Assignment 3 — Public Housing Inspections Star Schema

Heejae Roh (ID 002795339)

College of Professional Studies, Northeastern University

ALY 6030: Data Warehousing & SQL

Professor Venkata Duvvuri

December 6, 2023

Table of Contents

Introduction

In this analysis, I will distinguish between fact columns and dimension columns based on 'public_housing_inspection_data'. During the process, we will learn about the characteristics of facts and the dimensions that explain them. Finally, we will use window functions to construct a complex query capable of answering questions that stakeholders are curious about.

Analysis

1. About Fact

1) How many facts are there in this dataset?

- There are two facts which are COST_OF_INSPECTION_IN_DOLLARS, and INSPECTION_SCORE.

2) Which facts do you identify?

- COST_OF_INSPECTION_IN_DOLLARS shows the dollar amount of inspection cost of each INSPECTION_ID.
- INSPECTION_SCORE shows the score of inspection of each INSPECTION_ID.

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

- COST_OF_INSPECTION_IN_DOLLARS is addictive fact column. The most flexible and useful facts are fully additive; additive measures can be summed across any of the dimensions associated with the fact table (kimballgroup, n.d.). Average of COST_OF_INSPECTION_IN_DOLLARS is 25122.79 dollar.
- INSPECTION_SCORE is non-addictive fact column. It's similar to ratio of total score. Some measures are completely non-additive, such as ratios. A good approach for non-additive facts is, where possible, to store the fully additive components of

the non-additive measure and sum these components into the final answer set before calculating the final non-additive fact (kimballgroup, n.d.). Average of INSPECTION_SCORE is 83.37.

2. About Dimensions

Fact tables and dimension tables play different but important roles in a data warehouse. Fact tables contain numerical data, while dimension tables provide context and background information (Simplilearn, 2023). Date data is often stored in a separate date dimension (instead of a Date column in a fact table) (ibm, 2023).

1) How many dimensions are there in this dataset?

There are six dimensions in this dataset.

2) Which dimensions do you identify?

There are PUBLIC_HOUSING_AGENCY_NAME, INSPECTED_DEVELOPMENT_NAME, INSPECTED_DEVELOPMENT_ADDRESS, INSPECTED_DEVELOPMENT_CITY, INSPECTED_DEVELOPMENT_STATE, and INSPECTION_DATE.

3. Question about inspection level and cost

1) Senior management is interested in viewing the facts identified above, at both **the inspection level**, as well as a **periodic summary of inspection costs** for **each month**. 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?

- For inspection level: If the inspection level is based on INSPECTION SCORE, I will start the investigation first based on the INSPECTION_SCORE fact. This is because the inspection level can be obtained by determining a section based on INSPECTION_SCORE.
- For **periodic summary of inspection costs** for **each month**: Using the COST_OF_INSPECTION_IN_DOLLARS fact, I will use INSPECTION_DATE to group data by month and look at periodic details.

4. **Question about slowly changing dimensions**

1) Senior Management is also concerned with changes in the **names and addresses of the public housing agency names** since they tend to get merged with other agencies **on a frequent basis**.

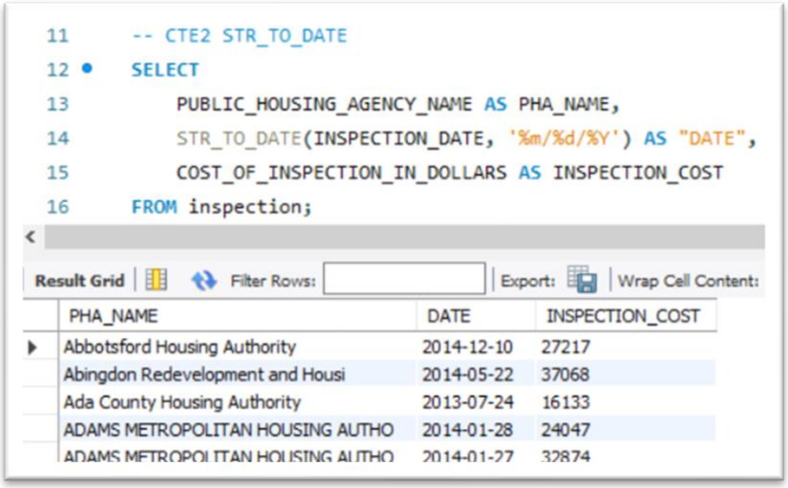
   Based on this context, how would handle this slowly changing dimension? Select from types 0,1,2, or 3 from the Kimball reading. Justify your answer.

- I want to manage SCD (slowly changing dimension) as type 3. Tables will be managed separately by CURRENT_NAME, OLD_NAME, CURRENT_ADDRESS, and OLD_ADDRESS. This is because previous company information may be needed and can be traced when it is necessary to find the person responsible for INSPECTION. Type 3 – Previous Value column: Track change to a specific attribute, add a column to show the previous value, which is updated as further changes occur (Whiteley, 2014).

5. Address the most recent and second resent scenario.

1) Finally, Senior Management is interested in a subset of this data, for only those PHAs that saw an increase in the $$ cost of performing an inspection in their jurisdiction. Since none of them are SQL programmers, they've asked your help in performing this analysis by providing a file as your final deliverable with the following columns:
Note that MR stands for "most recent":
PHA_NAME,
MR_INSPECTION_DATE,
MR_INSPECTION_COST,
SECOND_MR_INSPECTION_DATE,
SECOND_MR_INSPECTION_COST,
CHANGE_IN_COST
PERCENT_CHANGE_IN_COST

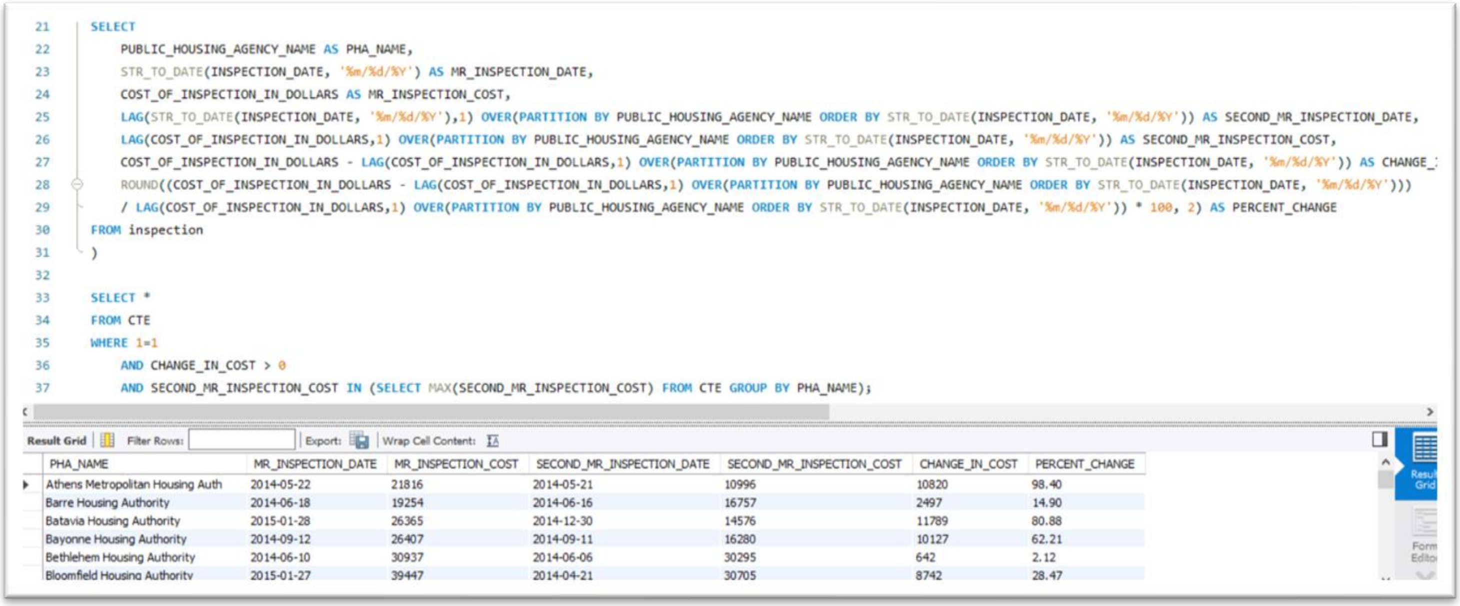
2) Management has asked that you perform this function using lead or lag functions in SQL. However, they're concerned that the files when imported into MySQL Workbench may not properly refer to dates using the correct format. If that is the case, they've asked you to investigate how best to convert dates from TEXT to Date format so that the lead/lag functions work as expected.



```
-- CTE2 STR_TO_DATE
SELECT
    PUBLIC_HOUSING_AGENCY_NAME AS PHA_NAME,
    STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y') AS "DATE",
    COST_OF_INSPECTION_IN_DOLLARS AS INSPECTION_COST
FROM inspection;
```

PHA_NAME	DATE	INSPECTION_COST
Abbotsford Housing Authority	2014-12-10	27217
Abingdon Redevelopment and Housi	2014-05-22	37068
Ada County Housing Authority	2013-07-24	16133
ADAMS METROPOLITAN HOUSING AUTHO	2014-01-28	24047
ADAMS METROPOLITAN HOUSING AUTHO	2014-01-27	32874

3) They’ve also asked that you filter your dataset to only those PHAs that saw an increase in $$ cost, and that you only list the PHA once with no duplicates to avoid noisy data. Naturally, this would also require you to filter out PHAs that only performed one inspection, so they’ve asked you to remove those as well.



```sql
SELECT
    PUBLIC_HOUSING_AGENCY_NAME AS PHA_NAME,
    STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y') AS MR_INSPECTION_DATE,
    COST_OF_INSPECTION_IN_DOLLARS AS MR_INSPECTION_COST,
    LAG(STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y'),1) OVER(PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y')) AS SECOND_MR_INSPECTION_DATE,
    LAG(COST_OF_INSPECTION_IN_DOLLARS,1) OVER(PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y')) AS SECOND_MR_INSPECTION_COST,
    COST_OF_INSPECTION_IN_DOLLARS - LAG(COST_OF_INSPECTION_IN_DOLLARS,1) OVER(PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y')) AS CHANGE_
    ROUND((COST_OF_INSPECTION_IN_DOLLARS - LAG(COST_OF_INSPECTION_IN_DOLLARS,1) OVER(PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y')))
    / LAG(COST_OF_INSPECTION_IN_DOLLARS,1) OVER(PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y')) * 100, 2) AS PERCENT_CHANGE
FROM inspection
)

SELECT *
FROM CTE
WHERE 1=1
    AND CHANGE_IN_COST > 0
    AND SECOND_MR_INSPECTION_COST IN (SELECT MAX(SECOND_MR_INSPECTION_COST) FROM CTE GROUP BY PHA_NAME);
```

PHA_NAME	MR_INSPECTION_DATE	MR_INSPECTION_COST	SECOND_MR_INSPECTION_DATE	SECOND_MR_INSPECTION_COST	CHANGE_IN_COST	PERCENT_CHANGE
Athens Metropolitan Housing Auth	2014-05-22	21816	2014-05-21	10996	10820	98.40
Barre Housing Authority	2014-06-18	19254	2014-06-16	16757	2497	14.90
Batavia Housing Authority	2015-01-28	26365	2014-12-30	14576	11789	80.88
Bayonne Housing Authority	2014-09-12	26407	2014-09-11	16280	10127	62.21
Bethlehem Housing Authority	2014-06-10	30937	2014-06-06	30295	642	2.12
Bloomfield Housing Authority	2015-01-27	39447	2014-04-21	30705	8742	28.47

Conclusion

In this analysis, we have effectively differentiated between fact columns and dimension columns within 'public_housing_inspection_data', uncovering the distinct characteristics of each. Through strategic approaches, including SQL window functions, I addressed key managerial concerns, offering insights into inspection costs and agency changes. This comprehensive examination not only clarifies the dataset's structure but also provides actionable intelligence for informed decision-making.

References

Kimball Group. (n.d.). Additive, Semi-Additive, Non-Additive Fact. Retrieved from https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/additive-semi-additive-non-additive-fact/

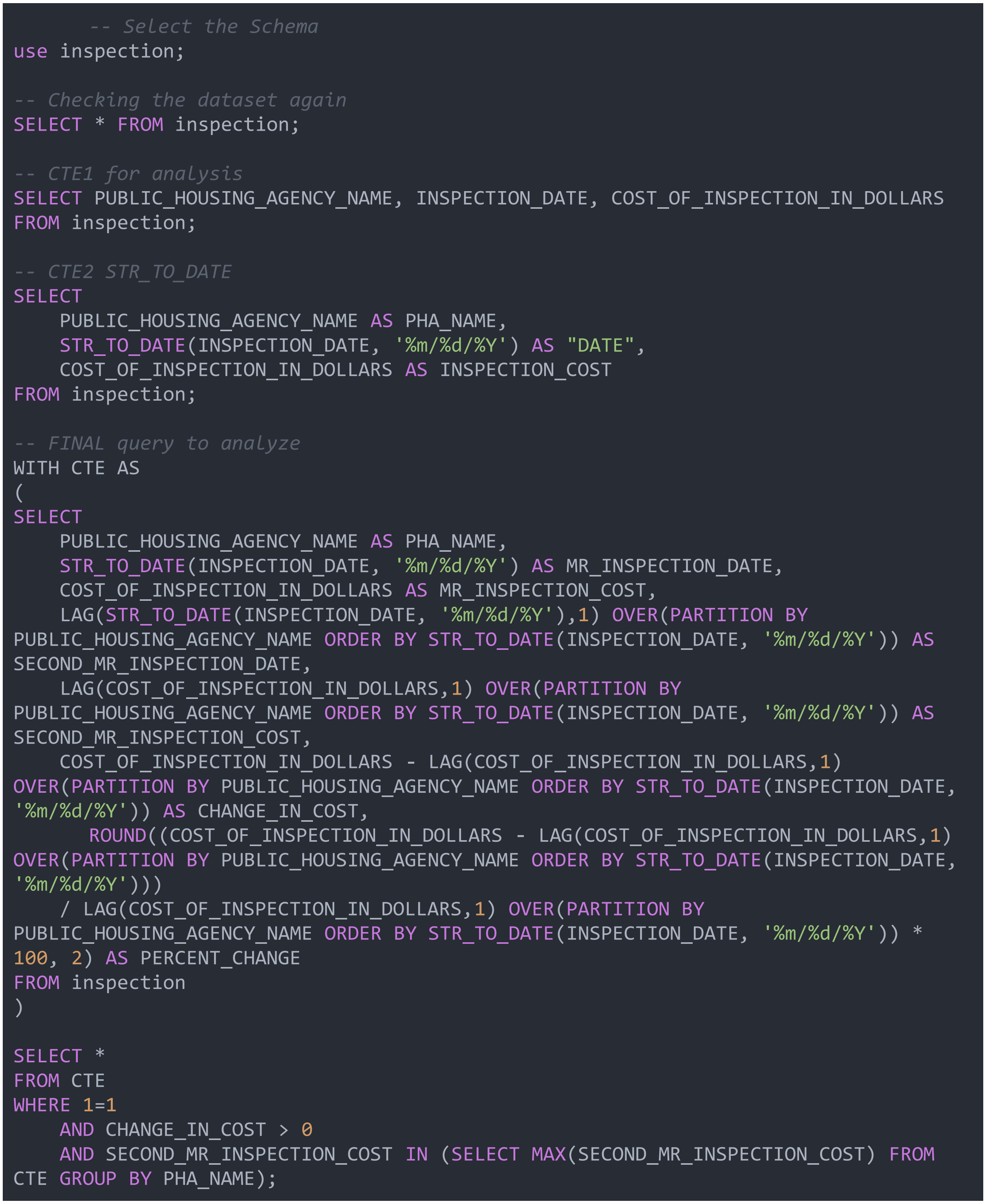
Simplilearn. (2023, June 6). Fact Table vs. Dimension Table. Retrieved from https://www.simplilearn.com/fact-table-vs-dimension-table-article

IBM. (2023). Identify dimensions. Retrieved from https://www.ibm.com/docs/en/ida/9.1?topic=phase-step-identify-dimensions

Whiteley, S. (2014, March 14). Introduction to Slowly Changing Dimensions (SCD) Types. Adatis. Retrieved from https://adatis.co.uk/introduction-to-slowly-changing-dimensions-scd-types/

Bluefoot. (2011, March 5). How to convert a string to date in MySQL. Stack Overflow. Retrieved from https://stackoverflow.com/questions/5201383/how-to-convert-a-string-to-date-in-mysql

SQL query



```
       -- Select the Schema
use inspection;

-- Checking the dataset again
SELECT * FROM inspection;

-- CTE1 for analysis
SELECT PUBLIC_HOUSING_AGENCY_NAME, INSPECTION_DATE, COST_OF_INSPECTION_IN_DOLLARS
FROM inspection;

-- CTE2 STR_TO_DATE
SELECT
    PUBLIC_HOUSING_AGENCY_NAME AS PHA_NAME,
    STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y') AS "DATE",
    COST_OF_INSPECTION_IN_DOLLARS AS INSPECTION_COST
FROM inspection;

-- FINAL query to analyze
WITH CTE AS
(
SELECT
    PUBLIC_HOUSING_AGENCY_NAME AS PHA_NAME,
    STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y') AS MR_INSPECTION_DATE,
    COST_OF_INSPECTION_IN_DOLLARS AS MR_INSPECTION_COST,
    LAG(STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y'),1) OVER(PARTITION BY
PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y')) AS
SECOND_MR_INSPECTION_DATE,
    LAG(COST_OF_INSPECTION_IN_DOLLARS,1) OVER(PARTITION BY
PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y')) AS
SECOND_MR_INSPECTION_COST,
    COST_OF_INSPECTION_IN_DOLLARS - LAG(COST_OF_INSPECTION_IN_DOLLARS,1)
OVER(PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE,
'%m/%d/%Y')) AS CHANGE_IN_COST,
       ROUND((COST_OF_INSPECTION_IN_DOLLARS - LAG(COST_OF_INSPECTION_IN_DOLLARS,1)
OVER(PARTITION BY PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE,
'%m/%d/%Y')))
    / LAG(COST_OF_INSPECTION_IN_DOLLARS,1) OVER(PARTITION BY
PUBLIC_HOUSING_AGENCY_NAME ORDER BY STR_TO_DATE(INSPECTION_DATE, '%m/%d/%Y')) *
100, 2) AS PERCENT_CHANGE
FROM inspection
)

SELECT *
FROM CTE
WHERE 1=1
    AND CHANGE_IN_COST > 0
    AND SECOND_MR_INSPECTION_COST IN (SELECT MAX(SECOND_MR_INSPECTION_COST) FROM
CTE GROUP BY PHA_NAME);
```