Logo

Description automatically generated

**College of Professional Studies**

**Northeastern University San Jose**

**MPS Analytics**

**Course: ALY6030: Data Warehousing & SQL**

**Assignment:**

Module 3 Assignment

**Submitted on:**

Mar 22, 2024

**Submitted to:**  **Submitted by:**

Prof: Kayal Chandrasekaran Nikshita Ranganathan

**Introduction**

The objective of this project is to develop a comprehensive dimensional model for analyzing the data collected from inspections of public housing developments. This model will assist the HUD in evaluating the effectiveness and financial implications of these inspections. By doing so, it aims to provide senior management with actionable insights for optimizing inspection processes and ensuring the highest standards of living conditions in public housing.

**Data Overview**

The dataset comprises detailed records of inspections carried out by Public Housing Agencies across the US. It includes essential data points such as the PHA names, housing development names and addresses, inspection dates, inspection scores (ranging from 0 to 100), and the costs incurred for each inspection. This dataset not only serves as a snapshot of the current state of public housing conditions but also provides a foundation for tracking performance over time, identifying areas for improvement, and assessing the financial impact of the inspection process on public resources.

A screenshot of a computer

Description automatically generated

**Questions**

1. **Facts:**

There are **two facts** in this dataset - COST\_OF\_INSPECTION\_IN\_DOLLARS and INSPECTION\_SCORE.

The facts identified are:

* COST\_OF\_INSPECTION\_IN\_DOLLARS: This represents the monetary expense incurred for conducting each inspection of a housing unit.
* INSPECTION\_SCORE: Denotes the evaluation score given to a housing development following an inspection, scaled from 0 to 100 percent.

Understanding the additive and non-additive nature of facts is essential for designing the star schema and fact tables.

COST\_OF\_INSPECTION\_IN\_DOLLARS is an addictive fact column. It can be summed up across different dimensions such as time (monthly, yearly), geographical location (city, state), or even by the PHA name. This allows for analyses such as total inspection costs over a period, total costs by PHA, or total costs by location.

On the other hand, INSPECTION\_SCORE is generally considered a non-additive fact because summing scores across properties or inspections does not provide useful information.

1. **Dimensions:**

There are **six dimensions** in this dataset.

**Dimensions Identified:**

* PUBLIC\_HOUSING\_AGENCY\_NAME - Refers to the organization overseeing the housing development.
* INSPECTED\_DEVELOPMENT\_NAME - Identifies the specific property under inspection.
* INSPECTED\_DEVELOPMENT\_ADDRESS - Provides the location details of the property.
* INSPECTED\_DEVELOPMENT\_CITY - Indicates the city in which the housing development is situated.
* INSPECTED\_DEVELOPMENT\_STATE - Specifies the state where the property is located.
* INSPECTION\_DATE - Marks the date on which the inspection took place.

1. **Facts for inspection level and inspection costs**

**Inspection Level – Transactional fact table**

For capturing detailed inspection activities, a Transaction Fact Table is suitable. This table will store detailed records for each inspection event. This table is organized such that each entry is linked to a unique INSPECTION\_ID, and it precisely captures the COST\_OF\_INSPECTION\_IN\_DOLLARS and INSPECTION\_SCORE. Transactional fact tables allow for deep-dive analyses into the data, enabling queries that can answer specific questions about inspection outcomes, trends, and anomalies at the most detailed level.

**Inspection Costs – Aggregate fact table**

This table will aggregate inspection costs by month for each PHA. Each row represents a specific month linked to a particular agency, with facts reflecting the aggregated (total/average) COST\_OF\_INSPECTION\_IN\_DOLLARS for that timeframe. By summarizing inspection costs by month, it enables Senior Management to quickly assess changes in inspection costs over time, identify patterns, and make informed decisions regarding resource allocation and policy adjustments.

1. **Type from Kimball's reading**

For addressing changes in the names and addresses of public housing agencies, which can frequently occur due to mergers with other agencies, it is recommended to employ a **Slowly Changing Dimension (SCD) Type 2 strategy**.

SCD Type 2 involves keeping a full historical record of dimension changes in the database. This means that for each change, rather than updating the existing record (as in SCD Type 1), a new record is added with the updated information. This approach allows for the preservation of historical data, enabling analyses that require understanding how data has changed over time, such as tracking the historical performance of a PHA before and after a merger or name change.

Justification for Choosing SCD Type 2:

* Historical Accuracy: SCD Type 2 maintains an accurate historical record, crucial for analyzing trends over time and understanding the impact of PHA mergers or name changes on housing inspections and scores.
* Auditing and Reporting: This approach provides a clear audit trail, showing when changes occurred and what the previous and new values were, enhancing data transparency.
* Flexibility in Analysis: It allows for more sophisticated analyses, including comparing inspection outcomes before and after specific events, such as agency mergers.
* No Data Loss: Unlike SCD Type 1, which overwrites existing information, SCD Type 2 ensures that no historical data is lost due to updates, preserving the integrity of past analyses.

Type 2 SCDs involve adding a few additional fields to the dimension tables, such as a version number, start date, and end date for each record. These fields enable queries to accurately identify the correct version of a PHA's information at any given point in time. This strategy ensures that the dimensional model can accommodate changes over time, providing a robust foundation for analysis and reporting.

1. **SQL code**

Initial analysis

The code begins with a CTE, which serves multiple purposes:

* **Date Conversion:** Converts the INSPECTION\_DATE from a string to a date format using STR\_TO\_DATE, facilitating proper chronological ordering.
* **Calculation of Inspection Costs and Dates:** Utilizes the LAG window function to retrieve the cost and date of the previous (second most recent) inspection for each PHA, enabling a direct comparison between the two most recent inspections.
* **Cost Change Calculation:** Determines the difference in inspection costs (CHANGE\_IN\_COST) between the most recent inspection and the one preceding it.
* **Percentage Change Calculation:** Calculates the percentage change in inspection costs (PERCENT\_CHANGE\_IN\_COST) between these two inspections, providing a normalized measure of cost variation.
* **Inspection Count and Ranking:** Counts the total number of inspections for each PHA (COUNT\_INSPECTION) and assigns a row number (INSPECTION\_ROW) to each inspection within a PHA, based on the inspection date in descending order.

**Checking the result of First CTE:**

**A screenshot of a computer

Description automatically generated**

**Filtering for Relevant Data**

A second CTE, named Filter, refines the dataset by applying two critical filters:

* **Positive Cost Change:** This selects only those records where there was an increase in the inspection cost (CHANGE\_IN\_COST > 0).
* **Minimum Inspection Requirement:** This filter ensures that only PHAs with more than one inspection (COUNT\_INSPECTION > 1) are considered, as a single inspection does not allow for cost comparison.

**Final Result – (448 rows)**

A screenshot of a computer

Description automatically generated

**Conclusion**

We successfully identified critical facts and dimensions within the dataset, distinguishing between additive and non-additive facts. Moreover, the application of advanced SQL techniques has enabled us to extract meaningful trends and patterns from the data. As HUD moves forward, this project will serve as a valuable asset in their ongoing efforts to ensure safe, affordable, and high-quality housing for all.

**References**

Kimball, R., & Ross, M. (1996). *The Data Warehouse Toolkit: the complete guide to dimensional modeling*. <http://ci.nii.ac.jp/ncid/BA75756421>

Alex The Analyst. (2020, December 8). *Advanced SQL Tutorial | CTE (Common Table Expression)* [Video]. YouTube. <https://www.youtube.com/watch?v=K1WeoKxLZ5o>

Gupta, R. (2019, December 9). *SQL Lag function overview and examples*. SQL Shack - Articles About Database Auditing, Server Performance, Data Recovery, and More. <https://www.sqlshack.com/sql-lag-function-overview-and-examples/>