



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.

# Reporting and Analytics Aggregates User's Guide

[How Do I View the Aggregation Query?](#)

---

## Contents

- 1 How Do I View the Aggregation Query?
  - 1.1 Using LogLevel=FINEST Logs Database Queries
  - 1.2 printQuery Logs RAA Queries

# How Do I View the Aggregation Query?

The Scheme files for the Genesys-provided hierarchies include high-level constructions that employ macros. These macros simplify development of SQL queries but make it difficult to see the actual queries that are passed to the RDBMS. This page describes where and how to view these queries.

## Using LogLevel=FINEST Logs Database Queries

To view the actual SQL queries for the interval-based aggregates (in which subhour data is stored in tables rather than views), you must submit a request to reaggregate data for any range of time and then run aggregation with the finest log level of detail:

### Procedure: Viewing SQL Queries for Interval-Based Aggregates

**Purpose:** Use this procedure to view the SQL queries for the interval-based aggregates.

#### Steps

1. Using the **-insertPendingAgg** runtime parameter (described in [Reaggregating Data over a Certain Time Range](#)), submit a request to reaggregate an existing range of data. The following command, for example, accomplishes this for all aggregates:

```
java -jar agg\GIMAgg.jar -user=<name> -pass=<password> -jdbcurl=<URL>  
    -insertPendingAgg <AGR_SET>:<START>:<END>
```

#### Tip

Your Info Mart does not have to contain contact center data, but it must be initialized in such a way that the DATE\_TIME table is populated.

2. Run aggregation (in either integrated or autonomous mode) with the log level set to FINEST:

```
java -jar agg\GIMAgg.jar -user=<name> -pass=<password> -jdbcurl=<url>  
    -log=<filename> -levelOfLog=.:FINEST
```

RAA outputs the results of this request and the SELECT statements issued for *all* database queries—including those for the interval-based aggregate tables.

### printQuery Logs RAA Queries

In autonomous mode, you can also specify the **-printQuery** runtime parameter on the command line:

- to view a particular query:

```
java -jar ./agg/GIMAgg.jar -printQuery <queryName>
```

where:

- <queryName> is any query that is known to the aggregation engine.

- to output a particular query to a file:

```
java -jar ./agg/GIMAgg.jar -printQuery <queryName> > <sqlfile>
```

where:

- <queryName> is any query that is known to the aggregation engine, or (in Release 8.5.003 and later) ALL , to print all existing queries.
- <sqlfile> is the name of the log file.

For example:

```
java -jar ./agg/GIMAgg.jar -printQuery QUEUE > logfile.sql
```

Executing this command with this parameter requires no connection to Info Mart. For more information about the **-printQuery** parameter, refer to its description in the *Reporting and Analytics Aggregates Deployment Guide*.