ALY6030 - Data Warehousing and SQL - Week 5

Professor Adam Jones

Nov 24, 2024

JAMES GEORGE

NUID: 002662779

Detailed Analysis Report: Public Housing Inspection Data

1.Facts

In the public housing inspection dataset, we have two distinct facts. These facts represent the measurable metrics that can be analyzed across various dimensions in our dataset.

The two facts identified in the dataset are COST_OF_INSPECTION_IN_DOLLARS and INSPECTION_SCORE. The COST_OF_INSPECTION_IN_DOLLARS represents the monetary amount spent on each inspection event, while INSPECTION_SCORE captures the numerical evaluation result of each inspection conducted at the housing facilities.

The COST_OF_INSPECTION_IN_DOLLARS is an additive fact, meaning its values can be meaningfully aggregated across any dimension in our dataset. This fact can be summed across time periods, geographic locations, or housing agencies to provide meaningful totals and enable cost analysis at various levels.

The INSPECTION_SCORE is a semi-additive fact. This classification as semi-additive is appropriate because while these scores can be averaged to provide meaningful insights, they cannot be summed across time periods. For instance, if a facility receives scores of 90 and 85 in two different inspections, summing these scores would not provide a meaningful metric, but averaging them would give us useful information about the facility's overall performance.

2.Dimensions

The dataset contains the following seven dimensions:

1. INSPECTION_ID
2. PUBLIC_HOUSING_AGENCY_NAME
3. INSPECTED_DEVELOPMENT_NAME
4. INSPECTED_DEVELOPMENT_ADDRESS
5. INSPECTED_DEVELOPMENT_CITY
6. INSPECTED_DEVELOPMENT_STATE
7. INSPECTION_DATE

INSPECTION_ID serves as a unique identifier dimension for each inspection record, allowing us to distinguish between individual inspection events.

PUBLIC_HOUSING_AGENCY_NAME functions as the organizational dimension, identifying which agency conducted each inspection. This dimension contains the full names of housing authorities responsible for the inspections.

INSPECTED_DEVELOPMENT_NAME acts as the property dimension, identifying the specific property being inspected. This provides information about which housing development underwent the inspection.

The geographic aspects are captured through three distinct but related dimensions: INSPECTED_DEVELOPMENT_ADDRESS provides the street-level location information, INSPECTED_DEVELOPMENT_CITY captures the city where the property is located, INSPECTED_DEVELOPMENT_STATE indicates the state of the property.

INSPECTION_DATE serves as our time dimension, recording when each inspection occurred. From our data, we can see dates formatted as MM/DD/YYYY before conversion to the proper DATE format.

An additional INSPECTED_DEVELOPMENT_ADDRESS field provides supplementary location reference information, offering another perspective on the property's location details.

Together, these dimensions provide the necessary context for analyzing both the cost and score facts, enabling multi-dimensional analysis of the inspection data.

3. Fact Table Type Analysis

Based on senior management's requirements for both detailed inspection-level data and periodic cost summaries, a dual fact table approach would be most effective. This approach would utilize both Transaction and Periodic Snapshot fact tables to meet the specific needs stated.

The Transaction fact table would serve as the foundational data store, capturing each inspection event. This directly addresses the requirement for inspection-level data, allowing each inspection record to be stored with its associated cost and score. This meets management's need to view individual inspection details.

Complementing this, the Periodic Snapshot fact table would provide aggregated monthly views of inspection costs, directly fulfilling management's requirement for periodic cost summaries by month. This structure addresses the specific need for monthly cost summaries without requiring repeated aggregation of transaction-level data.

So to summarize:

Two types of fact tables are recommended:

A. Transaction Fact Table

- Purpose: Store individual inspection records
- Justification:
  - Captures all the level inspection events
  - Supports analysis that could be drilled down
  - Maintains data granularity

B. Periodic Snapshot Fact Table

- Purpose: Monthly inspection cost summaries
- Justification:
    - Meets management's requirement for periodic summaries
    - Optimizes monthly reporting performance

4. Slowly Changing Dimension Management

Regarding the management of public housing agency names and addresses that undergo frequent mergers, implementing a Type 2 Slowly Changing Dimension (SCD) approach would be most appropriate. This recommendation is based directly on management's stated concern about tracking changes in agency names and addresses due to frequent mergers.

Type 2 SCD handling would maintain a historical record of agency changes, creating new records whenever an agency's name or address changes due to mergers. This directly addresses the need to track how agencies change over time due to mergers.

The choice of Type 2 over other SCD types is based on the specific requirements:

- Type 0 would not track merger changes at all
- Type 1 would lose the history of agency mergers
- Type 2 preserves the full history of agency changes
- Type 3 would only preserve one previous state, insufficient for tracking multiple mergers

The Type 2 approach ensures that when agencies merge, we maintain both the historical and current agency information, directly supporting the need to track these organizational changes over time. This addresses the specific requirement of handling frequent agency mergers while maintaining accurate historical records.

So to summarize:

I would recommend to use Type 2 SCD

Justification:

- Preserves historical accuracy of agency changes
- Maintains audit trail of mergers
- Supports historical analysis

5. Implementation Analysis: Cost Increase Assessment for Public Housing Agencies

Date Format Handling :

```
SET SQL_SAFE_UPDATES = 0;
UPDATE public_housing_inspection_data
SET INSPECTION_DATE = STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y');
ALTER TABLE public_housing_inspection_data
MODIFY COLUMN INSPECTION_DATE DATE;
SET SQL_SAFE_UPDATES = 1;
```

Analysis Query Implementation:

```
WITH RankedInspections AS (
   SELECT
      PUBLIC_HOUSING_AGENCY_NAME as PHA_NAME,
      INSPECTION_DATE,
      COST_OF_INSPECTION_IN_DOLLARS,
      LAG(INSPECTION_DATE) OVER (
         PARTITION BY PUBLIC_HOUSING_AGENCY_NAME
         ORDER BY INSPECTION_DATE DESC
      ) as PREV_INSPECTION_DATE,
      LAG(COST_OF_INSPECTION_IN_DOLLARS) OVER (
         PARTITION BY PUBLIC_HOUSING_AGENCY_NAME
         ORDER BY INSPECTION_DATE DESC
      ) as PREV_INSPECTION_COST,
      COUNT(*) OVER (
         PARTITION BY PUBLIC_HOUSING_AGENCY_NAME
      ) as inspection_count
   FROM public_housing_inspection_data
)
SELECT DISTINCT
   [columns as specified]
FROM RankedInspections
WHERE [filtering conditions]
ORDER BY PERCENT_CHANGE_IN_COST DESC;
```

Data Preparation and Date Format Handling

The initial phase of this analysis required careful handling of the date format conversion. When the data was imported into MySQL Workbench, the INSPECTION_DATE field was in text format (MM/DD/YYYY), which would not support proper date-based operations. To address this, we implemented a two-step date conversion process. First, we disabled the SQL safe update mode to allow for bulk updates, then used the STR_TO_DATE function to convert the text dates into MySQL's native date format. Following this, we altered the table structure to formally change the column type from VARCHAR to DATE, ensuring optimal performance for date-based operations.

Core Analysis Implementation

The central analysis was implemented using a Common Table Expression (CTE) with window functions, specifically utilizing the LAG function to compare consecutive inspections. The CTE, named RankedInspections, served as a foundation for comparing inspection costs across time periods for each PHA. Within this CTE, we partitioned the data by PUBLIC_HOUSING_AGENCY_NAME and ordered by INSPECTION_DATE in descending order to ensure we captured the most recent inspections first. This approach allowed us to handle the requirement of comparing the most recent inspection with its immediate predecessor.

Query Structure and Filtering

The main query was structured to handle multiple requirements simultaneously. It began with the selection of distinct records to eliminate duplicates, ensuring each PHA appeared only once in the final output. The WHERE clause implemented multiple filtering conditions: it excluded PHAs with NULL previous inspection costs (indicating only one inspection), ensured the current inspection cost was greater than the previous one (showing only increases), and verified that each PHA had multiple inspections through a count window function. The results were ordered by the percentage change in cost in descending order, highlighting the most significant cost increases first.

Results Analysis

The output data revealed insights into inspection cost trends. Looking at our results, we observed significant cost increases for some PHAs. From the data output, we can see that the highest increase was from a housing authority whose inspection costs rose from $10,142 to $38,841, representing a 282.97% increase. This was followed by similar high-percentage increases, with the top cases showing increases above 270%.

Data Export and Deliverable Creation

The final step involved exporting the analyzed data to a CSV file format. This file included all seven required columns: PHA_NAME, MR_INSPECTION_DATE, MR_INSPECTION_COST, SECOND_MR_INSPECTION_DATE, SECOND_MR_INSPECTION_COST, CHANGE_IN_COST, and PERCENT_CHANGE_IN_COST. The export process maintained the proper date formatting (YYYY-MM-DD) for both current and previous inspection dates, and the cost calculations were preserved with appropriate precision. The resulting CSV file provided management with the requested dataset showing cost increase information for each PHA, exactly as specified in the requirements.

This analysis provides senior management with the specific information they requested, formatted according to their requirements. The dataset enables the identification of PHAs that experienced inspection cost increases, with clear visibility into the magnitude of these increases.