tables are synchronized simultaneously.

Q: Is Genesys planning to develop a feature, now or in future versions, that would improve the handling of the huge `TIMESTEP` tables or provide a way to deactivate them?
A: Currently, nothing is planned, but we might consider a feature of disabling `TIMESTEP` tables. Please submit a feature request.

Q: What is the best way to start purging the `TIMESTEP` tables?
A: Depending on the scenario, you might want to schedule a task that would call a main purging procedure at regular intervals. For example, using these parameters: `WFM_PURGE_PROC` (`730`, `0`, `100000`)—where the values in brackets represent the `@DAYS_BACK_FOR_DAY`, `@DAYS_BACK_FOR_TIMESTEP`, and `@MAX_PURGE_CHUNK` tables, consecutively.