GEC BARTON HILL

DBMS PROJECT
– CRIME DATABASE –
FINAL REPORT

SUBMITTED BY:

GROUP 3:

DEVANANDA A	ROLL NO 33
LIZONA LOY PARAYIL	ROLL NO 47
KRISHNAPRIYA R	ROLL NO 46
M S GAURI SANKAR	ROLL NO 48
NIDHIN K BIJU	ROLL NO 55

SUBMITTED TO:

Mrs.JOSNA V R

INDEX

FUNCTIONAL REQUIREMENTS:

Scope

The system supports registering criminal profiles in the database. It allows updating criminal profiles whenever new information is available. Crimes can be recorded and linked directly to the criminals involved. Officers can be assigned to cases for proper investigation. Victim information can be recorded and managed. Witness statements can also be maintained in the system. The progress of each case is tracked from opening to closure. The system monitors criminal status, including whether the criminal is Arrested, Wanted, or Released.

1. Criminal Management

The system allows users to add criminal records into the database. Each criminal record includes a unique Criminal_ID for identification. The full name of the criminal is stored in the record. Age and gender of the criminal are maintained. The address of the criminal is recorded to help locate and contact them. The status of the criminal, which can be Arrested, Wanted, or Released, is tracked. A photograph of the criminal can be uploaded and associated with the record.

2. Crime Management

The system allows registering a new crime entry in the database. Each crime record has a unique Crime_ID to distinguish it from others. The type of crime, such as Theft, Assault, or Homicide, is stored. The date and time when the crime occurred are recorded. The location of the crime is maintained for investigation purposes. The severity level of the crime is indicated. A detailed description of the crime is stored to provide context for investigators.

3. Case Management

The system allows opening new cases and closing existing cases. Each case is assigned a unique Case_ID. Related Crime_IDs are linked to the case to indicate which crimes are under investigation. Criminal_IDs involved in the case are also associated with the case. An officer is assigned to the case using their Officer_ID. The status of the case can be Open, Closed, or On Hold. The start date of the case is recorded when it is opened. The end date of the case is recorded when it is closed.

4. Officer Management

The system maintains records for each officer. Each officer has a unique Officer_ID. The name of the officer is stored. The rank of the officer is recorded. Contact information for the officer is maintained. The police station assigned to the officer is recorded using Station_ID.

5. Station Management

The system stores information for each police station. Each station has a unique Station_ID. The name of the station is recorded. The location of the station is stored. The contact number for the station is maintained.

6. Victim Management

The system allows adding records of victims involved in crimes. Each victim has a unique Victim_ID. The full name of the victim is stored. The age and gender of the victim are recorded. The address of the victim is maintained. Contact information for the victim is stored. Victims are linked to the relevant cases in the system.

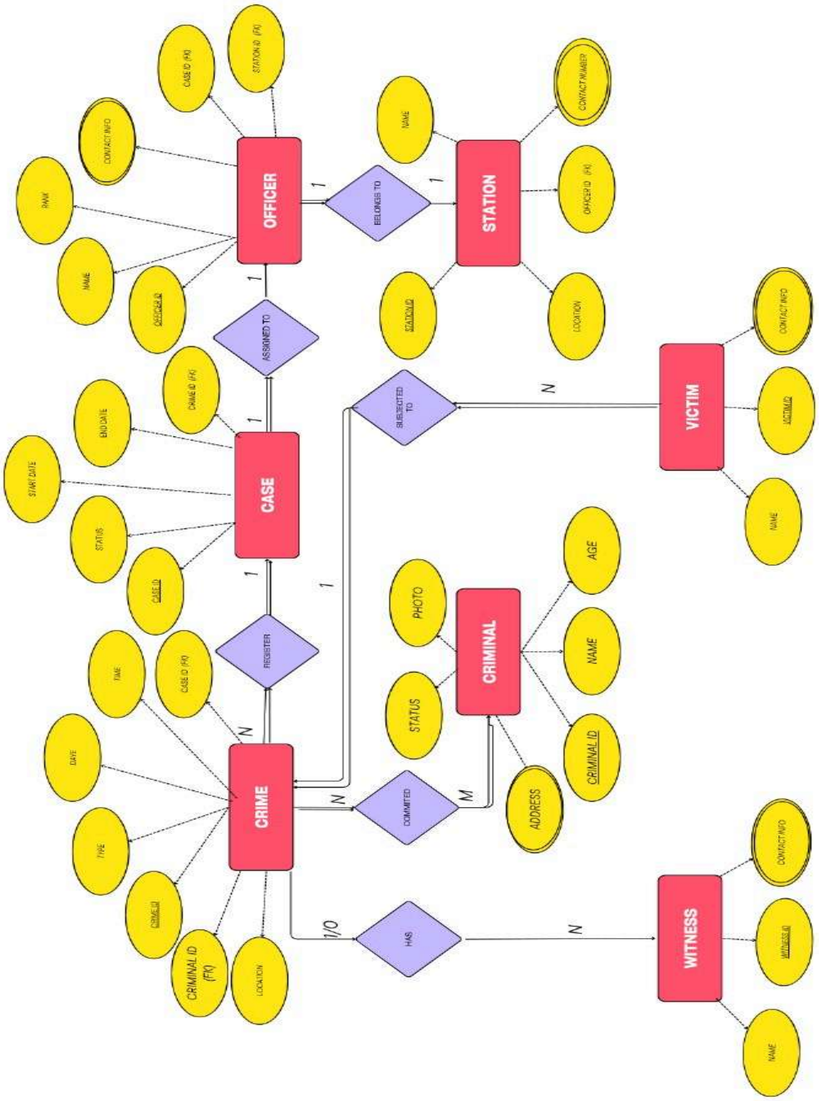
7. Witness Management

The system allows adding records for witnesses. Each witness has a unique Witness_ID. The full name of the witness is stored. The witness’s statement is recorded. Contact information for the witness is maintained. Witnesses are linked to the relevant cases in which they provided testimony.

8. Suspect Management

The system allows adding records for suspects. Each suspect has a unique AADHAAR number for identification. The full name of the suspect is recorded. The age and gender of the suspect are maintained. The address of the suspect is stored. The status of the suspect, which can be Guilty or Not Guilty, is tracked. A photograph of the suspect can be added to the record. Suspects are linked to the relevant crimes and cases.

ER DIAGRAM:


OFFICER
STATION
CASE
CRIME
CRIMINAL
VICTIM
WITNESS
BELONGS TO
ASSIGNED TO
SUBJECTED TO
REGISTER
COMMITED
HAS
CASE ID (FK)
STATION ID (FK)
CONTACT INFO
RANK
NAME
OFFICER ID
NAME
CONTACT NUMBER
OFFICER ID (FK)
STATION ID
LOCATION
CONTACT INFO
VICTIM ID
NAME
CRIME ID (FK)
END DATE
START DATE
STATUS
CASE ID
PHOTO
AGE
NAME
STATUS
CRIMINAL ID
ADDRESS
TIME
CASE ID (FK)
DATE
TYPE
CRIME ID
CRIMINAL ID (FK)
LOCATION
CONTACT INFO
WITNESS ID
NAME
1
1
1
1
1
1
N
N
N
M
1/0
N

SQL QUERY:

```
(PROCEDURES,TRIGGERS,)
CREATE DATABASE IF NOT EXISTS crimedatabase;
USE crimedatabase;

CREATE TABLE IF NOT EXISTS Station (
    station_id INT PRIMARY KEY AUTO_INCREMENT,
    station_name VARCHAR(100),
    location VARCHAR(150),
    phone VARCHAR(15)
);

CREATE TABLE IF NOT EXISTS Officer (
    officer_id INT PRIMARY KEY AUTO_INCREMENT,
    officer_name VARCHAR(100),
    officer_rank VARCHAR(50),
    station_id INT,
    contact_no VARCHAR(15),
    FOREIGN KEY (station_id) REFERENCES Station(station_id)
);

CREATE TABLE IF NOT EXISTS CaseFile (
    case_id INT PRIMARY KEY AUTO_INCREMENT,
    case_title VARCHAR(150),
    case_type VARCHAR(100),
    date_reported DATE,
    status VARCHAR(30),
    officer_id INT,
    FOREIGN KEY (officer_id) REFERENCES Officer(officer_id)
);

CREATE TABLE IF NOT EXISTS Crime (
    crime_id INT PRIMARY KEY AUTO_INCREMENT,
    case_id INT,
    crime_description TEXT,
    crime_date DATE,
    crime_location VARCHAR(150),
    FOREIGN KEY (case_id) REFERENCES CaseFile(case_id)
);

CREATE TABLE IF NOT EXISTS Criminal (
    criminal_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    gender VARCHAR(10),
    age INT,
    address VARCHAR(150),
    crime_id INT,
    FOREIGN KEY (crime_id) REFERENCES Crime(crime_id)
);

CREATE TABLE IF NOT EXISTS Victim (
    victim_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    age INT,
    contact_no VARCHAR(15),
    address VARCHAR(150),
    case_id INT,
    FOREIGN KEY (case_id) REFERENCES CaseFile(case_id)
);

CREATE TABLE IF NOT EXISTS Witness (
    witness_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    contact_no VARCHAR(15),
    statement TEXT,
    case_id INT,
    FOREIGN KEY (case_id) REFERENCES CaseFile(case_id)
);

CREATE TABLE IF NOT EXISTS Station_Log (
    log_id INT PRIMARY KEY AUTO_INCREMENT,
    officer_id INT,
    action VARCHAR(50),
    log_time DATETIME
);

CREATE TABLE IF NOT EXISTS Case_Log (
    log_id INT PRIMARY KEY AUTO_INCREMENT,
    case_id INT,
    old_status VARCHAR(30),
    new_status VARCHAR(30),
    updated_on DATETIME
);

CREATE TABLE IF NOT EXISTS Criminal_Log (
    log_id INT PRIMARY KEY AUTO_INCREMENT,
    criminal_id INT,
    log_message VARCHAR(255),
    log_time DATETIME
);
```

PROCEDURE:

```
DELIMITER //
CREATE PROCEDURE AddCaseWithOfficer(
   IN p_title VARCHAR(150),
   IN p_type VARCHAR(100),
   IN p_officer INT
)
BEGIN
   INSERT INTO CaseFile(case_title, case_type, date_reported, status, officer_id)
   VALUES(p_title, p_type, CURDATE(), 'Open', p_officer);

   SELECT * FROM CaseFile WHERE officer_id = p_officer ORDER BY case_id DESC LIMIT 3;
END //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE RegisterCriminal(
   IN p_name VARCHAR(100),
   IN p_gender VARCHAR(10),
   IN p_age INT,
   IN p_address VARCHAR(150),
   IN p_crime_id INT
)
BEGIN
   INSERT INTO Criminal(name, gender, age, address, crime_id)
   VALUES(p_name, p_gender, p_age, p_address, p_crime_id);

   SELECT c.crime_id, c.crime_description, cf.case_title, cf.status
   FROM Crime c
   JOIN CaseFile cf ON cf.case_id = c.case_id
   WHERE c.crime_id = p_crime_id;
END //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE AddVictim(
   IN p_name VARCHAR(100),
   IN p_age INT,
   IN p_contact VARCHAR(15),
   IN p_address VARCHAR(150),
   IN p_case INT
)
BEGIN
   INSERT INTO Victim(name, age, contact_no, address, case_id)
   VALUES(p_name, p_age, p_contact, p_address, p_case);
```

```
    SELECT name, contact_no FROM Victim WHERE case_id = p_case;
END //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE CloseCase(IN p_case INT)
BEGIN
    UPDATE CaseFile SET status = 'Closed' WHERE case_id = p_case;
    SELECT cf.case_id, cf.case_title, cf.status, o.officer_name
    FROM CaseFile cf
    JOIN Officer o ON cf.officer_id = o.officer_id
    WHERE cf.case_id = p_case;
END //
DELIMITER ;


DELIMITER //

CREATE PROCEDURE GetOfficerCases(IN p_officer_id INT)
BEGIN
    SELECT cf.case_id, cf.case_title, cf.case_type, cf.status, cf.date_reported
    FROM CaseFile cf
    WHERE cf.officer_id = p_officer_id
    ORDER BY cf.date_reported DESC;
END //

CREATE PROCEDURE UpdateCaseStatus(IN p_case_id INT, IN p_new_status VARCHAR(30))
BEGIN
    UPDATE CaseFile
    SET status = p_new_status
    WHERE case_id = p_case_id;

    SELECT case_id, case_title, status
    FROM CaseFile
    WHERE case_id = p_case_id;
END //

CREATE PROCEDURE DeleteCriminalRecord(IN p_criminal_id INT)
BEGIN
    DECLARE v_case_status VARCHAR(30);

    SELECT cf.status INTO v_case_status
    FROM Criminal cr
    JOIN Crime c ON cr.crime_id = c.crime_id
    JOIN CaseFile cf ON c.case_id = cf.case_id
    WHERE cr.criminal_id = p_criminal_id;

    IF v_case_status = 'Closed' THEN
```

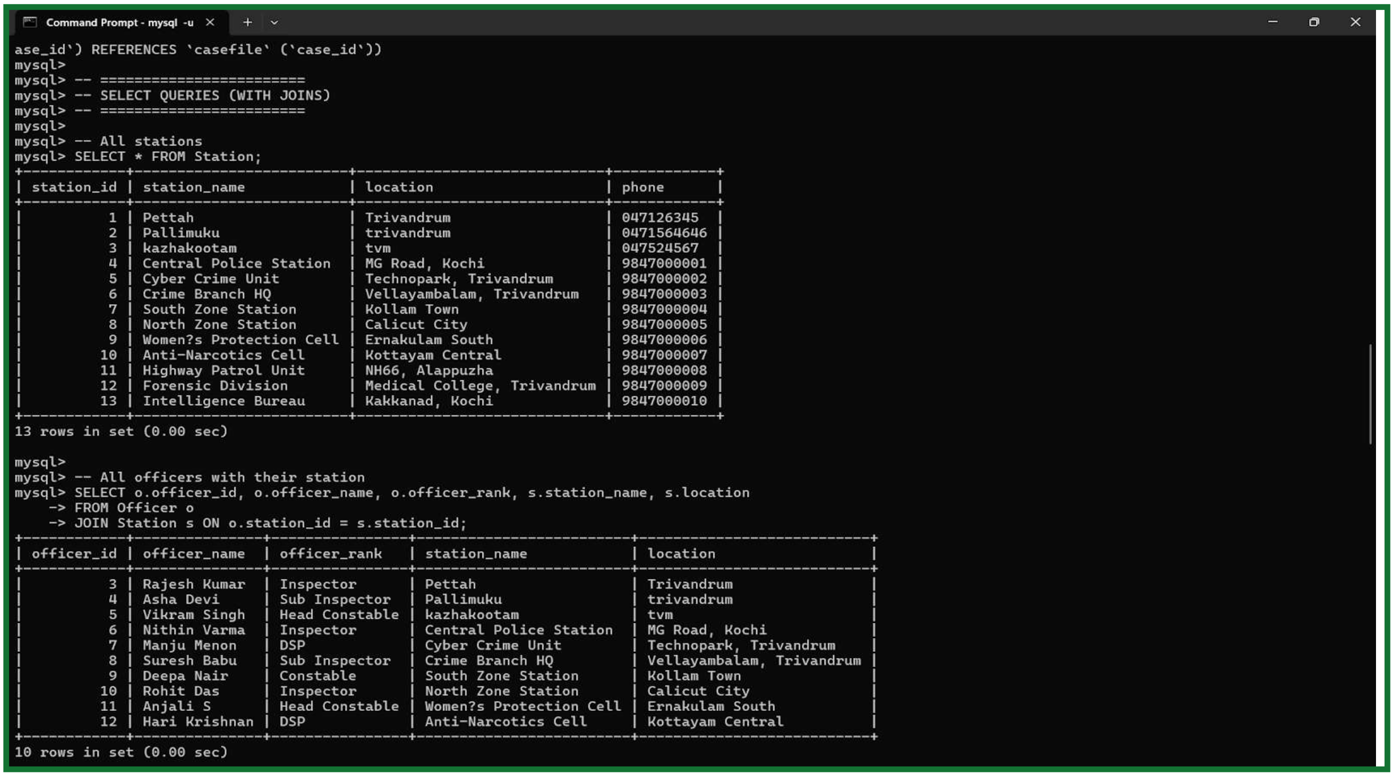
```
      DELETE FROM Criminal WHERE criminal_id = p_criminal_id;
      SELECT CONCAT('Criminal ID ', p_criminal_id, ' deleted successfully') AS Message;
   ELSE
      SELECT 'Cannot delete criminal record until case is closed' AS Message;
   END IF;
END //

CREATE PROCEDURE SearchCrimeByDate(IN p_start DATE, IN p_end DATE)
BEGIN
   SELECT c.crime_id, c.crime_description, c.crime_date, cf.case_title, o.officer_name
   FROM Crime c
   JOIN CaseFile cf ON c.case_id = cf.case_id
   JOIN Officer o ON cf.officer_id = o.officer_id
   WHERE c.crime_date BETWEEN p_start AND p_end
   ORDER BY c.crime_date;
END //

CREATE PROCEDURE GetCriminalHistory(IN p_criminal_id INT)
BEGIN
   SELECT cr.name AS Criminal_Name, cr.gender, cr.age, cr.address,
        c.crime_description, cf.case_title, cf.status, o.officer_name
   FROM Criminal cr
   JOIN Crime c ON cr.crime_id = c.crime_id
   JOIN CaseFile cf ON c.case_id = cf.case_id
   JOIN Officer o ON cf.officer_id = o.officer_id
   WHERE cr.criminal_id = p_criminal_id;
END //

CREATE PROCEDURE VictimReport()
BEGIN
   SELECT v.victim_id, v.name AS Victim_Name, v.age, v.contact_no, cf.case_title, cf.status
   FROM Victim v
   JOIN CaseFile cf ON v.case_id = cf.case_id
   ORDER BY cf.case_id;
END //
```

TRIGGER:

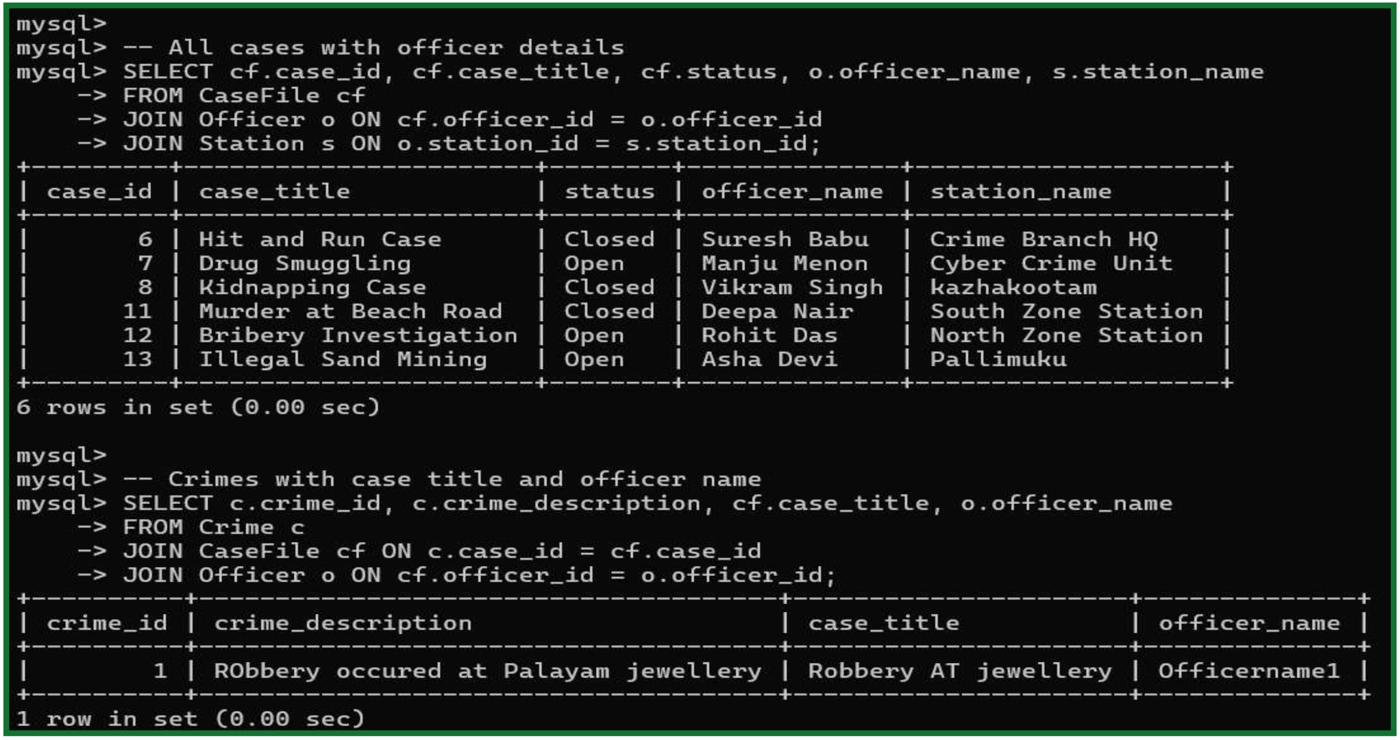
```
DELIMITER //
CREATE TRIGGER AfterOfficerInsert
AFTER INSERT ON Officer
FOR EACH ROW
BEGIN
   INSERT INTO Station_Log(officer_id, action, log_time)
   VALUES(NEW.officer_id, 'New Officer Added', NOW());
END //
DELIMITER ;

DELIMITER //
CREATE TRIGGER AfterCaseUpdate
AFTER UPDATE ON CaseFile
FOR EACH ROW
BEGIN
   IF OLD.status <> NEW.status THEN
      INSERT INTO Case_Log(case_id, old_status, new_status, updated_on)
      VALUES(NEW.case_id, OLD.status, NEW.status, NOW());
   END IF;
END //
DELIMITER ;

DELIMITER //
CREATE TRIGGER AfterCriminalInsert
AFTER INSERT ON Criminal
FOR EACH ROW
BEGIN
   INSERT INTO Criminal_Log(criminal_id, log_message, log_time)
   VALUES(NEW.criminal_id, CONCAT('Criminal Added: ', NEW.name), NOW());
END //
DELIMITER ;


CREATE TRIGGER BeforeCriminalDelete
BEFORE DELETE ON Criminal
FOR EACH ROW
BEGIN
   DECLARE v_status VARCHAR(30);

   SELECT cf.status INTO v_status
   FROM Crime c
   JOIN CaseFile cf ON c.case_id = cf.case_id
   WHERE c.crime_id = OLD.crime_id;

   IF v_status <> 'Closed' THEN
```

```
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Cannot delete criminal record until case is closed.';
    END IF;
END //

CREATE TRIGGER AfterCaseInsert
AFTER INSERT ON CaseFile
FOR EACH ROW
BEGIN
    INSERT INTO Case_Log(case_id, old_status, new_status, updated_on)
    VALUES(NEW.case_id, NULL, NEW.status, NOW());
END //

CREATE TRIGGER AfterVictimInsert
AFTER INSERT ON Victim
FOR EACH ROW
BEGIN
    INSERT INTO Case_Log(case_id, old_status, new_status, updated_on)
    VALUES(NEW.case_id, 'Victim Added', 'Victim Added', NOW());
END //

CREATE TRIGGER BeforeCaseUpdate
BEFORE UPDATE ON CaseFile
FOR EACH ROW
BEGIN
    IF OLD.status = 'Closed' AND NEW.status = 'Open' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Cannot reopen a closed case.';
    END IF;
END //

DELIMITER ;
```

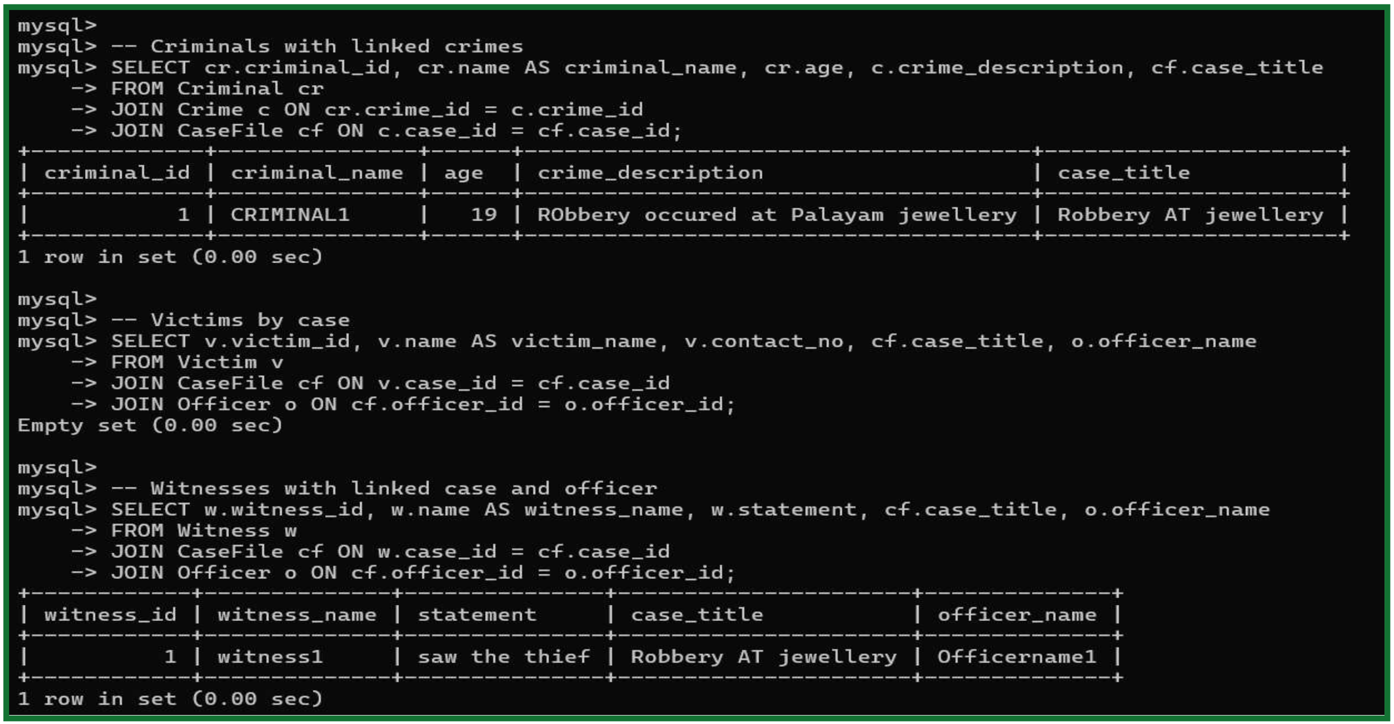
TABLES:



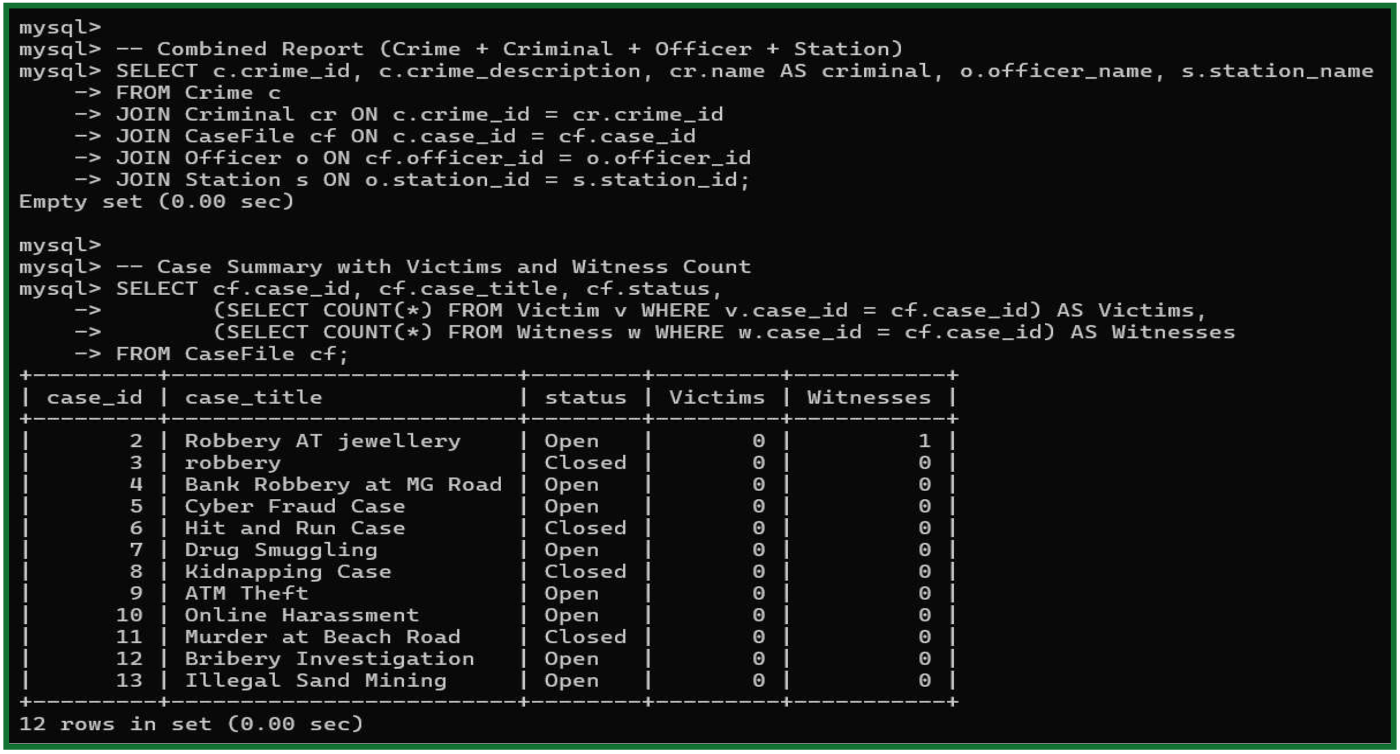
```
ase_id`) REFERENCES `casefile` (`case_id`))
mysql>
mysql> -- ========================
mysql> -- SELECT QUERIES (WITH JOINS)
mysql> -- ========================
mysql>
mysql> -- All stations
mysql> SELECT * FROM Station;
+------------+--------------------------+------------------------------+------------+
| station_id | station_name             | location                     | phone      |
+------------+--------------------------+------------------------------+------------+
|          1 | Pettah                   | Trivandrum                   | 047126345  |
|          2 | Pallimuku                | trivandrum                   | 0471564646 |
|          3 | kazhakootam              | tvm                          | 047524567  |
|          4 | Central Police Station   | MG Road, Kochi               | 9847000001 |
|          5 | Cyber Crime Unit         | Technopark, Trivandrum       | 9847000002 |
|          6 | Crime Branch HQ          | Vellayambalam, Trivandrum    | 9847000003 |
|          7 | South Zone Station       | Kollam Town                  | 9847000004 |
|          8 | North Zone Station       | Calicut City                 | 9847000005 |
|          9 | Women?s Protection Cell  | Ernakulam South              | 9847000006 |
|         10 | Anti-Narcotics Cell      | Kottayam Central             | 9847000007 |
|         11 | Highway Patrol Unit      | NH66, Alappuzha              | 9847000008 |
|         12 | Forensic Division        | Medical College, Trivandrum  | 9847000009 |
|         13 | Intelligence Bureau      | Kakkanad, Kochi              | 9847000010 |
+------------+--------------------------+------------------------------+------------+
13 rows in set (0.00 sec)

mysql>
mysql> -- All officers with their station
mysql> SELECT o.officer_id, o.officer_name, o.officer_rank, s.station_name, s.location
    -> FROM Officer o
    -> JOIN Station s ON o.station_id = s.station_id;
+------------+---------------+----------------+-------------------------+---------------------------+
| officer_id | officer_name  | officer_rank   | station_name            | location                  |
+------------+---------------+----------------+-------------------------+---------------------------+
|          3 | Rajesh Kumar  | Inspector      | Pettah                  | Trivandrum                |
|          4 | Asha Devi     | Sub Inspector  | Pallimuku               | trivandrum                |
|          5 | Vikram Singh  | Head Constable | kazhakootam             | tvm                       |
|          6 | Nithin Varma  | Inspector      | Central Police Station  | MG Road, Kochi            |
|          7 | Manju Menon   | DSP            | Cyber Crime Unit        | Technopark, Trivandrum    |
|          8 | Suresh Babu   | Sub Inspector  | Crime Branch HQ         | Vellayambalam, Trivandrum |
|          9 | Deepa Nair    | Constable      | South Zone Station      | Kollam Town               |
|         10 | Rohit Das     | Inspector      | North Zone Station      | Calicut City              |
|         11 | Anjali S      | Head Constable | Women?s Protection Cell | Ernakulam South           |
|         12 | Hari Krishnan | DSP            | Anti-Narcotics Cell     | Kottayam Central          |
+------------+---------------+----------------+-------------------------+---------------------------+
10 rows in set (0.00 sec)
```



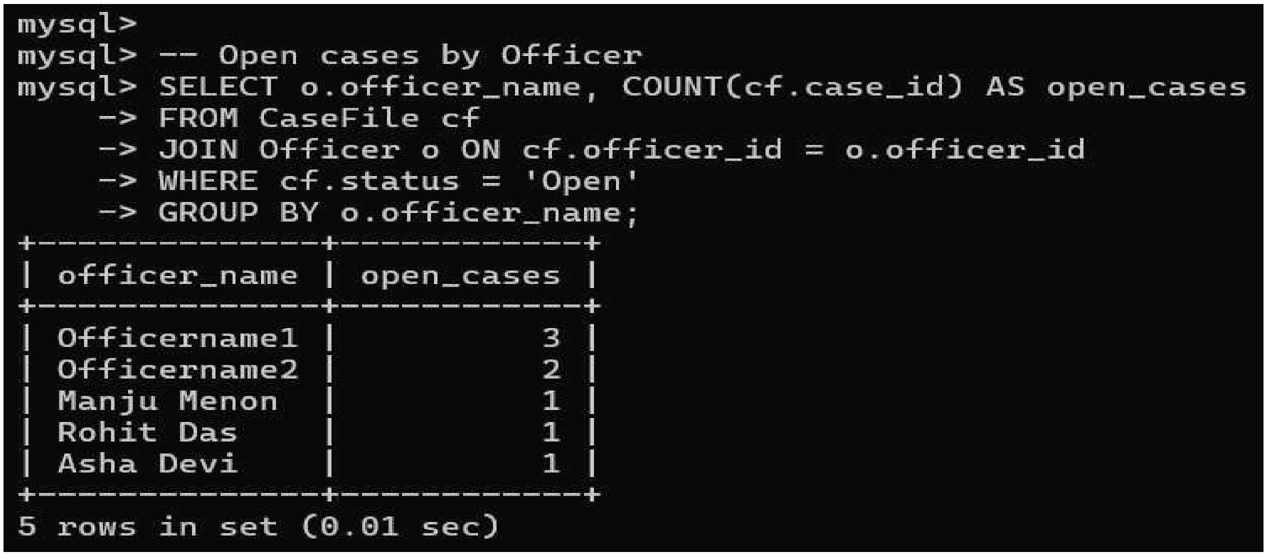
```
mysql>
mysql> -- All cases with officer details
mysql> SELECT cf.case_id, cf.case_title, cf.status, o.officer_name, s.station_name
    -> FROM CaseFile cf
    -> JOIN Officer o ON cf.officer_id = o.officer_id
    -> JOIN Station s ON o.station_id = s.station_id;
+---------+-----------------------+--------+--------------+--------------------+
| case_id | case_title            | status | officer_name | station_name       |
+---------+-----------------------+--------+--------------+--------------------+
|       6 | Hit and Run Case      | Closed | Suresh Babu  | Crime Branch HQ    |
|       7 | Drug Smuggling        | Open   | Manju Menon  | Cyber Crime Unit   |
|       8 | Kidnapping Case       | Closed | Vikram Singh | kazhakootam        |
|      11 | Murder at Beach Road  | Closed | Deepa Nair   | South Zone Station |
|      12 | Bribery Investigation | Open   | Rohit Das    | North Zone Station |
|      13 | Illegal Sand Mining   | Open   | Asha Devi    | Pallimuku          |
+---------+-----------------------+--------+--------------+--------------------+
6 rows in set (0.00 sec)

mysql>
mysql> -- Crimes with case title and officer name
mysql> SELECT c.crime_id, c.crime_description, cf.case_title, o.officer_name
    -> FROM Crime c
    -> JOIN CaseFile cf ON c.case_id = cf.case_id
    -> JOIN Officer o ON cf.officer_id = o.officer_id;
+----------+-------------------------------------+----------------------+--------------+
| crime_id | crime_description                   | case_title           | officer_name |
+----------+-------------------------------------+----------------------+--------------+
|        1 | RObbery occured at Palayam jewellery | Robbery AT jewellery | Officername1 |
+----------+-------------------------------------+----------------------+--------------+
1 row in set (0.00 sec)
```



```
mysql>
mysql> -- Criminals with linked crimes
mysql> SELECT cr.criminal_id, cr.name AS criminal_name, cr.age, c.crime_description, cf.case_title
    -> FROM Criminal cr
    -> JOIN Crime c ON cr.crime_id = c.crime_id
    -> JOIN CaseFile cf ON c.case_id = cf.case_id;
+-------------+---------------+------+--------------------------------------+----------------------+
| criminal_id | criminal_name | age  | crime_description                    | case_title           |
+-------------+---------------+------+--------------------------------------+----------------------+
|           1 | CRIMINAL1     |   19 | RObbery occured at Palayam jewellery | Robbery AT jewellery |
+-------------+---------------+------+--------------------------------------+----------------------+
1 row in set (0.00 sec)

mysql>
mysql> -- Victims by case
mysql> SELECT v.victim_id, v.name AS victim_name, v.contact_no, cf.case_title, o.officer_name
    -> FROM Victim v
    -> JOIN CaseFile cf ON v.case_id = cf.case_id
    -> JOIN Officer o ON cf.officer_id = o.officer_id;
Empty set (0.00 sec)

mysql>
mysql> -- Witnesses with linked case and officer
mysql> SELECT w.witness_id, w.name AS witness_name, w.statement, cf.case_title, o.officer_name
    -> FROM Witness w
    -> JOIN CaseFile cf ON w.case_id = cf.case_id
    -> JOIN Officer o ON cf.officer_id = o.officer_id;
+------------+--------------+---------------+----------------------+--------------+
| witness_id | witness_name | statement     | case_title           | officer_name |
+------------+--------------+---------------+----------------------+--------------+
|          1 | witness1     | saw the thief | Robbery AT jewellery | Officername1 |
+------------+--------------+---------------+----------------------+--------------+
1 row in set (0.00 sec)
```

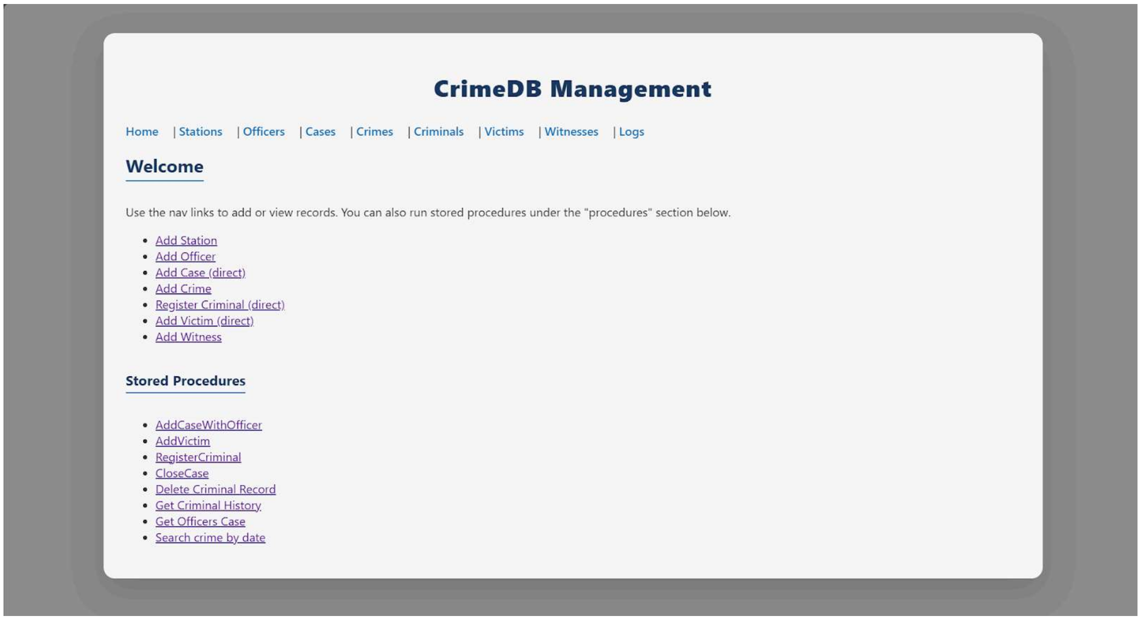


```
mysql>
mysql> -- Combined Report (Crime + Criminal + Officer + Station)
mysql> SELECT c.crime_id, c.crime_description, cr.name AS criminal, o.officer_name, s.station_name
    -> FROM Crime c
    -> JOIN Criminal cr ON c.crime_id = cr.crime_id
    -> JOIN CaseFile cf ON c.case_id = cf.case_id
    -> JOIN Officer o ON cf.officer_id = o.officer_id
    -> JOIN Station s ON o.station_id = s.station_id;
Empty set (0.00 sec)

mysql>
mysql> -- Case Summary with Victims and Witness Count
mysql> SELECT cf.case_id, cf.case_title, cf.status,
    ->        (SELECT COUNT(*) FROM Victim v WHERE v.case_id = cf.case_id) AS Victims,
    ->        (SELECT COUNT(*) FROM Witness w WHERE w.case_id = cf.case_id) AS Witnesses
    -> FROM CaseFile cf;
+---------+-------------------------+--------+---------+-----------+
| case_id | case_title              | status | Victims | Witnesses |
+---------+-------------------------+--------+---------+-----------+
|       2 | Robbery AT jewellery    | Open   |       0 |         1 |
|       3 | robbery                 | Closed |       0 |         0 |
|       4 | Bank Robbery at MG Road | Open   |       0 |         0 |
|       5 | Cyber Fraud Case        | Open   |       0 |         0 |
|       6 | Hit and Run Case        | Closed |       0 |         0 |
|       7 | Drug Smuggling          | Open   |       0 |         0 |
|       8 | Kidnapping Case         | Closed |       0 |         0 |
|       9 | ATM Theft               | Open   |       0 |         0 |
|      10 | Online Harassment       | Open   |       0 |         0 |
|      11 | Murder at Beach Road    | Closed |       0 |         0 |
|      12 | Bribery Investigation   | Open   |       0 |         0 |
|      13 | Illegal Sand Mining     | Open   |       0 |         0 |
+---------+-------------------------+--------+---------+-----------+
12 rows in set (0.00 sec)
```



```
mysql>
mysql> -- Open cases by Officer
mysql> SELECT o.officer_name, COUNT(cf.case_id) AS open_cases
    -> FROM CaseFile cf
    -> JOIN Officer o ON cf.officer_id = o.officer_id
    -> WHERE cf.status = 'Open'
    -> GROUP BY o.officer_name;
+--------------+------------+
| officer_name | open_cases |
+--------------+------------+
| Officername1 |          3 |
| Officername2 |          2 |
| Manju Menon  |          1 |
| Rohit Das    |          1 |
| Asha Devi    |          1 |
+--------------+------------+
5 rows in set (0.01 sec)
```

UI:



CrimeDB Management

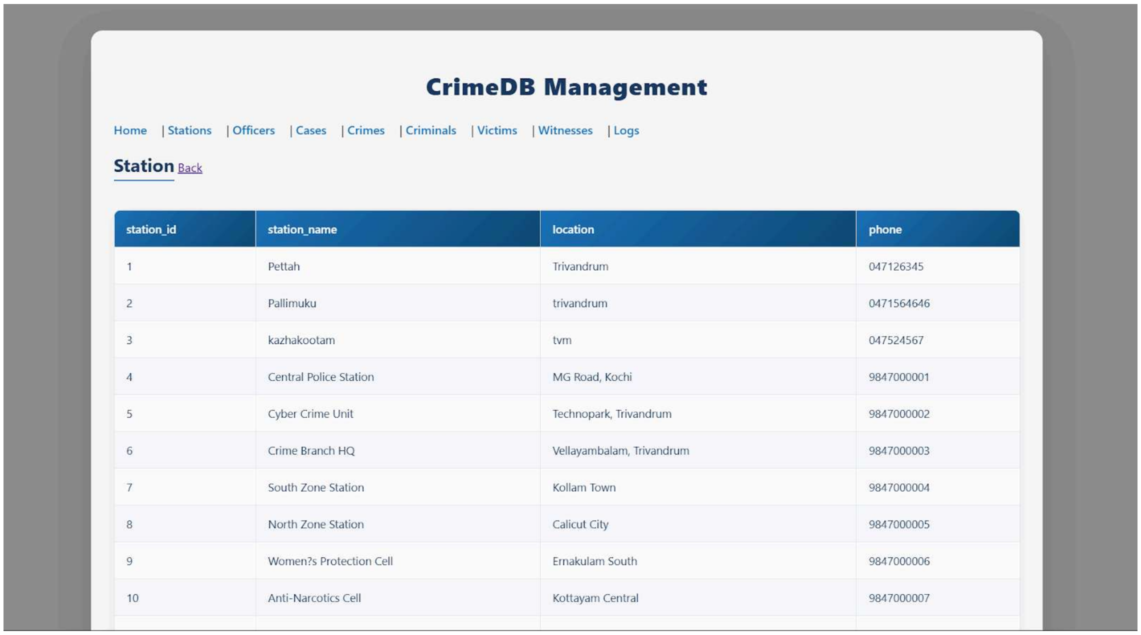
Welcome

Use the nav links to add or view records. You can also run stored procedures under the "procedures" section below.

- Add Station
- Add Officer
- Add Case (direct)
- Add Crime
- Register Criminal (direct)
- Add Victim (direct)
- Add Witness

Stored Procedures

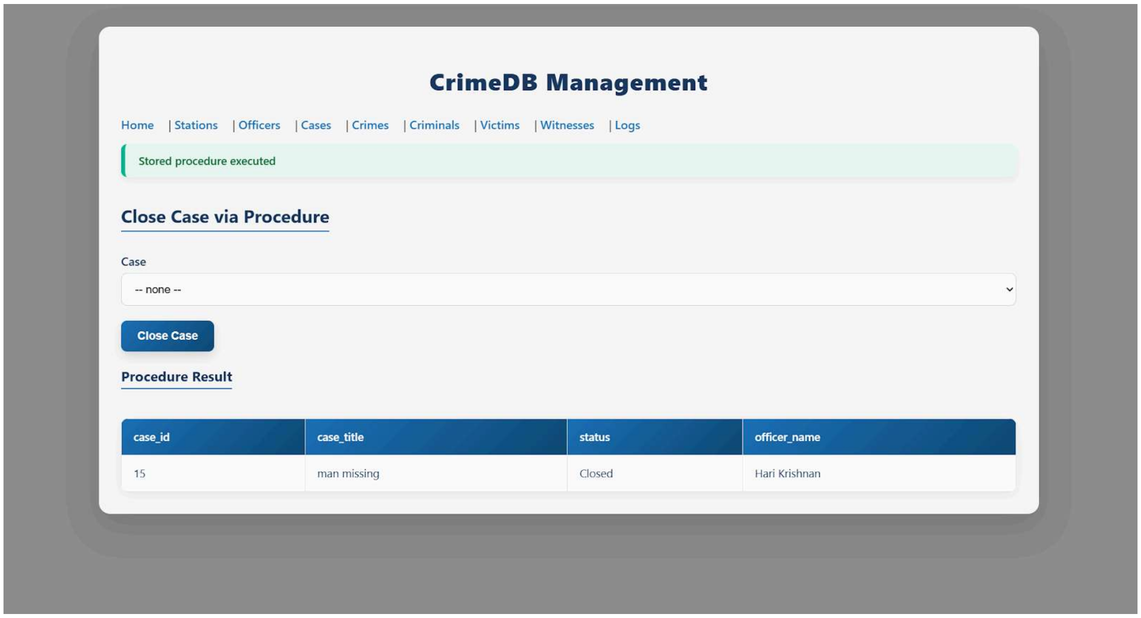
- AddCaseWithOfficer
- AddVictim
- RegisterCriminal
- CloseCase
- Delete Criminal Record
- Get Criminal History
- Get Officers Case
- Search crime by date



CrimeDB Management

Station Back

station_id	station_name	location	phone
1	Pettah	Trivandrum	047126345
2	Pallimuku	trivandrum	0471564646
3	kazhakootam	tvm	047524567
4	Central Police Station	MG Road, Kochi	9847000001
5	Cyber Crime Unit	Technopark, Trivandrum	9847000002
6	Crime Branch HQ	Vellayambalam, Trivandrum	9847000003
7	South Zone Station	Kollam Town	9847000004
8	North Zone Station	Calicut City	9847000005
9	Women?s Protection Cell	Ernakulam South	9847000006
10	Anti-Narcotics Cell	Kottayam Central	9847000007



CrimeDB Management

Home | Stations | Officers | Cases | Crimes | Criminals | Victims | Witnesses | Logs

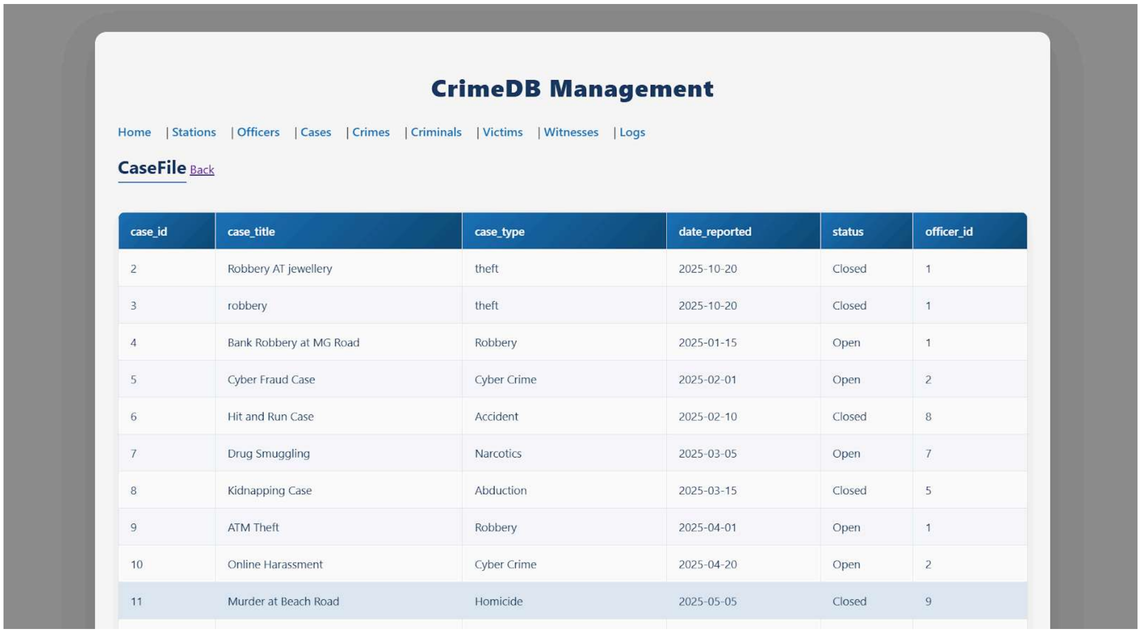
Stored procedure executed

Close Case via Procedure

Case: -- none --

Procedure Result

case_id	case_title	status	officer_name
15	man missing	Closed	Hari Krishnan



CrimeDB Management

Home | Stations | Officers | Cases | Crimes | Criminals | Victims | Witnesses | Logs

CaseFile Back

case_id	case_title	case_type	date_reported	status	officer_id
2	Robbery AT jewellery	theft	2025-10-20	Closed	1
3	robbery	theft	2025-10-20	Closed	1
4	Bank Robbery at MG Road	Robbery	2025-01-15	Open	1
5	Cyber Fraud Case	Cyber Crime	2025-02-01	Open	2
6	Hit and Run Case	Accident	2025-02-10	Closed	8
7	Drug Smuggling	Narcotics	2025-03-05	Open	7
8	Kidnapping Case	Abduction	2025-03-15	Closed	5
9	ATM Theft	Robbery	2025-04-01	Open	1
10	Online Harassment	Cyber Crime	2025-04-20	Open	2
11	Murder at Beach Road	Homicide	2025-05-05	Closed	9

CONCLUSION:

The Crime Database Management System is a comprehensive and well-structured database solution designed to efficiently record, manage, and retrieve criminal, crime, and case-related information. It offers a centralized platform for law enforcement agencies to track cases, monitor criminal activity, and maintain data integrity across multiple entities such as criminals, officers, victims, witnesses, and police stations.

The project successfully demonstrates the core principles of database design including normalization, referential integrity, and relational modeling through well-defined tables and entity relationships. By implementing Stored **Procedures**, the system automates key operations such as adding new cases, registering criminals, updating case statuses, and generating victim and officer reports. These procedures not only improve consistency and accuracy but also reduce manual effort and the likelihood of human errors.

The inclusion of **Triggers** adds a layer of intelligent automation by ensuring real-time updates and maintaining data accuracy. For instance, triggers log important activities like officer additions, case updates, and criminal record insertions, while also enforcing business rules such as preventing the deletion of records linked to open cases or reopening of closed cases. This ensures that the database remains secure, consistent, and fully auditable.

Additionally, the system leverages **SQL queries** to manage large datasets efficiently, supporting operations like case tracking, criminal history retrieval, and crime analysis based on specific parameters such as date or severity. The use of E-R diagrams provided a clear blueprint for the relational structure, ensuring smooth data flow and interconnectivity between different modules.

Overall, this project integrates procedural programming, relational database concepts, and data integrity mechanisms into a unified system that mirrors the real-world needs of modern law enforcement agencies. It highlights how effective database management can contribute to better crime tracking, transparency in case handling, and improved operational decision-making.

In conclusion, the Crime Database Management System stands as a robust, secure, and scalable solution that showcases the power of database automation through procedures, triggers, and structured design ensuring both reliability and efficiency in criminal record management.