MBIS623 Data Warehouse Design

The NYC311 Quality-of-Service Data Warehouse Development

Due: Sunday, 5 June 2022, 9:00 p.m.

Grade contribution: 30%

1. Overview

Data warehouses are intended for **bridging the information gap** where the data from operational databases are processed to support **high-level decision-making** processes. The MBIS623 Data Management course material examines the process of **data warehouse development** with multi-dimensional Star schema. The MBIS623 course material includes the NYC311 service request dataset with **millions of service request records** registered by the NYC311 call centre from 2010.

This project is focused on the development of the **basic extract-transform-load (ETL)** process for the dimensional modelling of the **quality-of-service data warehouse** to provide the suitable view of the NYC311 data. **Quality-of-service** in this data warehouse is **measured by the duration of time (hours) required** to close the requests received by the NYC311 call centre.

While there are many **commercial and open-source ETL and data warehouse tools** that could be used for our purposes, in this project we are **working with the fundamentals** of the process in SQL format to develop the essential skills and understanding of the ETL process and data warehouse design.

The **initial set of queries** for the creation of the NYC311 quality-of-service warehouse is **provided with this project**. Similarly, the **data warehouse with preloaded data** is available for exploration and familiarisation. However, the **service_request_all** table now has new data added to it, extending the total row count **beyond 28 million records**. These new records have been added after the **12th of February 2022**, which is **date('2022-02-12')** in SQL.

Please note the following details to accommodate the added service request:

- The original **service_request** table is available as an SQL view, which contains all records occurring prior to 12th of February 2022.
- The **service_request_all** table includes the complete set of service request records available to us.

The Dimension Tables

There are four dimension tables in our quality-of-service data warehouse:

- The **Agency** dimension table, **dim_agency**, includes the codes of the agencies which are allocated/directed to respond to the NYC311 request calls.
- The **Location** dimension table, **dim_location**, includes the ZIP codes associated only with the ZIP codes falling into the five NYC boroughs, as per the **zip_code_nyc_borough** reference data table.
- The **Request Type** dimension table, **dim_request_type**, includes the managed request types as per the **ref_sr_type_nyc311_open_data_26** reference data table.
- The **Year-Week** dimension table, **dim_yearweek**, which includes the years and weeks, combined as one value, covering the requests submitted prior to or in the week 6 of 2022 (**yearweek 202206**).

Please note that out of the four dimension tables only the **dim_yearweek** table will have to be extended to include the data that has not been loaded into the quality-of-service data warehouse.

The Fact Table

There is only one fact table in the quality-of-service warehouse: the **fact_service_quality** table. This table contains the aggregates of service requests categorised/grouped by **agency_id**, **location_zip**, **(request) type_id**, and yearweek values. The aggregates include the total count of service requests for a given category, the average time taken to complete a service request, calculated as the difference between **`Closed Date`** and **`Created Date`** values, the minimum and maximum times to complete the service request (again, calculated from the **`Closed Date`** and **`Created Date`** values).

The SQL queries to create the dimension tables, the fact table, and to insert data into the dimension and fact table are provided as .sql scripts alongside the project specification (this document) on UC LEARN.

2. Project Tasks

You are required to complete the ETL process to update the quality-of-service data warehouse which includes only the records collected on or before 12th of February 2022. While the **service_request_all** dataset has about half-a-million records added after 12th of February 2022, these records were not accounted in the data warehouse in its current state.

1. **dim_yearweek** dimension table extension:
   Develop SQL code to extend the **dim_yearweek** table with the values that would be required for the updates in the quality-of-service table to account for the added records.
   IMPORTANT: It is essential that the SQL code for this task only inserts the new values into the table, rather than overwriting the whole table.
2. **fact_service_quality** fact table extension:
   Develop SQL code to load the new values into the **fact_service_quality** table for the new service request values after 12th of February 2022.
   IMPORTANT: It is essential that the SQL code for this task only inserts the new rows into this table, rather than overwriting the whole table.
3. **Dashboard-style queries**: while dashboards may display graphs for visualisation purposes, underneath those dashboards and graphs are powered by efficient (SQL) queries which slice and dice the target data warehouse. Write three SQL queries to the following specifications:
   a. **Agency-wise breakdown** for quality-of-service (average duration):
      This query should return the average service request processing time for each agency (32 rows, one per agency).
   b. **Borough-wise breakdown** for quality-of-service (average duration):
      This query should return the average service request processing time in each NYC borough (5 rows, on per borough).
   c. **Monthly breakdown** for the **number of service requests**:
      This query should return the total number of requests for each month of the year, where the years are collapsed within each month (12 rows, one per month).
      Can you show the months written in words, rather than numbers?
4. **Optional (top 10/100 marks) no-data-warehouse dashboard query**:
   Choose any of the three dash-board-style queries to provide the same output **in absence of the data warehouse** – write this SQL query to work with the raw data (the **service_request_all** table) to provide the same output as the corresponding dashboard-style

query. This query would be quite a bit more complex than the corresponding dashboard-style query.

Please note that you may write these queries either on MySQL Server or on Snowflake. There are advantages to completing this project on Snowflake:

- *There are no restrictions associated with MySQL Server remote connection or data importing overhead if you work with your local MySQL Server on your hardware.*
- *Snowflake performance exceeds MySQL Server (local or remote) by miles – you will be able to do your work much faster, because you will get the results of your queries almost instantaneously.*
- *There are fewer restrictions on using extra space for creating extra/temporary tables for intermediate results and prototyping in your schema.*

Submission Requirements

The project submission must include **only the SQL file** with your queries, submitted via UC LEARN, the Data Warehouse Project drop box.

Each query should be set out in **an easy-to-read** way, **with comments and explanations** outlining the **details and assumptions** where necessary.

3. Marking Schedule

Please note the following allocation of marks for your project submissions:

Dimension (yearweek) table query	20%
Fact (quality-of-service) table	30%
Dashboard-style queries	
Agency-wise breakdown	10%
Borough-wise	20%
Monthly breakdown	10%
No-data-warehouse dashboard-style query	10%
Total:	100%

4. Getting Started and Seeking Clarifications

It is natural you will have questions to clarify some of the **requirements and some technical matters related to the dataset**, MySQL Server or Snowflake, and the rest of it all. Please note: don't email these questions to your course lecturer/coordinator/tutor; instead, please make sure to **post these questions via the Discussion Forum**.

Given the current context we are working in, all things around distance learning, COVID restrictions/regulation and the like, there is no resources to maintain one-on-one communication about course-related matters when it comes to the questions that may be of relevance to the rest of the class. The one-on-one emails to the course teaching team will not likely merit a response if they are suitable to be dealt with via the Discussion Forum.