**CS434 – Data Base Theory and Design**

**Project #5**

**Team Database Application (TDA): Part 5 – Querying and Schema Tuning**

**Team**

Lipika Baniya | 800794205 | [lbaniya@siue.edu](mailto:lbaniya@siue.edu)

The domain I would like to manage with the TDA is **Washington DC Crime Datasets 2024** by the District of Columbia Metropolitan Police Department (MPD).

**General Nature of application**

The main goal of an Entity Relationship Diagram (ER Diagram) is to explain the relationship between entities; it is a structural design of the database. Through the help of specialized symbols, it helps to define the relationship between entities. It is based on three main principles entities, attributes and relationships, these help to design the database that would be required before implementing the database. It is a systematic process to design a database as it would require analyzing all requirements.

**About Data**

Washington, D.C. has been facing significant challenges in ensuring public safety due to the varying and growing crime rates in different neighborhoods and time periods. It is important for law enforcement agencies to understand when and where crimes occur so that it can respond efficiently and allocate limited resources wisely. Imagine a robust database system that is designed to handle this task effectively, because without a data-driven approach and structured database, policing efforts may remain reactive, which would result in delays or gaps in coverage in high-risk areas. This database includes various entities, each representing a key component of crime data management.

In the previous project, I created the schema and inserted data into PostgreSQL based on the ER diagram. I used PostgreSQL to CREATE TABLE command and inserted data values and types for each entity’s attributes. I have attached screenshots of the entire dataset and separate files for each table relation which were included in Project 4.

**ER Diagram**

**A diagram of a crime investigation

AI-generated content may be incorrect.**

1. **Queries**
   1. **Neighborhood Clusters with highest number of reported theft incidents**

*Question: Which neighborhood clusters have the highest number of reported theft incidents?*

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1: Neighborhood Clusters with highest number of reported theft incidents

Figure 1 shows the result of the query, which lists neighborhood clusters along with the number of theft-related incidents.

**Explanation**

To identify which neighborhood clusters experienced the most theft incidents, I used the following tables

* Crime\_incident: this table is logbook of all crimes, including the offense\_id
* Offense: provides the offense type (e.g. theft, assault)
* Crime\_location links each incident to the neighborhood cluster
* Location: includes the detailed geographic attributes, including neighborhood cluster

The query:

* WHERE clause with LIKE ‘%theft%’ filters only those crimes that are theft related.
* Neighborhood\_cluster is grouped to count the number of thefts in each.
* Orders the result in descending order to show the clusters with the highest theft counts first.
  1. **Major time for crime to occur**

*Question: At what time during the day does the crime occur the most?*

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2: Major time for crime to occur

Figure 2 shows the result of the query, which displays the major time during the day does the crime occur the most.

**Explanation**

To find the major time for the crime to occur, I used the following tables

* Crime\_incident: this table includes all crime details, including shift.

The query:

* The shift is grouped to show at what time the crime occurs the most.
  1. **Top Crime Type in Each Block Group**

*Question: What is the major crime in each block group?*

*A screenshot of a computer

AI-generated content may be incorrect.*

Figure 3: Top Crime Type in Each Block Group

Figure 3 displays the major crime in each block group.

**Explanation**

**To find the major crime in each block group, the following tables were used:**

* Crime\_incident: the main record of incidents
* Offense: maps offense\_id to a name like “THEFT F/AUTO”.
* Crime\_location: links crimes to a location.
* Block\_group: gives geographic blocks via x\_block, y\_block.

The query:

* WITH offense\_counts AS is used to create Common Table Expression (CTE) to get temporary result named offense\_counts.
* SELECT is used to find the Type of offense, X and Y coordinates of block\_group and count of how many such offenses occurred.
* Row\_number() for each (x\_block, y\_block) pair, ranks the most frequent offense in each block.
* Group\_by is used to group offense\_type and block\_group to count number of times each offense happens in each block.
* The final selects query filters only the top-ranked offense type per block group.
  1. **Incidents with “Gun” involved**

*Question: How many incidents involved a "Gun" as the method?*

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 4: Incidents with "Gun" involved

Figure 4 shows the number of incidents that involved “gun” as the method.

**Explanation**

To find incidents that involved “Gun” as the method, I used the following the tables:

* Crime\_incident: this table includes all crime details, including shift.
* Method: this table includes method details, including method type (e.g. “KNIFE”, “GUN”)

The query:

* JOIN is used to join crime\_incident to the method table using method\_id.
* WHERE ILIKE “%gun%” is used to find methods that use gun.
* GROUP BY is used to count the number of incidents that involved gun.
* ORDER BY is used to display in descending order.
  1. **Crime Reported Late**

*Question: How many crimes were reported more than 3 days after they occurred?*

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5: crimes that were reported late

Figure 5 shows the number of crimes reported after 3 days than they occurred.

**Explanation**

To find the number of reports that were reported late, I used the queries:

* Crime\_incident:this table includes all crime details, including report\_date, start\_date.

The query:

* WHERE is used to subtract report\_date – start\_date to find interval of 3 days.

1. **Data Modification Queries**
   1. **Insert**

For the insert statement we are inputting into all the six table using dummy data (will be deleted during the delete command operation).

**A screenshot of a computer code

AI-generated content may be incorrect.**

Figure 6: Insert Query

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7: New tuple added to the DB

* 1. **Update: Updating Shift**

**A screenshot of a computer

AI-generated content may be incorrect.**

Figure 8: Updating shift

A screenshot of a phone

AI-generated content may be incorrect.

Figure 9: Updated record

* 1. **Delete**

Deleting the recently added dummy record from all the tables.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 10: Delete the records

1. **Create View**
   1. **Creating the view table**

A virtual table using existing schema has been created using create view as follows:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 11: Query for CREATE VIEW

* 1. **The Crime View Table**

**A screenshot of a computer

AI-generated content may be incorrect.**

Figure 12: Glimpse of the new Crime\_view Table

* 1. **Updating the View Table**

**A screenshot of a computer

AI-generated content may be incorrect.**

Figure 13: Updating View Table

The update view table query does not work and returned error; This is because the updated view cannot be inserted as it involves joins, and these are typically not updatable because the database cannot determine how to properly distribute new data across the base tables.

1. **Indexing**
   1. **Without indexing**

For the below query, the query time was 163 msec.

**A screenshot of a computer program

AI-generated content may be incorrect.**

* 1. **Creating index**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

* 1. **With indexing**

With indexing, the query time dropped to 158 msec

**A screenshot of a computer code

AI-generated content may be incorrect.**