Northeastern University

Assignment 3 —

Public Housing Inspections Star Schema

Yi Sun (ID 002246026)

College of Professional Studies, Northeastern University

ALY 6030: Data Warehousing

Professor: Adam Jones

Nov 27, 2024

Table of Contents

Introduction

This report analyzes inspection data from Public Housing Authorities (PHA) to help senior management review costs, scores, and related details. It includes data on inspection costs, dates, scores, project names, and addresses, along with key questions and their SQL solutions.

Part 1

1.How many facts are there in this dataset?

In this dataset, there are two key facts identified.

2. Which facts do you identify?

- **COST_OF_INSPECTION_IN_DOLLARS:** Represents the cost of conducting an inspection. This is an additive fact that can be aggregated (e.g., summed up).
- **INSPECTION_SCORE:** Represents the score of the inspection. This is also an additive fact that can be aggregated (e.g., averaged).

3.For the facts that you identify, what type of facts are they?

Both of these are **quantitative facts** that can undergo mathematical operations such as summation or averaging. They are **additive facts** because their values can be summed up.

Part 2

1.How many dimensions are there in this dataset?

The dataset contains the following 6 dimensions:

- **PUBLIC_HOUSING_AGENCY_NAME:** For analyzing inspection results across different housing agencies.
- **INSPECTED_DEVELOPMENT_NAME:** For identifying specific inspected developments.
- **INSPECTED_DEVELOPMENT_ADDRESS:** For locating the inspection activities.
- **INSPECTED_DEVELOPMENT_CITY:** For analyzing inspections by city.
- **INSPECTED_DEVELOPMENT_STATE:** For analyzing inspections by state.
- **INSPECTION_DATE:** For examining trends over time.

2.Which dimensions do you identify?

These are descriptive dimensions that provide context for the facts and support grouping, filtering, and aggregating the data during analysis

Part 3

Based on this context, if you were to store these data in a set of fact tables,which type (or types) of fact tables would you use and why?

If these data were to be stored in a set of fact tables, I would choose a measure fact table. The reasons are as followed:

- **Transaction Fact Table:** Each row in the dataset represents a single inspection event, including specific costs and scores. A transaction fact table organizes data by time, location, etc.
- **Summary Fact Table:** To analyze periodic trends, such as monthly inspection costs, a summary fact table can be created to aggregate total costs and average scores by month.

These fact tables would allow analysis from both granular and aggregated perspectives.

Part 4

Based on this context, how would handle this slowly changing dimension? Select from types 0,1,2, or 3 from the Kimball reading. Justify your answer.

Based on the characteristics of the data, different types of Slowly Changing Dimensions (SCD) can be chosen:

- SCD Type 0: No change, data remains constant.
- SCD Type 1: Update existing data by overwriting previous values.
- SCD Type 2: Retain historical records by creating a new record for each change, with valid dates to indicate the periods of validity.
- SCD Type 3: Only retain partial history, typically keeping the "current" and "previous" values.

For this dataset, SCD Type 2 is the appropriate choice. This is because dimensions like public housing agency name and inspected development name may change over time. Using SCD Type 2 allows us to retain historical records, track changes in dimensions, and ensure data integrity and accuracy in analysis.

Part 5

```sql
-- create database
DROP DATABASE IF EXISTS InspectionsData;
CREATE DATABASE InspectionsData;
USE InspectionsData;

-- create table
DROP TABLE IF EXISTS inspections;
CREATE TABLE inspections (
    INSPECTION_ID INT PRIMARY KEY,
    PUBLIC_HOUSING_AGENCY_NAME VARCHAR(255),
    COST_OF_INSPECTION_IN_DOLLARS DECIMAL(10, 2),
    INSPECTED_DEVELOPMENT_NAME VARCHAR(255),
    INSPECTED_DEVELOPMENT_ADDRESS VARCHAR(255),
    INSPECTED_DEVELOPMENT_CITY VARCHAR(100),
    INSPECTED_DEVELOPMENT_STATE CHAR(2),
    INSPECTION_DATE DATE,
    INSPECTION_SCORE INT
);
```

```sql
-- load csv file
SET GLOBAL LOCAL_INFILE=1;
LOAD DATA LOCAL INFILE '/Users/sunyi/Desktop/ALY6030/Module3_Public Housing Inspections Star Schema/public_housing_inspection_data.csv'
INTO TABLE `inspections`
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(INSPECTION_ID, PUBLIC_HOUSING_AGENCY_NAME, COST_OF_INSPECTION_IN_DOLLARS,
 INSPECTED_DEVELOPMENT_NAME, INSPECTED_DEVELOPMENT_ADDRESS,
 INSPECTED_DEVELOPMENT_CITY, INSPECTED_DEVELOPMENT_STATE,
 @INSPECTION_DATE, INSPECTION_SCORE)
SET INSPECTION_DATE = STR_TO_DATE(@INSPECTION_DATE, '%m/%d/%Y');


-- part 5
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 inspections
)


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;
```

	Time	Action	Response
✓ 8	16:10:15	WITH InspectionRanked AS (SELECT PUBLIC_HOUSING_AGENCY_NAME AS PHA_NAME, INSPECTION...	242 row(s) returned

Through the above code, I modified the date format, utilized the LAG function, and ultimately filtered the PHAs with increased costs, ensuring each PHA is listed only once.

Reference

Gosavi, R. (2024, May 8). *Understanding and implementing slowly changing dimensions (SCD) in SQL*. Medium. https://medium.com/@rahulgosavi.94/understanding-and-implementing-slowly-changing-dimensions-scd-in-sql-ec9fbbb73075