March 2021

Using a Semantic Model to Build and Execute Dynamic SQL Queries

Leigha Jarett

Using a Semantic Model to Build and Execute Dynamic SQL Queries

ABSTRACT

Having a semantic layer to define structured query language (SQL) based metrics, allows end-users of business intelligence (BI) products to compose SQL queries on the fly, using a graphical user interface (GUI). This disclosure describes the use of a semantic layer to define SQL-based metrics which allows end-users of business intelligence (BI) products to compose SQL queries on the fly, using a graphical user interface (UI), with support for utilizing the results of an initial query to automatically compose a new query. The use of a modeling layer to enable query composition on the fly makes such operation simple, performant, and easy to execute, without requiring building an entire new application.

KEYWORDS

- Dynamic SQL
- Business Intelligence (BI)
- Data warehouse
- Data analytics
- Semantic layer

BACKGROUND

Having a semantic layer to define structured query language (SQL) based metrics, allows end-users of business intelligence (BI) products to compose SQL queries on the fly, using a graphical user interface (GUI). In many use cases, users send a query to a database and use the results to compose a new query. An example of such a use case is querying the information schema for a project in a data warehouse. To understand metadata about a dataset, such a query

needs to be executed to obtain the information schema table related to the dataset. However, to retrieve the metadata for all the datasets in the project, a user would first need to query the information schema for the project to find all the dataset names, and then write a query for each dataset.

Technically competent users can typically address such use cases by using dynamic SQL which is a powerful technique within SQL scripting. However, non-technical end users may not be able to compose such queries for themselves. As a result, such users would either have to undertake time consuming and cumbersome operations or rely on technical users for help.

DESCRIPTION

This disclosure describes techniques that enable end users to build and execute dynamic SQL queries using a semantic model. The techniques described in this disclosure build upon functionality provided by data warehouse platforms (e.g., an EXECUTE IMMEDIATE statement) to run strings as SQL queries back in the database. Such commands are typically used when running statements that are not available at compile time, e.g., a column list based on a selection from the user.

Per the techniques described in this disclosure, end users can provide parameters such as table names, column names, values etc. from a GUI to compose dynamic SQL queries on the fly. Fig 1 illustrates an example process flow to enable end users to build and execute dynamic SQL queries using a semantic modeling layer.

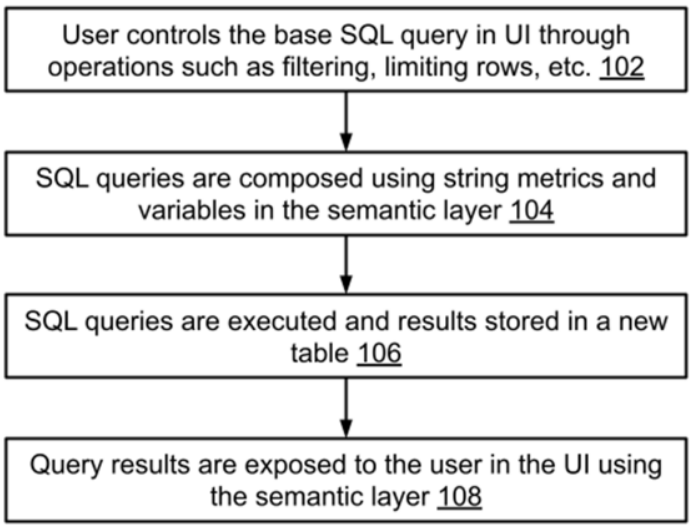

Fig. 1: Example process flow for building dynamic SQL queries using a semantic layer

A semantic model layer exposes a front-end user interface, e.g., via a business intelligence (BI application. An end user controls the base SQL query through operations such as selection, filtering, limiting rows etc. in the GUI (102). New string metrics are defined in the modeling layer and SQL queries are composed using string metrics and variables to represent the user inputs (104). The SQL queries are executed and the results are stored in a table (106). Query results are made available to the end user in the GUI through the semantic layer (108).

The technique described herein can be used by BI practitioners to create blocks of configurable code to be deployed in their tools. The end users can indicate their business needs through a user interface which are then utilized to modify the text of the SQL query through the semantic layer. The query is executed and the query results made available to the end users via the user interface.

Some example use cases for BI practitioners to use the described techniques include:

- Obtaining metadata for all the tables in a project by first querying the information schema on the project.

- Limiting the amount of data scanned. For example, if a user's goal is to compare the data for a particular week of the current year with data of the corresponding week from the previous year, fulfilling the query only needs two weeks of data. However, with date partitioning, the scan of an entire year's worth of data would still be necessary. Using the techniques described herein, an end user can be given access to a date table which can then be filtered on. The modeling layer can be configured to write one query for each day in the resulting dataset (two weeks - one from the current year, one from last year). This can improve performance and cut costs.

The use of a modeling layer to enable query composition on the fly makes such operation simple, performant, and easy to execute, without requiring building an entire new application. The techniques can be utilized in any database that supports EXECUTE IMMEDIATE functionality and a SQL-based semantic layer.

CONCLUSION

This disclosure describes the use of a semantic layer to define SQL-based metrics which allows end-users of business intelligence (BI) products to compose SQL queries on the fly, using a graphical user interface (UI), with support for utilizing the results of an initial query to automatically compose a new query. The use of a modeling layer to enable query composition on the fly makes such operation simple, performant, and easy to execute, without requiring building an entire new application.