MBIS623 Data Warehouse Assignment

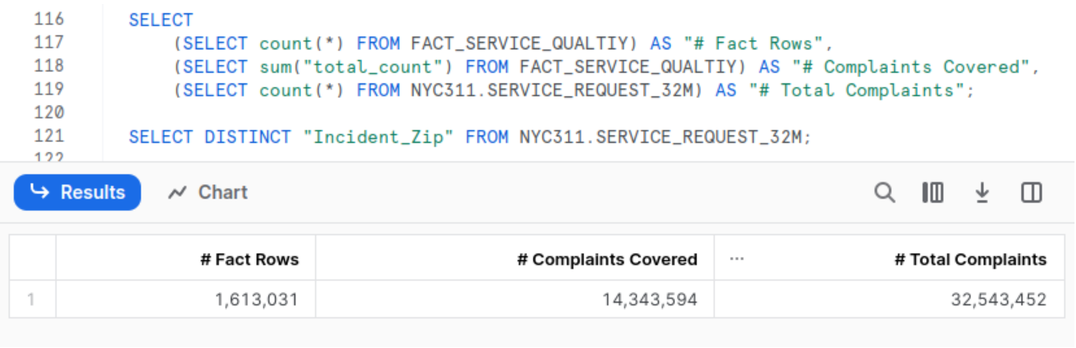
The NYC311 Data Warehouse Cleansing and Reporting

Due: Sunday, June 4th, 9:00 p.m.
Cut-off Date: Sunday, June 18th, 9:00 p.m.
Grade contribution: 30%

1. Overview

You are required to increase the coverage of the NYC 311 Call Centre data for reporting improvement. Consider the following points:

- In the tutorials we have created the dimension and fact tables for the NYC311 data warehouse. The extract-transform-load process (in our case it is just a single query) is used to populate the fact table.
- The query to populate the table JOINs four dimension table to bin the data by agency, year/week, location (ZIP) and complaint type.
- As this has been demonstrated in the tutorials, the data in the ZIP and complain type columns has many values that don't match the reference/dimension values.
- And so, out of the 32,5 M rows in the original service request table, the fact table created according to Tutorial 10 instructions, covers only 14,4 M service request:

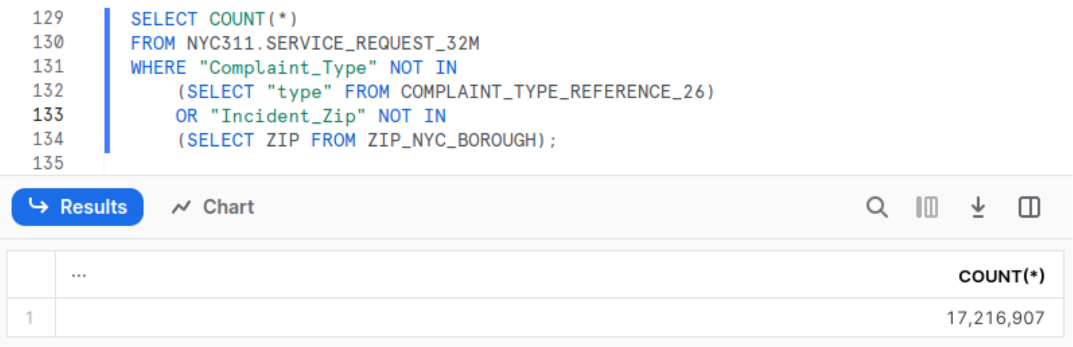


```
SELECT
    (SELECT count(*) FROM FACT_SERVICE_QUALTIY) AS "# Fact Rows",
    (SELECT sum("total_count") FROM FACT_SERVICE_QUALTIY) AS "# Complaints Covered",
    (SELECT count(*) FROM NYC311.SERVICE_REQUEST_32M) AS "# Total Complaints";

SELECT DISTINCT "Incident_Zip" FROM NYC311.SERVICE_REQUEST_32M;
```

	# Fact Rows	# Complaints Covered	...	# Total Complaints
1	1,613,031	14,343,594		32,543,452

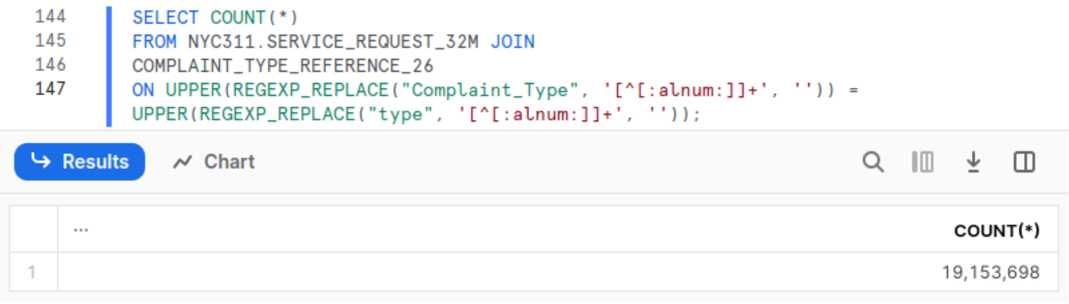
- This query confirms that about 17.2 M rows from the original 32 M rows table are left out of the JOIN query we use to populate the fact table simply because there is no matching data in the relevant columns:



```
SELECT COUNT(*)
FROM NYC311.SERVICE_REQUEST_32M
WHERE "Complaint_Type" NOT IN
    (SELECT "type" FROM COMPLAINT_TYPE_REFERENCE_26)
    OR "Incident_Zip" NOT IN
    (SELECT ZIP FROM ZIP_NYC_BOROUGH);
```

	...	COUNT(*)
1		17,216,907

- Your job is to increase the coverage and perform further data quality assessment and reporting. How do you increase the coverage? Let’s examine this query:



```
SELECT COUNT(*)
FROM NYC311.SERVICE_REQUEST_32M JOIN
COMPLAINT_TYPE_REFERENCE_26
ON UPPER(REGEXP_REPLACE("Complaint_Type", '[^[:alnum:]]+', '')) =
UPPER(REGEXP_REPLACE("type", '[^[:alnum:]]+', ''));
```

	...	COUNT(*)
1		19,153,698

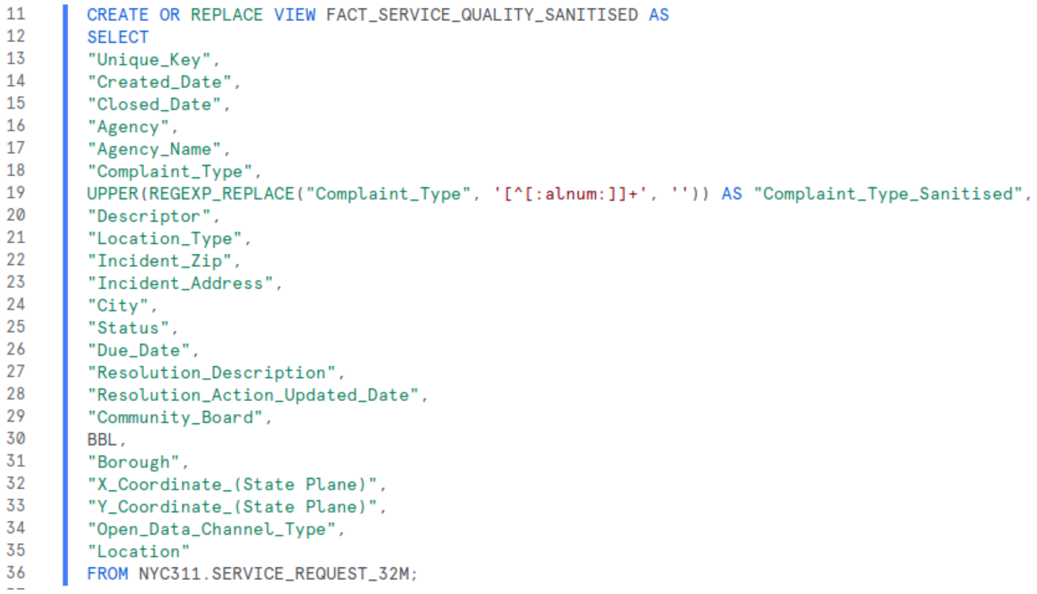
This query uses case conversion and regular expressions to remove everything but alphanumeric characters from the complaint type values in to increase the number of matching values in the service request table and the reference complaint type values. For example:

"PAINT / PLASTER" != "Paint - Plaster",

but after case conversion and removal of anything non-alphanumeric things look like this:

"PAINTPLASTER" == "PAINTPLASTER".

- And so, instead of creating a new table with clean(er) data we can create a view, like this:



```
CREATE OR REPLACE VIEW FACT_SERVICE_QUALITY_SANITISED AS
SELECT
"Unique_Key",
"Created_Date",
"Closed_Date",
"Agency",
"Agency_Name",
"Complaint_Type",
UPPER(REGEXP_REPLACE("Complaint_Type", '[^[:alnum:]]+', '')) AS "Complaint_Type_Sanitised",
"Descriptor",
"Location_Type",
"Incident_Zip",
"Incident_Address",
"City",
"Status",
"Due_Date",
"Resolution_Description",
"Resolution_Action_Updated_Date",
"Community_Board",
BBL,
"Borough",
"X_Coordinate_(State Plane)",
"Y_Coordinate_(State Plane)",
"Open_Data_Channel_Type",
"Location"
FROM NYC311.SERVICE_REQUEST_32M;
```

In this view (apart from dropping a number of columns which are not relevant to our goals at this stage) we add a new column, "Complaint_Type_Sanitised" which massages the complaint type values to improve our reference data matching in the ETL/populate JOIN query, where we would use this view instead of the original table.

- Similarly, we can add to this view another column, say, "Incident_Zip_Sanitized" which would massage the original data appropriately to increase the number of matching ZIP codes using appropriate functions to clean up (i.e., trim) the ZIP values.

2. Assignment Tasks and Expectations

Task One

Improve the request coverage in the fact table following the hints in the previous section and extending those with your own ideas.

Task Two

Write a query to identify request types that are recorded in the dataset as handled by more than one agency. You can use either the original 32 M table or the fact service quality table for this purpose.

Task Three

Write a query and produce a convincing chart/visualisation to show which NYC agencies are improving their service quality (based on faster request processing times) over time (months and years). Yes, you can modify the fact service quality table to include new columns.

Task Four

Write a query and produce a convincing chart/visualisation to show which NYC boroughs may be functioning better than others and are improving over time. You may use a subset of agencies for this purpose, choosing only a few criteria.

Submission Format

Put/paste all queries, charts/visualisations into an MS Word Document, along with clear explanation of your thinking and query development process.

Please note that the queries are to be pasted as text (not images), while the graphs can be pasted as screenshots, or better, saved appropriately as PNG — Snowflake offers this feature along with the chart editor.

Your explanation of the development process is intended to convince the markers of your clarity of understanding of your work — this is critical, so put your time into this. As a guide, two-to-three paragraphs explaining each query will be sufficient if written appropriately.

3. Marking Schedule

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

Task One: queries and explanation	30%
Task Two: query and explanation	15%
Task Three: query, explanation, and chart	30%
Task Four: query, explanation, and chart	25%
Total:	100%