Northeastern University

ALY 6030 Predictive Analytics

Module 4 Project

Public Housing Inspections Star Schema

Name: Nai-Cih (Nikki) Chen

College of Professional Studies, Northeastern University

Professor Adam Jones

November 22, 2024

Introduction

This report provides a comprehensive analysis of public housing inspection data, focusing on cost trends and changes for Public Housing Agencies (PHAs). The dataset contains critical information, including inspection dates, costs, and scores, enabling detailed insights into inspection performance and financial implications. Key areas of analysis include identifying PHAs with increasing inspection costs, tracking historical changes in agency details through Slowly Changing Dimensions (SCD), and leveraging SQL queries to extract meaningful patterns. Advanced techniques, such as window functions and cost trend analysis, were employed to ensure accurate and actionable results. The findings aim to support senior management in understanding cost dynamics, maintaining historical accuracy, and making informed decisions for resource allocation and operational improvements.

Analysis

Question 1: Facts

1. How many facts are there in this dataset?

There are **two primary facts** identified in the dataset.

2. Which facts do you identify?

- **Cost of Inspection (COST_OF_INSPECTION_IN_DOLLARS):**
  Represents the cost of each inspection, a measurable value.
- **Inspection Score (INSPECTION_SCORE):**
  Represents the evaluation score of the inspected housing development, a measurable value.

3. What type of facts are they?

Both **Cost of Inspection** and **Inspection Score** are **Additive Facts**:

- **Cost of Inspection:** Can be aggregated across time or regions to generate summary reports.
- **Inspection Score:** Can be averaged or used to calculate other aggregate measures, such as the highest or lowest scores.

Question 2: Dimensions

1. How many dimensions are there in this dataset?

There are **six primary dimensions** identified in the dataset.

2. Which dimensions do you identify?

- **Inspection Agency (PUBLIC_HOUSING_AGENCY_NAME):**
  Categorizes the cost and score by the agency conducting the inspections.
- **Development Name (INSPECTED_DEVELOPMENT_NAME):**
  Represents the specific target of the inspection.
- **Development Address (INSPECTED_DEVELOPMENT_ADDRESS):**

Provides the exact geographical location of the inspected housing development.

- **Development City (INSPECTED_DEVELOPMENT_CITY):**

  Allows classification of data by city, enabling geographical analysis.

- **Development State (INSPECTED_DEVELOPMENT_STATE):**

  Enables classification of data at a higher regional level for analysis.

- **Inspection Date (INSPECTION_DATE):**

  Serves as a time dimension, supporting time-series analysis by date, month, or year.

Question 3: Fact Table Design

Context:

Senior management is interested in viewing the facts identified above at both the **inspection level** and as a **periodic summary** of inspection costs for each month. The goal is to design a fact table structure that meets these needs.

Which type(s) of fact tables would you use and why?

Based on the context, there are two types of fact tables that would be appropriate for storing these data:

1. Transactional Fact Table (Inspection Level Data)

This type of fact table stores **detailed transaction-level data** and is suitable for capturing each individual inspection's information. It will include data such as:

- Inspection ID (INSPECTION_ID)
- Public Housing Agency Name (PUBLIC_HOUSING_AGENCY_NAME)
- Cost of Inspection (COST_OF_INSPECTION_IN_DOLLARS)
- Inspection Score (INSPECTION_SCORE)
- Inspection Date (INSPECTION_DATE)

Reason for choosing this type:

This table would allow senior management to view each individual inspection, giving them the most granular level of data for analysis, such as the cost and score for each specific inspection event. It supports detailed querying and analysis at the inspection level.

2. Snapshot Fact Table (Periodic Summary Data)

This type of fact table aggregates data over **periods** (e.g., monthly) and stores summary-level data for each time period. The data could include:

- Month (extracted from INSPECTION_DATE)
- Total Inspection Cost for the Month(SUM(COST_OF_INSPECTION_IN_DOLLARS))
- Average Inspection Score for the Month (AVG(INSPECTION_SCORE))

Reason for choosing this type:

This fact table would help senior management analyze trends over time by summarizing inspection costs and scores by month. It reduces the volume of data and supports high-level reporting, such as monthly cost analysis and performance comparisons across months.

Why these fact table types are appropriate:

- Transactional Fact Table enables detailed analysis at the inspection level and allows for more granular insights into individual inspections.
- Snapshot Fact Table provides a higher-level view for periodic trend analysis, allowing management to quickly review and compare inspection costs and scores by month.

By implementing both types of fact tables, the design can provide flexibility for analysis at both granular and summary levels, aligning with management's needs.Conclusion

Question 4: Slowly Changing Dimensions (SCD)

Context:

Senior management is concerned with the frequent changes in the **names and addresses** of public housing agencies, as they often merge with other agencies. This scenario is a classic example of a **slowly changing dimension** (SCD), where historical data must reflect changes in dimensional attributes over time.

How should we handle this slowly changing dimension?

To manage the slowly changing dimension (SCD) for the public housing agency names and addresses, we have several options based on how we want to track historical changes. The four common types of slowly changing dimensions are Type 0, 1, 2, and 3. Here's an overview of each:

- **Type 0: No Change**

The attribute remains unchanged, and any updates are ignored. This approach is used when historical accuracy is not required.

- **Type 1: Overwrite**

The record is updated with the new value, meaning only the most recent data is retained, and historical values are overwritten.

- **Type 2: Add New Record with Versioning**

A new record is created with a new surrogate key, preserving the history of changes using effective and end dates to track the timeline of each record.

- **Type 3: Add New Attributes for Tracking**

The old value is stored in a new column, so only the most recent change and the previous value are tracked.

Why Type 2 is the most appropriate choice?

Type 2 is ideal because it preserves historical data by creating a new record with a new surrogate key whenever a change occurs. This approach allows us to maintain the history of public housing agency names and addresses while reflecting the most current information. This is particularly useful for tracking the performance of agencies over time, even after they have undergone name or address changes.

Why not choose Types 0, 1, or 3?

- **Type 0 (No Change):**

This option is unsuitable because it ignores changes in the agency names and addresses. Given the frequency of changes, this would lead to outdated and inaccurate data, failing to reflect the true history of inspections.

- **Type 1 (Overwrite):**

Type 1 would overwrite the previous agency names and addresses with the new values, resulting in the loss of historical context. If only the most recent data is stored, we would lose the ability to associate past inspections with the correct agency at the time they were performed. This would severely limit the accuracy of historical analysis.

- **Type 3 (Limited History):**

While Type 3 could be used to track the most recent and previous changes, it would not capture the full history of changes over time. Since public housing agencies may change multiple times, Type 3's limited tracking does not provide sufficient historical data for comprehensive analysis.

Therefore, **Type 2** is the most suitable approach for handling slowly changing dimensions for public housing agencies. It allows us to keep a complete historical record of changes, ensuring that inspection data can be accurately linked to the agency responsible for the inspection at the time it was conducted.

Question 5: Data Analysis and Output

Context:

Senior management requires a detailed analysis of inspection costs, focusing on Public Housing Agencies (PHAs) with increasing inspection costs. The final deliverable includes essential columns for identifying cost changes and their percentage impact.

This query uses the **LAG** window function to compare the cost and date of each PHA's two most recent inspections.

Rows are filtered to include only PHAs with the following characteristics:

- At least two inspections (to ensure valid comparisons).
- An increase in inspection costs between the most recent and the second most recent inspection.

```sql
WITH InspectionRanked AS (
    SELECT PUBLIC_HOUSING_AGENCY_NAME AS PHA_NAME,
           INSPECTION_DATE,
           COST_OF_INSPECTION_IN_DOLLARS AS INSPECTION_COST,
           LAG(INSPECTION_DATE) OVER (
               PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY INSPECTION_DATE) AS SECOND_MR_INSPECTION_DATE,
           LAG(COST_OF_INSPECTION_IN_DOLLARS) OVER (
               PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY INSPECTION_DATE) AS SECOND_MR_INSPECTION_COST,
           ROW_NUMBER() OVER (PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY INSPECTION_DATE DESC) AS RN
    FROM public_housing_inspection
)
SELECT
    PHA_NAME,
    SECOND_MR_INSPECTION_DATE,
    SECOND_MR_INSPECTION_COST,
    INSPECTION_DATE AS MR_INSPECTION_DATE,
    INSPECTION_COST AS MR_INSPECTION_COST,
    (INSPECTION_COST - SECOND_MR_INSPECTION_COST) AS CHANGE_IN_COST,
    ROUND((INSPECTION_COST - SECOND_MR_INSPECTION_COST) / SECOND_MR_INSPECTION_COST * 100, 2) AS PERCENT_CHANGE_IN_COST
FROM InspectionRanked
WHERE SECOND_MR_INSPECTION_COST IS NOT NULL
AND INSPECTION_COST > SECOND_MR_INSPECTION_COST
AND RN = 1;
```

Results:

PHA_NAME	SECOND_MR_INSPECTION_DATE	SECOND_MR_INSPECTION_COST	MR_INSPECTION_DATE	MR_INSPECTION_COST	CHANGE_IN_COST	PERCENT_CHANGE_IN_COST
Akron Metropolitan Housing Autho	2014-10-08	15626.00	2014-10-09	25593.00	9967.00	63.78
Alachua County	2014-05-01	17019.00	2015-01-22	37345.00	20326.00	119.43
Alaska Housing Finance Corporati	2014-11-13	21366.00	2014-11-14	26342.00	4976.00	23.29
Albany Housing Authority	2015-01-09	30247.00	2015-01-12	31115.00	868.00	2.87
Alexandria Redevelopment & Housi	2014-04-18	14767.00	2014-05-09	29123.00	14356.00	97.22
ALLEGHENY COUNTY HOUSING AUTHORI	2015-02-02	36454.00	2015-02-02	37108.00	654.00	1.79
ALTOONA HOUSING AUTHORITY	2014-09-15	24813.00	2014-11-24	25750.00	937.00	3.78
ANNISTON HA	2014-08-21	10785.00	2014-12-30	31506.00	20721.00	192.13
ASHTABULA METROPOLITAN HOUSING A	2014-04-24	13920.00	2014-06-03	37948.00	24028.00	172.61
Athens Metropolitan Housing Auth	2014-05-21	10996.00	2014-05-22	21816.00	10820.00	98.40

Conculsion

By identifying PHAs with increasing inspection costs, the study highlights financial patterns requiring budget optimization attention. The application of Type 2 Slowly Changing Dimensions (SCD) ensures that historical changes in agency names and addresses are accurately tracked, preserving data integrity for long-term analysis. Through SQL-based analysis, including using LAG functions, the report identifies PHAs with significant cost increases while ensuring a focus on actionable data by excluding entries with insufficient inspections. The findings offer a robust foundation for senior management to address cost dynamics, monitor agency performance, and refine inspection strategies. This comprehensive approach demonstrates the importance of leveraging data to enhance decision-making and improve resource management in public housing operations.

References

1. Leis, V., Kundhikanjana, K., Kemper, A., & Neumann, T. (2015). Efficient processing of window functions in analytical SQL queries. Proceedings of the VLDB Endowment, 8(10), 1058-1069.

2. GeeksforGeeks. (2024, November 21). SQL LAG() function. https://www.geeksforgeeks.org/sql-server-lag-function-overview/

3. SQL: Window functions: Lead(). Codecademy. (n.d.). https://www.codecademy.com/resources/docs/sql/window-functions/lead

4. Wikimedia Foundation. (2024, October 2). Slowly changing dimension. Wikipedia. https://en.wikipedia.org/wiki/Slowly_changing_dimension#:~:text=These%20range%20from%20simple%20overwrites,not%20really%20changing%20at%20all.

5. Wikimedia Foundation. (2024b, November 19). Wikimedia Foundation. Wikipedia. https://en.wikipedia.org/wiki/Wikimedia_Foundation