

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



University of Management and Technology

Semester Project Report

Database Systems

Submitted by:

Talha Khalid (F2021266625)

Saad Hameed (F2021266496)

Muhammad Shujaat Ali (F2020266033)

Section: V-06

Semester: 4th (Spring 2023)

Submitted to:

Sir Muhammad Ehsan

Project Title

Criminal Database Management System

Table of contents

| | |
|---|----|
| ABSTRACT..... | 6 |
| Aim | 6 |
| Description..... | 7 |
| Features | 7 |
| Entities: | 9 |
| Technologies Used | 9 |
| Table Description/Attributes | 10 |
| Relationships..... | 11 |
| Entity Relationship Diagram | 13 |
| Relational Database Model Diagram | 14 |
| Data Flow Diagram..... | 15 |
| Relational Database Schema..... | 16 |
| Create Query: Creating all tables | 18 |
| Insert Query: Inserting Records | 21 |
| DDL Commands..... | 24 |
| DML Commands..... | 25 |
| Joins | 27 |
| Views..... | 31 |
| Stored Procedures | 33 |
| Conclusion..... | 34 |

Chapter 1: Introduction to the problem

ABSTRACT

The Criminal Database Management System (CDMS) is a comprehensive software solution aimed at facilitating the efficient management and organization of criminal records and related information. In today's complex law enforcement landscape, the need for an advanced system to store, retrieve, and analyze criminal data is paramount. The CDMS offers a user-friendly interface for law enforcement agencies to input, search, and update criminal records, including personal information, charges, convictions, and associated documents. The system tries to ensure data integrity and confidentiality.

Furthermore, the CDMS features powerful search and reporting capabilities, enabling agencies to quickly access relevant information, generate comprehensive reports, and aid investigations. By centralizing and streamlining criminal data management, the CDMS enhances operational effectiveness, improves data accuracy, and promotes information sharing among law enforcement entities. Ultimately, the implementation of this system leads to better crime prevention, detection, contributing to a safer and more secure society.

Overall, the Criminal Database Management System plays a crucial role in strengthening crime prevention, detection, and apprehension.

These were the general reasons that why a criminal database management system should be interduce in society. So, now in our case, we have seen a lot of times on our news channels and in our society that still, the same old File management system is used for storing records in the Police Stations which is quite tiresome and is not suitable according to the upcoming age. In our Police Stations, still the Police store criminal records in physical files. And when they want to search for a particular file, it takes too much time.

However, with the Criminal database management system that we are proposing, it would be so easy to enter the record of a criminal, like his age, his crime, his previous history in the Police Station. And if we want to search any criminal, it can also be done in micro-seconds. There is just the need of a simple query to be entered and all the information related to the criminal would be displayed. So, I think this would be a revolution to the system currently being used and that's what we are striving for.

Aim

So basically, we're creating a criminal database system for Pakistan. The aim is to help law enforcement agencies keep track of criminals, arrests, crimes, sentences, victims, witnesses, and evidence. This will make it easier for them to solve crimes and bring justice to victims. Plus, it'll also help in reducing crime rates and facilitating efficient investigations. We hope this project will contribute to making Pakistan a safer place for everyone.

Description

The Criminal Database System project aims to design and implement a database that can store and manage information related to crimes, criminals, victims, officers, investigations, and evidence. The database system will be developed using the relational database model and will be used by law enforcement agencies and other authorized personnel to access criminal records and information related to criminal cases.

Features

The Criminal Database System project will include the following features:

- 1. Data Entry and Management:** The system will allow authorized users to enter and manage data related to criminal cases, investigations, evidence, and other related information. The data will be stored in the database tables and can be easily accessed, updated, and deleted by authorized users.
- 2. Case Management:** The system will allow authorized users to manage criminal cases by adding, updating, and deleting case information. Users will be able to view the details of a particular case, such as the case name, date, registered location, and related criminal records.
- 3. Criminal Records Management:** The system will store and manage information related to criminals, such as their name, date of birth, gender, height, weight, and any known aliases.
- 4. Evidence Management:** The system will store and manage information related to evidence, such as the type of evidence, its location, and the case to which it is related.
- 5. Investigation Management:** The system will allow authorized users to manage criminal investigations by adding, updating, and deleting investigation information. Users will be able to view the details of an investigation, such as the start and end date, the officer assigned to the investigation, and the related case.
- 6. Reporting:** The system will generate reports on criminal cases, investigations, evidence, and other related information. Users will be able to filter the data based on various parameters such as date range, location, type of crime, and other criteria.

Chapter 2: Logical Database Design

Entities:

The database design for the Criminal Database System project will be based on the relational model. The ERD will include the following entities:

- Officer
- Arrest
- Crime
- Criminal
- Criminal Alias
- Jail
- Sentence
- Cases
- Victim
- Witness
- Suspect
- Evidence
- Investigation

The relationships between the entities will be defined using foreign keys and primary keys.

Technologies Used:

The Criminal Database System project will be developed using the following technologies:

- **SQL Server Management Studio:** The database system will be implemented using the SQL Server Management Studio relational database management system.
- **MySQL**

Table Description/Attributes

1. **Officer:** This table stores information about law enforcement officers, including their badge number, name, and rank.
2. **Arrest:** This table stores information about arrests, including the arrest ID, date, time, location, reason, and the badge number of the officer who made the arrest.
3. **Crime:** This table stores information about crimes, including the crime ID, type, location, date, and time.
4. **Criminal:** This table stores information about criminals, including their criminal ID, name, date of birth, gender, race, height, and weight.
5. **Criminal_Alias:** This table stores information about aliases used by criminals, including the alias ID, alias name, and the criminal ID that it belongs to.
6. **Jail:** This table stores information about jails, including their jail ID, name, and location.
7. **Sentence:** This table stores information about sentences, including the sentence ID, length, the crime ID that it pertains to, and the jail ID where the sentence is being served.
8. **Cases:** This table stores information about criminal cases, including the case ID, name, location, date, and time.
9. **Victim:** This table stores information about victims, including the victim ID, name, gender, race, age, and the case ID that they pertain to.
10. **Witness:** This table stores information about witnesses, including the witness ID, name, gender, race, age, and the case ID that they pertain to.
11. **Suspect:** This table stores information about suspects, including the suspect ID, name, gender, race, age, and the case ID that they pertain to.
12. **Evidence:** This table stores information about evidence, including the evidence ID, type, location, and the case ID that it pertains to.

13. **Investigation:** This table stores information about investigations, including the start date, end date, the badge number of the officer in charge of the investigation, and the case ID that the investigation pertains to.

Relationships

The relationships between the entities can be explained as follows:

1. **Relation between Officer and Arrest table**

The Officer table has a many-to-many relationship with the Arrest table. An officer can make many arrests, on the other hand, an arrest can be made by one or more officers. Therefore, the Officer table's **primary key (Badge_Number)** is used as a **foreign key** in the Arrest table to associate each arrest with the corresponding officer.

2. **Relation between Arrest and Crime table**

The Arrest table has a many-to-many relationship with the Crime table. An arrest corresponds to one or more crimes, on the other hand, a crime can have multiple arrests. Therefore, the Crime table's **primary key (Crime_Id)** is used as a **foreign key** in the Arrest table to associate each arrest with the corresponding crime.

3. **Relation between Criminal and Crime table**

The Criminal table has a many-to-many relationship with the Crime table. One or many criminals can involve in one or multiple crimes. Therefore, the Criminal table's **primary key (Criminal_Id)** is used as a **foreign key** in the Crime table to associate each criminal with the corresponding crime.

4. **Relation between Criminal and Criminal_Alias table**

The Criminal table has a many-to-many relationship with the Criminal_Alias table. A criminal can have one or many aliases, but each alias corresponds to one or more criminals. Therefore, the Criminal table's **primary key (Criminal_Id)** is used as a **foreign key** in the Criminal_Alias table to associate each alias with the corresponding criminal.

5. **Relation between Sentence, Crime and Jail table**

The Sentence table has a many-to-many relationship with the Crime and Jail tables. A sentence corresponds to one or more crimes and jails, but a crime or jail can have multiple sentences. Therefore, the Crime and Jail tables' **primary keys (Crime_Id and Jail_Id, respectively)** are used as **foreign keys** in the Sentence table to associate each sentence with the corresponding crime and jail.

6. **Relation between Cases, Victim, Witness, Suspect and Evidence table**

The Cases table has a many-to-many relationship with the Victim, Witness, Suspect and Evidence tables. A case can have one or multiple victims, witnesses, suspects, evidence, but each victim, witness, suspect, evidence corresponds to one or many cases. Therefore, the Cases table's **primary key (Case_Id)** is used as a **foreign key** in each of these tables to associate each record with the corresponding case.

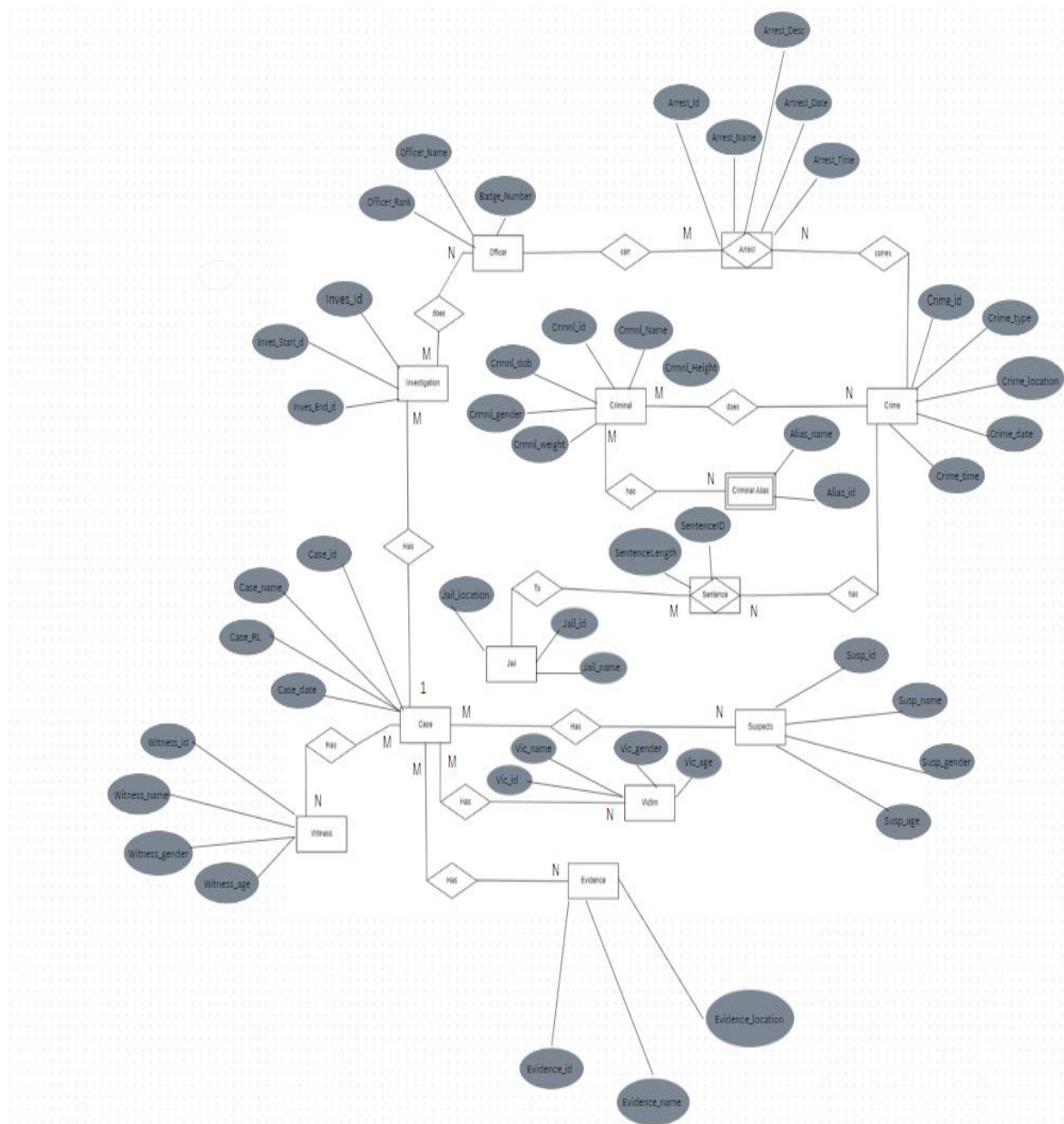
7. **Relation between Cases and Investigation table**

The Cases table has a one-to-many relationship with the Investigation table. A case can have one or multiple investigations and each investigation corresponds one case. Therefore, the Cases table's **primary key (Case_Id)** is used as a **foreign key** in the Investigation table to associate each record with the corresponding case.

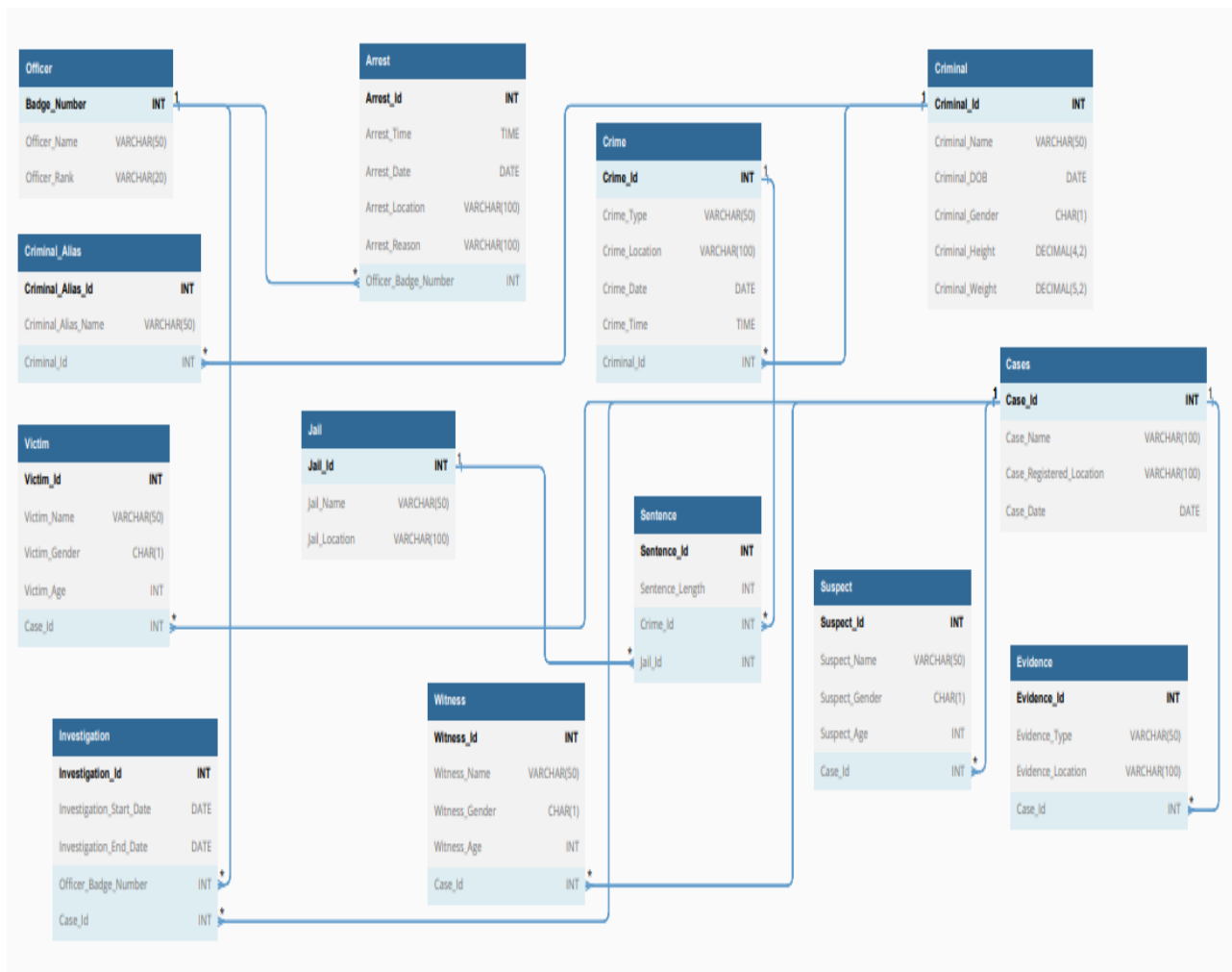
8. **Relation between Investigation and Officer table**

The Investigation table has a many-to-many relationship with the Officer table. An investigation corresponds to one or multiple officers, but one or more officers can have multiple investigations. Therefore, the Officer tables' **primary keys (Badge_Number respectively)** are used as **foreign key** in the Investigation table to associate each investigation with the corresponding office.

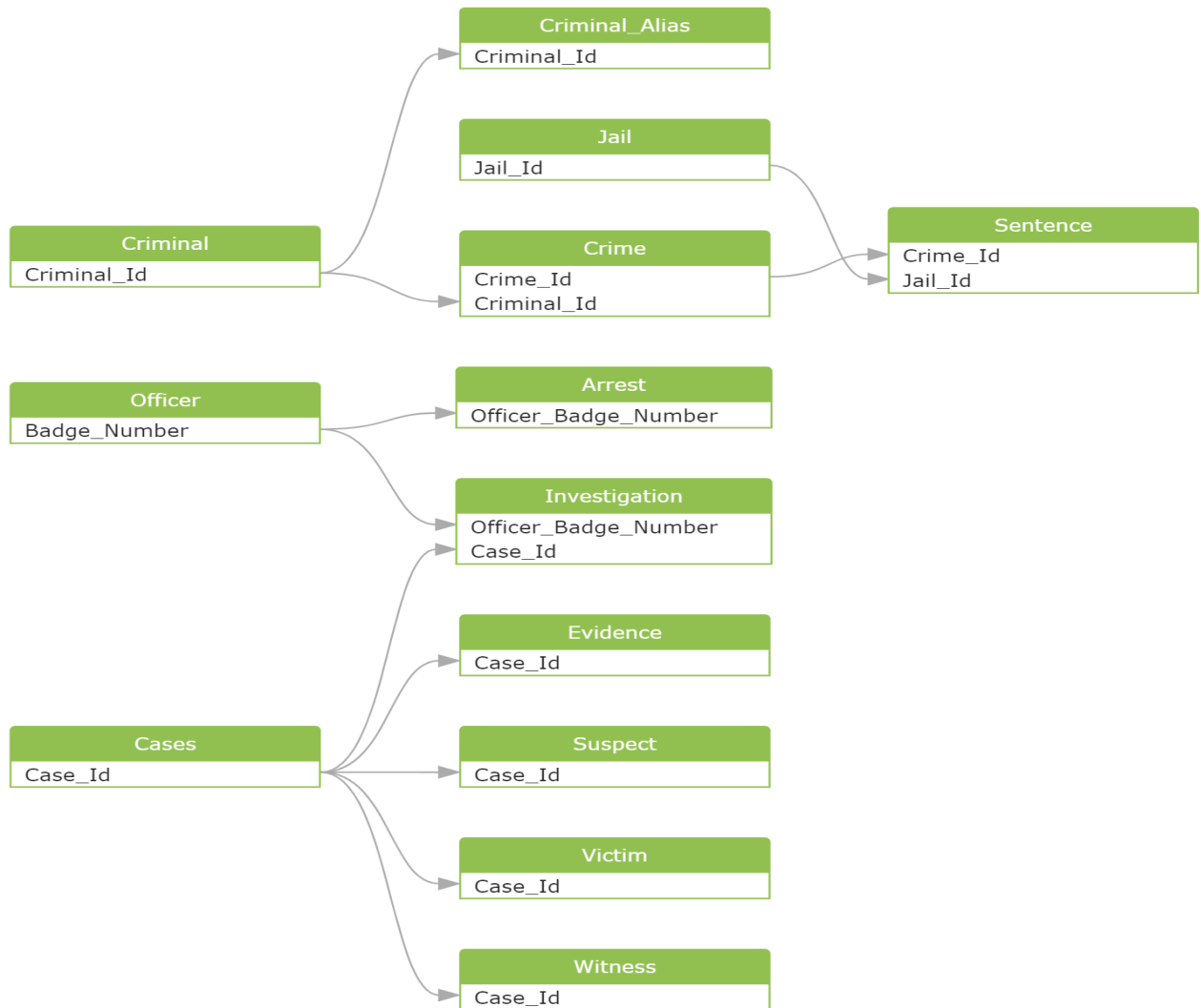
Entity Relationship Diagram



Relational Database Model Diagram



Data Flow Diagram



Relational Database Schema

The relational database schema for Criminal Management database is as follows:

1. **Officer** (Badge_Number, Officer_Name, Officer_Rank)
2. **Arrest** (Arrest_Id, Arrest_Time, Arrest_Date, Arrest_Location, Arrest_Reason, Officer_Badge_Number)
3. **Crime** (Crime_Id, Crime_Type, Crime_Location, Crime_Date, Crime_Time, Criminal_Id)
4. **Criminal** (Criminal_Id, Criminal_Name, Criminal_DOB, Criminal_Gender, Criminal_Height, Criminal_Weight)
5. **Criminal_Alias** (Criminal_Alias_Id, Criminal_Alias_Name, Criminal_Id)
6. **Jail** (Jail_Id, Jail_Name, Jail_Location);
7. **Sentence** (Sentence_Id, Sentence_Length, Crime_Id, Jail_Id)
8. **Cases** (Case_Id, Case_Name, Case_Registered_Location, Case_Date)
9. **Victim** (Victim_Id, Victim_Name, Victim_Gender, Victim_Age, Case_Id)
10. **Witness** (Witness_Id, Witness_Name, Witness_Gender, Witness_Age, Case_Id)
11. **Suspect** (Suspect_Id, Suspect_Name, Suspect_Gender, Suspect_Age, Case_Id)
12. **Evidence** (Evidence_Id, Evidence_Type, Evidence_Location, Case_Id)
13. **Investigation** (Officer_Badge_Number, Case_Id, Investigation_Start_Date, Investigation_End_Date)

Chapter 3: Physical Database Design

Create Query: Creating all tables

```
CREATE TABLE Officer(  
    Badge_Number INT PRIMARY KEY,  
    Officer_Name VARCHAR(50) NOT NULL,  
    Officer_Rank VARCHAR(20)  
);  
  
CREATE TABLE Arrest(  
    Arrest_Id INT PRIMARY KEY,  
    Arrest_Time TIME NOT NULL,  
    Arrest_Date DATE NOT NULL,  
    Arrest_Location VARCHAR(100) NOT NULL,  
    Arrest_Reason VARCHAR(100) NOT NULL,  
    Officer_Badge_Number INT,  
    FOREIGN KEY (Officer_Badge_Number) REFERENCES Officer(Badge_Number)  
);  
  
CREATE TABLE Crime(  
    Crime_Id INT PRIMARY KEY,  
    Crime_Type VARCHAR(50) NOT NULL,  
    Crime_Location VARCHAR(100) NOT NULL,  
    Crime_Date DATE NOT NULL,  
    Crime_Time TIME NOT NULL,  
    Criminal_Id INT,  
    FOREIGN KEY (Criminal_Id) REFERENCES Criminal(Criminal_Id)  
);  
  
CREATE TABLE Criminal(  
    Criminal_Id INT PRIMARY KEY,  
    Criminal_Name VARCHAR(50) NOT NULL,  
    Criminal_DOB DATE NOT NULL,  
    Criminal_Gender CHAR(1) NOT NULL,  
    Criminal_Height DECIMAL(4,2) NOT NULL,  
    Criminal_Weight DECIMAL(5,2) NOT NULL  
);
```

```
CREATE TABLE Criminal_Alias(  
    Criminal_Alias_Id INT PRIMARY KEY,  
    Criminal_Alias_Name VARCHAR(50) NOT NULL,  
    Criminal_Id INT,  
    FOREIGN KEY (Criminal_Id) REFERENCES Criminal(Criminal_Id)  
);
```

```
CREATE TABLE Jail(  
    Jail_Id INT PRIMARY KEY,  
    Jail_Name VARCHAR(50) NOT NULL,  
    Jail_Location VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Sentence(  
    Sentence_Id INT PRIMARY KEY,  
    Sentence_Length INT NOT NULL,  
    Crime_Id INT,  
    Jail_Id INT,  
    FOREIGN KEY (Crime_Id) REFERENCES Crime(Crime_Id),  
    FOREIGN KEY (Jail_Id) REFERENCES Jail(Jail_Id)  
);
```

```
CREATE TABLE Cases(  
    Case_Id INT PRIMARY KEY,  
    Case_Name VARCHAR(100) NOT NULL,  
    Case_Registered_Location VARCHAR(100) NOT NULL,  
    Case_Date DATE NOT NULL,  
);
```

```
CREATE TABLE Victim(  
    Victim_Id INT PRIMARY KEY,  
    Victim_Name VARCHAR(50) NOT NULL,  
    Victim_Gender CHAR(1) NOT NULL,  
    Victim_Age INT NOT NULL,  
    Case_Id INT,  
    FOREIGN KEY (Case_Id) REFERENCES Cases(Case_Id)  
);
```

```
CREATE TABLE Witness(  
    Witness_Id INT PRIMARY KEY,  
    Witness_Name VARCHAR(50) NOT NULL,  
    Witness_Gender CHAR(1) NOT NULL,  
    Witness_Age INT NOT NULL,  
    Case_Id INT,  
    FOREIGN KEY (Case_Id) REFERENCES Cases(Case_Id)  
);
```

```
CREATE TABLE Suspect(  
    Suspect_Id INT PRIMARY KEY,  
    Suspect_Name VARCHAR(50) NOT NULL,  
    Suspect_Gender CHAR(1) NOT NULL,  
    Suspect_Age INT NOT NULL,  
    Case_Id INT,  
    FOREIGN KEY (Case_Id) REFERENCES Cases(Case_Id)  
);
```

```
CREATE TABLE Evidence(  
    Evidence_Id INT PRIMARY KEY,  
    Evidence_Type VARCHAR(50) NOT NULL,  
    Evidence_Location VARCHAR(100) NOT NULL,  
    Case_Id INT,  
    FOREIGN KEY (Case_Id) REFERENCES Cases(Case_Id)  
);
```

```
CREATE TABLE Investigation(  
    Investigation_Start_Date DATE NOT NULL,  
    Investigation_End_Date DATE NOT NULL,  
    Officer_Badge_Number INT,  
    Case_Id INT,  
    PRIMARY KEY (Badge_Number, Case_Id),  
    FOREIGN KEY (Officer_Badge_Number) REFERENCES Officer(Badge_Number),  
    FOREIGN KEY (Case_Id) REFERENCES Cases(Case_Id)  
);
```

Insert Query: Inserting Records

```
INSERT INTO Officer
VALUES
(1, 'John Smith', 'Captain'),
(2, 'David Wilson', 'Major'),
(3, 'Emily Johnson', 'Lieutenant'),
(4, 'Samantha Davis', 'Colonel'),
(5, 'Michael Thompson', 'Lieutenant Colonel');
```

```
INSERT INTO Arrest
VALUES
(5678, '2023-05-10', '09:45', 'Main Street, City Center', 'Theft',1),
(2345, '2023-05-12', '02:30', 'Elm Street, East District', 'Assault',2),
(7890, '2023-05-02', '08:20', 'Oak Lane, South District', 'Drugs',1),
(1234, '2023-05-05', '11:10', 'Maple Avenue, North District', 'Robbery',4),
(4567, '2023-05-11', '02:14', 'Park Avenue, West District', 'Assault',2);
```

```
INSERT INTO Criminal
VALUES
(1,'John Smith', '2003-05-01', 'M', 1.83, 75.56),
(2,'Emily Johnson', '2002-05-02', 'F', 0.9, 60.87),
(3,'David Wilson', '2001-05-03', 'M', 1.12, 80.76),
(4,'Samantha Davis', '2003-05-04', 'F', 2.00, 55.22),
(5,'Michael Thompson', '2003-05-05', 'M', 1.31, 90.12);
```

```
INSERT INTO Crime
VALUES
(1,'Theft','Lahore','2023-05-02','05:34',1),
(2,'Assault','Islamabad','2023-05-11','05:11',2),
(3,'Robbery','Karachi','2023-06-21','02:33',3),
(4,'Theft','Lahore','2023-04-09','07:21',4),
(5,'Harrasment','Karachi','2023-05-04','05:44',5);
```

```
INSERT INTO Criminal_Alias
VALUES
(1,'Emily',1),
(2,'John',2),
(3,'Michael',3),
(4,'Charlotte',4),
(5,'Bruce',5);
```

```
INSERT INTO Jail
VALUES
(1,'Maplewood Correctional Facility','City Center, Maplewood'),
(2,'Riverside County Detention Center','Riverside, California'),
(3,'Oakridge Penitentiary','Oakridge, New York'),
(4,'Evergreen Correctional Institution','Evergreen, Texas'),
(5,'Pinehurst Correctional Center','Pinehurst, Illinois');
```

```
INSERT INTO Sentence
VALUES
(1,3,1,1),
(2,2,2,2),
(3,3,3,3),
(4,4,4,4),
(5,2,5,5);
```

```
INSERT INTO Cases
VALUES
(1,'John Smith','New York','2023-05-01'),
(2,'Emily Johnson','Los Angeles','2023-05-02'),
(3,'David Wilson','Chicago','2023-05-03'),
(4,'Samantha Davis','Houston','2023-05-04'),
(5,'Michael Thompson','Miami','2023-05-05');
```

```
INSERT INTO Victim
VALUES
(1,'John Smith','M',30,1),
(2,'Emily Johnson','F',25,2),
(3,'David Wilson','M',35,3),
(4,'Samantha Davis','F',28,4),
(5,'Michael Thompson','M',32,5);
```

```
INSERT INTO Witness
VALUES
(1,'Saad','M',20,1),
(2,'Talha','M',20,2),
(3,'Laiba','F',19,3),
(4,'Noman','M',26,4),
(5,'Shoaib','M',22,5);
```

```
INSERT INTO Suspect  
VALUES
```

```
(1,'Clark','M',26,1),  
(2,'Hector','M',44,2),  
(3,'Victoria','F',27,3),  
(4,'Fudge','M',22,4),  
(5,'Robertson','M',17,5);
```

```
INSERT INTO Evidence  
VALUES
```

```
(1,'Fingerprint', 'Forensic Lab',1),  
(2,'DNA Sample', 'Crime Scene A',2),  
(3,'Weapon', 'Evidence Locker B',3),  
(4,'Surveillance Footage', 'Security Room',4),  
(5,'Documents', 'Investigation Office',5);
```

```
INSERT INTO Investigation  
VALUES
```

```
(1,'2023-01-01', '2023-01-05',1,1),  
(2,'2023-02-10', '2023-02-15',2,2),  
(3,'2023-03-20', '2023-03-25',3,3),  
(4,'2023-04-05', '2023-04-10',4,4),  
(5,'2023-05-15', '2023-05-20',5,5);
```

DDL Commands

1. Renaming the column

```
/* Renaming a column */  
Alter table Officer  
rename COLUMN Badge_Number to Badge_No;
```

2. Adding a new column

```
/* Adding a new cloumn in a table */  
Alter table Officer  
add COLUMN Officer_CNIC varchar(255);
```

3. Changing data-type

```
/* Changing the data type of a column */  
ALTER TABLE Suspect  
ALTER COLUMN Suspect_Gender varchar(12);
```

(We can also use **Modify** instead of Alter before column)

4. Dropping a column

```
/* Dropping a column */  
Alter table Officer  
drop COLUMN Officer_Rank;
```

5. Rename the table

```
/* Rename the table*/  
RENAME TABLE Officer TO Off;
```

6. Truncate

```
/* Remove data from the table */  
truncate Criminal_Alias;
```


DML Commands

1. Select

```
select * from Criminal_Alias;  
select * from Arrest;
```

| Results | | Messages | |
|---------|-------------------|---------------------|-------------|
| | Criminal_Alias_Id | Criminal_Alias_Name | Criminal_Id |
| 1 | 1 | Emily | 1 |
| 2 | 2 | John | 2 |
| 3 | 3 | Michael | 3 |
| 4 | 4 | Charlotte | 4 |
| 5 | 5 | Bruce | 5 |

| | Arrest_Id | Arrest_Time | Arrest_Date | Arrest_Location | Arrest_Reason | Officer_Badge_Number |
|---|-----------|------------------|-------------|------------------------------|---------------|----------------------|
| 1 | 1234 | 00:00:00.0000000 | 1900-01-01 | Maple Avenue, North District | Robbery | 4 |
| 2 | 2345 | 00:00:00.0000000 | 1900-01-01 | Elm Street, East District | Assault | 2 |
| 3 | 4567 | 00:00:00.0000000 | 1900-01-01 | Park Avenue, West District | Assault | 2 |
| 4 | 5678 | 00:00:00.0000000 | 1900-01-01 | Main Street, City Center | Theft | 1 |
| 5 | 7890 | 00:00:00.0000000 | 1900-01-01 | Oak Lane, South District | Drugs | 1 |

✓ Query executed successfully.

2. Insert

```
INSERT INTO Suspect  
VALUES  
(1, 'Clark', 'M', 26, 1),  
(2, 'Hector', 'M', 44, 2),  
(3, 'Victoria', 'F', 27, 3),  
(4, 'Fudge', 'M', 22, 4),  
(5, 'Robertson', 'M', 17, 5);
```

Results

Messages

| | Suspect_Id | Suspect_Name | Suspect_Gender | Suspect_Age | Case_Id |
|---|------------|--------------|----------------|-------------|---------|
| 1 | 1 | Clark | M | 26 | 1 |
| 2 | 2 | Hector | M | 44 | 2 |
| 3 | 3 | Victoria | F | 27 | 3 |
| 4 | 4 | Fudge | M | 22 | 4 |
| 5 | 5 | Robertson | M | 17 | 5 |

3. Update

```
update Criminal set Criminal_Name = 'Luca Modrich' where Criminal_Id = 1;
```

Before

| Results | | Messages | | | | |
|---------|-------------|------------------|--------------|-----------------|-----------------|-----------------|
| | Criminal_Id | Criminal_Name | Criminal_DOB | Criminal_Gender | Criminal_Height | Criminal_Weight |
| 1 | 1 | John Smith | 2003-05-01 | M | 1.83 | 75.56 |
| 2 | 2 | Emily Johnson | 2002-05-02 | F | 0.90 | 60.87 |
| 3 | 3 | David Wilson | 2001-05-03 | M | 1.12 | 80.76 |
| 4 | 4 | Samantha Davis | 2003-05-04 | F | 2.00 | 55.22 |
| 5 | 5 | Michael Thompson | 2003-05-05 | M | 1.31 | 90.12 |

After

| Results | | Messages | | | | |
|---------|-------------|------------------|--------------|-----------------|-----------------|-----------------|
| | Criminal_Id | Criminal_Name | Criminal_DOB | Criminal_Gender | Criminal_Height | Criminal_Weight |
| 1 | 1 | Luca Modrich | 2003-05-01 | M | 1.83 | 75.56 |
| 2 | 2 | Emily Johnson | 2002-05-02 | F | 0.90 | 60.87 |
| 3 | 3 | David Wilson | 2001-05-03 | M | 1.12 | 80.76 |
| 4 | 4 | Samantha Davis | 2003-05-04 | F | 2.00 | 55.22 |
| 5 | 5 | Michael Thompson | 2003-05-05 | M | 1.31 | 90.12 |

4. Delete

```
delete from Criminal_Alias where Criminal_Alias_Id = 1;
```

| Results | | Messages | |
|---------|-------------------|---------------------|-------------|
| | Criminal_Alias_Id | Criminal_Alias_Name | Criminal_Id |
| 1 | 2 | John | 2 |
| 2 | 3 | Michael | 3 |
| 3 | 4 | Charlotte | 4 |
| 4 | 5 | Bruce | 5 |

Joins

1. Inner Join

```
SELECT Criminal.Criminal_Id, Criminal.Criminal_name, Criminal_Alias.Criminal_Alias_name,  
Crime.Crime_Type  
FROM ((Criminal  
INNER JOIN Crime ON Criminal.Criminal_Id = Crime.Criminal_Id)  
INNER JOIN Criminal_Alias ON Criminal.Criminal_Id = Criminal_Alias.Criminal_Id);
```

Results

Messages

| | Criminal_Id | Criminal_name | Criminal_Alias_name | Crime_Type |
|---|-------------|------------------|---------------------|------------|
| 1 | 2 | Emily Johnson | John | Assault |
| 2 | 3 | David Wilson | Michael | Robbery |
| 3 | 4 | Samantha Davis | Charlotte | Theft |
| 4 | 5 | Michael Thompson | Bruce | Harrasment |

2. Left Join

```
SELECT Criminal.Criminal_Id, Criminal.Criminal_name, Criminal_Alias.Criminal_Alias_name,  
Crime.Crime_Type  
FROM ((Criminal  
left JOIN Crime ON Criminal.Criminal_Id = Crime.Criminal_Id)  
left JOIN Criminal_Alias ON Criminal.Criminal_Id = Criminal_Alias.Criminal_Id);
```

| Results | | Messages | | |
|---------|-------------|------------------|---------------------|------------|
| | Criminal_Id | Criminal_name | Criminal_Alias_name | Crime_Type |
| 1 | 1 | John Smith | NULL | Theft |
| 2 | 2 | Emily Johnson | John | Assault |
| 3 | 3 | David Wilson | Michael | Robbery |
| 4 | 4 | Samantha Davis | Charlotte | Theft |
| 5 | 5 | Michael Thompson | Bruce | Harrasment |
| 6 | 10 | Woods Kaka | NULL | NULL |
| 7 | 12 | Mori Chen | NULL | NULL |
| 8 | 14 | Laura Baye | NULL | NULL |

Query executed successfully.

3. Right Join

```
SELECT Criminal.Criminal_Id, Criminal.Criminal_name, Criminal_Alias.Criminal_Alias_name,  
Crime.Crime_Type  
FROM ((Criminal  
right JOIN Crime ON Criminal.Criminal_Id = Crime.Criminal_Id)  
right JOIN Criminal_Alias ON Criminal.Criminal_Id = Criminal_Alias.Criminal_Id);
```

| Results | | Messages | | |
|---------|-------------|------------------|---------------------|------------|
| | Criminal_Id | Criminal_name | Criminal_Alias_name | Crime_Type |
| 1 | 2 | Emily Johnson | John | Assault |
| 2 | 3 | David Wilson | Michael | Robbery |
| 3 | 4 | Samantha Davis | Charlotte | Theft |
| 4 | 5 | Michael Thompson | Bruce | Harrasment |

Query executed successfully.

4. Full Join

```
SELECT Criminal.Criminal_Id, Criminal.Criminal_name, Criminal_Alias.Criminal_Alias_name,  
Crime.Crime_Type  
FROM ((Criminal  
full JOIN Crime ON Criminal.Criminal_Id = Crime.Criminal_Id)  
full JOIN Criminal_Alias ON Criminal.Criminal_Id = Criminal_Alias.Criminal_Id);
```

| Results | | Messages | | |
|---------|-------------|------------------|---------------------|------------|
| | Criminal_Id | Criminal_name | Criminal_Alias_name | Crime_Type |
| 1 | 1 | John Smith | NULL | Theft |
| 2 | 2 | Emily Johnson | John | Assault |
| 3 | 3 | David Wilson | Michael | Robbery |
| 4 | 4 | Samantha Davis | Charlotte | Theft |
| 5 | 5 | Michael Thompson | Bruce | Harrasment |
| 6 | 10 | Woods Kaka | NULL | NULL |
| 7 | 12 | Mori Chen | NULL | NULL |
| 8 | 14 | Laura Baye | NULL | NULL |

✓ Query executed successfully.

5. Natural Join

```
SELECT * from Crime NATURAL JOIN Criminal;
```

SQL ⓘ

Run Save

```
248  
249  
250 SELECT * from Crime NATURAL JOIN Criminal;  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260
```

Output

1|Theft|Lahore|2023-05-02|05:34|1|John Smith|2003
2|Assault|Islamabad|2023-05-11|05:11|2|Emily John
3|Robbery|Karachi|2023-06-21|02:33|3|David Wilson
4|Theft|Lahore|2023-04-09|07:21|4|Samantha Davis|
5|Harrasment|Karachi|2023-05-04|05:44|5|Michael T

[Execution complete with exit code 0]

6. Self-Join

```
SELECT c1.*, c2.Crime_Id AS Related_Crime_Id, c2.Crime_Date AS Related_Crime_Date
FROM Crime c1
JOIN Crime c2 ON c1.Crime_Type = c2.Crime_Type AND c1.Crime_Location = c2.Crime_Location
WHERE c1.Crime_Id <> c2.Crime_Id;
```

100 %

Results Messages

| | Crime_Id | Crime_Type | Crime_Location | Crime_Date | Crime_Time | Criminal_Id | Related_Crime_Id | Related_Crime_Date |
|---|----------|------------|----------------|------------|------------------|-------------|------------------|--------------------|
| 1 | 1 | Theft | Lahore | 2023-05-02 | 05:34:00.0000000 | 1 | 4 | 2023-04-09 |
| 2 | 4 | Theft | Lahore | 2023-04-09 | 07:21:00.0000000 | 4 | 1 | 2023-05-02 |

7. Cross Join

```
SELECT * from Crime CROSS JOIN Criminal;
```

100 %

Results Messages

| | Crime_Id | Crime_Type | Crime_Location | Crime_Date | Crime_Time | Criminal_Id | Criminal_Id | Criminal_Name | Criminal_DOB | Criminal_Gender | Criminal_Height | Criminal_Weight |
|----|----------|------------|----------------|------------|------------------|-------------|-------------|------------------|--------------|-----------------|-----------------|-----------------|
| 1 | 1 | Theft | Lahore | 2023-05-02 | 05:34:00.0000000 | 1 | 1 | John Smith | 2003-05-01 | M | 1.83 | 75.56 |
| 2 | 1 | Theft | Lahore | 2023-05-02 | 05:34:00.0000000 | 1 | 2 | Emily Johnson | 2002-05-02 | F | 0.90 | 60.87 |
| 3 | 1 | Theft | Lahore | 2023-05-02 | 05:34:00.0000000 | 1 | 3 | David Wilson | 2001-05-03 | M | 1.12 | 80.76 |
| 4 | 1 | Theft | Lahore | 2023-05-02 | 05:34:00.0000000 | 1 | 4 | Samantha Davis | 2003-05-04 | F | 2.00 | 55.22 |
| 5 | 1 | Theft | Lahore | 2023-05-02 | 05:34:00.0000000 | 1 | 5 | Michael Thompson | 2003-05-05 | M | 1.31 | 90.12 |
| 6 | 1 | Theft | Lahore | 2023-05-02 | 05:34:00.0000000 | 1 | 10 | Woods Kaka | 1998-02-17 | M | 1.56 | 66.10 |
| 7 | 1 | Theft | Lahore | 2023-05-02 | 05:34:00.0000000 | 1 | 12 | Mori Chen | 2000-07-21 | F | 1.20 | 46.10 |
| 8 | 1 | Theft | Lahore | 2023-05-02 | 05:34:00.0000000 | 1 | 14 | Laura Baye | 1988-06-24 | F | 1.30 | 54.21 |
| 9 | 2 | Assault | Islamabad | 2023-05-11 | 05:11:00.0000000 | 2 | 1 | John Smith | 2003-05-01 | M | 1.83 | 75.56 |
| 10 | 2 | Assault | Islamabad | 2023-05-11 | 05:11:00.0000000 | 2 | 2 | Emily Johnson | 2002-05-02 | F | 0.90 | 60.87 |
| 11 | 2 | Assault | Islamabad | 2023-05-11 | 05:11:00.0000000 | 2 | 3 | David Wilson | 2001-05-03 | M | 1.12 | 80.76 |

Query executed successfully.

DELL\SQLEXPRESS (16.0 RTM) | DELL\Talha (79) | master | 00:00:00

Views

```
GO
CREATE VIEW my_first_view
AS
SELECT Criminal.Criminal_Id, Criminal.Criminal_name,
Criminal_Alias.Criminal_Alias_name, Crime.Crime_Type
FROM ((Criminal
INNER JOIN Crime ON Criminal.Criminal_Id = Crime.Criminal_Id)
INNER JOIN Criminal_Alias ON Criminal.Criminal_Id = Criminal_Alias.Criminal_Id);

GO
Select * from my_first_view;
```

| Results | | Messages | | |
|---------|-------------|------------------|---------------------|------------|
| | Criminal_Id | Criminal_name | Criminal_Alias_name | Crime_Type |
| 1 | 2 | Emily Johnson | John | Assault |
| 2 | 3 | David Wilson | Michael | Robbery |
| 3 | 4 | Samantha Davis | Charlotte | Theft |
| 4 | 5 | Michael Thompson | Bruce | Harrasment |

```
GO
CREATE VIEW Public_User
AS
select Criminal.Criminal_name, Crime.Crime_Type
from Criminal
Inner Join Crime on Criminal.Criminal_Id = Crime.Criminal_Id;

GO
Select * from Public_User;
```

| Results | | Messages | |
|---------|------------------|------------|--|
| | Criminal_name | Crime_Type | |
| 1 | John Smith | Theft | |
| 2 | Emily Johnson | Assault | |
| 3 | David Wilson | Robbery | |
| 4 | Samantha Davis | Theft | |
| 5 | Michael Thompson | Harrasment | |

```

GO
CREATE VIEW Court_Records
AS
Select Officer.Badge_Number, Officer.Officer_Name, Cases.Case_Id, Cases.Case_Name,
Cases.Case_Date, Investigation.Investigation_Start_Date,
Investigation.Investigation_End_Date, Suspect.Suspect_Name, Witness.Witness_Name,
Evidence.Evidence_Type
from (((Investigation

Inner Join Officer on Investigation.Officer_Badge_Number = Officer.Badge_Number)
Inner Join Cases on Investigation.Case_Id = Cases.Case_Id)
Inner Join Witness on Witness.Case_Id = Cases.Case_Id)
Inner Join Suspect on Suspect.Case_Id = Cases.Case_Id)
Inner Join Evidence on Evidence.Case_Id = Cases.Case_Id);

GO
Select * from Court_Records;

```

| | Badge_Number | Officer_Name | Case_Id | Case_Name | Case_Date | Investigation_Start_Date | Investigation_End_Date | Suspect_Name | Witness_Name | Evidence_Type |
|---|--------------|------------------|---------|------------------|------------|--------------------------|------------------------|--------------|--------------|----------------------|
| 1 | 1 | John Smith | 1 | John Smith | 2023-05-01 | 2023-01-01 | 2023-01-05 | Clark | Saad | Fingerprint |
| 2 | 2 | David Wilson | 2 | Emily Johnson | 2023-05-02 | 2023-02-10 | 2023-02-15 | Hector | Talha | DNA Sample |
| 3 | 3 | Emily Johnson | 3 | David Wilson | 2023-05-03 | 2023-03-20 | 2023-03-25 | Victoria | Laiba | Weapon |
| 4 | 4 | Samantha Davis | 4 | Samantha Davis | 2023-05-04 | 2023-04-05 | 2023-04-10 | Fudge | Noman | Surveillance Footage |
| 5 | 5 | Michael Thompson | 5 | Michael Thompson | 2023-05-05 | 2023-05-15 | 2023-05-20 | Robertson | Shoab | Documents |

Stored Procedures

```
GO
CREATE PROCEDURE GetCourtRecords
AS
BEGIN
    SELECT Officer.Badge_Number, Officer.Officer_Name, Cases.Case_Id, Cases.Case_Name,
    Cases.Case_Date, Investigation.Investigation_Start_Date,
    Investigation.Investigation_End_Date, Suspect.Suspect_Name, Witness.Witness_Name
    FROM (((Investigation
    INNER JOIN Officer ON Investigation.Officer_Badge_Number = Officer.Badge_Number)
    INNER JOIN Cases ON Investigation.Case_Id = Cases.Case_Id)
    INNER JOIN Suspect ON Cases.Case_Id = Suspect.Case_Id)
    INNER JOIN Witness ON Cases.Case_Id = Witness.Case_Id)
END;

EXEC GetCourtRecords;
```

| | Badge_Number | Officer_Name | Case_Id | Case_Name | Case_Date | Investigation_Start_Date | Investigation_End_Date | Suspect_Name | Witness_Name |
|---|--------------|------------------|---------|------------------|------------|--------------------------|------------------------|--------------|--------------|
| 1 | 1 | John Smith | 1 | John Smith | 2023-05-01 | 2023-01-01 | 2023-01-05 | Clark | Saad |
| 2 | 2 | David Wilson | 2 | Emily Johnson | 2023-05-02 | 2023-02-10 | 2023-02-15 | Hector | Talha |
| 3 | 3 | Emily Johnson | 3 | David Wilson | 2023-05-03 | 2023-03-20 | 2023-03-25 | Victoria | Laiba |
| 4 | 4 | Samantha Davis | 4 | Samantha Davis | 2023-05-04 | 2023-04-05 | 2023-04-10 | Fudge | Noman |
| 5 | 5 | Michael Thompson | 5 | Michael Thompson | 2023-05-05 | 2023-05-15 | 2023-05-20 | Robertson | Shoaib |

```
GO
CREATE PROCEDURE totalCrimes
    @total INT OUTPUT
AS
BEGIN
    SELECT @total = COUNT(*) FROM Crime;
END;

DECLARE @total_C INT;
EXEC totalCrimes @total_C OUTPUT;
SELECT @total_C AS Total_Crimes;
```

| | Total_Crimes |
|---|--------------|
| 1 | 5 |

Conclusion

The Criminal Database System project aims to provide an efficient and secure way for law enforcement agencies and other authorized personnel to manage and access criminal records and information related to criminal cases. The system will be developed using the relational database model and will include features such as user authentication, data entry and management, case management, criminal records management, evidence management, investigation management, and reporting.