UE20CS301

Database Management System

DBMS Mini Project Report

Forensics Database

Submitted By:

Renita Kurian

PES1UG20CS331

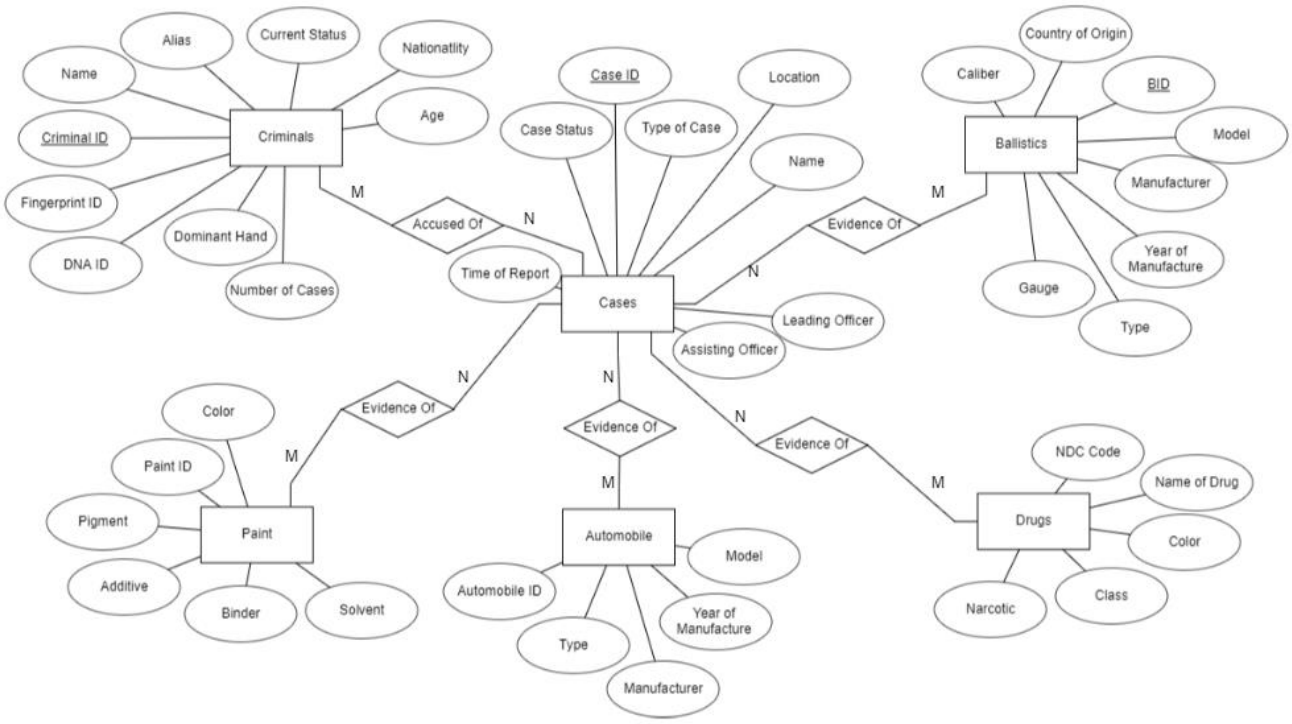
V Semester Section - F

Description and Scope

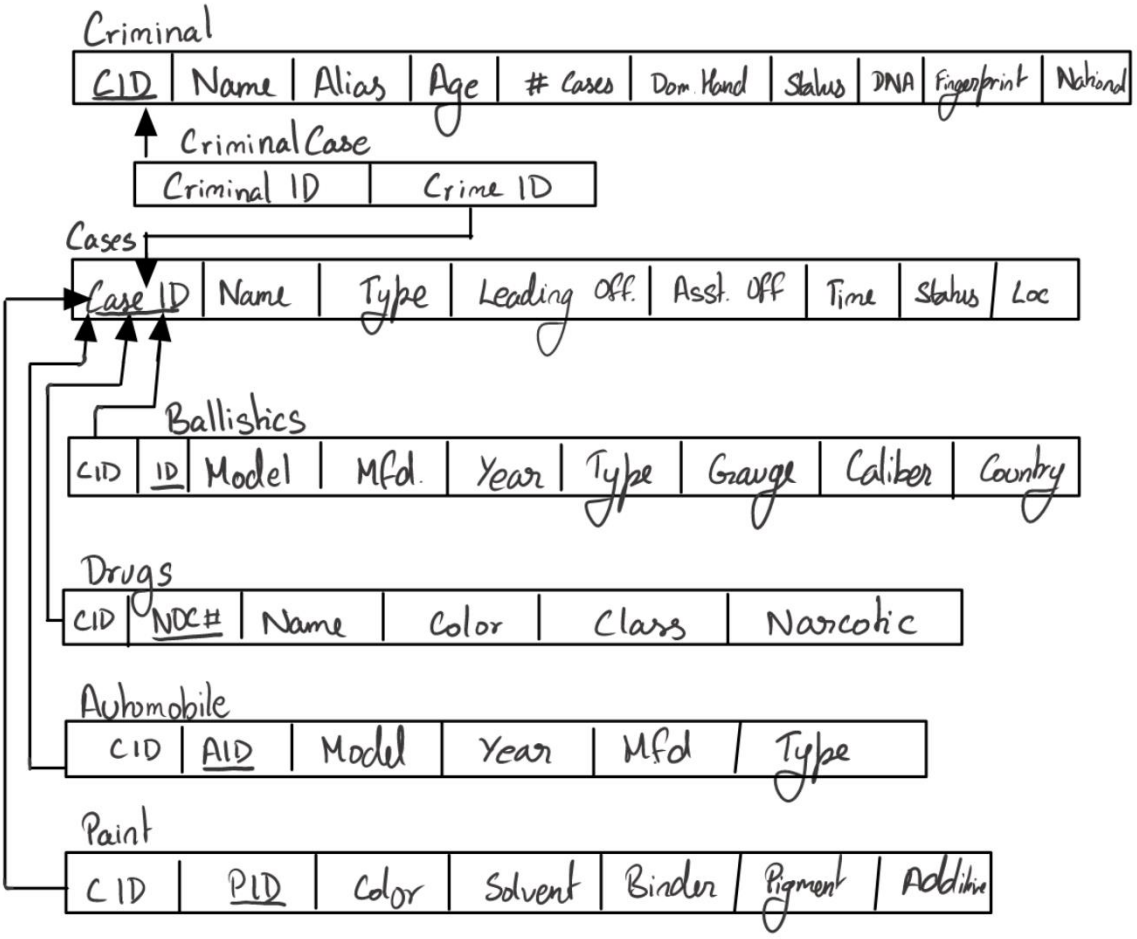
A forensic database system can be used to keep track of incoming evidence instances and their results. Forensic scientists can enter the obtained results from the evidence after analysing the samples. These results can then be viewed by police officers and other officials working on the case.

This project allows forensic scientist and police officers to view and also add new evidence. It supports all CRUD operations. Users can also run SQL queries of their own as well as see the results of predefined queries.

ER Diagram


Criminals
Alias
Current Status
Nationality
Name
Criminal ID
Age
Fingerprint ID
Dominant Hand
DNA ID
Number of Cases
M
Accused Of
N
Cases
Case ID
Location
Case Status
Type of Case
Name
Time of Report
Leading Officer
Assisting Officer
Ballistics
Country of Origin
Caliber
BID
Model
Manufacturer
Year of Manufacture
Gauge
Type
Evidence Of
M
N
Evidence Of
N
M
Paint
Color
Paint ID
Pigment
Additive
Binder
Solvent
Evidence Of
N
M
Automobile
Automobile ID
Model
Year of Manufacture
Type
Manufacturer
Evidence Of
N
M
Drugs
NDC Code
Name of Drug
Color
Narcotic
Class

Relational Schema



Building the Database – DDL

Creating the forensic database

```
MariaDB [(none)]> create database Forensics_331;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> use Forensics_331
Database changed
MariaDB [Forensics_331]>
```

Creating the Cases Table

```
CREATE TABLE CASES(CaseID varchar(255) NOT NULL UNIQUE, TypeOfCase
varchar(255),NameOfCase varchar(255), LeadingOfficer varchar(255),
AsstOfficer varchar(255), TimeOfReport datetime NOT NULL, Loc varchar(255),
statusOfCase varchar(255), PRIMARY KEY(CaseID));
```

```
MariaDB [Forensics_331]> CREATE TABLE CASES(CaseID varchar(255) NOT NULL UNIQUE, TypeOfCase varchar(255),
NameOfCase varchar(255), LeadingOfficer varchar(255), AsstOfficer varchar(255), TimeOfReport datetime NOT
 NULL, Loc varchar(255), statusOfCase varchar(255), PRIMARY KEY(CaseID));
Query OK, 0 rows affected (0.039 sec)
```

Creating the Criminal Table

```
CREATE TABLE CRIMINAL(CID varchar(255) NOT NULL UNIQUE, CName varchar(255)
NOT NULL, Alias varchar(255), Age int, NoOfCases int, DominantHand
varchar(255), CurrentStatus varchar(255), DNAID varchar(50), FingerprintID
varchar(50), nationality varchar(255), PRIMARY KEY(CID));
```

```
MariaDB [Forensics_331]> -- Criminal Data
MariaDB [Forensics_331]> CREATE TABLE CRIMINAL(CID varchar(255) NOT NULL UNIQUE, CName varchar(255) NOT NULL,
 Alias varchar(255), Age int, NoOfCases int, DominantHand varchar(255), CurrentStatus varchar(255), DNAID var
char(50), FingerprintID varchar(50), nationality varchar(255), PRIMARY KEY(CID));
Query OK, 0 rows affected (0.020 sec)
```

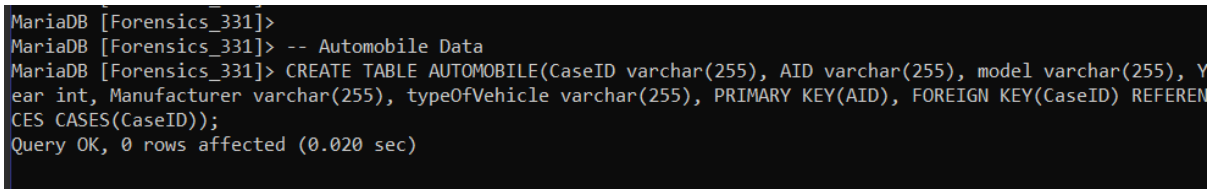
Creating the Criminal-Crime Table

```
CREATE TABLE CriminalCase(CriminalID varchar(255), CrimeID varchar(255),
FOREIGN KEY(CriminalID) REFERENCES CRIMINAL(CID), FOREIGN KEY(CrimeID)
REFERENCES CASES(CaseID));
```

```
MariaDB [Forensics_331]> -- Criminal Case
MariaDB [Forensics_331]> CREATE TABLE CriminalCase(CriminalID varchar(255), CrimeID varchar(255), FOREIGN KEY
(CriminalID) REFERENCES CRIMINAL(CID), FOREIGN KEY(CrimeID) REFERENCES CASES(CaseID));
Query OK, 0 rows affected (0.022 sec)
```

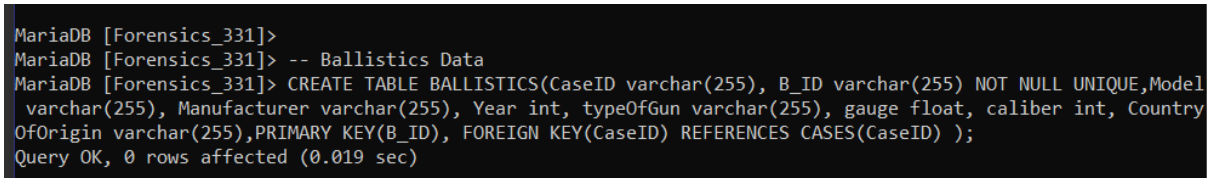
Creating Automobile Table

```
CREATE TABLE AUTOMOBILE(CaseID varchar(255), AID varchar(255), model
varchar(255), Year int, Manufacturer varchar(255), typeOfVehicle
varchar(255), PRIMARY KEY(AID), FOREIGN KEY(CaseID) REFERENCES
CASES(CaseID));
```



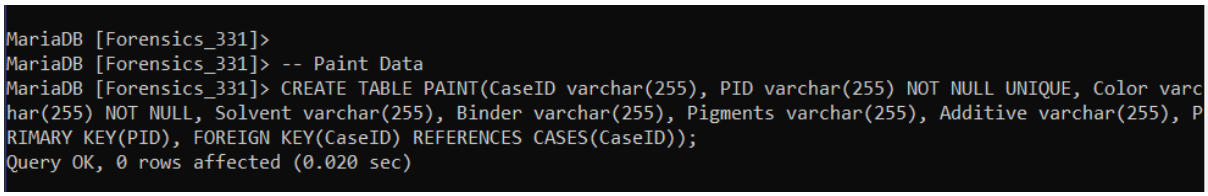
Creating Ballistics Table

```
CREATE TABLE BALLISTICS(CaseID varchar(255), B_ID varchar(255) NOT NULL
UNIQUE,Model varchar(255), Manufacturer varchar(255), Year int, typeOfGun
varchar(255), gauge float, caliber int, CountryOfOrigin
varchar(255),PRIMARY KEY(B_ID), FOREIGN KEY(CaseID) REFERENCES
CASES(CaseID) );
```



Creating Paint Table

```
CREATE TABLE PAINT(CaseID varchar(255), PID varchar(255) NOT NULL UNIQUE,
Color varchar(255) NOT NULL, Solvent varchar(255), Binder varchar(255),
Pigments varchar(255), Additive varchar(255), PRIMARY KEY(PID), FOREIGN
KEY(CaseID) REFERENCES CASES(CaseID));
```



Populating the database

Inserting values into Cases using load data command

```
Load data infile 'cases.csv' into table CASES
columns terminated by ','
optionally enclosed by '"'
escaped by '"'
lines terminated by '\n'
ignore 1 lines;
```

```
MariaDB [Forensics_331]> Load data infile 'Cases.csv' into table CASES
    -> columns terminated by ','
    -> optionally enclosed by '"'
    -> escaped by '"'
    -> lines terminated by '\n'
    -> ignore 1 lines;
Query OK, 4 rows affected, 3 warnings (0.014 sec)
Records: 4  Deleted: 0  Skipped: 0  Warnings: 3

MariaDB [Forensics_331]> select * from cases;
+--------+------------+---------------------+----------------+--------------+---------------------+-----------+-------------------+
| CaseID | TypeOfCase | NameOfCase          | LeadingOfficer | AsstOfficer  | TimeOfReport        | Loc       | statusOfCase      |
+--------+------------+---------------------+----------------+--------------+---------------------+-----------+-------------------+
| L102   | Larceny    | City National Fraud | Jake Peralta   | Charles Boyle | 2022-11-20 13:54:25 | New York  | Archived"
"M12   |
| M46    | Murder     | Tuomi Murder        | Ray Holt       | Charles Boyle | 2022-11-20 13:49:08 | Milwaukee | Archived"
"M53   |
| M981   | Murder     | Capo Murder         | Jake Peralta   | Amy Santiago  | 2022-11-20 14:16:28 | Chicago   | Ongoing"
"QT09P1 |
| T22    | Theft      | Private Jet Theft   | Jake Peralta   | Charles Boyle | 2022-11-20 13:54:59 | New York  | "Archived"
     |
+--------+------------+---------------------+----------------+--------------+---------------------+-----------+-------------------+
4 rows in set (0.001 sec)
```

Inserting values into Criminal Table using load data command

```
Load data infile 'criminal.csv' into table CRIMINAL
columns terminated by ','
optionally enclosed by '"'
escaped by '"'
lines terminated by '\n'
ignore 1 lines;
```

```
MariaDB [Forensics_331]> -- Inserting data into criminals table
MariaDB [Forensics_331]> Load data infile 'criminal.csv' into table CRIMINAL
    -> columns terminated by ','
    -> optionally enclosed by '"'
    -> escaped by '"'
    -> lines terminated by '\n'
    -> ignore 1 lines;
Query OK, 3 rows affected, 2 warnings (0.007 sec)
Records: 3  Deleted: 0  Skipped: 0  Warnings: 2

MariaDB [Forensics_331]> select * from criminal;
+------+---------------+--------------------+------+-----------+--------------+---------------+----------------+-----------------+-----------------+
| CID  | CName         | Alias              | Age  | NoOfCases | DominantHand | CurrentStatus | DNAID          | FingerprintID   | nationality     |
+------+---------------+--------------------+------+-----------+--------------+---------------+----------------+-----------------+-----------------+
| AK47 | Anita Kurian  | Crime Potato       |   51 |         3 | Left         | Prison        | DSLFUKG4059LFGH | DKFJGHOF213KSJVH | British"
"RX12  |
| SK24 | Anna Sorokin  | Anna Delvey        |   27 |         6 | Unknown      | Prison        | OHJF23943JG    | XC245XXM        | German"
"Z12QA1 |
| ZK53 | Jeffrey Dahmer | Milwaukee Cannibal |   25 |        17 | Left         | Dead          | OERNO123JG     | ZWGKGJ123DFOG   | "American"
     |
+------+---------------+--------------------+------+-----------+--------------+---------------+----------------+-----------------+-----------------+
3 rows in set (0.001 sec)
```

Inserting values into drugs table using insert Command

```
INSERT INTO `drugs` (`CaseID`, `NDC_No`, `dname`, `color`, `class`,
`narcotic`) VALUES
('T22', '6745103120', 'Lexapro', 'blue', 'analgesic', 'yes'),
('M53', '6745718120', 'Ketamine', 'white', 'inhalants', 'yes'),
('M981', '6998813120', 'Heroin', 'white', 'opioid', 'no'),
('QT09P1', '8861238761', 'Axypenetril', 'Pink', 'Hallucinogins', 'no'),
('QT09P1', '97234698', 'Nescipixinol', 'Green', 'Inhalants', 'no');
```

```
MariaDB [Forensics_331]> INSERT INTO `drugs` (`CaseID`, `NDC_No`, `dname`, `color`, `class`, `narcotic`) VALUES
    -> ('T22', '6745103120', 'Lexapro', 'blue', 'analgesic', 'yes'),
    -> ('M53', '6745718120', 'Ketamine', 'white', 'inhalants', 'yes'),
    -> ('M981', '6998813120', 'Heroin', 'white', 'opioid', 'no'),
    -> ('QT09P1', '8861238761', 'Axypenetril', 'Pink', 'Hallucinogins', 'no'),
    -> ('QT09P1', '97234698', 'Nescipixinol', 'Green', 'Inhalants', 'no');
Query OK, 5 rows affected (0.006 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Inserting values into ballistics table using insert command

```
INSERT INTO `ballistics` (`CaseID`, `B_ID`, `Model`, `Manufacturer`,
`Year`, `typeOfGun`, `gauge`, `caliber`, `CountryOfOrigin`) VALUES
('M981', 'H39', 'Automag II', 'AMT', 1970, 'Handgun', 410, 9, 'USA'),
('M981', 'H9', 'Glock 21', 'Glock', 1970, 'Handgun', -1, 10, 'Austria'),
('M46', 'S123PW', 'Glock 20', 'Glock', 1980, 'Pistol', 9, -1, 'Austria'),
('M981', 'SH09', 'Benneli M1', 'Benneli Armi', 1986, 'Shotgun', 20, -1,
'Italy'),
('M981', 'SH23', 'Benneli M3', 'Benneli Armi', 1989, 'Shotgun', 20, -1,
'Italy');
```

```
MariaDB [Forensics_331]> INSERT INTO `ballistics` (`CaseID`, `B_ID`, `Model`, `Manufacturer`, `Year`, `typeOfGun`, `gaug
e`, `caliber`, `CountryOfOrigin`) VALUES
    -> ('M981', 'H39', 'Automag II', 'AMT', 1970, 'Handgun', 410, 9, 'USA'),
    -> ('M981', 'H9', 'Glock 21', 'Glock', 1970, 'Handgun', -1, 10, 'Austria'),
    -> ('M46', 'S123PW', 'Glock 20', 'Glock', 1980, 'Pistol', 9, -1, 'Austria'),
    -> ('M981', 'SH09', 'Benneli M1', 'Benneli Armi', 1986, 'Shotgun', 20, -1, 'Italy'),
    -> ('M981', 'SH23', 'Benneli M3', 'Benneli Armi', 1989, 'Shotgun', 20, -1, 'Italy');
Query OK, 5 rows affected (0.007 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Join Queries

Cars associated with cases

```
select NameOfCase, model, Manufacturer from (AUTOMOBILE JOIN CASES ON
AUTOMOBILE.CaseID=CASES.CaseID);
```

```
MariaDB [Forensics_331]> select NameOfCase, model, Manufacturer from (AUTOMOBILE JOIN CASES ON AUTOMOBILE.CaseID=CASES.C
aseID);
+------------------+----------------+--------------+
| NameOfCase       | model          | Manufacturer |
+------------------+----------------+--------------+
| Hicks Murder     | Hemi 66 Charger | Dodge        |
| Private Jet Theft | Model S        | Tesla        |
| Private Jet Theft | Model X        | Tesla        |
+------------------+----------------+--------------+
3 rows in set (0.001 sec)
```

Criminals and the Cases they've been accused of

```
select CName, NameOfCase from (CriminalCase JOIN Criminal ON
CriminalCase.CriminalID=Criminal.CID) JOIN CASES ON
CASES.CaseID=CriminalCase.CrimeID;
```

```
MariaDB [Forensics_331]> select CName, NameOfCase from (CriminalCase JOIN Criminal ON CriminalCase.CriminalID=Criminal.C
ID) JOIN CASES ON CASES.CaseID=CriminalCase.CrimeID;
+----------------+------------------+
| CName          | NameOfCase       |
+----------------+------------------+
| Ted Bundy      | Ann Heally Murder |
| Anna Sorokin   | Private Jet Theft |
| Jeffrey Dahmer | Hicks Murder     |
| Jeffrey Dahmer | Tuomi Murder     |
+----------------+------------------+
4 rows in set (0.001 sec)
```

Criminals and the type of Cases they've been accused of

```
select DISTINCT CName, TypeOfCase  from (CriminalCase JOIN Criminal ON
CriminalCase.CriminalID=Criminal.CID) JOIN CASES ON
CASES.CaseID=CriminalCase.CrimeID;
```

```
MariaDB [Forensics_331]> select DISTINCT CName, TypeOfCase  from (CriminalCase JOIN Criminal ON CriminalCase.CriminalID=
Criminal.CID) JOIN CASES ON CASES.CaseID=CriminalCase.CrimeID;
+----------------+------------+
| CName          | TypeOfCase |
+----------------+------------+
| Ted Bundy      | Murder     |
| Anna Sorokin   | Theft      |
| Jeffrey Dahmer | Murder     |
+----------------+------------+
3 rows in set (0.001 sec)
```

Criminals and the officers investigating them

```
select DISTINCT CName, LeadingOfficer from (CASES JOIN (CRIMINALCASE JOIN
CRIMINAL ON CRIMINALCASE.CriminalID=CRIMINAL.CID) ON
Cases.CaseID=CriminalCase.CrimeID)

UNION

select DISTINCT CName, AsstOfficer from (CASES JOIN (CRIMINALCASE JOIN
CRIMINAL ON CRIMINALCASE.CriminalID=CRIMINAL.CID) ON
Cases.CaseID=CriminalCase.CrimeID)
```

```
MariaDB [Forensics_331]> select DISTINCT CName, LeadingOfficer from (CASES JOIN (CRIMINALCASE JOIN CRIMINAL ON CRIMINALC
ASE.CriminalID=CRIMINAL.CID) ON Cases.CaseID=CriminalCase.CrimeID)
    -> UNION
    -> select DISTINCT CName, AsstOfficer from (CASES JOIN (CRIMINALCASE JOIN CRIMINAL ON CRIMINALCASE.CriminalID=CRIMIN
AL.CID) ON Cases.CaseID=CriminalCase.CrimeID)
    -> ;
+---------------+---------------+
| CName         | LeadingOfficer |
+---------------+---------------+
| Ted Bundy     | Amy Santiago   |
| Anna Sorokin  | Jake Peralta   |
| Jeffrey Dahmer | Jake Peralta   |
| Jeffrey Dahmer | Ray Holt       |
| Ted Bundy     | Gina Linetti   |
| Anna Sorokin  | Charles Boyle  |
| Jeffrey Dahmer | Rosa Diaz      |
| Jeffrey Dahmer | Charles Boyle  |
+---------------+---------------+
8 rows in set (0.005 sec)
```

Aggregate Functions

Cases with drug evidence and number of drug evidence instances for each

```
select NameOfCase, count(*) from (DRUGS NATURAL JOIN CASES) group by
CaseID;
```

```
MariaDB [Forensics_331]> select NameOfCase, count(*) from (DRUGS NATURAL JOIN CASES) group by CaseID;
+---------------------+----------+
| NameOfCase          | count(*) |
+---------------------+----------+
| Ann Heally Murder   |        1 |
| Capo Murder         |        1 |
| Greenlane 15 Robbery |        2 |
| Private Jet Theft   |        1 |
+---------------------+----------+
4 rows in set (0.001 sec)
```

Paints grouped by solvent

```
select Solvent, count(*) from Paint group by Solvent;
```

```
MariaDB [Forensics_331]> select Solvent, count(*) from Paint group by Solvent;
+---------+----------+
| Solvent | count(*) |
+---------+----------+
| Benzene |        2 |
| Toluene |        2 |
+---------+----------+
2 rows in set (0.001 sec)
```

Number of Non-Narcotic and Narcotic drugs

```
select narcotic, count(*) from DRUGS group by narcotic;
```

```
MariaDB [Forensics_331]> select narcotic, count(*) from DRUGS group by narcotic;
+----------+----------+
| narcotic | count(*) |
+----------+----------+
| no       |        3 |
| yes      |        2 |
+----------+----------+
2 rows in set (0.001 sec)
```

Number of cases for each Location

```
select Loc, count(*) from CASES group by Loc;
```

```
MariaDB [Forensics_331]> select Loc, count(*) from CASES group by Loc;
+------------+----------+
| Loc        | count(*) |
+------------+----------+
| Brooklyn   |        1 |
| Chicago    |        1 |
| Milwaukee  |        1 |
| New York   |        1 |
| Ohio       |        1 |
| Washington |        1 |
+------------+----------+
6 rows in set (0.001 sec)
```

Set Operations

Names of blue narcotic drugs

```
select dname from DRUGS where narcotic="yes"
INTERSECT
select dname from DRUGS where color="blue";
```

```
MariaDB [Forensics_331]> select dname from DRUGS where narcotic="yes"
    -> INTERSECT
    -> select dname from DRUGS where color="blue";
+---------+
| dname   |
+---------+
| Lexapro |
+---------+
1 row in set (0.003 sec)

MariaDB [Forensics_331]>
```

Handguns manufactured in 1970

```
select model, Manufacturer, gauge, caliber from BALLISTICS where Year=1970
INTERSECT
select model, Manufacturer, gauge, caliber from BALLISTICS where
typeOfGun="Handgun";
```

```
MariaDB [Forensics_331]> select model, Manufacturer, gauge, caliber from BALLISTICS where Year=1970
    -> INTERSECT
    -> select model, Manufacturer, gauge, caliber from BALLISTICS where typeOfGun="Handgun";
+------------+--------------+-------+---------+
| model      | Manufacturer | gauge | caliber |
+------------+--------------+-------+---------+
| Automag II | AMT          |   410 |       9 |
| Glock 21   | Glock        |    -1 |      10 |
+------------+--------------+-------+---------+
2 rows in set (0.004 sec)
```

Cases lead by Jake Peralta and Assisted by Amy Santiago

```
select NameOfCase from CASES where LeadingOfficer="Jake Peralta"
INTERSECT
select NameOfCase from CASES where AsstOfficer="Amy Santiago";
```

```
MariaDB [Forensics_331]> select NameOfCase from CASES where LeadingOfficer="Jake Peralta"
    -> INTERSECT
    -> select NameOfCase from CASES where AsstOfficer="Amy Santiago";
+-------------+
| NameOfCase  |
+-------------+
| Capo Murder |
+-------------+
1 row in set (0.001 sec)
```

Cars manufactured by Audi or Dodge

```
select model from AUTOMOBILE where Manufacturer="Audi"
UNION
select model from AUTOMOBILE where Manufacturer="Dodge"
```

```
MariaDB [Forensics_331]> select model from AUTOMOBILE where Manufacturer="Audi"
    -> UNION
    -> select model from AUTOMOBILE where Manufacturer="Dodge"
    -> ;
+-----------------+
| model           |
+-----------------+
| Hemi 66 Charger |
+-----------------+
1 row in set (0.002 sec)
```

Functions

Function to return number of cases lead by an officer

```
DELIMITER $$
CREATE FUNCTION number_of_cases(officer varchar(255))
RETURNS int
DETERMINISTIC
BEGIN
    DECLARE case_count int;

    SELECT count(CaseID) into case_count
      FROM CASES
      WHERE LeadingOfficer = officer;

      RETURN case_count;
END; $$
DELIMITER ;
```

```
MariaDB [Forensics_331]> select distinct LeadingOfficer, number_of_cases(LeadingOfficer) from Cases;
+---------------+---------------------------------+
| LeadingOfficer | number_of_cases(LeadingOfficer) |
+---------------+---------------------------------+
| Jake Peralta   |                               3 |
| Ray Holt       |                               1 |
| Amy Santiago   |                               1 |
| Michaela Stone |                               1 |
+---------------+---------------------------------+
4 rows in set (0.001 sec)
```

Function to return number of criminals given status – ie, the function can return no. of prison, active etc.

```
DELIMITER $$
CREATE FUNCTION number_of_criminals(stat varchar(255))
RETURNS int
DETERMINISTIC
BEGIN
    DECLARE c int;

    SELECT count(CID) into c
      FROM Criminal
      WHERE CurrentStatus = stat;

      RETURN c;
END; $$
DELIMITER ;
```

```
MariaDB [Forensics_331]> select distinct CurrentStatus, number_of_criminals(CurrentStatus) from CRIMINAL;
+---------------+-----------------------------------+
| CurrentStatus | number_of_criminals(CurrentStatus) |
+---------------+-----------------------------------+
| Unknown       |                                 2 |
| Prison        |                                 1 |
| Dead          |                                 1 |
+---------------+-----------------------------------+
3 rows in set (0.001 sec)
```

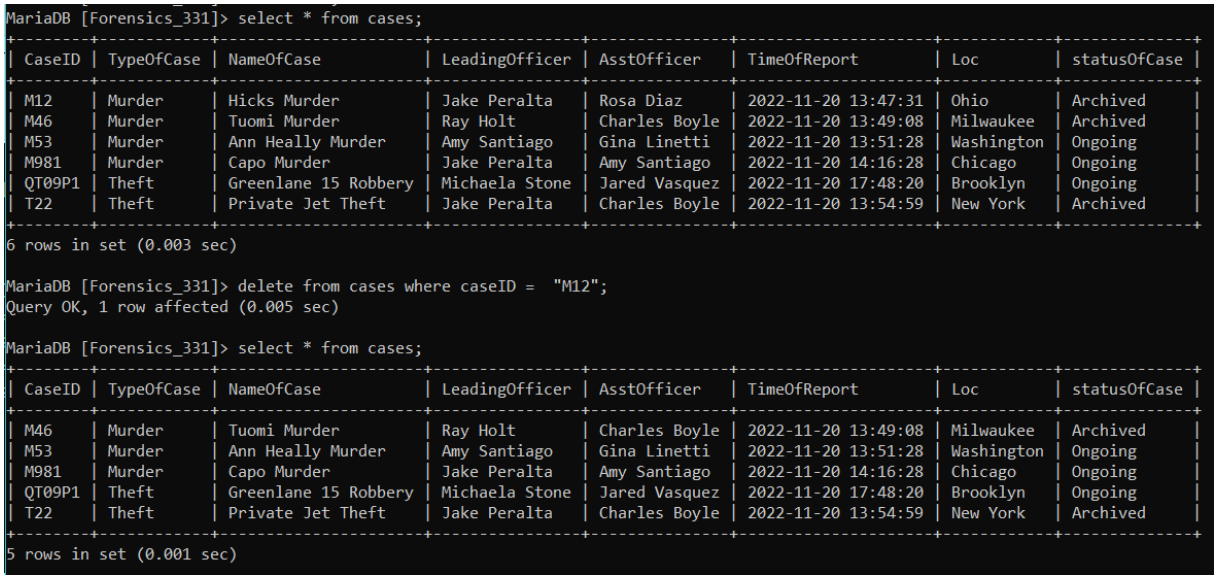
Trigger

Trigger to allow case deletion

```
DELIMITER $$
CREATE TRIGGER delCase
BEFORE DELETE
ON CASES FOR EACH ROW
BEGIN
    delete from CriminalCase where CrimeID = old.CaseID;
    delete from DRUGS where CaseID = old.CaseID;
    delete from BALLISTICS where CaseID = old.CaseID;
    delete from PAINT where CaseID = old.CaseID;
    delete from AUTOMOBILE where CaseID = old.CaseID;
END $$
DELIMITER ;
```

```
MariaDB [Forensics_331]> DELIMITER $$
MariaDB [Forensics_331]> CREATE TRIGGER delCase
    -> BEFORE DELETE
    -> ON CASES FOR EACH ROW
    -> BEGIN
    ->     delete from CriminalCase where CrimeID = old.CaseID;
    ->     delete from DRUGS where CaseID = old.CaseID;
    ->     delete from BALLISTICS where CaseID = old.CaseID;
    ->     delete from PAINT where CaseID = old.CaseID;
    ->     delete from AUTOMOBILE where CaseID = old.CaseID;
    -> END $$
Query OK, 0 rows affected (0.005 sec)

MariaDB [Forensics_331]>
MariaDB [Forensics_331]> DELIMITER ;
MariaDB [Forensics_331]>
```

The above trigger allows users to delete case information. Without this trigger it is not possible to delete case information due to foreign key parent constraints. This trigger deals with that by first deleting all evidence related to a case before deleting the case record.



```
MariaDB [Forensics_331]> select * from cases;
+--------+-----------+----------------------+----------------+--------------+---------------------+------------+--------------+
| CaseID | TypeOfCase | NameOfCase           | LeadingOfficer | AsstOfficer  | TimeOfReport        | Loc        | statusOfCase |
+--------+-----------+----------------------+----------------+--------------+---------------------+------------+--------------+
| M12    | Murder     | Hicks Murder         | Jake Peralta   | Rosa Diaz    | 2022-11-20 13:47:31 | Ohio       | Archived     |
| M46    | Murder     | Tuomi Murder         | Ray Holt       | Charles Boyle | 2022-11-20 13:49:08 | Milwaukee  | Archived     |
| M53    | Murder     | Ann Heally Murder    | Amy Santiago   | Gina Linetti | 2022-11-20 13:51:28 | Washington | Ongoing      |
| M981   | Murder     | Capo Murder          | Jake Peralta   | Amy Santiago | 2022-11-20 14:16:28 | Chicago    | Ongoing      |
| QT09P1 | Theft      | Greenlane 15 Robbery | Michaela Stone | Jared Vasquez | 2022-11-20 17:48:20 | Brooklyn   | Ongoing      |
| T22    | Theft      | Private Jet Theft    | Jake Peralta   | Charles Boyle | 2022-11-20 13:54:59 | New York   | Archived     |
+--------+-----------+----------------------+----------------+--------------+---------------------+------------+--------------+
6 rows in set (0.003 sec)

MariaDB [Forensics_331]> delete from cases where caseID =  "M12";
Query OK, 1 row affected (0.005 sec)

MariaDB [Forensics_331]> select * from cases;
+--------+-----------+----------------------+----------------+--------------+---------------------+------------+--------------+
| CaseID | TypeOfCase | NameOfCase           | LeadingOfficer | AsstOfficer  | TimeOfReport        | Loc        | statusOfCase |
+--------+-----------+----------------------+----------------+--------------+---------------------+------------+--------------+
| M46    | Murder     | Tuomi Murder         | Ray Holt       | Charles Boyle | 2022-11-20 13:49:08 | Milwaukee  | Archived     |
| M53    | Murder     | Ann Heally Murder    | Amy Santiago   | Gina Linetti | 2022-11-20 13:51:28 | Washington | Ongoing      |
| M981   | Murder     | Capo Murder          | Jake Peralta   | Amy Santiago | 2022-11-20 14:16:28 | Chicago    | Ongoing      |
| QT09P1 | Theft      | Greenlane 15 Robbery | Michaela Stone | Jared Vasquez | 2022-11-20 17:48:20 | Brooklyn   | Ongoing      |
| T22    | Theft      | Private Jet Theft    | Jake Peralta   | Charles Boyle | 2022-11-20 13:54:59 | New York   | Archived     |
+--------+-----------+----------------------+----------------+--------------+---------------------+------------+--------------+
5 rows in set (0.001 sec)
```

The above picture shows the table before and after deleting. As shown above the case “M12” has been successfully deleted.

Trigger to delete and backup criminal records

```
DELIMITER $$
CREATE TRIGGER delCriminal
BEFORE DELETE
ON CRIMINAL FOR EACH ROW
BEGIN
    DECLARE cid, criminalname, a, d, n varchar(255);
    DECLARE ncases int;
    DECLARE c1 CURSOR FOR SELECT CID, CName, Alias, NoOfCases,
DominantHand, nationality from CRIMINAL where CID = old.CID;
    open c1;
    fetch c1 into cid, criminalname, a, ncases, d, n;
    insert into criminalBackup values(cid, criminalName, a, ncases, d, n);
    close c1;
    delete from CriminalCase where CriminalID = old.CID;
END $$
DELIMITER ;
```

```
MariaDB [Forensics_331]> DELIMITER $$
MariaDB [Forensics_331]> CREATE TRIGGER delCriminal
    -> BEFORE DELETE
    -> ON CRIMINAL FOR EACH ROW
    -> BEGIN
    ->     DECLARE id, criminalname, a, d, n varchar(255);
    ->     DECLARE ncases int;
    ->     DECLARE c1 CURSOR FOR SELECT CID, CName, Alias, NoOfCases, DominantHand, nationality from CRIMINAL where CID
= old.CID;
    ->     open c1;
    ->     fetch c1 into id, criminalname, a, ncases, d, n;
    ->     insert into criminalBackup values(id, criminalName, a, ncases, d, n);
    ->     close c1;
    ->     delete from CriminalCase where CriminalID = old.CID;
    -> END $$
Query OK, 0 rows affected (0.012 sec)

MariaDB [Forensics_331]>
MariaDB [Forensics_331]> DELIMITER ;
```

Result for above trigger

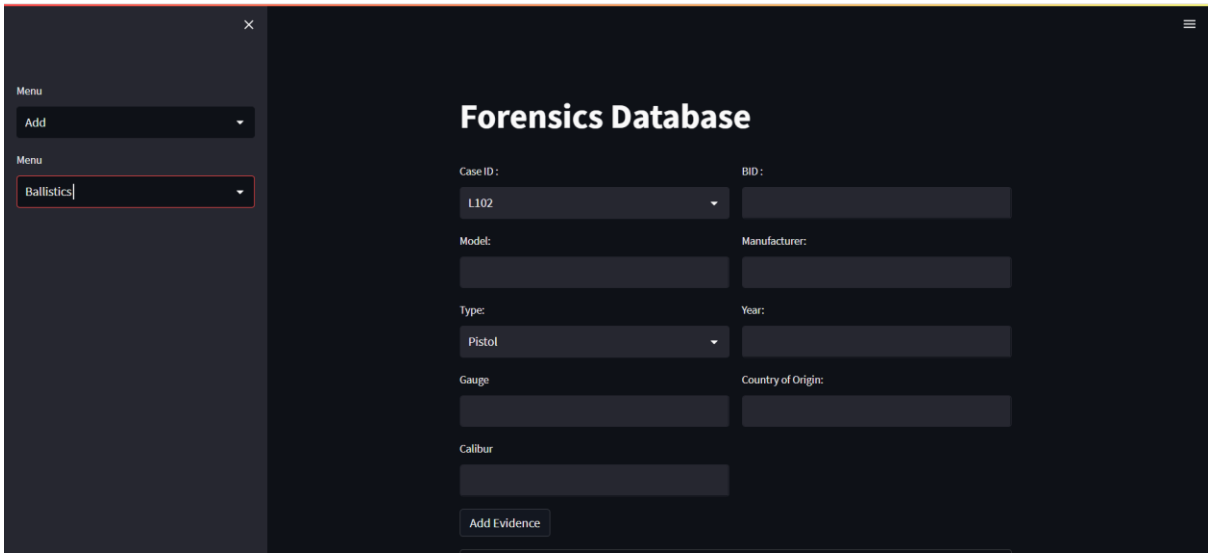
```
MariaDB [Forensics_331]> select * from criminal;
+-------+---------------+-------------------+------+-----------+--------------+---------------+------------+---------------+-------------+
| CID   | CName         | Alias             | Age  | NoOfCases | DominantHand | CurrentStatus | DNAID      | FingerprintID | nationality |
+-------+---------------+-------------------+------+-----------+--------------+---------------+------------+---------------+-------------+
| 213FD | Joe Palmer    | None              |   23 |         1 | Unknown      | Prison        | SGH23RSDV  | SIFDGH123     | Russian     |
| ZK53  | Jeffrey Dahmer | Milwaukee Cannibal |   25 |        17 | Left         | Dead          | OERNO123JG | ZWGKGJ123DFOG | American    |
+-------+---------------+-------------------+------+-----------+--------------+---------------+------------+---------------+-------------+
2 rows in set (0.001 sec)

MariaDB [Forensics_331]> delete from criminal where CName="Joe Palmer";
Query OK, 1 row affected (0.014 sec)

MariaDB [Forensics_331]> select * from criminalBackup;
+-------+-------------+---------------+-----------+--------------+-------------+
| ID    | CriminalName | Alias         | NoOfCases | dominantHand | Nationality |
+-------+-------------+---------------+-----------+--------------+-------------+
| NULL  | Luke Wrenner | Sticky Fingers |         2 | Right        | American    |
| NULL  | Anna Sorokin | Anna Delvey    |         6 | Unknown      | German      |
| RX12  | Ted Bundy    | Lady Killer    |        50 | Left         | American    |
| 213FD | Joe Palmer   | None           |         1 | Unknown      | Russian     |
+-------+-------------+---------------+-----------+--------------+-------------+
4 rows in set (0.001 sec)

MariaDB [Forensics_331]>
```

Frontend –

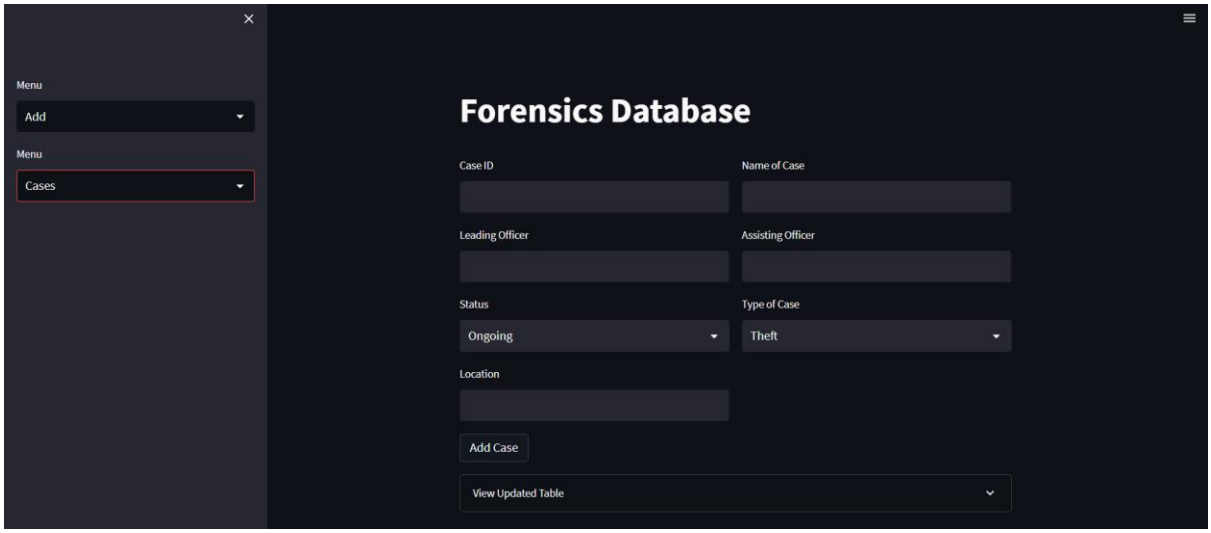
The application allows users to perform CRUD operations, run predefined queries and also provides a terminal to run custom queries. The user is presented with a sidebar and the main page on opening the website. The sidebar allows the user to choose between, add, view, edit, remove, predefined queries and CMD option. On choosing an option, the user is taken to the respective page for further steps.

Create Operations

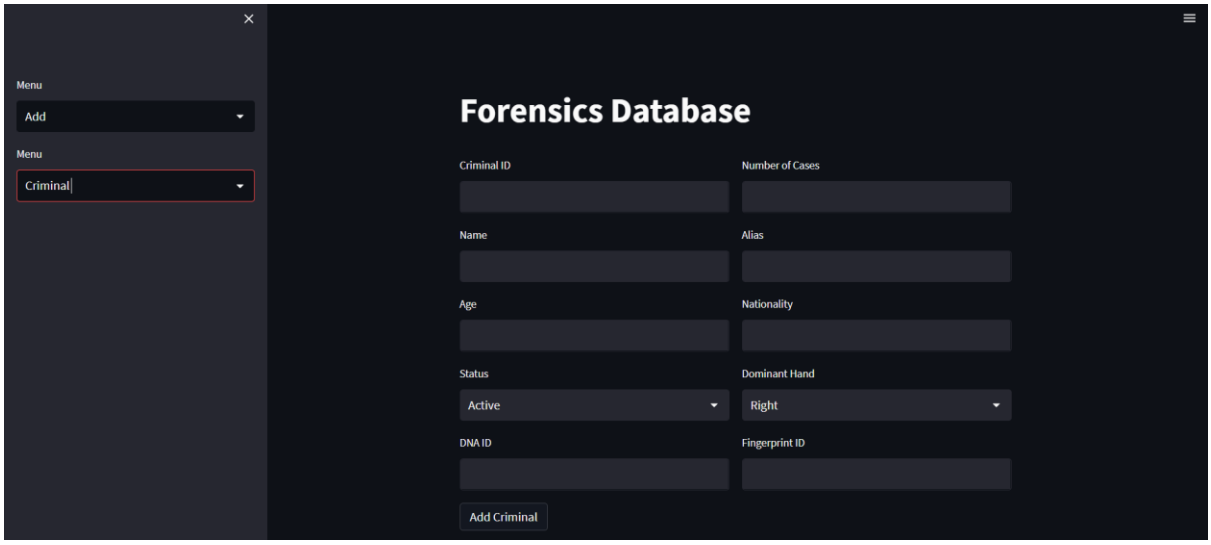
On selecting the add option, the user is presented with a second drop down list to choose a table for which new record is to be added. The user is then provided with the relevant input boxes to add a new record.



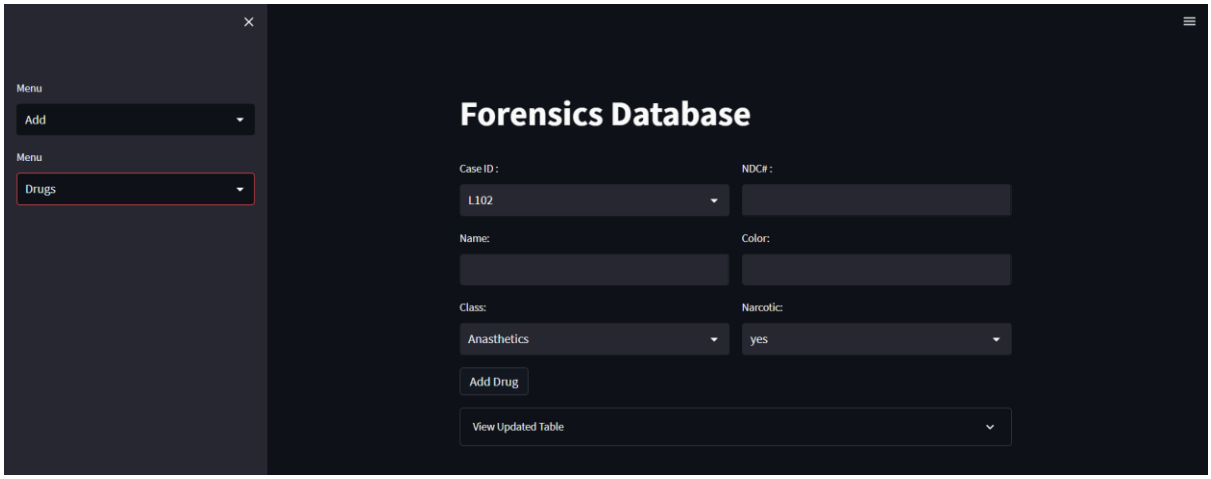
The above picture shows the UI to insert values in the Ballistics tables



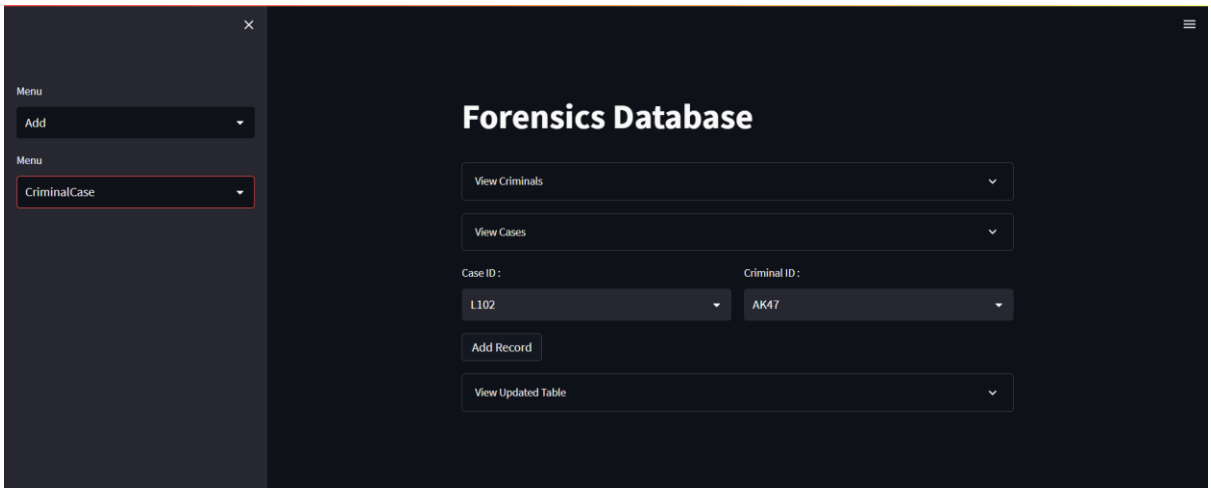
UI to insert a new case



UI to insert a new criminal to database



UI to insert a new drug evidence to database

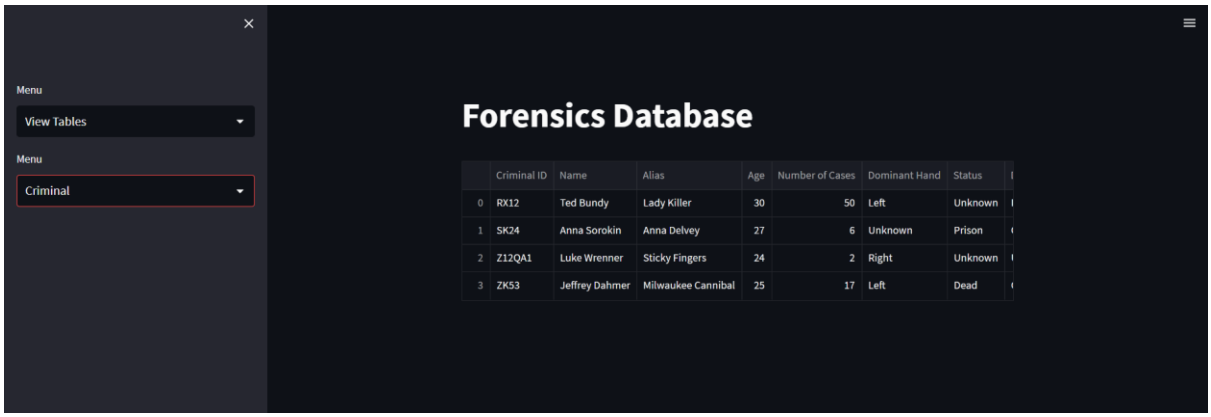


UI to add a new criminal-case record to database

Read Operation

On selecting the view option, the user is once again presented with a new drop-down list to select a table. Once a table is selected the user can see the values in the desired table.

UI to view Drugs table

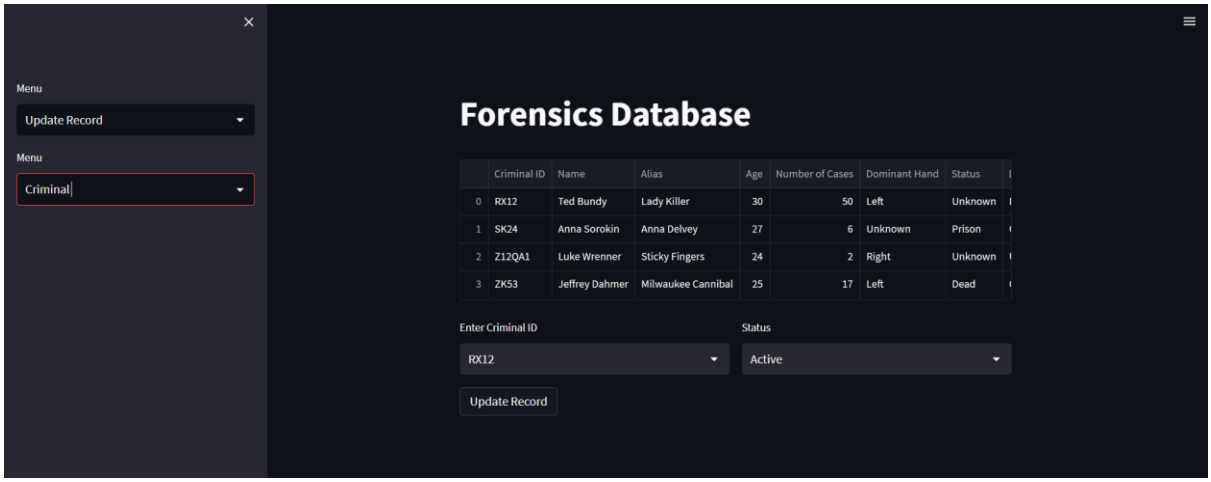


UI to view Criminal table

Update Option

The UI also allows the user to perform update operations. The user can update the status of a criminal or a crime. The user is made to choose the table from the drop-drown list in the sidebar. Once the table is chosen the user can select the relevant id and status from the respective drop-drown lists.

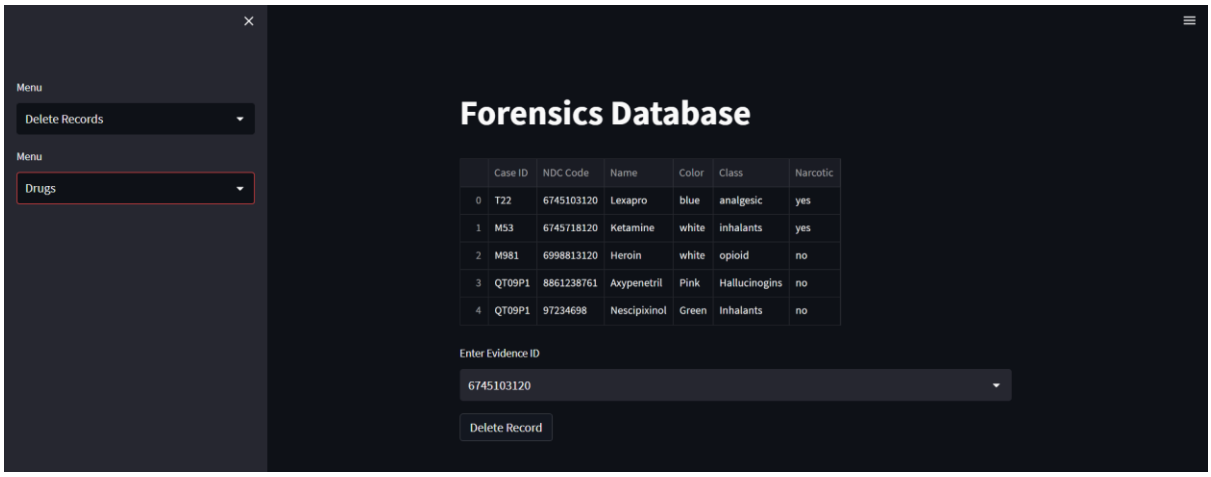
UI to Edit case table



UI to edit criminal table

Delete Operations

The user also has the option to delete evidence from the database using the UI. The user can navigate between the different tables using the dropdown in the sidebar. The user then selects the id of the record to be deleted from the list. Once the id is selected the record can be deleted by clicking on the delete record button.

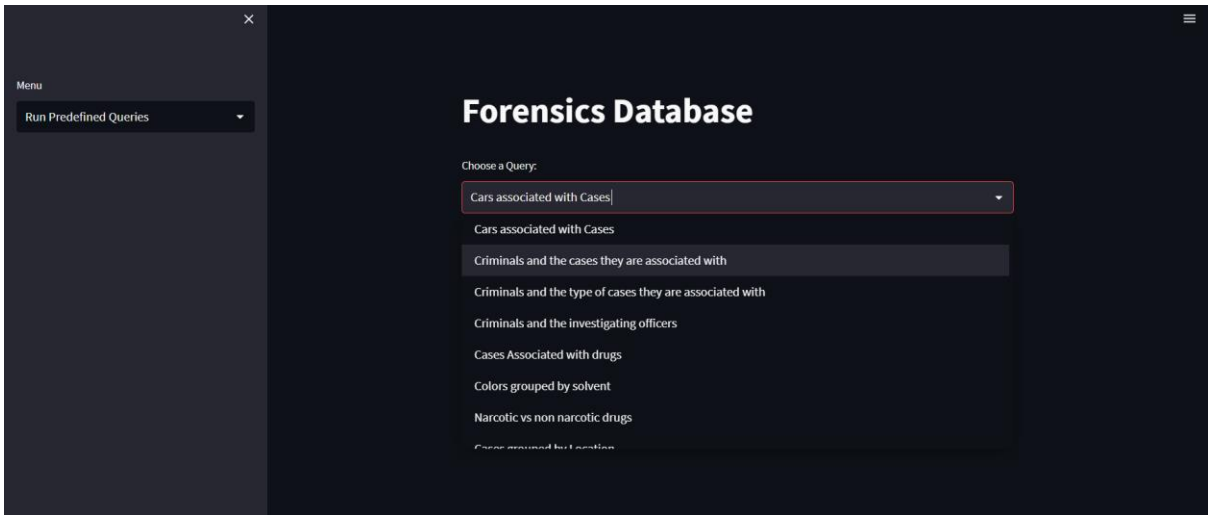


UI to delete drug evidence instance from the table

UI to delete case from the table

Run Predefined Queries

The user can also run queries from a list. These queries include criminals and the name of cases they're accused of, criminal and the types of crimes they've committed, etc.



UI displaying the queries available

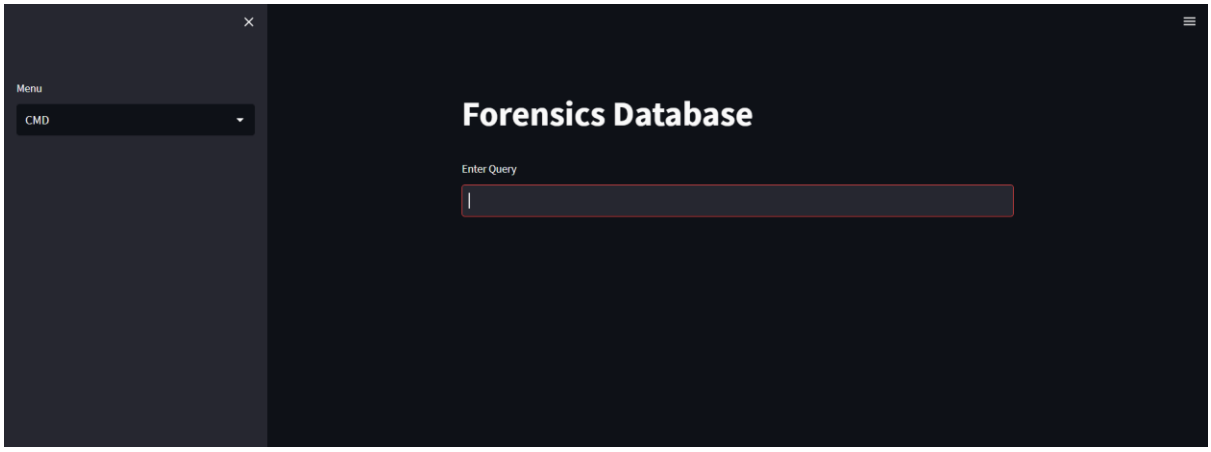
UI displaying the result of a selected query

Cases grouped by location

Cases grouped by type

CMD

The user can enter the desired SQL query in the input box and see the results



UI to input user’s query

UI displaying custom query result