

# Crime Insights : A Data-Driven Approach to Public Safety

Rajashree Shanmuganathan  
*rajashre*  
*rajashre@buffalo.edu*

Sindhura Velmurugan  
*svelmuru*  
*svelmuru@buffalo.edu*

Sadhana Saravanan  
*sadhanas*  
*sadhanas@buffalo.edu*

## I. PROBLEM STATEMENT

The increasing volume and complexity of crime data from 2020 to the present pose significant challenges for law enforcement agencies, policymakers, and researchers in analyzing trends, identifying patterns, and making data-driven decisions. While crime data is available in formats like Excel, these tools are insufficient for handling large datasets, performing complex queries, and enabling real-time analysis. This project aims to create a robust, scalable database system to store, manage, and analyze crime data efficiently. The database will enable users to answer critical questions such as:

- What are the trends and patterns in crime over time?
- Which areas or neighborhoods are most affected by specific types of crime?
- How do socioeconomic factors correlate with crime rates?
- What are the seasonal or temporal variations in criminal activity?

A database is essential for this project rather than an Excel file because a database is a superior choice over an Excel file for crime data management due to its scalability, data integrity, efficiency, multi-user access, and security. While Excel struggles with large datasets and can become sluggish or prone to crashes, a database efficiently handles millions of records with structured indexing and optimized storage. Databases enforce constraints, ensuring data accuracy and reducing redundancies that are common in spreadsheets. Additionally, SQL enables powerful querying capabilities, allowing users to filter, sort, and analyze data quickly—tasks that are manual and inefficient in Excel. Unlike Excel, which lacks proper concurrency control, a database allows multiple users to access and update records simultaneously without conflicts. Furthermore, databases provide robust security features, including access controls and user permissions, ensuring that sensitive crime data remains protected from unauthorized access or accidental modifications.

### A. Background of the Problem

Crime is a major societal issue that affects public safety, economic stability, and community well-being. Law enforcement agencies and policymakers depend on accurate and timely crime data to detect patterns, allocate resources, and implement effective crime prevention strategies. However, traditional crime data management methods, such as Excel spreadsheets or paper-based records, pose significant challenges. These include data redundancy, human errors, difficulty in handling large datasets, and the inability to

perform complex queries efficiently. As crime rates fluctuate and urban populations grow, the need for a more efficient and scalable system to manage and analyze crime data becomes increasingly critical.

One of the major challenges faced by law enforcement is the inability to quickly retrieve and analyze crime trends. Manual data handling can result in delayed responses to emerging threats and inefficient deployment of police resources. Additionally, the lack of integration between different agencies leads to fragmented data, making it difficult to track repeat offenders or identify crime hotspots. These inefficiencies can hinder proactive policing and result in ineffective crime control strategies. Given the critical role that data plays in crime analysis and prevention, adopting a structured database system is essential for improving decision-making and operational efficiency.

### B. Significance of the Problem

This project addresses the growing need for a centralized, efficient, and accessible system to analyze crime data. By transforming raw data into actionable insights, the database will contribute to:

- Improving public safety through data-driven policing strategies.
- Informing policy decisions to reduce crime rates.
- Enhancing transparency and accountability in law enforcement.
- Supporting academic research on crime and its societal impacts.

### C. Potential Contribution to the Problem

This project aims to develop a database-driven crime analysis system that will significantly enhance data management, retrieval, and analysis capabilities for law enforcement agencies. By transitioning from traditional spreadsheet-based management to a relational database, the project will improve data accuracy, reduce redundancy, and provide advanced analytical capabilities. Law enforcement officers will be able to efficiently filter, sort, and analyze crime reports, enabling them to identify patterns and trends that may otherwise go unnoticed.

The contribution of this project is crucial as it will facilitate real-time crime monitoring, allowing authorities to make data-driven decisions for crime prevention. Predictive analytics can help identify high-risk areas, optimize patrol routes, and allocate resources more effectively, ultimately leading to improved public safety. Furthermore, the system will enable better coordination between law enforcement agencies by providing centralized and structured access to crime data.

Beyond law enforcement, researchers and policymakers can utilize the database to study long-term crime trends, assess the effectiveness of crime prevention strategies, and develop policies that address root causes of crime. In contrast to traditional methods, which are often time-consuming and error-prone, this database-driven approach will provide a scalable, secure, and efficient solution for crime data analysis. By leveraging technology, this project has the potential to significantly enhance crime prevention efforts and contribute to the creation of safer communities.

## II. TARGET USERS

### A. Primary Users

- **Law Enforcement Agencies** : Police departments will use the database to track crime trends, manage cases, and allocate resources effectively.
- **General Public**: Citizens and community organizations can access anonymized data to understand crime in their neighborhoods and advocate for change.
- **Researchers**: Academics and social scientists will use the database to study crime patterns, correlations, and impacts.

### B. Database Admins

- **Data Scientists and Analysts**: Responsible for maintaining the database, performing complex queries, and generating insights.
- **Data Modellers** : Ensuring the database is secure, scalable, and accessible to authorized users.

1) *Real Life Scenario*: A city police department is struggling with a recent spike in burglaries. Using the database, crime analysts can quickly filter data by crime type, location, and time period to identify patterns. They discover that most burglaries occur in specific neighborhoods during late evenings. Based on this insight, the department increases patrols in those areas during peak times, leading to a significant reduction in burglary rates. Meanwhile, policymakers use the database to evaluate the effectiveness of this intervention and allocate funding for additional street lighting in high-risk areas.

## III. ENTITY-RELATIONSHIP DIAGRAM

The ER diagram below represents the structure of the crime database, showcasing the relationships between different entities.

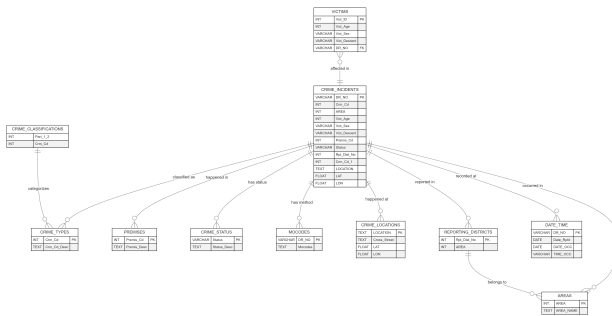


Fig. 1. Entity-Relationship Diagram for Crime Database

The following are the 11 relations present in the ER diagram:

The relationships between different tables in the ER diagram are as follows:

- **Crime\_Incidents** (DR\_NO, Crm\_Cd, AREA, Vict\_Age, Vict\_Sex, Vict\_Descent, Premis\_Cd, Status, Rpt\_Dist\_No, LOCATION, LAT, LON)
  - **One-to-Many**: *Crime\_Incidents* → *Victims* (DR\_NO)
  - **Many-to-One**: *Crime\_Incidents* → *Crime\_Types* (Crm\_Cd)
  - **Many-to-One**: *Crime\_Incidents* → *Premises* (Premis\_Cd)
  - **Many-to-One**: *Crime\_Incidents* → *Reporting\_Districts* (Rpt\_Dist\_No)
  - **Many-to-One**: *Crime\_Incidents* → *Areas* (AREA)
- **Victims** (Vict\_ID, Vict\_Age, Vict\_Sex, Vict\_Descent, DR\_NO (FK))
  - **Many-to-One**: *Victims* → *Crime\_Incidents* (DR\_NO)
- **Crime\_Types** (Crm\_Cd, Crm\_Cd\_Desc)
  - **One-to-Many**: *Crime\_Types* → *Crime\_Incidents* (Crm\_Cd)
- **Premises** (Premis\_Cd, Premis\_Desc)
  - **One-to-Many**: *Premises* → *Crime\_Incidents* (Premis\_Cd)
- **Crime\_Status** (Status, Status\_Desc)
  - **One-to-Many**: *Crime\_Status* → *Crime\_Incidents* (Status)
- **MOCodes** (DR\_NO, MOCodes)
  - **One-to-Many**: *Crime\_Incidents* → *MOCodes* (DR\_NO)
- **Crime\_Locations** (LOCATION, Cross\_Street, LAT, LON)
  - **One-to-Many**: *Crime\_Locations* → *Crime\_Incidents* (LOCATION)
- **Reporting\_Districts** (Rpt\_Dist\_No, AREA (FK))
  - **One-to-Many**: *Reporting\_Districts* → *Crime\_Incidents* (Rpt\_Dist\_No)
- **Date\_Time** (DR\_NO, Date\_Rptd, DATE\_OCC, TIME\_OCC, AREA (FK))
  - **One-to-One**: *Crime\_Incidents* → *Date\_Time* (DR\_NO)
- **Areas** (AREA, AREA\_NAME)
  - **One-to-Many**: *Areas* → *Crime\_Incidents* (AREA)
- **Crime\_Classifications** (Crm\_Cd, Part\_1\_2)
  - **One-to-Many**: *Crime\_Classifications* → *Crime\_Types* (Crm\_Cd)

## IV. FUNCTIONAL DEPENDENCIES (FDs) FOR EACH RELATION

### A. Crime\_Incidents (Primary Key: DR\_NO)

**FDs:**

DR\_NO → Crm\_Cd, AREA, Vict\_Age, Vict\_Sex, Vict\_Descent, Premis\_Cd, Status, Rpt\_Dist\_No, LOCATION, LAT, LON

**BCNF Check:**

Since DR\_NO is a superkey, this relation is in BCNF.

*B. Crime\_Types (Primary Key: Crm\_Cd)*

**FDs:**

$\text{Crm\_Cd} \rightarrow \text{Crm\_Cd\_Desc}$

**BCNF Check:**

Since Crm\_Cd is a superkey, this relation is in BCNF.

*C. Areas (Primary Key: AREA)*

**FDs:**

$\text{AREA} \rightarrow \text{AREA\_NAME}$

**BCNF Check:**

Since AREA is a superkey, this relation is in BCNF.

*D. Victims (Primary Key: Vict\_ID)*

**FDs:**

$\text{Vict\_ID} \rightarrow \text{Vict\_Age}, \text{Vict\_Sex}, \text{Vict\_Descent}, \text{DR\_NO}$

**BCNF Check:**

Since Vict\_ID is a superkey, this relation is in BCNF.

*E. Premises (Primary Key: Premis\_Cd)*

**FDs:**

$\text{Premis\_Cd} \rightarrow \text{Premis\_Desc}$

**BCNF Check:**

Since Premis\_Cd is a superkey, this relation is in BCNF.

*F. Crime\_Status (Primary Key: Status)*

**FDs:**

$\text{Status} \rightarrow \text{Status\_Desc}$

**BCNF Check:**

Since Status is a superkey, this relation is in BCNF.

*G. Mocodes (Primary Key: DR\_NO)*

**FDs:**

$\text{DR\_NO} \rightarrow \text{Mocodes}$

**BCNF Check:**

Since DR\_NO is a superkey, this relation is in BCNF.

*H. Crime\_Locations (Primary Key: LOCATION)*

**FDs:**

$\text{LOCATION} \rightarrow \text{Cross\_Street}, \text{LAT}, \text{LON}$

**BCNF Check:**

Since LOCATION is a superkey, this relation is in BCNF.

*I. Reporting\_Districts (Primary Key: Rpt\_Dist\_No)*

**FDs:**

$\text{Rpt\_Dist\_No} \rightarrow \text{AREA}$

**BCNF Check:**

Since Rpt\_Dist\_No is a superkey, this relation is in BCNF.

*J. Crime\_Classifications (Primary Key: Crm\_Cd)*

**FDs:**

$\text{Crm\_Cd} \rightarrow \text{Part\_1\_2}$

**BCNF Check:**

Since Crm\_Cd is a superkey, this relation is in BCNF.

*K. Date\_Time (Primary Key: DR\_NO)*

**FDs:**

$\text{DR\_NO} \rightarrow \text{Date\_Rptd}, \text{Date\_Occ}, \text{Time\_Occ}$

**BCNF Check:**

Since DR\_NO is a superkey, this relation is in BCNF.

V. PRIMARY KEYS AND FOREIGN KEYS FOR EACH RELATION

*A. Crime\_Incidents*

**Primary Key:** DR\_NO

**Foreign Keys:**

- Crm\_Cd  $\rightarrow$  Crime\_Types (Crime classification)
- AREA  $\rightarrow$  Areas (Crime location)
- Premis\_Cd  $\rightarrow$  Premises (Location type)
- Rpt\_Dist\_No  $\rightarrow$  Reporting\_Districts (District association)

**Justification:** Each crime incident has a unique identifier (DR\_NO). The foreign keys ensure referential integrity by linking crime incidents to crime types, areas, premises, and reporting districts.

*B. Crime\_Types*

**Primary Key:** Crm\_Cd

**Justification:** Each crime type is uniquely identified by its code.

*C. Areas*

**Primary Key:** AREA

**Justification:** Each geographical area has a unique identifier.

*D. Victims*

**Primary Key:** Vict\_ID

**Foreign Keys:**

- DR\_NO  $\rightarrow$  Crime\_Incidents (Linking victims to crime incidents)

**Justification:** Each victim has a unique ID, and the foreign key ensures that each victim is linked to a crime.

*E. Premises*

**Primary Key:** Premis\_Cd

**Justification:** Each type of premise where crimes occur has a unique identifier.

*F. Crime\_Status*

**Primary Key:** Status

**Justification:** Each crime status (e.g., pending, closed) is uniquely identified.

*G. Mocodes*

**Primary Key:** DR\_NO

**Justification:** Each crime (DR\_NO) can have multiple modus operandi codes.

*H. Crime\_Locations*

**Primary Key:** LOCATION

**Justification:** Each crime location is uniquely identified.

### *I. Reporting\_Districts*

**Primary Key:** Rpt\_Dist\_No

**Foreign Keys:**

- AREA → Areas (Each district belongs to an area)

**Justification:** Each reporting district has a unique number, and it belongs to an area.

### *J. Crime\_Classifications*

**Primary Key:** Crm\_Cd

**Justification:** Each crime classification has a unique identifier.

### *K. Date\_Time*

**Primary Key:** DR\_NO

**Justification:** Each crime incident (DR\_NO) has an associated date and time.

## VI. LIST OF RELATIONS AND ATTRIBUTES

### *A. Crime\_Incidents*

**Attribute:** DR\_NO

**Datatype:** VARCHAR(20)

**Description:** Unique identifier for each crime incident

**Null Allowed:** No

**Default Value:** -

**Attribute:** Crm\_Cd

**Datatype:** INT

**Description:** Crime classification code

**Null Allowed:** No

**Default Value:** -

**Attribute:** AREA

**Datatype:** INT

**Description:** Area where the crime happened

**Null Allowed:** No

**Default Value:** -

**Attribute:** Vict\_Age

**Datatype:** INT

**Description:** Age of the victim (if available)

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** Vict\_Sex

**Datatype:** VARCHAR(10)

**Description:** Gender of the victim (M, F, X)

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** Vict\_Descent

**Datatype:** VARCHAR(10)

**Description:** Ethnicity of the victim

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** Premis\_Cd

**Datatype:** INT

**Description:** Code for the premise where crime occurred

**Null Allowed:** No

**Default Value:** -

**Attribute:** Status

**Datatype:** VARCHAR(20)

**Description:** Crime status (e.g., pending, closed)

**Null Allowed:** No

**Default Value:** -

**Attribute:** Rpt\_Dist\_No

**Datatype:** INT

**Description:** Reporting district number

**Null Allowed:** No

**Default Value:** -

**Attribute:** LOCATION

**Datatype:** VARCHAR(255)

**Description:** Description of the crime location

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** LAT

**Datatype:** FLOAT

**Description:** Latitude coordinates

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** LON

**Datatype:** FLOAT

**Description:** Longitude coordinates

**Null Allowed:** Yes

**Default Value:** NULL

#### **Foreign Key Constraints:**

Crm\_Cd → Crime\_Types(Crm\_Cd) (ON DELETE CASCADE)

AREA → Areas(AREA) (ON DELETE SET NULL)

Premis\_Cd → Premises(Premis\_Cd) (ON DELETE SET NULL)

Rpt\_Dist\_No → Reporting\_Districts(Rpt\_Dist\_No) (ON DELETE SET NULL)

### *B. Crime\_Types*

**Attribute:** Crm\_Cd

**Datatype:** INT (PK)

**Description:** Unique identifier for crime type

**Null Allowed:** No

**Default Value:** -

**Attribute:** Crm\_Cd\_Desc

**Datatype:** TEXT

**Description:** Description of the crime type

**Null Allowed:** No

**Default Value:** -

**Foreign Key Constraints:** None

### *C. Victims*

**Attribute:** Vict\_ID

**Datatype:** INT (PK)

**Description:** Unique ID for each victim

**Null Allowed:** No

**Default Value:** -

**Attribute:** Vict\_Age

**Datatype:** INT

**Description:** Age of the victim

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** Vict\_Sex

**Datatype:** VARCHAR(10)

**Description:** Gender of the victim (M, F, X)

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** Vict\_Descent

**Datatype:** VARCHAR(10)

**Description:** Ethnicity of the victim

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** DR\_NO

**Datatype:** VARCHAR(20)

**Description:** Crime incident ID related to the victim

**Null Allowed:** No

**Default Value:** -

**Foreign Key Constraints:**

DR\_NO → Crime\_Incidents(DR\_NO) (ON DELETE CASCADE)

#### *D. Premises*

**Attribute:** Premis\_Cd

**Datatype:** INT (PK)

**Description:** Unique ID for premise type

**Null Allowed:** No

**Default Value:** -

**Attribute:** Premis\_Desc

**Datatype:** TEXT

**Description:** Description of premise type

**Null Allowed:** No

**Default Value:** -

**Foreign Key Constraints:** None

#### *E. Reporting\_Districts*

**Attribute:** Rpt\_Dist\_No

**Datatype:** INT (PK)

**Description:** Reporting district number

**Null Allowed:** No

**Default Value:** -

**Attribute:** AREA

**Datatype:** INT

**Description:** Area ID related to this district

**Null Allowed:** No

**Default Value:** -

**Foreign Key Constraints:**

AREA → Areas(AREA) (ON DELETE CASCADE)

#### *F. Crime\_Status*

**Attribute:** Status

**Datatype:** VARCHAR(20) (PK)

**Description:** Status of the crime (e.g., open, closed)

**Null Allowed:** No

**Default Value:** -

**Attribute:** Status\_Desc

**Datatype:** TEXT

**Description:** Description of crime status

**Null Allowed:** No

**Default Value:** -

**Foreign Key Constraints:** None

#### *G. Crime\_Locations*

**Attribute:** LOCATION

**Datatype:** TEXT (PK)

**Description:** Unique identifier for location

**Null Allowed:** No

**Default Value:** -

**Attribute:** Cross\_Street

**Datatype:** TEXT

**Description:** Cross street information

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** LAT

**Datatype:** FLOAT

**Description:** Latitude coordinates

**Null Allowed:** Yes

**Default Value:** NULL

**Attribute:** LON

**Datatype:** FLOAT

**Description:** Longitude coordinates

**Null Allowed:** Yes

**Default Value:** NULL

**Foreign Key Constraints:** None

### VII. FOREIGN KEY ACTION SUMMARY

#### • ON DELETE CASCADE:

- **Crime\_Incidents** → **Victims**: If a crime incident is deleted, the corresponding victim record is also removed. Each crime record is associated with a victim, and if the crime is removed, victim details should not persist independently.
- **Crime\_Incidents** → **Crime\_Types**: If a crime type is deleted, all crime incidents of that type are also deleted. A crime must always have a valid type. If a type is removed, related crimes should also be removed to maintain consistency.
- **Crime\_Incidents** → **Reporting\_Districts**: If a reporting district is deleted, all crimes reported

in that district are deleted. Crimes are linked to a reporting district, and if the district no longer exists, it makes sense to remove associated incidents rather than leaving them with an invalid reference.

## • ON DELETE SET NULL:

- **Crime\_Incidents** → **Areas**: If an area is deleted, the corresponding area reference in the crime incidents table is set to NULL. This ensures that crime incidents are not deleted but are instead disassociated from the deleted area.
- **Crime\_Incidents** → **Premises**: If a premise is deleted, the corresponding premise reference in the crime incidents table is set to NULL. This ensures that crime incidents are not deleted but are instead disassociated from the deleted premise.

## VIII. INDEXING AND QUERY EXECUTION ANALYSIS

### A. Problematic Query 1: Search Crimes by Area Name and Crime Description

1) *Query Description*: We want to find crimes reported in a particular area (e.g., Hollywood) with specific keywords (e.g., BURGLARY) in the crime description.

2) *Initial Query Performance (Before Optimization)*: Initially, no proper indexes existed on the AREA\_NAME or Crm\_Cd\_Desc, resulting in sequential scans and slower performance.

```

2 < EXPLAIN ANALYZE
3 SELECT ci.dr_no, ct.crm_cd_desc, a.area_name
4 FROM Crime_Incidents ci
5 JOIN Crime_Types ct ON ci.crm_cd = ct.crm_cd
6 JOIN Areas a ON ci.area = a.area
7 WHERE a.area_name = 'Hollywood' AND ct.crm_cd_desc LIKE 'BURGLARY%';
8
9 /* Solution using Indexing */
10 CREATE INDEX idx_area_name ON Areas (AREA_NAME);
11 CREATE INDEX idx_crm_cd_desc ON Crime_Types (Crm_Cd_Desc text_pattern_ops);
12
13 DROP INDEX idx_area_name;

```

Step	Operation	Cost	Rows	Width	Actual Time	Loops
1	Hash Join	(cost=3.91..3458.74 rows=191 width=69)	191	69	0.078..28.055 rows=1074	loops=1
2	Hash Cond	(ci.area = a.area)				
3	Hash Join	(cost=2.64..3444.82 rows=4009 width=41)	4009	41	0.041..26.879 rows=13151	loops=1
4	Hash Cond	(ci.crm_cd = ct.crm_cd)				
5	Seq Scan on crime_incidents ci	(cost=0.00..3096.90 rows=127290 width=18)	127290	18	0.006..12.891 rows=127290	loops=1
6	Hash	(cost=2.59..2.59 rows=4 width=31)	4	31	0.025..0.025 rows=4	loops=1
7	Buckets: 1024 Batches: 1 Memory Usage: 9kB					
8	Seq Scan on crime_types ct	(cost=0.00..2.59 rows=4 width=31)	4	31	0.017 rows=4	loops=1
9	Filter	(crm_cd_desc ~ 'BURGLARY%':text)				
10	Rows Removed by Filter: 123					
11	Hash	(cost=1.26..1.26 rows=1 width=36)	1	36	0.021 rows=1	loops=1
12	Buckets: 1024 Batches: 1 Memory Usage: 9kB					
13	Seq Scan on areas a	(cost=0.00..1.26 rows=1 width=36)	1	36	0.011 rows=1	loops=1
14	Filter	(area_name = 'Hollywood':text)				
15	Rows Removed by Filter: 20					
16	Planning Time	0.571 ms				
17	Execution Time	28.129 ms				

**Execution Time:** Approximately 28 ms.

3) *Optimization Strategy*: To improve query performance, the following indexes were created:

- CREATE INDEX idx\_area\_name ON Areas (AREA\_NAME);
- CREATE INDEX idx\_crm\_cd\_desc ON Crime\_Types (Crm\_Cd\_Desc text\_pattern\_ops);

4) *Query Performance After Indexing*: After applying the indexes, index scans were performed instead of sequential scans, significantly improving performance.

```

17 < EXPLAIN ANALYZE
18 SELECT ci.dr_no, ct.crm_cd_desc, a.area_name
19 FROM Crime_Incidents ci
20 JOIN Crime_Types ct ON ci.crm_cd = ct.crm_cd
21 JOIN Areas a ON ci.area = a.area
22 WHERE a.area_name = 'Hollywood' AND ct.crm_cd_desc LIKE 'BURGLARY%';
23
24

```

Step	Operation	Cost	Rows	Width	Actual Time	Loops
1	Hash Join	(cost=3.91..3458.74 rows=191 width=69)	191	69	0.166..24.554 rows=1074	loops=1
2	Hash Cond	(ci.area = a.area)				
3	Hash Join	(cost=2.64..3444.82 rows=4009 width=41)	4009	41	0.111..23.355 rows=13151	loops=1
4	Hash Cond	(ci.crm_cd = ct.crm_cd)				
5	Seq Scan on crime_incidents ci	(cost=0.00..3096.90 rows=127290 width=18)	127290	18	0.018..9.965 rows=127290	loops=1
6	Hash	(cost=2.59..2.59 rows=4 width=31)	4	31	0.077 rows=4	loops=1
7	Buckets: 1024 Batches: 1 Memory Usage: 9kB					
8	Seq Scan on crime_types ct	(cost=0.00..2.59 rows=4 width=31)	4	31	0.045 rows=4	loops=1
9	Filter	(crm_cd_desc ~ 'BURGLARY%':text)				
10	Rows Removed by Filter: 123					
11	Hash	(cost=1.26..1.26 rows=1 width=36)	1	36	0.031 rows=1	loops=1
12	Buckets: 1024 Batches: 1 Memory Usage: 9kB					
13	Seq Scan on areas a	(cost=0.00..1.26 rows=1 width=36)	1	36	0.019 rows=1	loops=1
14	Filter	(area_name = 'Hollywood':text)				
15	Rows Removed by Filter: 20					
16	Planning Time	2.470 ms				
17	Execution Time	24.579 ms				

**Execution Time:** Approximately 24 ms.

### B. Problematic Query 2: Search by Latitude and Longitude Range

1) *Query Description*: We want to find crimes that occurred within a specific latitude and longitude boundary.

2) *Initial Query Performance (Before Optimization)*: Initially, the Crime\_Incidents table had no spatial indexes, leading to full table scans and slower query execution.

```

28 /* Problematic Query 2 */
29 < EXPLAIN ANALYZE
30 SELECT dr_no, lat, lon
31 FROM Crime_Incidents
32 WHERE lat BETWEEN 34.95 AND 34.18
33 AND lon BETWEEN -118.38 AND -118.28;
34
35

```

Step	Operation	Cost	Rows	Width	Actual Time	Loops
1	Seq Scan on crime_incidents	(cost=0.00..4369.80 rows=14456 width=26)	14456	26	0.022..23.977 rows=16018	loops=1
2	Filter	(lat >= 34.05::double precision) AND (lat <= 34.17::double precision) AND (lon >= -118.2::double precision) AND (lon <= -118.27::double precision)				
3	Rows Removed by Filter: 117272					
4	Planning Time	1.317 ms				
5	Execution Time	24.507 ms				

**Execution Time:** Approximately 25 ms.

3) *Optimization Strategy*: To improve query performance, a composite index on latitude and longitude was created:

- CREATE INDEX idx\_lat\_lon ON Crime\_Incidents (LAT, LON);

4) *Query Performance After Indexing*: After creating the composite index, the database used index range scans, which reduced the query time substantially.

```

27 /* Solution using Indexing */
28 CREATE INDEX idx_lat_lon ON Crime_Incidents (LAT, LON);
29
30 < EXPLAIN ANALYZE
31 SELECT dr_no, lat, lon
32 FROM Crime_Incidents
33 WHERE lat BETWEEN 34.95 AND 34.18
34 AND lon BETWEEN -118.38 AND -118.28;
35
36

```

Step	Operation	Cost	Rows	Width	Actual Time	Loops
1	Bitmap Heap Scan on crime_incidents	(cost=890.48..3003.60 rows=14456 width=26)	14456	26	2.235..8.713 rows=16018	loops=1
2	Recheck Cond	(lat >= 34.05::double precision) AND (lat <= 34.17::double precision) AND (lon >= -118.2::double precision) AND (lon <= -118.27::double precision)				
3	Heap Blocks	exact=1789				
4	Bitmap Index Scan on idx_lat_lon	(cost=0.00..888.88 rows=14456 width=0)	14456	0	2.072 rows=16018	loops=1
5	Index Cond	(lat >= 34.05::double precision) AND (lat <= 34.17::double precision) AND (lon >= -118.2::double precision) AND (lon <= -118.27::double precision)				
6	Planning Time	0.326 ms				
7	Execution Time	9.382 ms				

**Execution Time:** Approximately 9 ms.

### C. Problematic Query 3: Find Top 5 Crime Types in a Specific Area

1) *Query Description*: We want to find the **Top 5 most frequently reported crime types** in a specific area, for example, Downtown.

2) *Initial Query Performance (Before Optimization)*: Initially, no indexes existed on AREA or Crm\_Cd columns, leading to sequential scans and slower performance, especially when grouping and ordering by crime type.

37	/* Problematic Query3 */
38	EXPLAIN ANALYZE
39	SELECT ct.Crm_Cd_Desc, COUNT(*) AS total_crimes
40	FROM Crime_Incidents ci
41	JOIN Crime_Types ct ON ci.Crm_Cd = ct.Crm_Cd
42	JOIN Areas a ON ci.AREA = a.AREA
43	WHERE a.AREA_NAME = 'Downtown'
44	GROUP BY ct.Crm_Cd_Desc
45	ORDER BY total_crimes DESC
46	LIMIT 5;
47	
Data Output Messages Notifications	
Showing 10	
QUERY PLAN	
test	
3	Sort Key: (count(*)) DESC
4	Sort Method: quicksort Memory: 254B
5	→ HashAggregate (cost=3550.26..3551.62 rows=127 width=33) (actual time=0.070..0.072 rows=0 loops=1)
6	Group Key: ct.crm_cd_desc
7	Batches: 1 Memory Usage: 40kB
8	→ Hash Join (cost=5.13..3520.04 rows=6061 width=27) (actual time=0.066..0.068 rows=0 loops=1)
9	Hash Cond: (ci.crm_cd = ct.crm_cd)
10	→ Hash Join (cost=1.27..3499.74 rows=6061 width=4) (actual time=0.066..0.067 rows=0 loops=1)
11	Hash Cond: (ci.area = a.area)
12	→ Seq Scan on crime_incidents ci (cost=0.00..3096.90 rows=127290 width=6) (actual time=0.024..0.024 rows=1 loop..)
13	→ Hash Scan on areas a (cost=1.26 rows=1 width=4) (actual time=0.019..0.020 rows=0 loops=1)
14	Buckets: 1024 Batches: 1 Memory Usage: 8kB
15	→ Seq Scan on areas a (cost=0.00..1.26 rows=1 width=4) (actual time=0.019..0.019 rows=0 loops=1)
16	Filter: (area_name = 'Downtown') text
17	Rows Removed by Filter: 21
18	→ Hash (cost=2.27..2.27 rows=127 width=31) (never executed)
19	→ Seq Scan on crime_types ct (cost=0.00..2.27 rows=127 width=31) (never executed)
20	Planning Time: 4.616 ms
21	Execution Time: 1.133 ms

**Execution Time:** Approximately 1.2 ms.

3) *Optimization Strategy:* To optimize the query, the following indexes were created:

- CREATE INDEX idx\_crime\_incidents\_area  
ON Crime\_Incidents (AREA);
- CREATE INDEX idx\_crime\_types\_crncd  
ON Crime\_Types (Crm\_Cd);

These indexes improved the join operations between Crime\_Incidents, Crime\_Types, and Areas, enabling faster filtering, grouping, and sorting.

4) *Query Performance After Indexing:* After creating the indexes, the query execution utilized index scans instead of full sequential scans, which significantly improved the performance.

49	/* Solution using Indexing */
50	CREATE INDEX idx_crime_incidents_area ON Crime_Incidents (AREA);
51	CREATE INDEX idx_crime_types_crncd ON Crime_Types (Crm_Cd);
52	
53	
54	
55	EXPLAIN ANALYZE
56	SELECT ct.Crm_Cd_Desc, COUNT(*) AS total_crimes
57	FROM Crime_Incidents ci
58	JOIN Crime_Types ct ON ci.Crm_Cd = ct.Crm_Cd
59	JOIN Areas a ON ci.AREA = a.AREA
60	WHERE a.AREA_NAME = 'Downtown'
61	GROUP BY ct.Crm_Cd_Desc
62	ORDER BY total_crimes DESC
63	LIMIT 5;
Data Output Messages Notifications	
Showing 10	
QUERY PLAN	
test	
7	Batches: 1 Memory Usage: 40kB
8	→ Hash Join (cost=76.12..3053.20 rows=6061 width=27) (actual time=0.115..0.117 rows=0 loops=1)
9	Hash Cond: (ci.crm_cd = ct.crm_cd)
10	→ Nested Loop (cost=71.27..3032.90 rows=6061 width=4) (actual time=0.075..0.076 rows=0 loops=1)
11	→ Seq Scan on areas a (cost=0.00..1.26 rows=1 width=4) (actual time=0.014..0.014 rows=0 loops=1)
12	Filter: (area_name = 'Downtown') text
13	Rows Removed by Filter: 21
14	→ Bitmap Heap Scan on crime_incidents ci (cost=71.27..1971.03 rows=6061 width=6) (never executed)
15	Recheck Cond: (area = a.area)
16	→ Bitmap Index Scan on idx_crime_incidents_area (cost=0.00..69.75 rows=6061 width=0) (never executed)
17	Index Cond: (area = a.area)
18	→ Hash (cost=2.27..2.27 rows=127 width=31) (actual time=0.093..0.094 rows=127 loops=1)
19	Buckets: 1024 Batches: 1 Memory Usage: 17kB
20	→ Seq Scan on crime_types ct (cost=0.00..2.27 rows=127 width=31) (actual time=0.026..0.052 rows=127 loop..)
21	Planning Time: 0.681 ms
22	Execution Time: 0.243 ms

**Execution Time:** Approximately 0.2 ms.

IX. REFERENCES

The dataset used for this project is sourced from the official U.S. government open data portal. It contains crime incident reports from 2020 to the present, including various attributes such as crime type, location, victim details, and more.

The dataset can be accessed at the following link: **Crime Data from 2020 to Present**