Genesys Info Mart User's Guide

Representing Dates and Times of Day

5/6/2025

Representing Dates and Times of Day

Because of the large volume of data handled by Genesys Info Mart, most SQL queries of a fact table are constrained by date and time. This page describes how Genesys Info Mart represents dates and times of day.

About dates and times of day

Dates and times of day are stored in the START_TS and END_TS fields, which mark the start and end of each handling stage. The START_DATE_TIME_KEY and END_DATE_TIME_KEY reference the DATE_TIME dimension, which exists in all fact tables. Dates and times are stored in Coordinated Universal Time (UTC) format. You can express local, enterprise, or tenant time using custom DATE_TIME dimensions that offset the UTC time by a specified amount of time.

Because 0 equals 1970 January 1 in UTC, each custom DATE_TIME dimension table will associate this UTC key with a different local time that is relevant for your enterprise. This enables you to use the same keys to create reports in different time zones.

> ### Tip
> For instructions on configuring custom DATE_TIME tables, see Creating Custom Calendars in the *Genesys Info Mart Deployment Guide*.

How dates and times can be constrained

Each fact table row has a surrogate key, START_DATE_TIME_KEY, which references the DATE_TIME dimension that represents its start date and time. This surrogate key can constrain the fact table rows by start date and time of day. Similarly, the END_DATE_TIME_KEY can be used to constrain the fact table rows by end date and time of day.

Each fact table row contains measurements that represent the start date and time of day, and the end date and time of day. These measurements can constrain fact table rows by any arbitrary time span, based on whether the fact table row:

- Starts and ends within the time span.
- Starts before, and ends within, the time span.
- Starts within, and ends after, the time span.
- Starts before, and ends after, the time span.

In any case, you must create the appropriate database indexes in order to efficiently retrieve the data you want.

All fact tables have surrogate key references to the DATE_TIME dimension that represent the 15-minute date and time interval in which a fact started and ended.

The DATE_TIME dimension is useful for constraining based on an arbitrary range of 15-minute time intervals, because this single dimension includes both date and time of day. The dimension keys increase regularly each 15 minutes.

Example: Working with timestamps and the DATE_TIME dimension

The following example illustrates how Genesys Info Mart represents the date and time of an inbound call in local time.

An inbound call arrives at a contact center in San Francisco on October 21, 2009 at 5:05 PM local time (PDT). This time corresponds to 1:05 AM on October 22, 2009 in the UTC GMT time zone, or 1256173500 seconds, expressed in UTC integer format. This integer is stored in the START_TS field in the table containing data about the call.

The start time of the call falls into a 15-minute time interval that begins on October 22, 2009 at 1:00 AM in the UTC GMT time zone, or 1256173200 seconds in UTC integer format. This integer is stored in the START_DATE_TIME_KEY field in the tables containing data related to the call. The value is a surrogate key that can be used to link to the corresponding DATE_TIME_KEY field in any DATE_TIME_CUSTOM dimension. These custom tables contain text labels for the day of the week, month, year, and so on, in whichever local time zone formats your business requires.

In this example, a DATE_TIME_CUSTOM table has been created for the Pacific time zone containing labels in local PDT format. The START_DATE_TIME_KEY field in the fact table containing the UTC integer 1256173200 (corresponding to 5:00 pm PDT), can be used to link to this DATE_TIME_CUSTOM dimension. The correct text labels for the Pacific time zone can then be retrieved for your reports.

Calculating timestamps

To show timestamps in reports converted to a particular time zone, use a simple calculation combining the START_TS (or END_TS) field of a fact table with the DATE_TIME_KEY and CAL_DATE fields of the DATE_TIME_CUSTOM table created for that time zone.

For example, to convert the timestamp value, 1256173500, from the example above, where the time of call arrival is stored in UTC seconds format in the START_TS field of the corresponding INTERACTION_RESOURCE_FACT (IRF) row in a Microsoft SQL Server RDBMS, execute the following query on the DATE_TIME_CUSTOM dimension and IRF table:

```
select DTC.CAL_DATE + CAST ((IRF.START_TS - DTC.DATE_TIME_KEY) as float) / CAST (86400 as
float)
from DATE_TIME_CUSTOM DTC, INTERACTION_RESOURCE_FACT IRF
where DTC.DATE_TIME_KEY = IRF.START_DATE_TIME_KEY
```

The resulting value is October 21, 2009 at 5:05 pm in PDT time zone.

To make the same conversion in an Oracle RDBMS, execute the following query:

```
select DTC.CAL_DATE + (IRF.START_TS - DTC.DATE_TIME_KEY) / 86400
from DATE_TIME_CUSTOM DTC, INTERACTION_RESOURCE_FACT IRF
where DTC.DATE_TIME_KEY = IRF.START_DATE_TIME_KEY
```

Calendar years and week-numbering years

There are two available ways to number the weeks in a year:

- Full-week numbering — In this system, weeks always contain seven days and always start on the day of the week specified as Day 1 in the first-day-of-week configuration option. This system supports the ISO-8601 week configuration used in the European Union and Russia.
- Simple-week numbering — In this system, the weeks calendar matches the calendar year. Week 1 begins on January 1. As a result, the first day of the week differs each year. Most of the time, Weeks 1 or 52 will have fewer than seven days. This is the functionality used in previous releases of Genesys Info Mart.

Table fields for full-week numbering

The DATE_TIME table contains several fields that are used to support the full-week numbering system.

- WEEK_YEAR — This field stores a Week Numbering year. This year may be different from Calendar year. For example, in ISO-8601, 31 December of 2007 is Week 1 Day 1 of 2008. So in this case we have 2007 as the Calendar year and 2008 as the Week Numbering year.
- LABEL_YYYY_WE_D — The label for the day of the week.
- LABEL_TZ — This field stores the time zone offset.

Configuration options for week-numbering

A number of configuration options control how week-numbering is done. For more information, see the descriptions of the [date-time] options in the *Genesys Info Mart Configuration Options Reference*.

> **Important**
>
> In deployments that include Reporting and Analytics Aggregates (RAA) or custom aggregation that uses the DATE_TIME table, Genesys strongly recommends that you do not change [date-time] options that set basic features of the calendar — such as the time zone — after the aggregation engine has started to aggregate data. If you do, at the very least, you will likely need to re-aggregate data.

Maintaining the calendar dimensions

The DATE_TIME dimension is a default calendar that is set up when Genesys Info Mart is initialized and that needs to be populated ahead of time on a continuing basis. Maintenance of the DATE_TIME dimension is controlled by the date-time-min-days-ahead and date-time-max-days-ahead configuration options in the **[date-time]** configuration section. Similarly, maintenance of any custom calendar dimension(s) is controlled by equivalent options in the applicable **[date-time-*]** configuration section(s).

Job_MaintainGIM adds records to the calendar table if the last existing record is earlier than <current-date> + date-time-min-days-ahead. Records are added until <current-date> + date-time-max-days-ahead.

Example

For example, take a scenario in which **date-time-min-days-ahead** is set to 183, **date-time-max-days-ahead** is set to 366, and today is March 30, 2011. In other words, <current-date> + date-time-min-days-ahead = September 29, 2011.

- **Case 1:** DATE_TIME is populated until January 1, 2012.
  Since (March 30, 2011 + 183) < January 1, 2012, **Job_MaintainGIM** will not add any records to the calendar table.
- **Case 2:** DATE_TIME is populated until June 1, 2011.
  Since (March 30, 2011 + 183) > June 1, 2011, **Job_MaintainGIM** will add records to the calendar table until (March 30, 2011 + 366) = March 30, 2012.