Northeastern University
College *of* Professional Studies

Public Housing Inspections Star Schema

MIN ZENG

ALY6030 Data Warehousing & SQL

Instructor: Adam Jones

11/20/2024

Introduction

This assignment focuses on analyzing and designing a star schema using the **public_housing_inspection_data.csv** dataset, which contains information on inspections of public housing developments. The dataset provides details such as inspection costs, dates, scores, and information about the inspected developments, including their locations and the housing agencies responsible. The goal is to answer key business questions by identifying facts and dimensions, creating fact tables, and designing a strategy to handle changes in agency names and addresses. Additionally, senior management has requested an SQL-based analysis to identify public housing agencies (PHAs) with increasing inspection costs, leveraging advanced SQL functions like lead and lag.

Analysis

1. Answer the questions below.

- How many facts are there in this dataset?

  There are two primary facts in the dataset.

- Which facts do you identify?
  - Cost of Inspection (COST_OF_INSPECTION_IN_DOLLARS): The monetary cost incurred for conducting an inspection.
  - Inspection Score (INSPECTION_SCORE): A quantitative score representing the inspection outcome.

- For the facts that you identify, what type of facts are they?
  - Cost of Inspection: This is an additive fact because it can be summed up across inspections, months, or agencies to provide total costs.
  - Inspection Score: This is a non-additive fact because it cannot be summed meaningfully across dimensions but can be averaged or aggregated for meaningful insights.

2. Answer the questions below.

- How many dimensions are there in this dataset?

  There are five main dimensions in the dataset.

- Which dimensions do you identify?
  - Public Housing Agency (PUBLIC_HOUSING_AGENCY_NAME): Identifies the agency responsible for the inspections.
  - Date (INSPECTION_DATE): Represents the date of inspection.
  - Development Name (INSPECTED_DEVELOPMENT_NAME): Identifies the specific development being inspected.

- Location (INSPECTED_DEVELOPMENT_ADDRESS, INSPECTED_DEVELOPMENT_CITY, INSPECTED_DEVELOPMENT_STATE): Provides geographical information about the development.
- Inspection ID (INSPECTION_ID): Acts as a unique identifier for each inspection event.

3. Answer the question below.

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?

To organize this data, I would implement two types of fact tables: a **Transaction Fact Table** to record individual inspection events, including costs and scores at the most detailed level, and a **Periodic Snapshot Fact Table** to aggregate monthly inspection costs for each agency. The transaction fact table offers granular data for in-depth analysis, while the periodic snapshot fact table allows senior management to observe and evaluate inspection costs over time, facilitating the identification of monthly trends and patterns.

4. Answer the question below.

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.

For changes in the names and addresses of public housing agencies, I recommend using Type 2 Slowly Changing Dimension (SCD). This approach is effective because it preserves history, allowing a complete historical record of changes, which is essential for tracking mergers, address updates, and restructuring over time (Kimball & Ross, 2013). Type 2 also provides granularity in analysis by adding a new record for each change, enabling accurate tracking of historical and current agency performance. Each record typically includes a start date, end date, and a flag to indicate the active record (Kimball & Ross, 2013).

5. Address the scenario below.

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

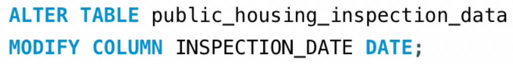
Management has asked that you perform this function using lead or lag functions in SQL.

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.

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.

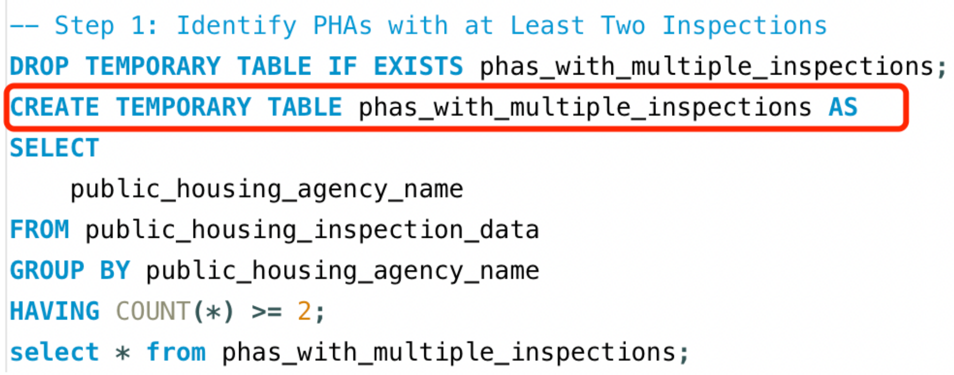
1) Convert Dates



```
ALTER TABLE public_housing_inspection_data
MODIFY COLUMN INSPECTION_DATE DATE;
```

2) Identify PHAs with at Least Two Inspections

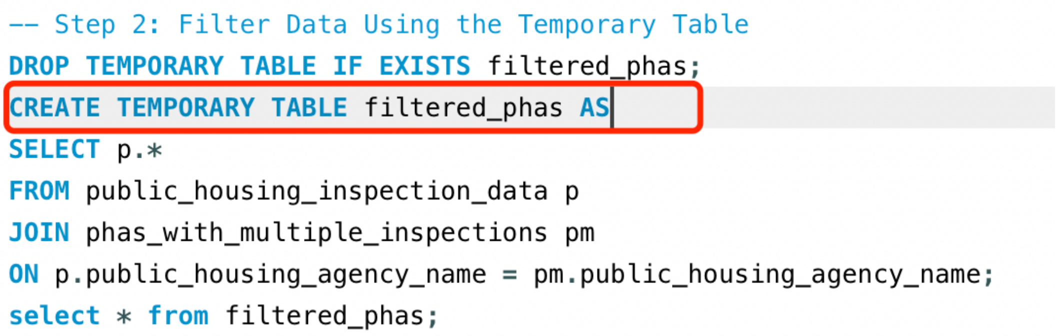
I created table called **‘phas_with_multiple_inspections’** to just show PHAs with at Least Two Inspections.



```
-- Step 1: Identify PHAs with at Least Two Inspections
DROP TEMPORARY TABLE IF EXISTS phas_with_multiple_inspections;
CREATE TEMPORARY TABLE phas_with_multiple_inspections AS
SELECT
    public_housing_agency_name
FROM public_housing_inspection_data
GROUP BY public_housing_agency_name
HAVING COUNT(*) >= 2;
select * from phas_with_multiple_inspections;
```

3) Use JOIN to create my filtered data

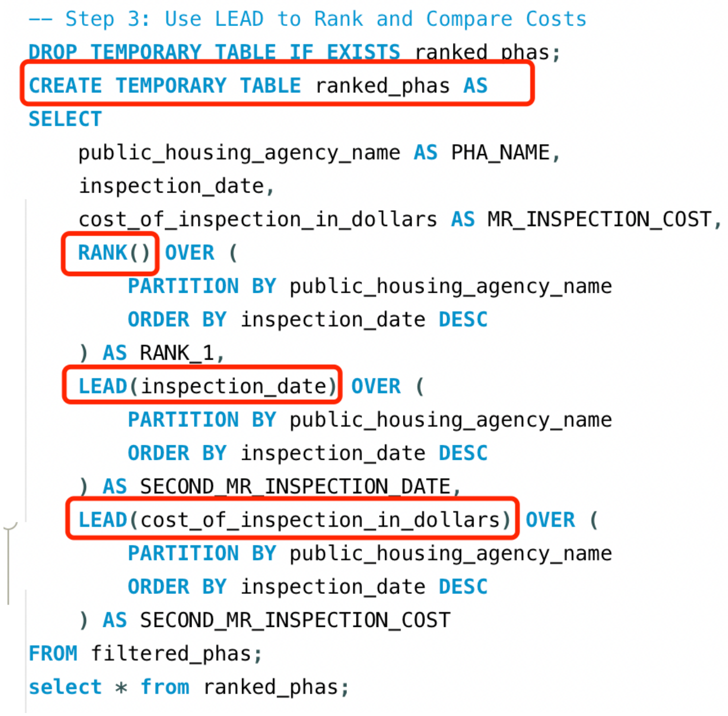
I created table named ‘filtered_phas’ to filter the date that I will use in the next step.



```
-- Step 2: Filter Data Using the Temporary Table
DROP TEMPORARY TABLE IF EXISTS filtered_phas;
CREATE TEMPORARY TABLE filtered_phas AS
SELECT p.*
FROM public_housing_inspection_data p
JOIN phas_with_multiple_inspections pm
ON p.public_housing_agency_name = pm.public_housing_agency_name;
select * from filtered_phas;
```

4) Use LEAD to Rank and Compare Costs

I used rank(), lead to sort the inspection_date and second_mr_inspection_date, second_mr_inspection_cost.

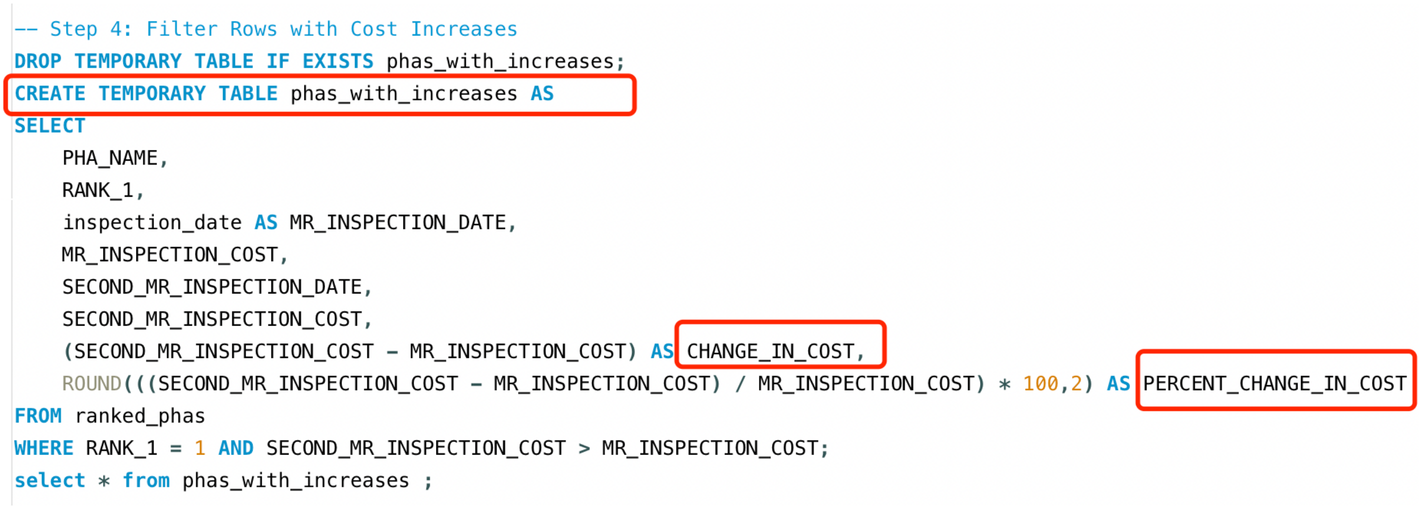


```
-- Step 3: Use LEAD to Rank and Compare Costs
DROP TEMPORARY TABLE IF EXISTS ranked_phas;
CREATE TEMPORARY TABLE ranked_phas AS
SELECT
    public_housing_agency_name AS PHA_NAME,
    inspection_date,
    cost_of_inspection_in_dollars AS MR_INSPECTION_COST,
    RANK() OVER (
        PARTITION BY public_housing_agency_name
        ORDER BY inspection_date DESC
    ) AS RANK_1,
    LEAD(inspection_date) OVER (
        PARTITION BY public_housing_agency_name
        ORDER BY inspection_date DESC
    ) AS SECOND_MR_INSPECTION_DATE,
    LEAD(cost_of_inspection_in_dollars) OVER (
        PARTITION BY public_housing_agency_name
        ORDER BY inspection_date DESC
    ) AS SECOND_MR_INSPECTION_COST
FROM filtered_phas;
select * from ranked_phas;
```

5) Filter Rows with Cost Increases

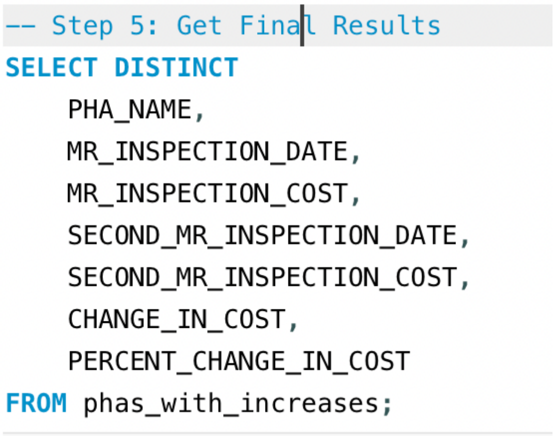
Fisrt, I compared (SECOND_MR_INSPECTION_COST - MR_INSPECTION_COST) AS CHANGE_IN_COST and ROUND(((SECOND_MR_INSPECTION_COST - MR_INSPECTION_COST) / MR_INSPECTION_COST) * 100,2) AS PERCENT_CHANGE_IN_COST;

Then, I just keep the rank =1 row and SECOND_MR_INSPECTION_COST > MR_INSPECTION_COST.

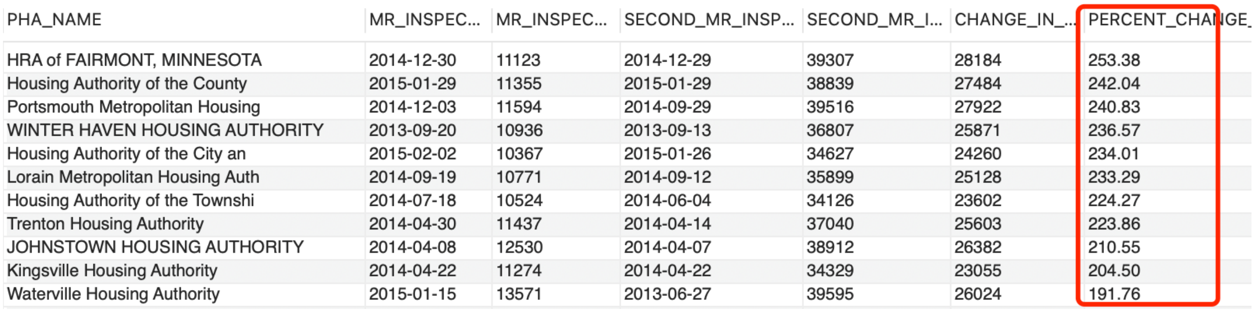


```
-- Step 4: Filter Rows with Cost Increases
DROP TEMPORARY TABLE IF EXISTS phas_with_increases;
CREATE TEMPORARY TABLE phas_with_increases AS
SELECT
    PHA_NAME,
    RANK_1,
    inspection_date AS MR_INSPECTION_DATE,
    MR_INSPECTION_COST,
    SECOND_MR_INSPECTION_DATE,
    SECOND_MR_INSPECTION_COST,
    (SECOND_MR_INSPECTION_COST - MR_INSPECTION_COST) AS CHANGE_IN_COST,
    ROUND(((SECOND_MR_INSPECTION_COST - MR_INSPECTION_COST) / MR_INSPECTION_COST) * 100,2) AS PERCENT_CHANGE_IN_COST
FROM ranked_phas
WHERE RANK_1 = 1 AND SECOND_MR_INSPECTION_COST > MR_INSPECTION_COST;
select * from phas_with_increases ;
```

6) Get Final Results



```
-- Step 5: Get Final Results
SELECT DISTINCT
     PHA_NAME,
     MR_INSPECTION_DATE,
     MR_INSPECTION_COST,
     SECOND_MR_INSPECTION_DATE,
     SECOND_MR_INSPECTION_COST,
     CHANGE_IN_COST,
     PERCENT_CHANGE_IN_COST
FROM phas_with_increases;
```



PHA_NAME	MR_INSPEC...	MR_INSPEC...	SECOND_MR_INSP...	SECOND_MR_I...	CHANGE_IN_...	PERCENT_CHANGE_
HRA of FAIRMONT, MINNESOTA	2014-12-30	11123	2014-12-29	39307	28184	253.38
Housing Authority of the County	2015-01-29	11355	2015-01-29	38839	27484	242.04
Portsmouth Metropolitan Housing	2014-12-03	11594	2014-09-29	39516	27922	240.83
WINTER HAVEN HOUSING AUTHORITY	2013-09-20	10936	2013-09-13	36807	25871	236.57
Housing Authority of the City an	2015-02-02	10367	2015-01-26	34627	24260	234.01
Lorain Metropolitan Housing Auth	2014-09-19	10771	2014-09-12	35899	25128	233.29
Housing Authority of the Townshi	2014-07-18	10524	2014-06-04	34126	23602	224.27
Trenton Housing Authority	2014-04-30	11437	2014-04-14	37040	25603	223.86
JOHNSTOWN HOUSING AUTHORITY	2014-04-08	12530	2014-04-07	38912	26382	210.55
Kingsville Housing Authority	2014-04-22	11274	2014-04-22	34329	23055	204.50
Waterville Housing Authority	2015-01-15	13571	2013-06-27	39595	26024	191.76

The result above shows the filtered results with percentage change between the second most recent and the most recent inspection costs.

The red box highlights the percentage changes, and we can observe that most rows exceed 200% (except the last one, which is 191.76%).

Key Insights for Senior Management

1. Significant Cost Increases:

   - Several Public Housing Agencies (PHAs) experienced drastic increases in inspection costs, with some exceeding 200%. The highest increase was 253.38% (HRA of FAIRMONT, MINNESOTA).

2. Operational and Budgetary Risks:
   - These cost spikes could strain budgets and indicate operational inefficiencies or external factors like regulatory changes or rising labor costs.

3. Geographic Trends:
   - There may be regional patterns driving these increases. For example, multiple PHAs from certain states showed significant changes.

4. Actionable Recommendations:
   - Investigate top PHAs with extreme cost changes to identify root causes.
   - Optimize inspection processes to control costs and avoid future spikes.
   - Monitor trends regionally to proactively address recurring issues.

By addressing these cost anomalies, we can improve budget planning and reduce operational risks.

Reference

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd ed.). Wiley.