Crime Analytics with SQL: Building A Relational Investigation System

Table of Contents

Introduction 3

Organizational Overview 3

Entities and Attributes 4

Business Rules 8

Entity Relationship Diagram 8

Relational Model 10

Enhanced Entity Relationship Diagram (After Relational Model) 12

Generating the Mock Data 15

Creating and Loading the Tables 15

Investigation Scenario 16

Solving the Case using SQL Queries 17

- Query 1 - Homicide and Kidnapping Occurred in Commercial Area 17
- Query 2 - List of Officers Assigned to Drug-Related Cases 17
- Query 3 - More victims with fatal injuries found in similar locations 18
- Query 4 - Suspects at large, affiliated with a group, involved in Homicide or Kidnapping 19
- Query 5 – Drug Traces and Severe Victim Injuries 20
- Query 6 – Open Cases with High Ransom or Stolen Value 21
- Query 7 - Creating Alerts for Toxic Forensic Entries 22

Conclusion 24

Benefits Gained 24

Challenges Encountered During the Project 25

Introduction

In today's world, the rise in complex and organized crimes requires law enforcement agencies to rely on more than just manual records and intuition. With vast amounts of data generated from crime reports, forensic labs, and field investigations, there is a critical need for a centralized and structured system that can organize, connect, and analyze this information efficiently.

This project focuses on building a crime investigation system for the Crime Data Management Bureau (CDMB) - a hypothetical organization designed to manage violent crime data. By applying principles of database design and SQL querying, we simulate how investigators can use digital tools to trace patterns, link suspects to crime scenes, analyze forensic results, and ultimately solve cases. The aim is to show how well-modeled data, and powerful queries can turn raw information into actionable insights that support real-time decision-making in criminal justice.

Organizational Overview

Company name: Crime Data Management Bureau (CDMB) **(Hypothetical, of course)**

Business Sector: Law Enforcement

Description: The Crime Data Management Bureau (CDMB) is a centralized database system designed to track and manage violent crimes, including homicides, kidnappings, organized crime, and other high-profile felonies. This system will assist law enforcement agencies, crime analysts, and judicial bodies in identifying patterns, tracking suspects, and improving national security. Our database focuses specifically on violent crimes and serial offenses, establishing connections between victims, suspects, and known criminal activities. By analyzing these relationships, the system will support criminal profiling, predictive policing, and intelligence gathering, ultimately aiding in ongoing investigations and helping prevent future felonies.

Area of Focus: The project specifically focuses on serious crime investigation and data management, ensuring that detailed and structured records of reported crimes, victims, suspects, crime scenes, forensic evidence, and law enforcement officers are maintained. The system will help track crime patterns, support forensic analysis, and aid in case resolution.

Reason for choosing this organization: We selected the crime data management sector due to its crucial role in maintaining law and order. A well-structured database for crime tracking enhances investigative efficiency, supports forensic analysis, and aids in better decision-making by law enforcement agencies. Additionally, managing crime data is a real-world challenge faced by law enforcement globally, and creating a structured system helps us explore practical applications of database design in public safety.

Entities and Attributes

Entity Descriptions:

- **Crime:** This entity describes specific details about criminal events, including the date, type, description, and weapons used. It tracks the status of each case, from open investigations to closed or dismissed cases.
- **Victim:** This entity stores information about individuals who have been harmed or affected by a crime. It includes personal details like name, date of birth, address, and information about injuries sustained.
- **Suspect:** This entity holds data about individuals believed to be involved in criminal activities. It encompasses personal identification, criminal history, known affiliations, and their current legal status.
- **Crime Scene:** This entity describes the location where a crime occurred, detailing the type of location, address, and evidence found. It also notes the presence of security cameras and other relevant scene details.
- **Forensics:** This entity contains the results of scientific analysis conducted on evidence from a crime scene. It includes DNA findings, postmortem results, blood type analysis, and toxicology reports, aiding in the investigation.
- **Officer:** This entity stores information about law enforcement personnel involved in crime investigations. It includes personal details, badge numbers, precinct assignments, ranks, and specialized areas of expertise.

A list of all the entities, attributes and their description and domain constraints:

Entities	Attribute Name	Data Type	Description	Domain Constraint
Crime	CrimeID	INT	A unique identifier for each crime record.	Primary Key, NOT NULL, Unique
	CrimeDate	DATE	The date when the crime occurred.	NOT NULL
	CrimeDescription	VARCHAR(1000)	A narrative account of the events of the crime.	NULL allowed
	WeaponUsed	VARCHAR(100)	The instrument or means used to commit the crime.	NULL allowed
	CaseStatus	VARCHAR(30)	Case status: Open/Closed/Cold/Dismissed	NOT NULL
	CrimeType	VARCHAR(30)	The category of the crime: Suicide/Homicide, Kidnapping/Robbery	CHECK (CrimeType IN ('Suicide','Homicide', 'Kidnapping', 'Robbery')
	CrimeType- Suicide: MethodUsed	VARCHAR(30)	The method used for suicide (e.g. Hanging, Poison, etc.)	NOT NULL
	CrimeType- Suicide: SuicideNoteFound	BOOLEAN	Identifies whether suicide note was found or not	NOT NULL
	CrimeType- Homicide: CauseOfDeath	VARCHAR(100)	Identifies cause of death(e.g. Strangulation, Blunt Force Trauma, etc.)	NOT NULL

	CrimeType- Homicide: NumberOfVictims	INT	Total number of victims of the crime	NOT NULL
	CrimeType- Kidnapping: RansomDemamded	INT	Amount of ransom demanded (in $)	NULL allowed
	CrimeType- Kidnapping: SuspectedMotive	VARCHAR(30)	Identifies probable motive of the kidnapping	NOT NULL
	CrimeType- Kidnapping: VictimRecovered	BOOLEAN	Identifies whether the victim was recovered(Y/N)	NOT NULL
	CrimeType- Robbery: StolenValue	INT	Total value of goods stolen (in $)	NOT NULL
Victim	VictimID	INT	A unique identifier for each victim.	Primary Key, NOT NULL, Unique
	Name	VARCHAR(50)	The victim's given name and last name.	NULL allowed
	DateOfBirth	DATE	The victim's birth date.	NULL allowed
	Address	VARCHAR(255)	The victim's residential address.	NULL allowed
	Phone_Number	VARCHAR(15)	The victim's contact number.	NULL allowed
	Injury_Type	VARCHAR(10)	The nature and severity of the victim's injuries (e.g., Minor, Major, Fatal).	CHECK (InjuryType IN ('Minor', 'Major', 'Fatal'))
	Injury_Type- Minor: RecoveryTime	INT	Days required for recovery	NOT NULL
	Injury_Type- Major: InjuryDescription	VARCHAR(255)	Detailed description of the injury	NOT NULL
	Injury_Type- Fatal: TimeOfDeath	DATETIME	Date & time of death	NOT NULL
	Injury_Type- Fatal:AutopsyConducted	BOOLEAN	Whether autopsy was conducted (Y/N)	NOT NULL
Suspect	SuspectID	INT	A unique identifier for each suspect.	Primary Key, NOT NULL, Unique
	Name	VARCHAR(50)	The suspect's given name and last name.	NULL allowed
	DateOfBirth	DATE	The suspect's birth date.	NULL allowed
	Address	VARCHAR(255)	The suspect's residential address.	NULL allowed
	Alias (Multivalued Attribute)	VARCHAR(100)	Other names known to be used by the suspect.	NOT NULL
	CriminalRecord	BOOLEAN	Indicates whether the suspect has a prior criminal history (Y/N).	NULL allowed
	KnownAffiliations (Multivalued Attribute)	VARCHAR(100)	Groups or organizations the suspect is associated with (e.g., Gangs, Cartels).	NOT NULL
	CurrentStatus	VARCHAR(30)	The suspect's legal situation (e.g., Arrested, At Large, Deceased).	CHECK (CurrentStatus IN ('Arrested', 'At Large', 'Deceased'))
	CurrentStatus- Arrested: ArrestDate	DATE	Date of arrest	NOT NULL
	CurrentStatus- Arrested: DetentionFacility	VARCHAR(255)	Detention facility name where the suspect is held	NOT NULL

	CurrentStatus-Deceased: DateOfDeath	DATE	Date when the suspect died	NULL allowed
	CurrentStatus-Deceased: CauseOfDeath	VARCHAR(30)	Cause of death of suspect (e.g. Natural, accident, etc.)	NULL allowed
	CurrentStatus-AtLarge: LastKnownLocation	VARCHAR(255)	Location at which suspect was last known to be present	NULL allowed
Crime_Scene	SceneID	INT	A unique identifier for each crime scene.	Primary Key, NOT NULL, Unique
	LocationType	VARCHAR(255)	Details about the physical location of the crime scene (e.g., Abandoned Warehouse, Residential Home).	CHECK (CurrentStatus IN ('Residential', 'Commercial', 'Abandoned','Government Facility,'Public Place'))
	LocationType-Residential: NumberOfOccupants	INT	Number of occupants of the residential crime scene	NULL allowed
	LocationType-Residential: OwnerName	VARCHAR(50)	Name of the residential property owner	NULL allowed
	LocationType-Commercial:SecurityLevel	VARCHAR(30)	Levels of security: High, medium, low, none.	NULL allowed
	LocationType-Commercial: NumberOfVictims	INT	Number of total victims on the crime scene	NULL allowed
	LocationType-Abandoned:LastKnownActivity	VARCHAR(255)	Description of the last known activity at the abandoned location	NULL allowed
	Address	VARCHAR(255)	Address of the crime scene	NOT NULL
	Evidence_Collected (Multivalued Attribute)	VARCHAR(255)	A record of the physical evidence found at the scene (e.g., DNA, Bullet Casings, Blood Samples)	NOT NULL
	SecurityCamerasPresent	BOOLEAN	Indicates whether security camera was present (Y/N)	NULL allowed
Forensics	ForensicsReportID	INT	A unique identifier for each forensic report.	Primary Key, NOT NULL, Unique
	DNAFound	BOOLEAN	Indicates whether DNA evidence was found at the scene (Y/N).	NULL allowed
	PostMortem_Performed	BOOLEAN	Indicates whether postmortem has been ferformed yet (Y/N)	NULL allowed
	BloodTypeFound	BOOLEAN	Indicates whether blood type has been found yet (Y/N)	NULL allowed
	ToxicologyReportResult	VARCHAR(30)	Whether Poison/Drug/Alcohol were found in the toxicology report	CHECK (ToxicologyReportResult IN ('Poison', 'Drug', 'Alcohol'))
	ToxicologyReportResult-Poison: PoisonName	VARCHAR(50)	Common name of the poison(if identified)	NOT NULL

	ToxicologyReportResult-Poison: PossibleSource	VARCHAR(50)	Source from where poison was obtained	NULL allowed
	ToxicologyReportResult-Drug: DrugName	VARCHAR(50)	Name of the drug (if identified)	NOT NULL
	ToxicologyReportResult-Drug: MethodOfIntake	VARCHAR(50)	Method of intake of the drug (if identified)	NULL allowed
	ToxicologyReportResult-Alcohol: BloodAlcoholContent	INT	Blood alcohol content identified	NOT NULL
	EstimatedDeathTime	DATETIME	Estimated time of death of victim(s)	NULL allowed
Officer	OfficerID	INT	A unique identifier for each law enforcement officer.	Primary Key, NOT NULL, Unique
	Name	VARCHAR(50)	The officer's given name.	NOT NULL
	BadgeNumber	INT	The officer's identification number.	NOT NULL, Unique
	Precinct	VARCHAR(100)	The police station or jurisdiction the officer is assigned to.	NULL allowed
	OfficerRank	VARCHAR(50)	The officer's position in the law enforcement hierarchy.	NULL allowed
	Specialization	VARCHAR(50)	The officer's area of expertise (BehavioralAnalysis, Kidnapping, Robbery&Theft).	CHECK (Specialization IN ('Behavioural Analysis', 'Kidnapping', 'Robbery & Theft', 'Financial Crime'))
	Specialization-BehavioralAnalysis: YearsOfExperience	INT	Number of years that the officer has experience in solving homicide cases	NOT NULL
	Specialization-Kidnapping: Case_Success_Rate	INT	A function of no. of cases solved from the no. of cases assigned	NOT NULL
	Specialization-Robbery&Theft: NoOfCasesSolved	INT	Total no. of cases solved by the Officer	NOT NULL

Business Rules

The business rules used in this system are:

- Each Crime ***can*** have ***many*** Victims
- Each victim ***must*** be associated with ***exactly one*** crime

- Each crime ***must*** have ***at least one (many)*** suspect(suspects)
- Each suspect ***can*** be linked to ***many*** crimes

- Each crime ***must*** be assigned to ***exactly one*** officer
- Each officer ***can*** be assigned to ***many*** crimes

- Each Supervisor ***must*** supervise ***at least one(many)*** officer
- Each officer ***must*** be supervised by ***only one*** Supervisor

- Each crime ***must*** have ***at least one(many)*** crime scenes
- Each crime scene ***must*** be associated with ***exactly one*** crime

- Each crime ***can*** have ***many*** Forensics reports
- Each Forensics report ***must*** be produced for ***only one*** Crime

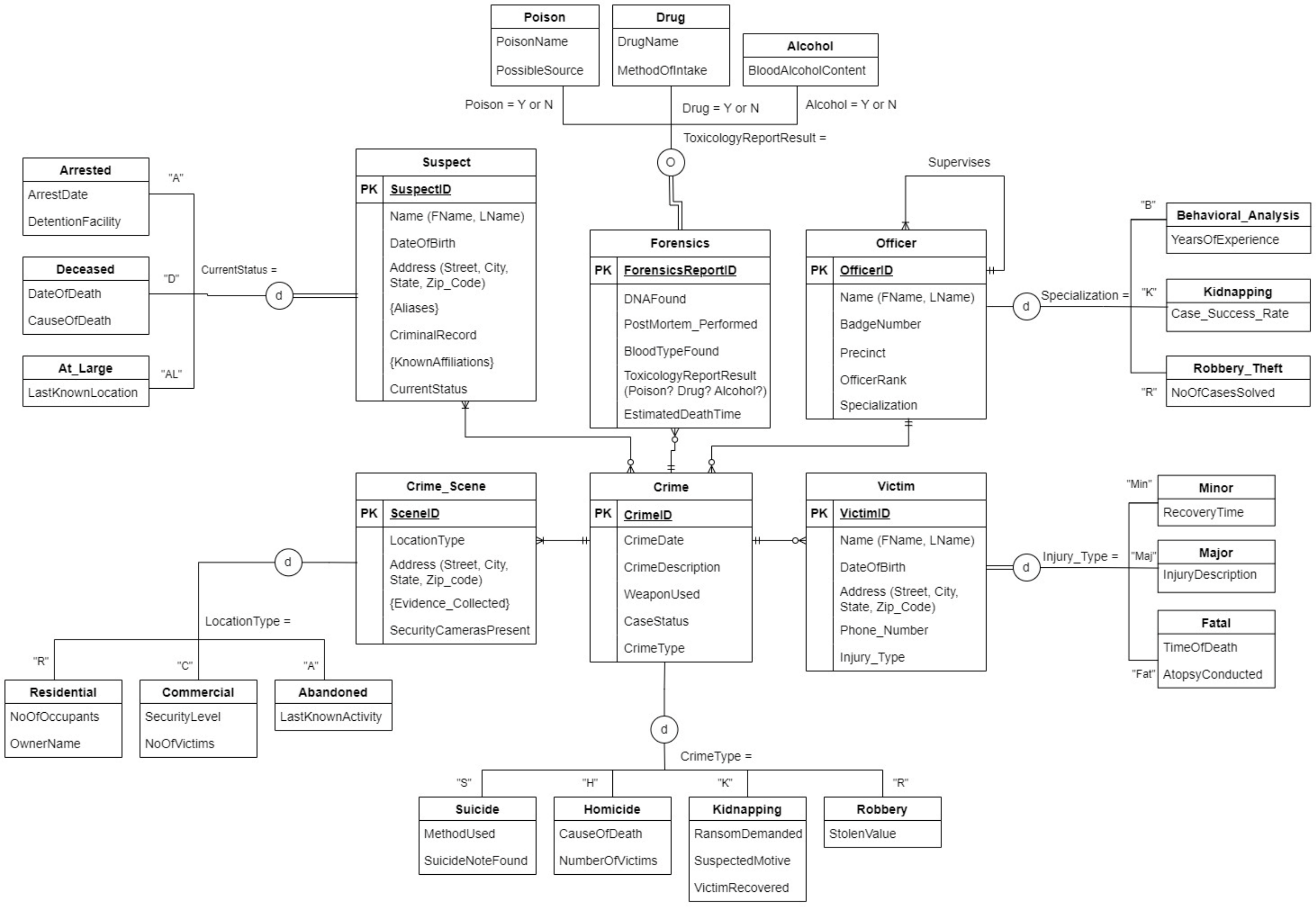
Entity Relationship Diagram

The following E-R diagram represents the core structure of the Crime Data Management Bureau (CDMB) system. It captures the real-world relationships between crimes, suspects, victims, officers, forensic reports, and scenes. Each relationship in the diagram is directly aligned with key business rules, which ensure that the data model reflects logical and operational expectations of a real criminal investigation system.

The E-R diagram also includes:

- Specialization (subtypes) for entities like Suspect, Victim, Crime, Scene, Officer, and Forensics
- Multivalued attributes such as Aliases and Affiliations for suspects
- Enforcement of rules like disjoint/overlap and total/partial specialization, helping to preserve data integrity and model real-life classification logic

Together, the E-R diagram and these rules provide a strong foundation for relational design, enabling advanced query execution, forensic analysis, and structured investigation tracking within the CDMB system.


Poison
PoisonName
PossibleSource
Drug
DrugName
MethodOfIntake
Alcohol
BloodAlcoholContent
Poison = Y or N
Drug = Y or N
Alcohol = Y or N
ToxicologyReportResult =
O
Arrested
ArrestDate
DetentionFacility
"A"
Deceased
DateOfDeath
CauseOfDeath
"D"
At_Large
LastKnownLocation
"AL"
CurrentStatus =
d
Suspect
PK SuspectID
Name (FName, LName)
DateOfBirth
Address (Street, City, State, Zip_Code)
{Aliases}
CriminalRecord
{KnownAffiliations}
CurrentStatus
Forensics
PK ForensicsReportID
DNAFound
PostMortem_Performed
BloodTypeFound
ToxicologyReportResult (Poison? Drug? Alcohol?)
EstimatedDeathTime
Supervises
Officer
PK OfficerID
Name (FName, LName)
BadgeNumber
Precinct
OfficerRank
Specialization
d
Specialization =
"B"
Behavioral_Analysis
YearsOfExperience
"K"
Kidnapping
Case_Success_Rate
Robbery_Theft
"R"
NoOfCasesSolved
Crime_Scene
PK SceneID
LocationType
Address (Street, City, State, Zip_code)
{Evidence_Collected}
SecurityCamerasPresent
d
LocationType =
"R"
Residential
NoOfOccupants
OwnerName
"C"
Commercial
SecurityLevel
NoOfVictims
"A"
Abandoned
LastKnownActivity
Crime
PK CrimeID
CrimeDate
CrimeDescription
WeaponUsed
CaseStatus
CrimeType
Victim
PK VictimID
Name (FName, LName)
DateOfBirth
Address (Street, City, State, Zip_Code)
Phone_Number
Injury_Type
d
Injury_Type =
"Min"
Minor
RecoveryTime
"Maj"
Major
InjuryDescription
Fatal
TimeOfDeath
"Fat"
AtopsyConducted
d
CrimeType =
"S"
Suicide
MethodUsed
SuicideNoteFound
"H"
Homicide
CauseOfDeath
NumberOfVictims
"K"
Kidnapping
RansomDemanded
SuspectedMotive
VictimRecovered
"R"
Robbery
StolenValue

Relational Model

The relational model for our Crime Database Management and Investigation System (CDMB) was developed by systematically converting the ER diagram into a normalized database schema that adheres to Third Normal Form (3NF). Our primary objective was to eliminate data redundancy, ensure data integrity, and support flexible and accurate SQL queries for crime investigation workflows.

Entity to Table Conversion

Each entity from our ER diagram was mapped to a distinct relational table with clearly defined primary keys. Attributes were assigned appropriate data types and domain constraints. To meet 3NF requirements:

- We ensured that all non-key attributes are fully dependent on the primary key.
- We eliminated any partial dependencies (which could occur in composite keys) and transitive dependencies (where one non-key attribute depends on another non-key attribute).

Handling Composite & Multivalued Attributes

Some entities, such as Crime Scene and Suspect, contained composite or multivalued attributes. These were restructured as follows:

- Crime_Scene_Evidence was created to store multiple pieces of evidence linked to a scene (SceneID), ensuring that Evidence_Collected is managed independently.
- Suspect_Aliases and Suspect_Affiliations tables were introduced to handle suspects with multiple aliases or known gang affiliations, using SuspectID as a foreign key.

Many-to-Many Relationships

Many-to-many relationships were implemented using junction tables with composite primary keys:

- Crime_Suspect captures all links between suspects and crimes.
- This approach supports scenarios where one suspect may be involved in multiple crimes, and a crime may involve multiple suspects.

Unary Relationship (Officer Reports to Supervisor)

The Officer table includes a SupervisorID column, which acts as a self-referencing foreign key to OfficerID. This unary relationship models internal reporting structures, allowing us to track which officers supervise others.

Subtypes & Specialization

- Victims were categorized based on injury severity. We created separate subtype tables:
  - Victim_Minor (e.g., includes recovery time)
  - Victim_Major (e.g., includes injury description)
  - Victim_Fatal (e.g., includes time of death and autopsy info)

These tables each have a foreign key linking back to the Victim table, ensuring that one victim appears in only one subtype based on business rules.

- Forensics Reports were modeled with overlapping subtypes:
    - Forensics_Drug
    - Forensics_Poison
    - Forensics_Alcohol

  A single Forensics entry (e.g., a toxicology report) can appear in more than one of these subtype tables if multiple substances are involved.

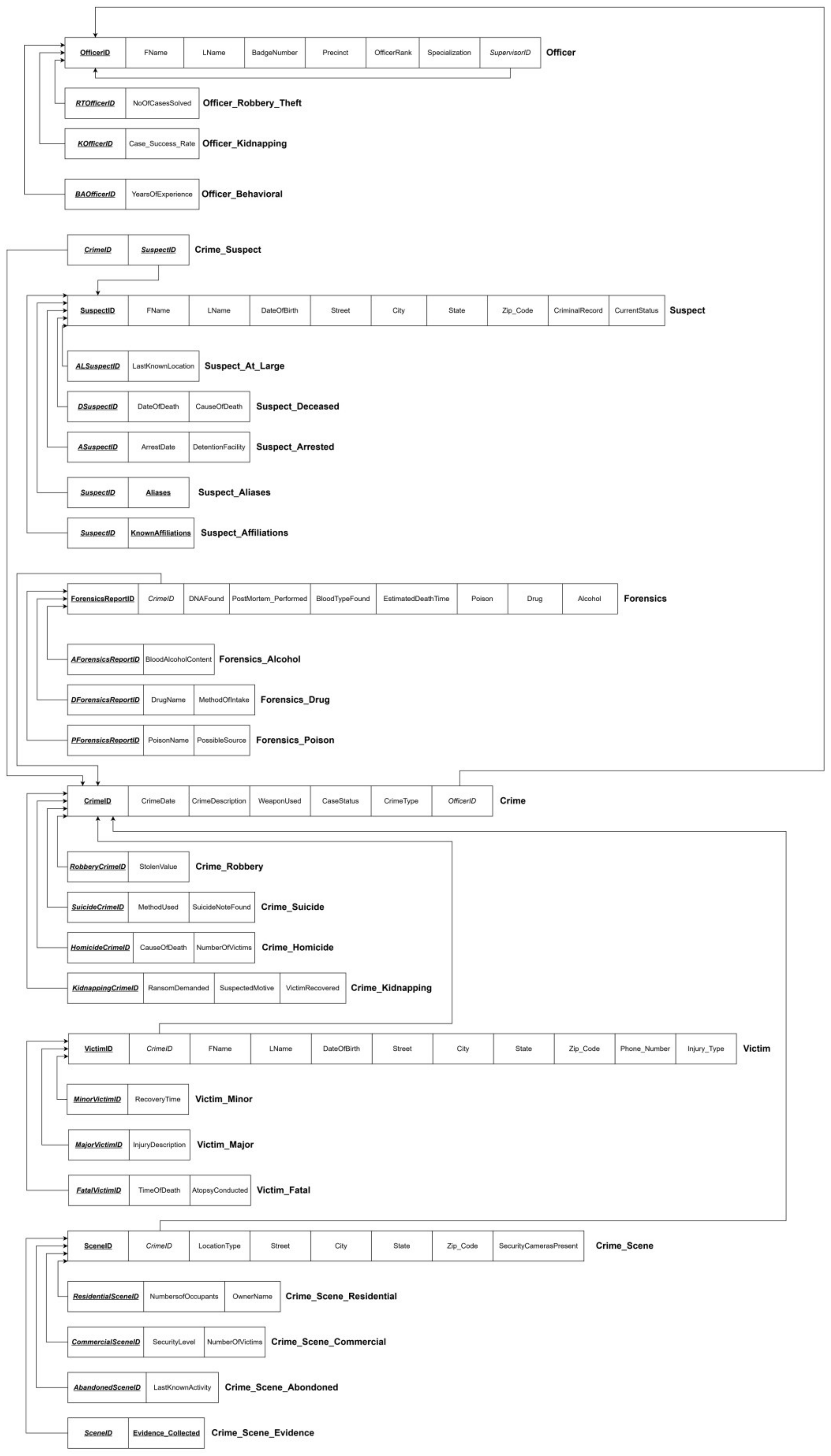
- Crime scenes were further subclassed based on location context:
    - Crime_Scene_Residential,
    - Crime_Scene_Commercial
    - Crime_Scene_Abandoned

  Each extend the base Crime_Scene table and contain attributes relevant to the type of location (e.g., number of occupants, security level, or last known activity).

Final Design Characteristics

- A total of 20+ relational tables were created to capture the full scope of the CDMB system.
- Each table enforces primary key and foreign key constraints to maintain referential integrity.
- We incorporated domain constraints to restrict values (e.g., boolean for Drug presence, enumerated injury types).
- The relational model ensures modularity and scalability, making it adaptable to real-world forensic or law enforcement systems.

A professional visual diagram of this model was created using draw.io and It clearly shows all entities, keys, relationships, and structural decisions.


OfficerID FName LName BadgeNumber Precinct OfficerRank Specialization SupervisorID Officer
RTOfficerID NoOfCasesSolved Officer_Robbery_Theft
KOfficerID Case_Success_Rate Officer_Kidnapping
BAOfficerID YearsOfExperience Officer_Behavioral
CrimeID SuspectID Crime_Suspect
SuspectID FName LName DateOfBirth Street City State Zip_Code CriminalRecord CurrentStatus Suspect
ALSuspectID LastKnownLocation Suspect_At_Large
DSuspectID DateOfDeath CauseOfDeath Suspect_Deceased
ASuspectID ArrestDate DetentionFacility Suspect_Arrested
SuspectID Aliases Suspect_Aliases
SuspectID KnownAffiliations Suspect_Affiliations
ForensicsReportID CrimeID DNAFound PostMortem_Performed BloodTypeFound EstimatedDeathTime Poison Drug Alcohol Forensics
AForensicsReportID BloodAlcoholContent Forensics_Alcohol
DForensicsReportID DrugName MethodOfIntake Forensics_Drug
PForensicsReportID PoisonName PossibleSource Forensics_Poison
CrimeID CrimeDate CrimeDescription WeaponUsed CaseStatus CrimeType OfficerID Crime
RobberyCrimeID StolenValue Crime_Robbery
SuicideCrimeID MethodUsed SuicideNoteFound Crime_Suicide
HomicideCrimeID CauseOfDeath NumberOfVictims Crime_Homicide
KidnappingCrimeID RansomDemanded SuspectedMotive VictimRecovered Crime_Kidnapping
VictimID CrimeID FName LName DateOfBirth Street City State Zip_Code Phone_Number Injury_Type Victim
MinorVictimID RecoveryTime Victim_Minor
MajorVictimID InjuryDescription Victim_Major
FatalVictimID TimeOfDeath AtopsyConducted Victim_Fatal
SceneID CrimeID LocationType Street City State Zip_Code SecurityCamerasPresent Crime_Scene
ResidentialSceneID NumbersofOccupants OwnerName Crime_Scene_Residential
CommercialSceneID SecurityLevel NumberOfVictims Crime_Scene_Commercial
AbandonedSceneID LastKnownActivity Crime_Scene_Abondoned
SceneID Evidence_Collected Crime_Scene_Evidence

Enhanced Entity Relationship Diagram (After Relational Model)

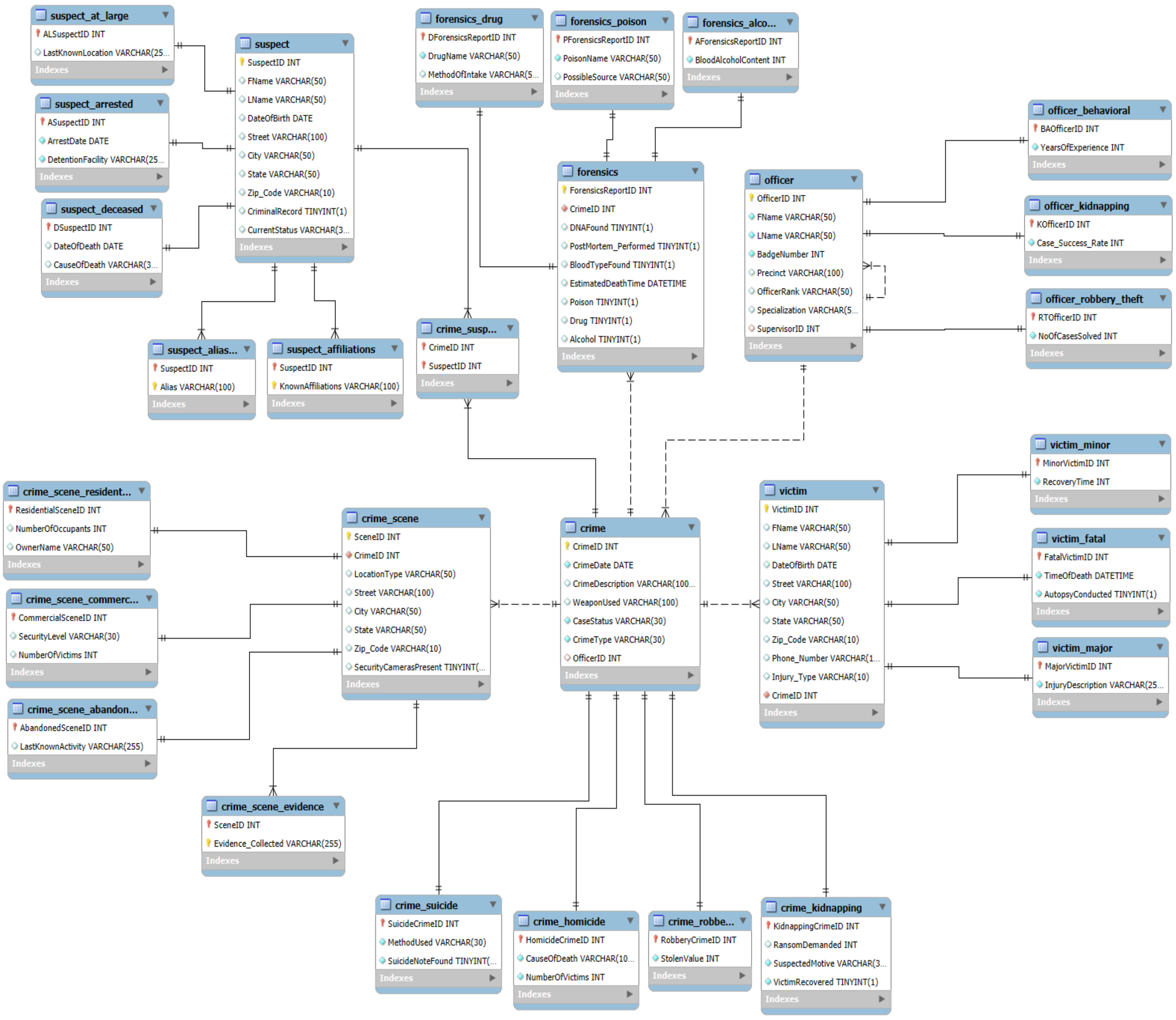
To validate our relational model and ensure correct implementation in MySQL, we used MySQL Workbench's **reverse engineering feature** to generate an Enhanced ER (EER) diagram. Unlike the earlier ER diagram, which was conceptual, this EER diagram reflects the actual database schema including table structures, keys, data types, and foreign key constraints.

The primary purpose of creating this diagram was to:

- Visually confirm that our relational model was implemented correctly
- Verify referential integrity, ensuring that all relationships (especially foreign keys) were properly enforced
- Check mapping of subtypes and junction tables in the live schema
- Provide a technical reference for database navigation and query design

One key insight gained from this diagram was the ability to see the full structure at a glance, including how central tables like Crime connect to others. This was especially useful for spotting connection paths across multiple layers - such as how forensic reports relate back to scenes and suspects through crime IDs.

Compared to the earlier ER and relational diagrams, this visual representation gave us confidence that the database was normalized, logically consistent, and ready for data loading and advanced querying.


suspect_at_large
ALSuspectID INT
LastKnownLocation VARCHAR(25...
Indexes
suspect_arrested
ASuspectID INT
ArrestDate DATE
DetentionFacility VARCHAR(25...
Indexes
suspect_deceased
DSuspectID INT
DateOfDeath DATE
CauseOfDeath VARCHAR(3...
Indexes
suspect
SuspectID INT
FName VARCHAR(50)
LName VARCHAR(50)
DateOfBirth DATE
Street VARCHAR(100)
City VARCHAR(50)
State VARCHAR(50)
Zip_Code VARCHAR(10)
CriminalRecord TINYINT(1)
CurrentStatus VARCHAR(3...
Indexes
suspect_alias...
SuspectID INT
Alias VARCHAR(100)
Indexes
suspect_affiliations
SuspectID INT
KnownAffiliations VARCHAR(100)
Indexes
forensics_drug
DForensicsReportID INT
DrugName VARCHAR(50)
MethodOfIntake VARCHAR(5...
Indexes
forensics_poison
PForensicsReportID INT
PoisonName VARCHAR(50)
PossibleSource VARCHAR(50)
Indexes
forensics_alco...
AForensicsReportID INT
BloodAlcoholContent INT
Indexes
forensics
ForensicsReportID INT
CrimeID INT
DNAFound TINYINT(1)
PostMortem_Performed TINYINT(1)
BloodTypeFound TINYINT(1)
EstimatedDeathTime DATETIME
Poison TINYINT(1)
Drug TINYINT(1)
Alcohol TINYINT(1)
Indexes
crime_susp...
CrimeID INT
SuspectID INT
Indexes
officer
OfficerID INT
FName VARCHAR(50)
LName VARCHAR(50)
BadgeNumber INT
Precinct VARCHAR(100)
OfficerRank VARCHAR(50)
Specialization VARCHAR(5...
SupervisorID INT
Indexes
officer_behavioral
BAOfficerID INT
YearsOfExperience INT
Indexes
officer_kidnapping
KOfficerID INT
Case_Success_Rate INT
Indexes
officer_robbery_theft
RTOfficerID INT
NoOfCasesSolved INT
Indexes
victim_minor
MinorVictimID INT
RecoveryTime INT
Indexes
victim_fatal
FatalVictimID INT
TimeOfDeath DATETIME
AutopsyConducted TINYINT(1)
Indexes
victim_major
MajorVictimID INT
InjuryDescription VARCHAR(25...
Indexes
victim
VictimID INT
FName VARCHAR(50)
LName VARCHAR(50)
DateOfBirth DATE
Street VARCHAR(100)
City VARCHAR(50)
State VARCHAR(50)
Zip_Code VARCHAR(10)
Phone_Number VARCHAR(1...
Injury_Type VARCHAR(10)
CrimeID INT
Indexes
crime_scene_resident...
ResidentialSceneID INT
NumberOfOccupants INT
OwnerName VARCHAR(50)
Indexes
crime_scene_commerc...
CommercialSceneID INT
SecurityLevel VARCHAR(30)
NumberOfVictims INT
Indexes
crime_scene_abandon...
AbandonedSceneID INT
LastKnownActivity VARCHAR(255)
Indexes
crime_scene
SceneID INT
CrimeID INT
LocationType VARCHAR(50)
Street VARCHAR(100)
City VARCHAR(50)
State VARCHAR(50)
Zip_Code VARCHAR(10)
SecurityCamerasPresent TINYINT(...
Indexes
crime
CrimeID INT
CrimeDate DATE
CrimeDescription VARCHAR(100...
WeaponUsed VARCHAR(100)
CaseStatus VARCHAR(30)
CrimeType VARCHAR(30)
OfficerID INT
Indexes
crime_scene_evidence
SceneID INT
Evidence_Collected VARCHAR(255)
Indexes
crime_suicide
SuicideCrimeID INT
MethodUsed VARCHAR(30)
SuicideNoteFound TINYINT(...
Indexes
crime_homicide
HomicideCrimeID INT
CauseOfDeath VARCHAR(10...
NumberOfVictims INT
Indexes
crime_robbe...
RobberyCrimeID INT
StolenValue INT
Indexes
crime_kidnapping
KidnappingCrimeID INT
RansomDemanded INT
SuspectedMotive VARCHAR(3...
VictimRecovered TINYINT(1)
Indexes

Generating the Mock Data

Since real crime data is sensitive and often restricted, we used mock data to simulate realistic scenarios for our project. The dataset was generated using Mockaroo, a data generation platform that allows users to create structured datasets based on custom-defined attributes and constraints.

We carefully designed each entity — such as Crime, Victim, Suspect, Crime Scene, Forensics, and Officer — to reflect the attributes and relationships outlined in our Relational diagram. To ensure referential integrity and realistic query outcomes, domain rules and foreign key relationships were implemented by selecting the "dataset column" type in Mockaroo. This allowed us to generate values that accurately reference other entities, aligning with our database schema and supporting complex investigation logic.

This approach ensured that our data remained ethical and secure, while still offering complexity and variability needed to demonstrate real-world crime investigation workflows.

Creating and Loading the Tables

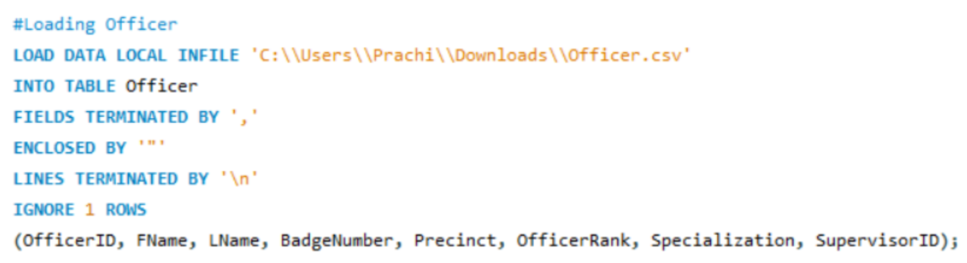
Creating Tables

All 29 tables were created in MySQL using `CREATE TABLE` statements. Each table was designed based on our Enhanced ER model, with appropriate `PRIMARY KEY` and `FOREIGN KEY` constraints explicitly defined as discussed in class. Dummy values were first used to test structural integrity before full population.

Loading Data into Tables

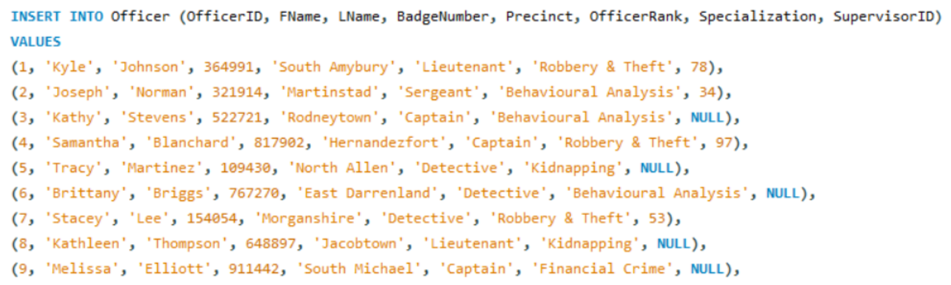
We explored two methods:

- **Method 1 – CSV Import:** We initially used the `LOAD DATA LOCAL INFILE` command to load data from CSV files. This method helped load bulk data for larger tables like `Officer`, `Crime`, `Victim`, etc. But this method was not efficient if someone else wants to look at the SQL code & understand what's happening as the person would have to also manage all 29 .csv files, edit the location of those files according to their unique system and then load them to use the SQL file.



```
#Loading Officer
LOAD DATA LOCAL INFILE 'C:\\Users\\Prachi\\Downloads\\Officer.csv'
INTO TABLE Officer
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(OfficerID, FName, LName, BadgeNumber, Precinct, OfficerRank, Specialization, SupervisorID);
```

- **Method 2 – Manual INSERT:** We wrote detailed `INSERT INTO` statements to input data manually for selected tables. This helped us control relational integrity and test query responses. Ths would also be efficient to share the SQL file, as the user would be able to run all the queries to load the tables in their system's memory.



```sql
INSERT INTO Officer (OfficerID, FName, LName, BadgeNumber, Precinct, OfficerRank, Specialization, SupervisorID)
VALUES
(1, 'Kyle', 'Johnson', 364991, 'South Amybury', 'Lieutenant', 'Robbery & Theft', 78),
(2, 'Joseph', 'Norman', 321914, 'Martinstad', 'Sergeant', 'Behavioural Analysis', 34),
(3, 'Kathy', 'Stevens', 522721, 'Rodneytown', 'Captain', 'Behavioural Analysis', NULL),
(4, 'Samantha', 'Blanchard', 817902, 'Hernandezfort', 'Captain', 'Robbery & Theft', 97),
(5, 'Tracy', 'Martinez', 109430, 'North Allen', 'Detective', 'Kidnapping', NULL),
(6, 'Brittany', 'Briggs', 767270, 'East Darrenland', 'Detective', 'Behavioural Analysis', NULL),
(7, 'Stacey', 'Lee', 154054, 'Morganshire', 'Detective', 'Robbery & Theft', 53),
(8, 'Kathleen', 'Thompson', 648897, 'Jacobtown', 'Lieutenant', 'Kidnapping', NULL),
(9, 'Melissa', 'Elliott', 911442, 'South Michael', 'Captain', 'Financial Crime', NULL),
```

Hence, with the help of ChatGPT, we generated clean, consistent SQL code to load our relational data. All tables were successfully populated, and major tables (like `Officer`, `Crime`, `Victim`, `Suspect`) contain 100+ entries each. This enabled the execution of complex queries and ensured referential integrity throughout the system.

Investigation Scenario

In our investigation, we analyzed the case of an **unidentified young woman found dead** in an **abandoned commercial warehouse**. The scene showed clear **signs of physical assault and no security surveillance**, indicating a planned violent act. The forensic team later reported drug traces in the victim's system, raising the possibility of substance-related intent. Since the victim's identity was unknown, we expanded the scope to explore links with **other unresolved crimes**, particularly **kidnapping and robbery cases**.

- Location: Commercial warehouse on the outskirts, marked abandoned
- Evidence Found: Blood samples, drug presence, and possible weapon residues
- No CCTV Footage: Scene lacked surveillance, increasing complexity
- Unknown Victim: No ID, leading to correlation with missing persons
- Investigation Angle: Homicide suspected, but kidnapping or robbery not ruled out
- Goal: Link victim profile, forensics, and suspect records to solve the case

This scenario gave us a realistic setup to test our data model, apply investigative SQL queries, and simulate how digital systems assist in solving serious crimes.

Solving the Case using SQL Queries

With the database structure in place and the investigation scenario clearly defined, we proceeded to design a series of SQL queries to simulate how law enforcement might explore different angles of the case. Each query is carefully crafted to address a specific investigative question. The queries make full use of the relational model through joins, subqueries, views, and triggers, allowing us to extract meaningful insights and progress toward solving the case. Below, we present each query with its purpose, SQL code, output, and interpretation:

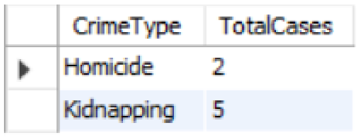
Query 1 - Homicide and Kidnapping Occurred in Commercial Area

Purpose: To identify patterns of serious violent crimes (specifically homicide and kidnapping) in commercial areas, which could signal the need for increased patrol or public safety measures in business-heavy districts.

SQL Query:

```sql
SELECT C.CrimeType, COUNT(*) AS TotalCases
FROM crime C
JOIN crime_scene CS ON C.CrimeID = CS.CrimeID
WHERE CS.LocationType = 'Commercial'
  AND C.CrimeType IN ('Homicide', 'Kidnapping')
GROUP BY C.CrimeType
HAVING COUNT(*) > 1;
```

Output:



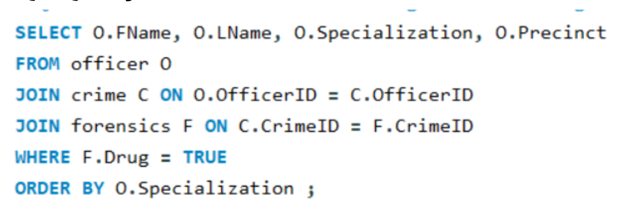
CrimeType	TotalCases
Homicide	2
Kidnapping	5

Interpretation:
The results show that kidnapping and homicide cases are significantly present in commercial areas, with 5 and 2 cases respectively.
This indicates the presence of high-risk crimes in zones typically perceived as business safe. This insight can guide law enforcement in focusing surveillance and prevention strategies in these areas.

Query 2 - List of Officers Assigned to Drug-Related Cases

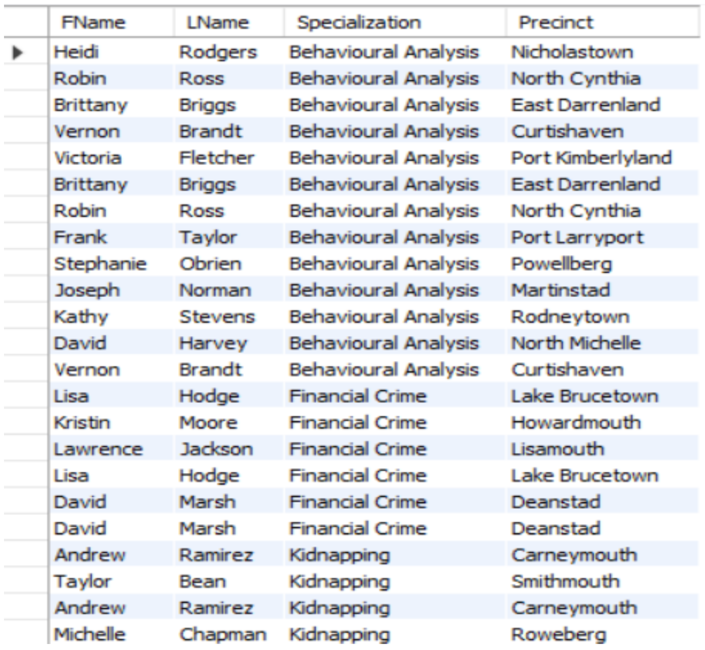
Purpose: To track down officers responsible for active drug-linked investigations, helping departments coordinate better and escalate cases when necessary.

SQL Query:



```
SELECT O.FName, O.LName, O.Specialization, O.Precinct
FROM officer O
JOIN crime C ON O.OfficerID = C.OfficerID
JOIN forensics F ON C.CrimeID = F.CrimeID
WHERE F.Drug = TRUE
ORDER BY O.Specialization ;
```

Output:



FName	LName	Specialization	Precinct
Heidi	Rodgers	Behavioural Analysis	Nicholastown
Robin	Ross	Behavioural Analysis	North Cynthia
Brittany	Briggs	Behavioural Analysis	East Darrenland
Vernon	Brandt	Behavioural Analysis	Curtishaven
Victoria	Fletcher	Behavioural Analysis	Port Kimberlyland
Brittany	Briggs	Behavioural Analysis	East Darrenland
Robin	Ross	Behavioural Analysis	North Cynthia
Frank	Taylor	Behavioural Analysis	Port Larryport
Stephanie	Obrien	Behavioural Analysis	Powellberg
Joseph	Norman	Behavioural Analysis	Martinstad
Kathy	Stevens	Behavioural Analysis	Rodneytown
David	Harvey	Behavioural Analysis	North Michelle
Vernon	Brandt	Behavioural Analysis	Curtishaven
Lisa	Hodge	Financial Crime	Lake Brucetown
Kristin	Moore	Financial Crime	Howardmouth
Lawrence	Jackson	Financial Crime	Lisamouth
Lisa	Hodge	Financial Crime	Lake Brucetown
David	Marsh	Financial Crime	Deanstad
David	Marsh	Financial Crime	Deanstad
Andrew	Ramirez	Kidnapping	Carneymouth
Taylor	Bean	Kidnapping	Smithmouth
Andrew	Ramirez	Kidnapping	Carneymouth
Michelle	Chapman	Kidnapping	Roweberg

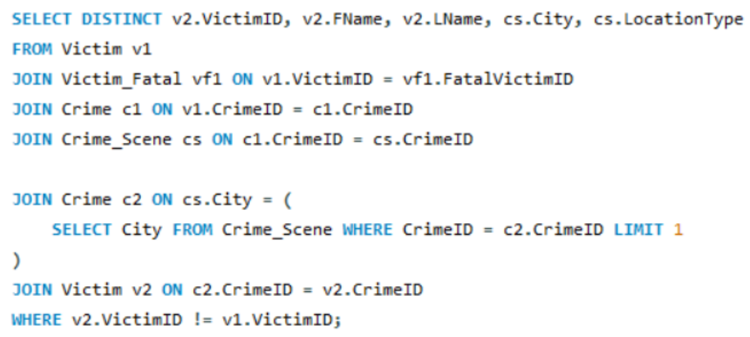
Interpretation:

- The output lists officers from various precincts working on drug-related crimes.
- This centralized view assists internal coordination and resource allocation by understanding which officers specialize in what domains are currently focused.

Query 3 - More victims with fatal injuries found in similar locations

Purpose: To identify whether certain locations tend to have repeated crimes linked to fatal injuries.

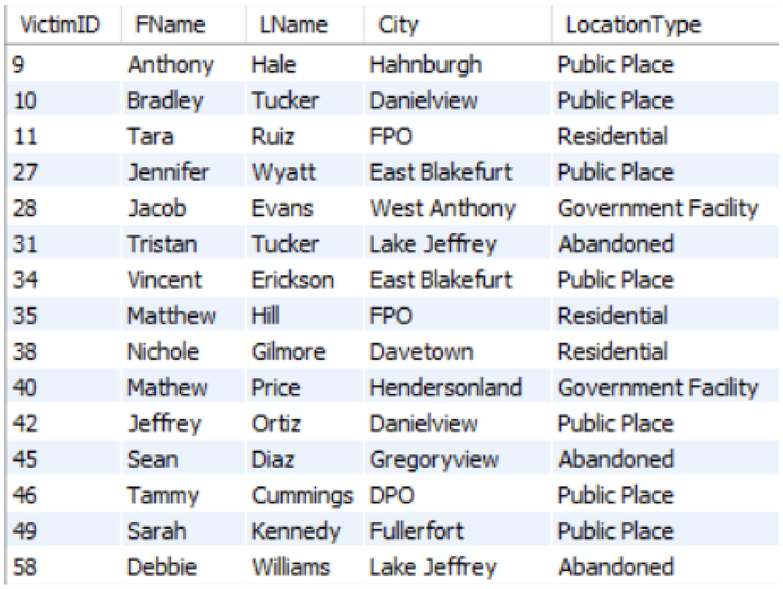
SQL Query:



```
SELECT DISTINCT v2.VictimID, v2.FName, v2.LName, cs.City, cs.LocationType
FROM Victim v1
JOIN Victim_Fatal vf1 ON v1.VictimID = vf1.FatalVictimID
JOIN Crime c1 ON v1.CrimeID = c1.CrimeID
JOIN Crime_Scene cs ON c1.CrimeID = cs.CrimeID

JOIN Crime c2 ON cs.City = (
    SELECT City FROM Crime_Scene WHERE CrimeID = c2.CrimeID LIMIT 1
)
JOIN Victim v2 ON c2.CrimeID = v2.CrimeID
WHERE v2.VictimID != v1.VictimID;
```

Output:



VictimID	FName	LName	City	LocationType
9	Anthony	Hale	Hahnburgh	Public Place
10	Bradley	Tucker	Danielview	Public Place
11	Tara	Ruiz	FPO	Residential
27	Jennifer	Wyatt	East Blakefurt	Public Place
28	Jacob	Evans	West Anthony	Government Facility
31	Tristan	Tucker	Lake Jeffrey	Abandoned
34	Vincent	Erickson	East Blakefurt	Public Place
35	Matthew	Hill	FPO	Residential
38	Nichole	Gilmore	Davetown	Residential
40	Mathew	Price	Hendersonland	Government Facility
42	Jeffrey	Ortiz	Danielview	Public Place
45	Sean	Diaz	Gregoryview	Abandoned
46	Tammy	Cummings	DPO	Public Place
49	Sarah	Kennedy	Fullerfort	Public Place
58	Debbie	Williams	Lake Jeffrey	Abandoned

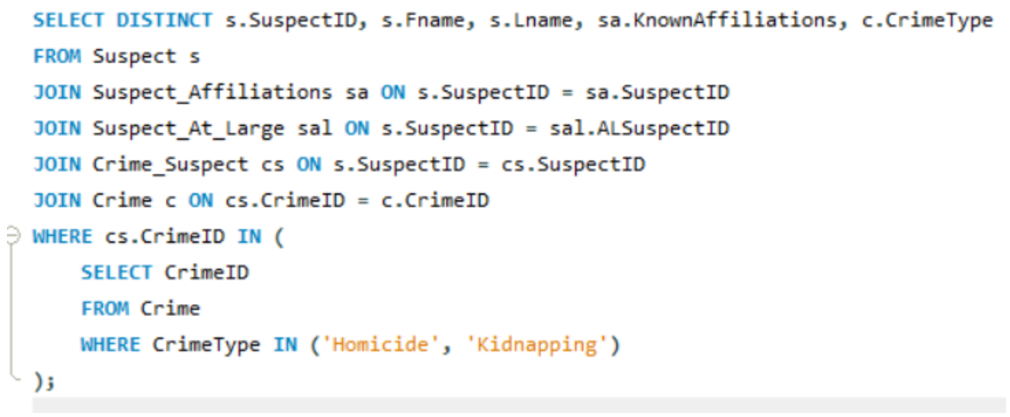
Interpretation:

- This query starts from victims who suffered fatal injuries and finds **other victims** connected to **different crimes in the same city**.
- It helps detect **geographical patterns** — whether certain cities or location types like **public places**, **residential areas**, or **government sites** repeatedly appear in fatal cases.
- This can alert investigators to **hotspots of criminal activity** or **recurring danger zones**.

Query 4 - Suspects at large, affiliated with a group, involved in Homicide or Kidnapping

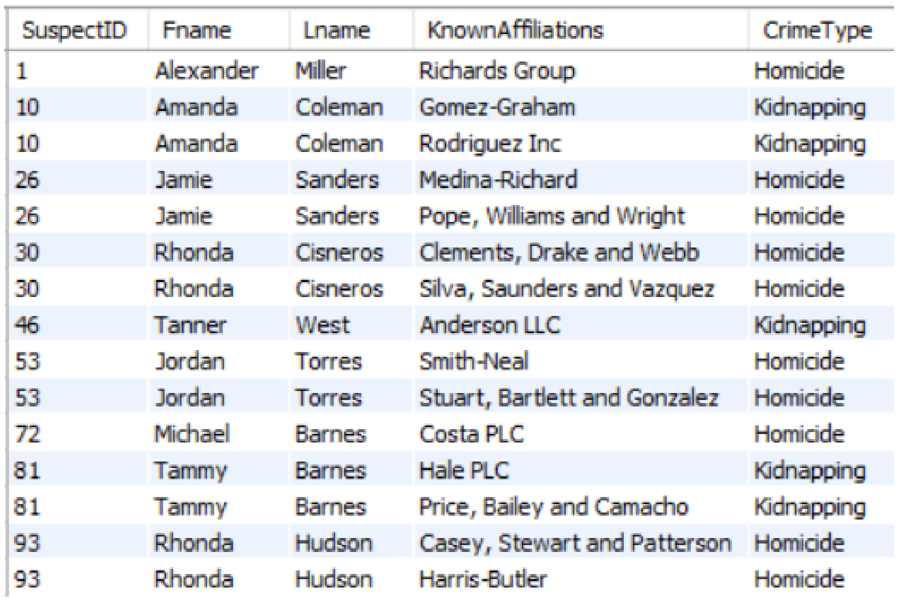
Purpose: To flag high-risk suspects who are still free, have known affiliations, and are involved in serious violent crimes.

SQL Query:



```
SELECT DISTINCT s.SuspectID, s.Fname, s.Lname, sa.KnownAffiliations, c.CrimeType
FROM Suspect s
JOIN Suspect_Affiliations sa ON s.SuspectID = sa.SuspectID
JOIN Suspect_At_Large sal ON s.SuspectID = sal.ALSuspectID
JOIN Crime_Suspect cs ON s.SuspectID = cs.SuspectID
JOIN Crime c ON cs.CrimeID = c.CrimeID
WHERE cs.CrimeID IN (
    SELECT CrimeID
    FROM Crime
    WHERE CrimeType IN ('Homicide', 'Kidnapping')
);
```

Output:



SuspectID	Fname	Lname	KnownAffiliations	CrimeType
1	Alexander	Miller	Richards Group	Homicide
10	Amanda	Coleman	Gomez-Graham	Kidnapping
10	Amanda	Coleman	Rodriguez Inc	Kidnapping
26	Jamie	Sanders	Medina-Richard	Homicide
26	Jamie	Sanders	Pope, Williams and Wright	Homicide
30	Rhonda	Cisneros	Clements, Drake and Webb	Homicide
30	Rhonda	Cisneros	Silva, Saunders and Vazquez	Homicide
46	Tanner	West	Anderson LLC	Kidnapping
53	Jordan	Torres	Smith-Neal	Homicide
53	Jordan	Torres	Stuart, Bartlett and Gonzalez	Homicide
72	Michael	Barnes	Costa PLC	Homicide
81	Tammy	Barnes	Hale PLC	Kidnapping
81	Tammy	Barnes	Price, Bailey and Camacho	Kidnapping
93	Rhonda	Hudson	Casey, Stewart and Patterson	Homicide
93	Rhonda	Hudson	Harris-Butler	Homicide

Interpretation: This query combines four conditions — the suspect is:

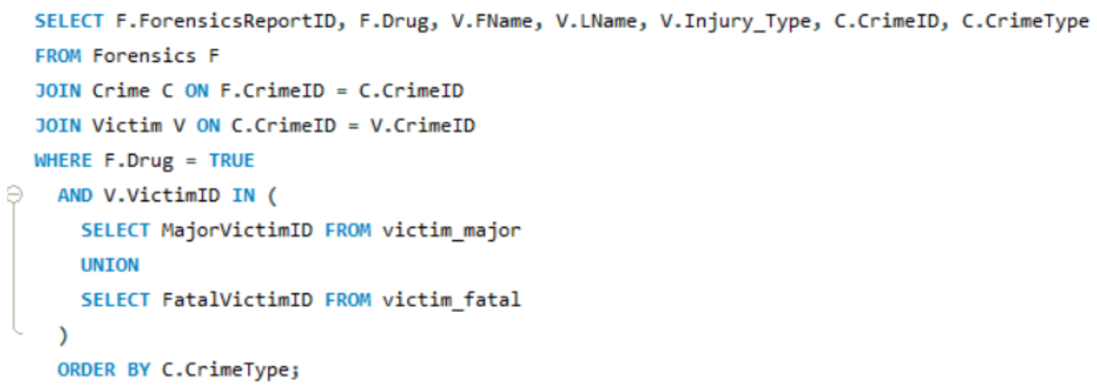
- *At large* (not yet apprehended),
- *Part of a known group or affiliation*,
- *Linked to a crime* through the `Crime_Suspect` table,
- And the crime is either a **Homicide** or **Kidnapping**.

This result can help law enforcement prioritize suspects who are **still roaming**, may be operating **within a network**, and are involved in **high-severity cases**. It's a proactive profiling method to focus resources on those most likely to re-offend or act in coordination.

Query 5 – Drug Traces and Severe Victim Injuries

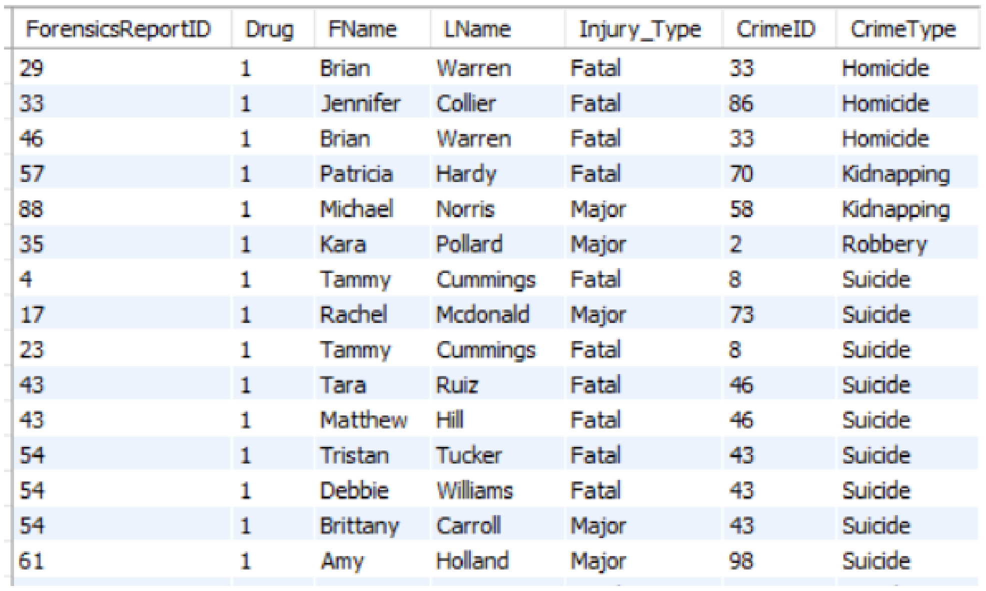
Purpose: To connect drug-related forensic evidence with severe victim outcomes, allowing us to detect patterns and understand if the presence of drugs is linked to high levels of violence.

SQL Query:



```sql
SELECT F.ForensicsReportID, F.Drug, V.FName, V.LName, V.Injury_Type, C.CrimeID, C.CrimeType
FROM Forensics F
JOIN Crime C ON F.CrimeID = C.CrimeID
JOIN Victim V ON C.CrimeID = V.CrimeID
WHERE F.Drug = TRUE
  AND V.VictimID IN (
    SELECT MajorVictimID FROM victim_major
    UNION
    SELECT FatalVictimID FROM victim_fatal
  )
  ORDER BY C.CrimeType;
```

Output:



ForensicsReportID	Drug	FName	LName	Injury_Type	CrimeID	CrimeType
29	1	Brian	Warren	Fatal	33	Homicide
33	1	Jennifer	Collier	Fatal	86	Homicide
46	1	Brian	Warren	Fatal	33	Homicide
57	1	Patricia	Hardy	Fatal	70	Kidnapping
88	1	Michael	Norris	Major	58	Kidnapping
35	1	Kara	Pollard	Major	2	Robbery
4	1	Tammy	Cummings	Fatal	8	Suicide
17	1	Rachel	Mcdonald	Major	73	Suicide
23	1	Tammy	Cummings	Fatal	8	Suicide
43	1	Tara	Ruiz	Fatal	46	Suicide
43	1	Matthew	Hill	Fatal	46	Suicide
54	1	Tristan	Tucker	Fatal	43	Suicide
54	1	Debbie	Williams	Fatal	43	Suicide
54	1	Brittany	Carroll	Major	43	Suicide
61	1	Amy	Holland	Major	98	Suicide

Interpretation:

- The output shows multiple forensic reports where the drug presence is confirmed and the victims had major or fatal injuries.
- Notably, drug traces are found across Homicide, Kidnapping, and Suicide cases - indicating that drug involvement is not limited to one crime type and can point to a possible recurring factor in violent offenses.

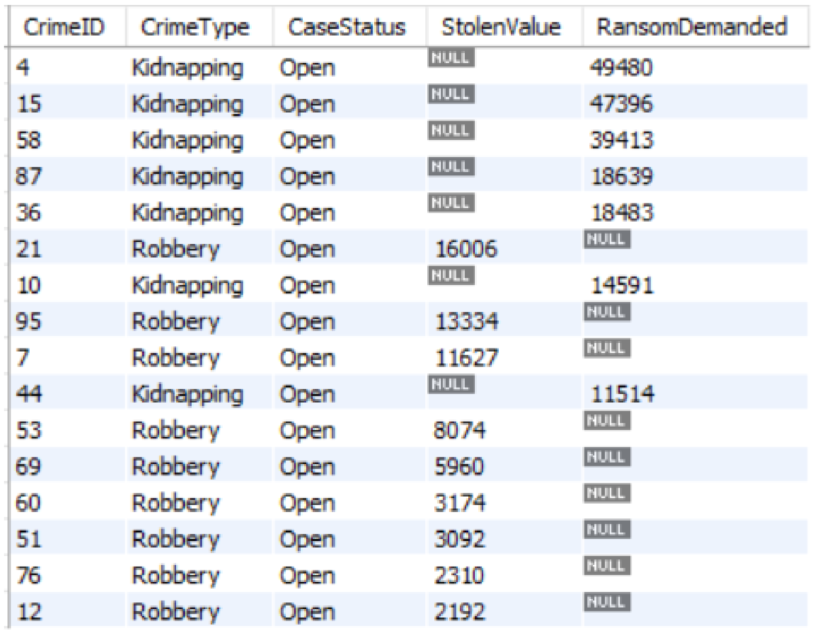
Query 6 – Open Cases with High Ransom or Stolen Value

Purpose: To help investigators explore whether the current homicide victim could be linked to financially motivated crimes like unresolved kidnappings or robberies, especially where the victim might still be unidentified or missing.

SQL Query:

```
CREATE VIEW View_Open_HighValue_Cases AS
SELECT C.CrimeID, C.CrimeType, C.CaseStatus, R.StolenValue, K.RansomDemanded
FROM Crime C
LEFT JOIN Crime_Robbery R ON C.CrimeID = R.RobberyCrimeID
LEFT JOIN Crime_Kidnapping K ON C.CrimeID = K.KidnappingCrimeID
WHERE C.CaseStatus = 'Open' AND C.CrimeType IN ('Robbery', 'Kidnapping')
ORDER BY
    GREATEST(IFNULL(R.StolenValue, 0), IFNULL(K.RansomDemanded, 0)) DESC;
```

Output:



CrimeID	CrimeType	CaseStatus	StolenValue	RansomDemanded
4	Kidnapping	Open	NULL	49480
15	Kidnapping	Open	NULL	47396
58	Kidnapping	Open	NULL	39413
87	Kidnapping	Open	NULL	18639
36	Kidnapping	Open	NULL	18483
21	Robbery	Open	16006	NULL
10	Kidnapping	Open	NULL	14591
95	Robbery	Open	13334	NULL
7	Robbery	Open	11627	NULL
44	Kidnapping	Open	NULL	11514
53	Robbery	Open	8074	NULL
69	Robbery	Open	5960	NULL
60	Robbery	Open	3174	NULL
51	Robbery	Open	3092	NULL
76	Robbery	Open	2310	NULL
12	Robbery	Open	2192	NULL

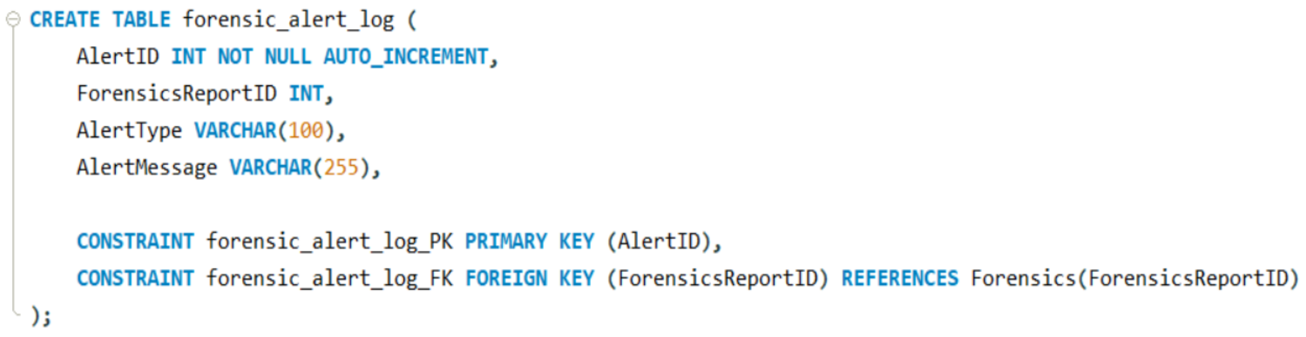
Interpretation:

- The output lists several open cases with substantial financial stakes — especially kidnapping cases with ransom demands above $45,000 and robberies with stolen values above $10,000.
- These cases deserve attention because they may be related to unresolved victim identities or criminal motivations.

Query 7 - Creating Alerts for Toxic Forensic Entries

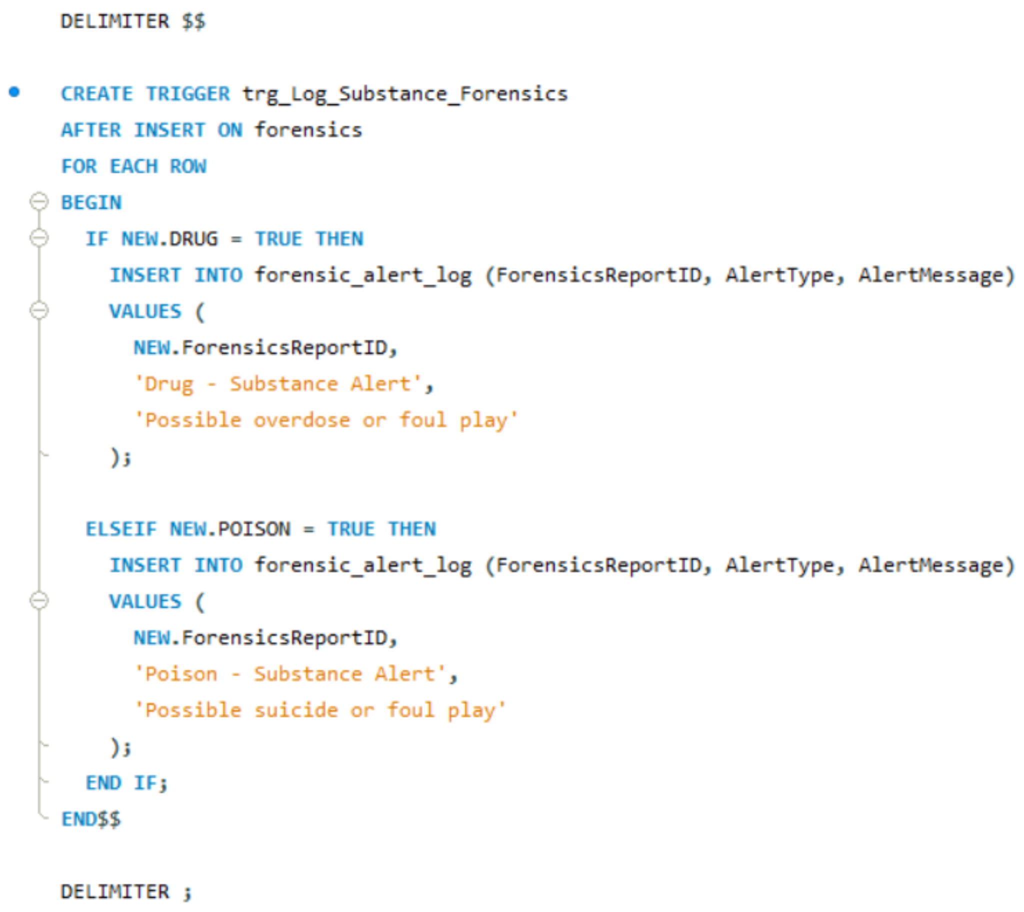
Purpose: To automate the detection of high-risk forensic evidence and notify analysts or officers when toxic substances like Drug or Poison are found, allowing early intervention or prioritization.

SQL Query:



```
CREATE TABLE forensic_alert_log (
    AlertID INT NOT NULL AUTO_INCREMENT,
    ForensicsReportID INT,
    AlertType VARCHAR(100),
    AlertMessage VARCHAR(255),

    CONSTRAINT forensic_alert_log_PK PRIMARY KEY (AlertID),
    CONSTRAINT forensic_alert_log_FK FOREIGN KEY (ForensicsReportID) REFERENCES Forensics(ForensicsReportID)
);
```



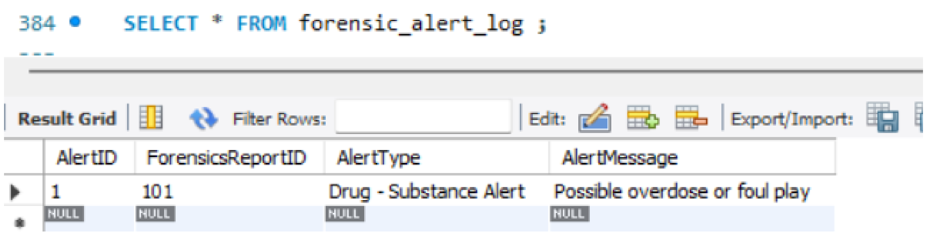
```
DELIMITER $$

CREATE TRIGGER trg_Log_Substance_Forensics
AFTER INSERT ON forensics
FOR EACH ROW
BEGIN
  IF NEW.DRUG = TRUE THEN
    INSERT INTO forensic_alert_log (ForensicsReportID, AlertType, AlertMessage)
    VALUES (
      NEW.ForensicsReportID,
      'Drug - Substance Alert',
      'Possible overdose or foul play'
    );

  ELSEIF NEW.POISON = TRUE THEN
    INSERT INTO forensic_alert_log (ForensicsReportID, AlertType, AlertMessage)
    VALUES (
      NEW.ForensicsReportID,
      'Poison - Substance Alert',
      'Possible suicide or foul play'
    );
  END IF;
END$$

DELIMITER ;
```

Output:



```
SELECT * FROM forensic_alert_log ;
```

AlertID	ForensicsReportID	AlertType	AlertMessage
1	101	Drug - Substance Alert	Possible overdose or foul play
NULL	NULL	NULL	NULL

Interpretation:

- The output shows that CrimeID 101 (The current case of a murder added as a new record) triggered an alert due to the presence of drug evidence.
- This confirms that our alert system works as intended — flagging forensic entries the moment they are inserted, allowing the case to be escalated for review.
- Given that CrimeID 101 involves an unidentified female victim with signs of drug traces, this alert helps prioritize the case as high-risk and possibly connected to organized or substance-driven crime.

Conclusion

This project successfully demonstrated how relational database design and SQL queries can be used to simulate real-world crime investigation workflows. By building a robust data model for the Crime Data Management Bureau (CDMB), we were able to manage, link, and analyze data related to crimes, suspects, victims, forensics, and law enforcement officers.

The SQL queries we developed enabled us to explore multiple angles of a complex homicide case, uncover relationships between data points, and generate actionable insights such as identifying at-large suspects, flagging high-risk forensic evidence, and tracking criminal patterns in specific locations.

The project highlighted the power of structured data and query logic in criminal investigations, reinforcing how well-built databases can support real-time decision-making and long-term public safety strategies.

Benefits Gained

Working on the CDMB project allowed us to apply theoretical database concepts in a highly practical and realistic setting. Below are the key benefits we experienced during the course of this project:

- **Practical Application of Concepts:** We were able to apply advanced database concepts such as normalization, subtype modeling, and multivalued attributes in a real-world scenario.

- **Strong SQL Skill Building:** Writing joins, nested subqueries, views, and triggers deepened our understanding of SQL for analytical problem-solving.

- **Data Modeling Confidence:** Creating and validating our ER diagram, relational schema, and EER model strengthened our ability to model complex domains with multiple relationships and constraints.

- **Analytical Thinking:** Designing queries that mimicked how law enforcement solves crimes helped us think logically, make data-driven assumptions, and validate hypotheses.

- **Automation Readiness:** Implementing alerts and views prepared us for understanding how systems can go from passive data storage to active investigation support.

Challenges Encountered During the Project

While the project was rewarding, it also came with a range of challenges that tested our problem-solving and adaptability. These challenges occurred both during the database design phase and while loading and querying the data. However, each issue helped us learn and improve our approach. The main challenges we encountered, along with how we addressed them, are listed below:

- **Issues with Loading Data via CSV:** While bulk loading using .csv files was initially faster, it became problematic due to system-specific file paths and the need to load 29 separate files in the correct order. We shifted to manual INSERT INTO statements, which improved control, readability, and ease of sharing the SQL file.

- **Errors in Generating the EER Diagram through Reverse Engineering:** MySQL Workbench generated incorrect 1:M relationships between subtype and supertype tables instead of the intended 1:1 mapping. We manually adjusted these relationships to reflect correct subtype implementation and ensure the model aligned with our design rules.

- **Query Logic Returning No Results:** Some initial queries didn't yield any output due to mismatches in mock data. We revised our SQL conditions and expanded the dataset to ensure better coverage across entities.

- **Modeling Specialization Correctly:** Applying disjoint vs. overlap and total vs. partial specialization took time to reason through. We referred back to the business rules and real-world logic to finalize the correct structure.

- **Ensuring Referential Integrity During Data Generation:** Maintaining foreign key relationships in synthetic data was complex. We used Mockaroo's dataset linking features and carefully sequenced inserts to preserve data consistency.

- **Trigger Functionality Testing:** Ensuring the alert trigger worked correctly for forensic entries with drugs or poison required trial and error. We validated the logic by inserting test cases and reviewing the resulting log table to confirm success.

Despite the challenges faced, this project proved to be a valuable learning experience that strengthened our technical, analytical, and problem-solving abilities. It gave us a deeper appreciation for how structured data and thoughtful design can support critical decision-making in real-world domains like law enforcement. We leave this project with a solid understanding of database systems and their role in solving complex, data-driven problems.