### UE20CS301

## Database Management System

## DBMS Mini Project Report

## Forensics Database

Submitted By:

Renita Kurian

PES1UG20CS331

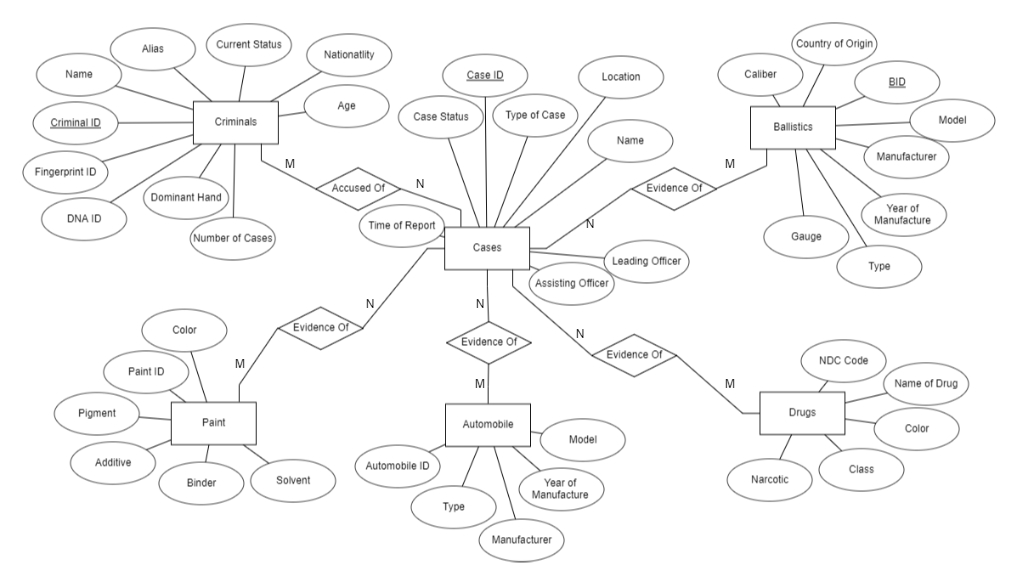
V Semester Section - F

### Description and Scope

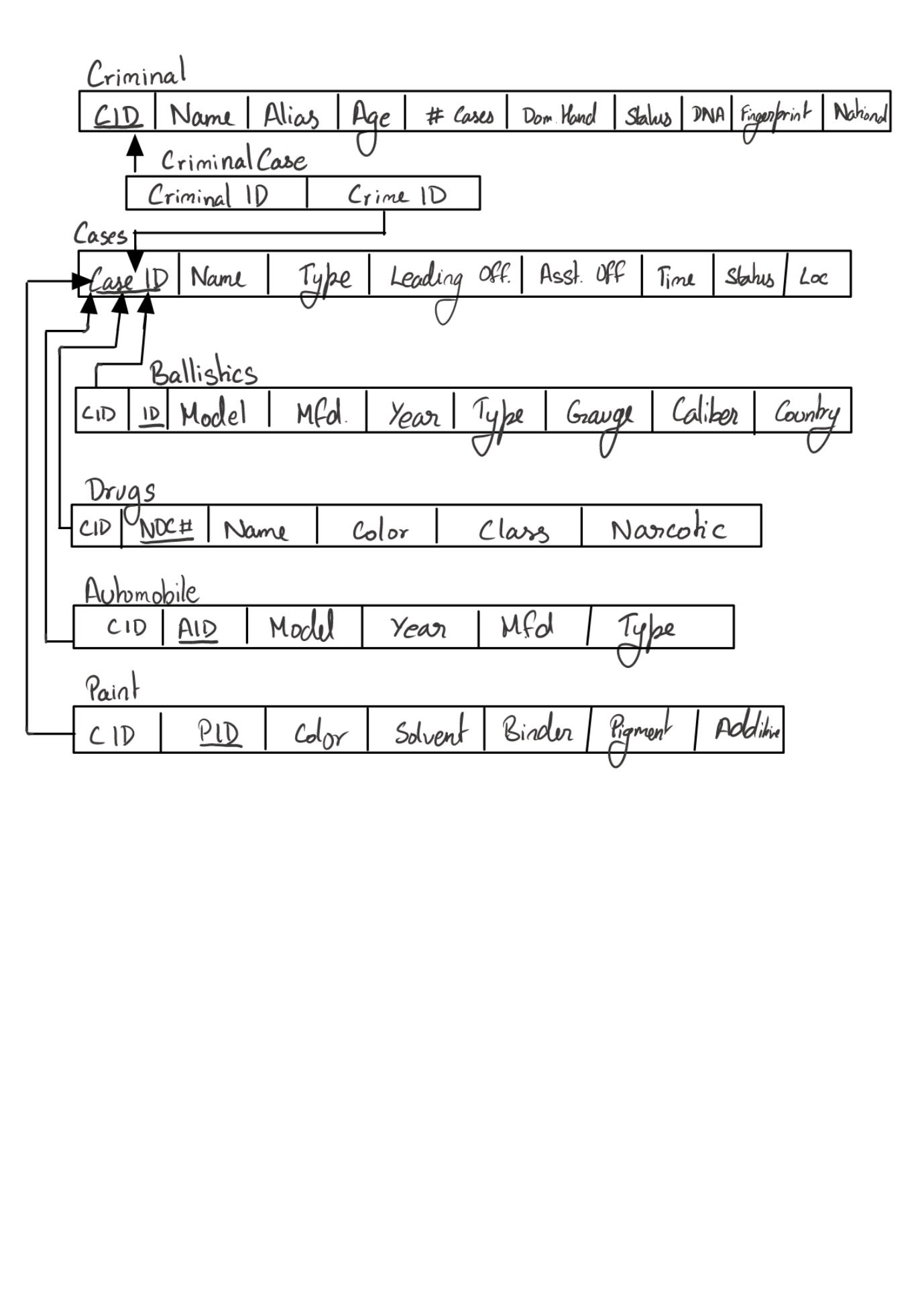
A forensic database system can be used to keep track of incoming evidence instances and their results. Forensic scientists can enter the obtained results from the evidence after analysing the samples. These results can then be viewed by police officers and other officials working on the case.

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

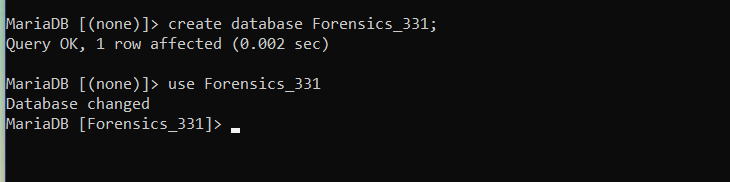


### Relational Schema



### Building the Database – DDL

Creating the forensic database



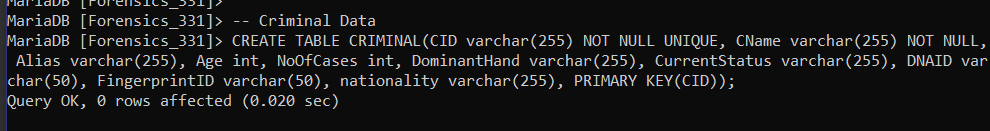
Creating the Cases Table

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));

### 

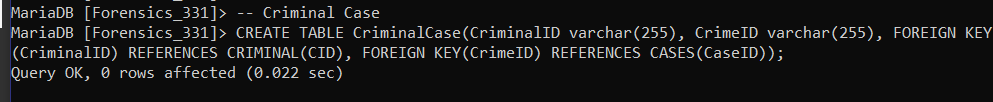
Creating the Criminal Table

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));



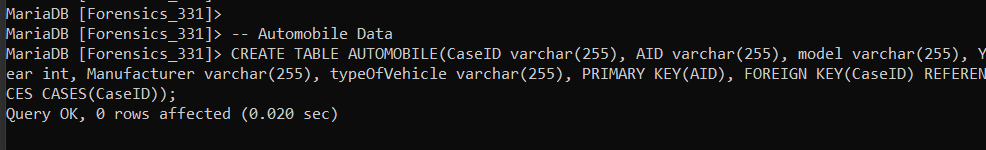
Creating the Criminal-Crime Table

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



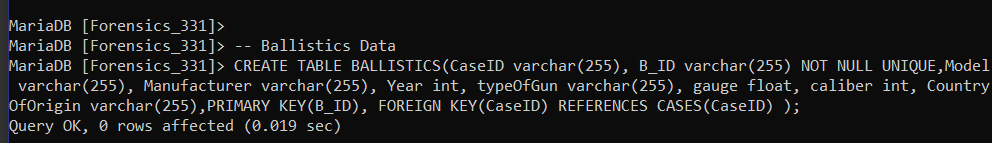
Creating Automobile Table

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));



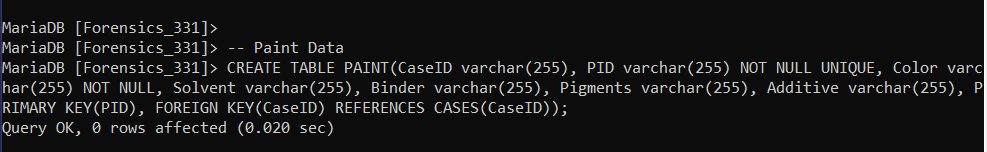
Creating Ballistics Table

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

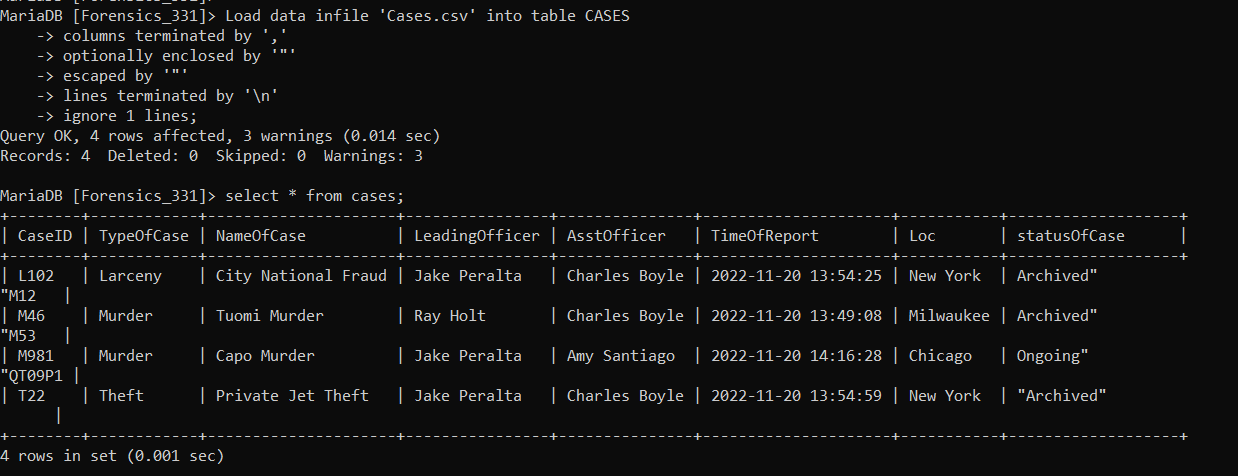
columns terminated by ','

optionally enclosed by '"'

escaped by '"'

lines terminated by '\n'

ignore 1 lines;



Inserting values into Criminal Table using load data command

Load data infile 'criminal.csv' into table CRIMINAL

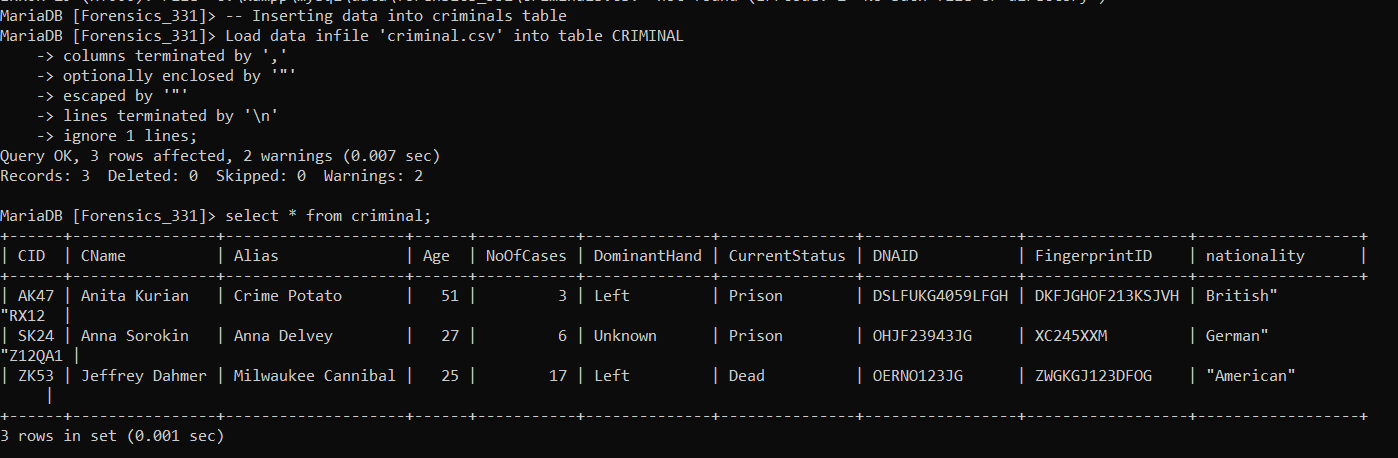
columns terminated by ','

optionally enclosed by '"'

escaped by '"'

lines terminated by '\n'

ignore 1 lines;



Inserting values into drugs table using insert Command

INSERT INTO `drugs` (`CaseID`, `NDC\_No`, `dname`, `color`, `class`, `narcotic`) VALUES

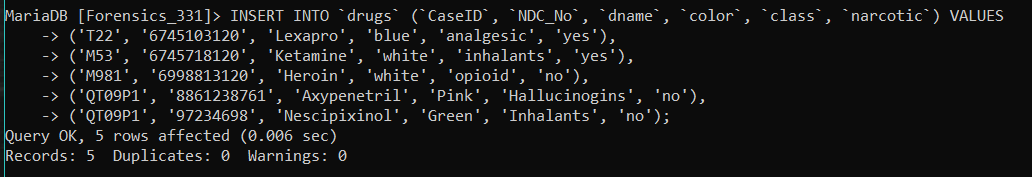
('T22', '6745103120', 'Lexapro', 'blue', 'analgesic', 'yes'),

('M53', '6745718120', 'Ketamine', 'white', 'inhalants', 'yes'),

('M981', '6998813120', 'Heroin', 'white', 'opioid', 'no'),

('QT09P1', '8861238761', 'Axypenetril', 'Pink', 'Hallucinogins', 'no'),

('QT09P1', '97234698', 'Nescipixinol', 'Green', 'Inhalants', 'no');



Inserting values into ballistics table using insert command

INSERT INTO `ballistics` (`CaseID`, `B\_ID`, `Model`, `Manufacturer`, `Year`, `typeOfGun`, `gauge`, `caliber`, `CountryOfOrigin`) VALUES

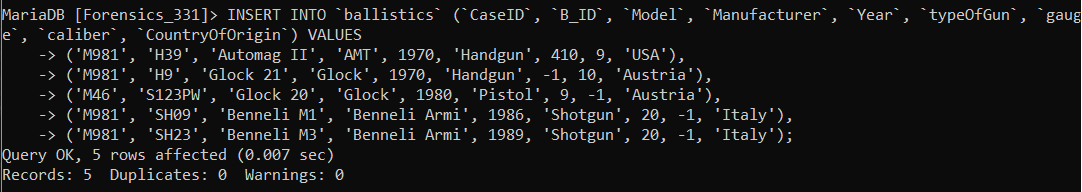
('M981', 'H39', 'Automag II', 'AMT', 1970, 'Handgun', 410, 9, 'USA'),

('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'),

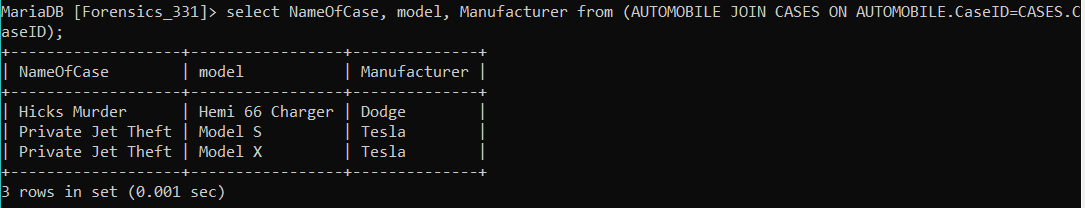
('M981', 'SH23', 'Benneli M3', 'Benneli Armi', 1989, 'Shotgun', 20, -1, 'Italy');



### Join Queries

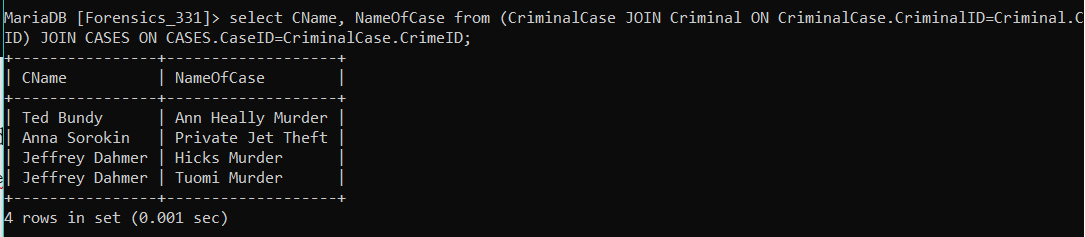
Cars associated with cases

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



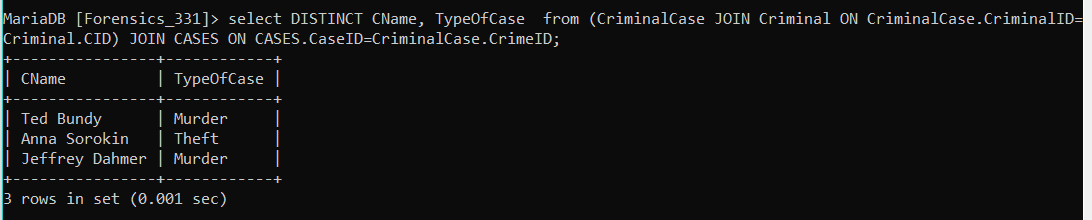
Criminals and the Cases they’ve been accused of

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



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;

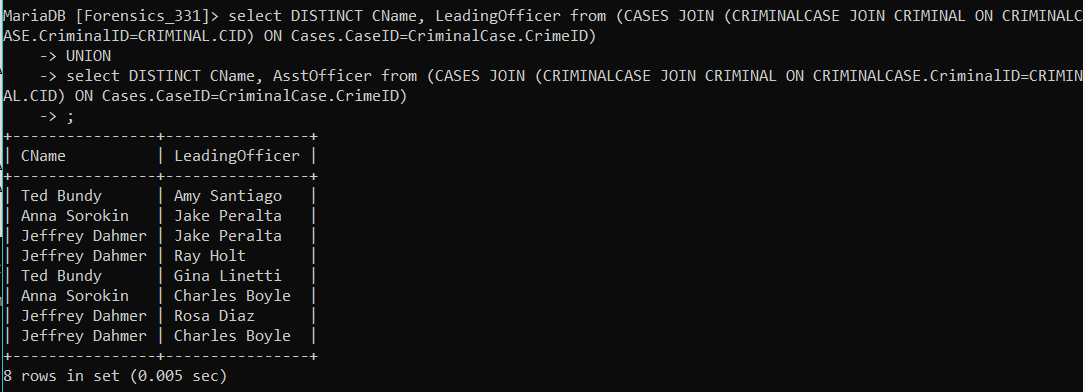


Criminals and the officers investigating them

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

UNION

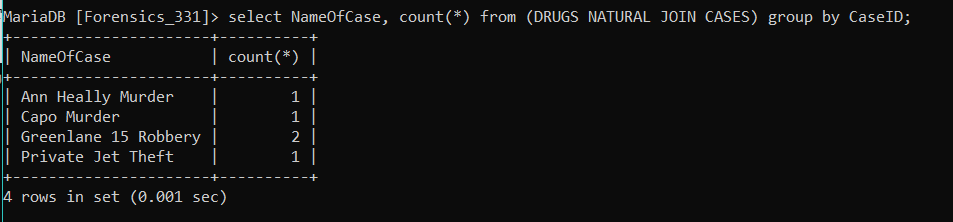
select DISTINCT CName, AsstOfficer from (CASES JOIN (CRIMINALCASE JOIN CRIMINAL ON CRIMINALCASE.CriminalID=CRIMINAL.CID) ON Cases.CaseID=CriminalCase.CrimeID)



### Aggregate Functions

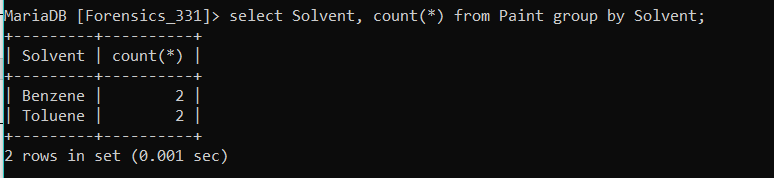
Cases with drug evidence and number of drug evidence instances for each

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



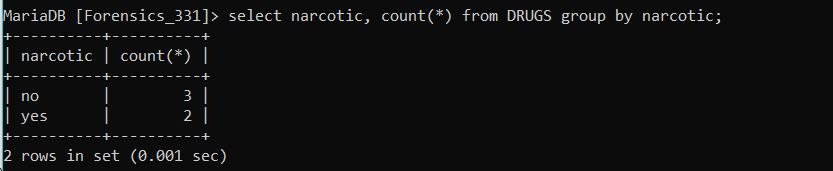
Paints grouped by solvent

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



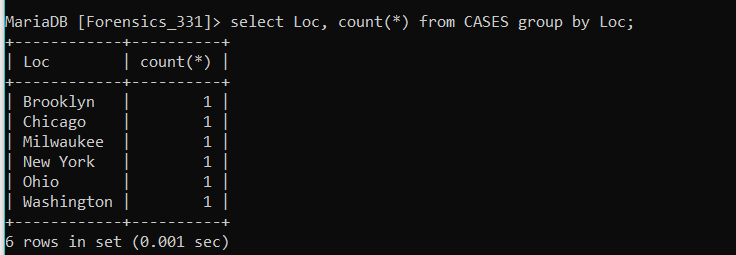
Number of Non-Narcotic and Narcotic drugs

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



Number of cases for each Location

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



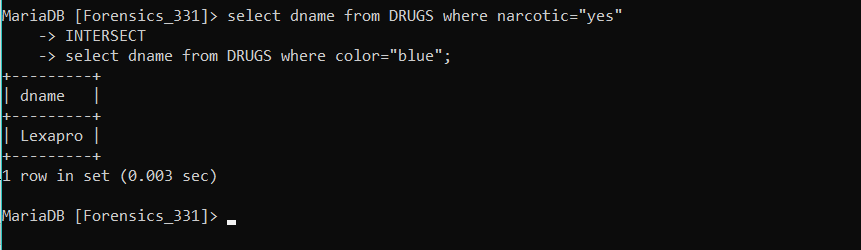
### Set Operations

Names of blue narcotic drugs

select dname from DRUGS where narcotic="yes"

INTERSECT

select dname from DRUGS where color="blue";

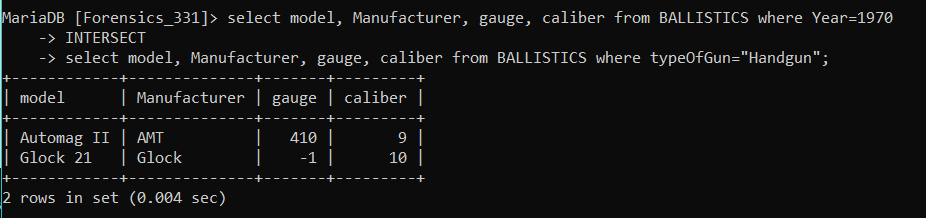


Handguns manufactured in 1970

select model, Manufacturer, gauge, caliber from BALLISTICS where Year=1970

INTERSECT

select model, Manufacturer, gauge, caliber from BALLISTICS where typeOfGun="Handgun";

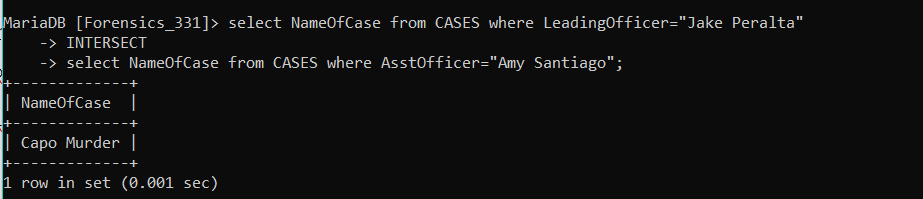


Cases lead by Jake Peralta and Assisted by Amy Santiago

select NameOfCase from CASES where LeadingOfficer="Jake Peralta"

INTERSECT

select NameOfCase from CASES where AsstOfficer="Amy Santiago";

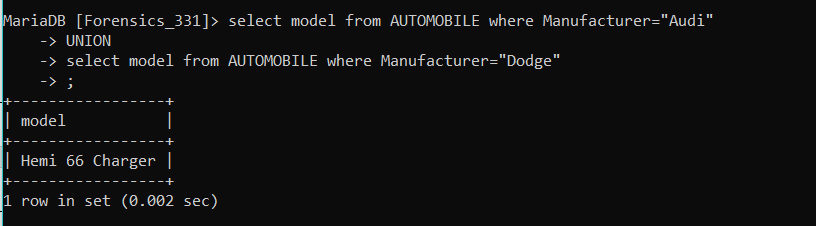


Cars manufactured by Audi or Dodge

select model from AUTOMOBILE where Manufacturer="Audi"

UNION

select model from AUTOMOBILE where Manufacturer="Dodge"



### 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

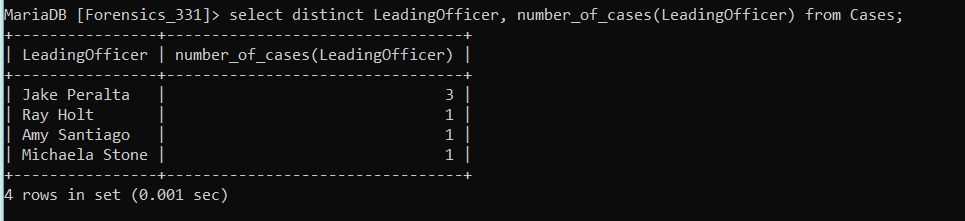
FROM CASES

WHERE LeadingOfficer = officer;

RETURN case\_count;

END; $$

DELIMITER ;



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

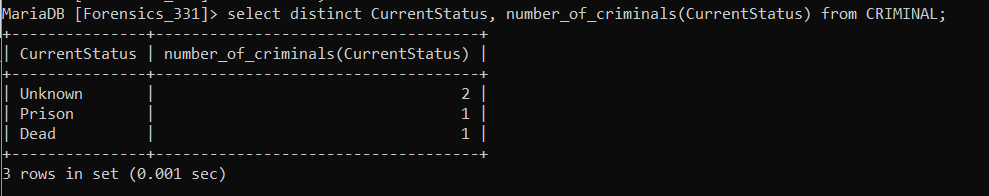
FROM Criminal

WHERE CurrentStatus = stat;

RETURN c;

END; $$

DELIMITER ;



### 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;

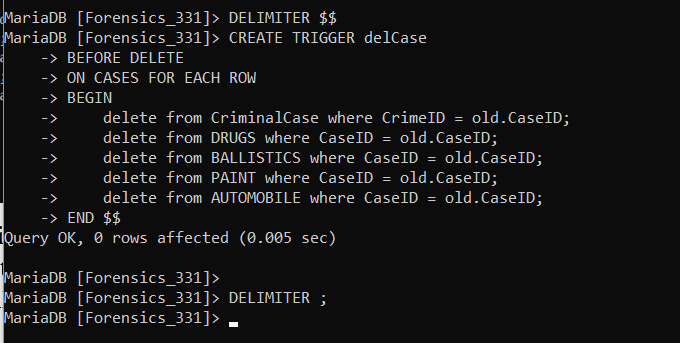
delete from BALLISTICS where CaseID = old.CaseID;

delete from PAINT where CaseID = old.CaseID;

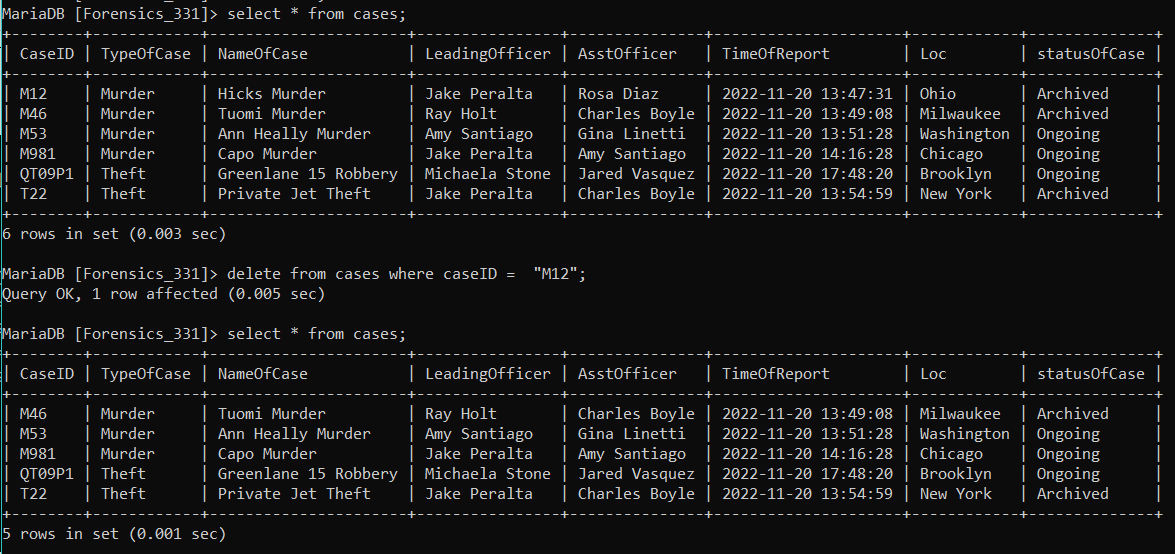
delete from AUTOMOBILE where CaseID = old.CaseID;

END $$

DELIMITER ;



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.



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;

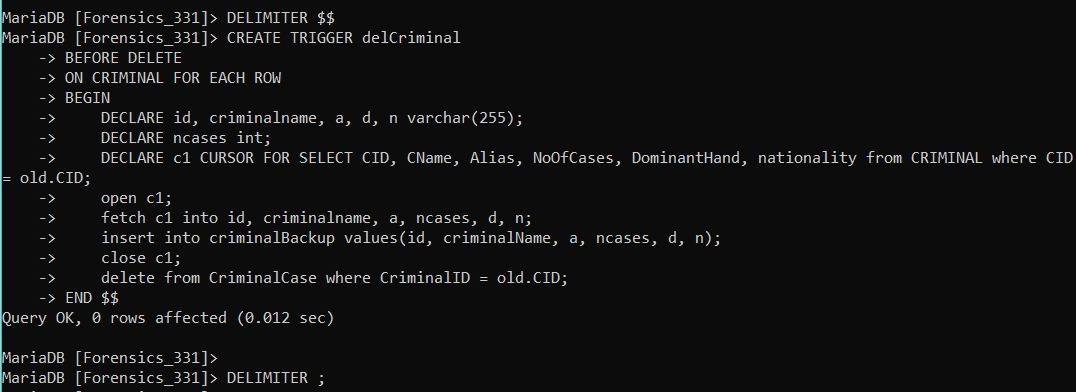
insert into criminalBackup values(cid, criminalName, a, ncases, d, n);

close c1;

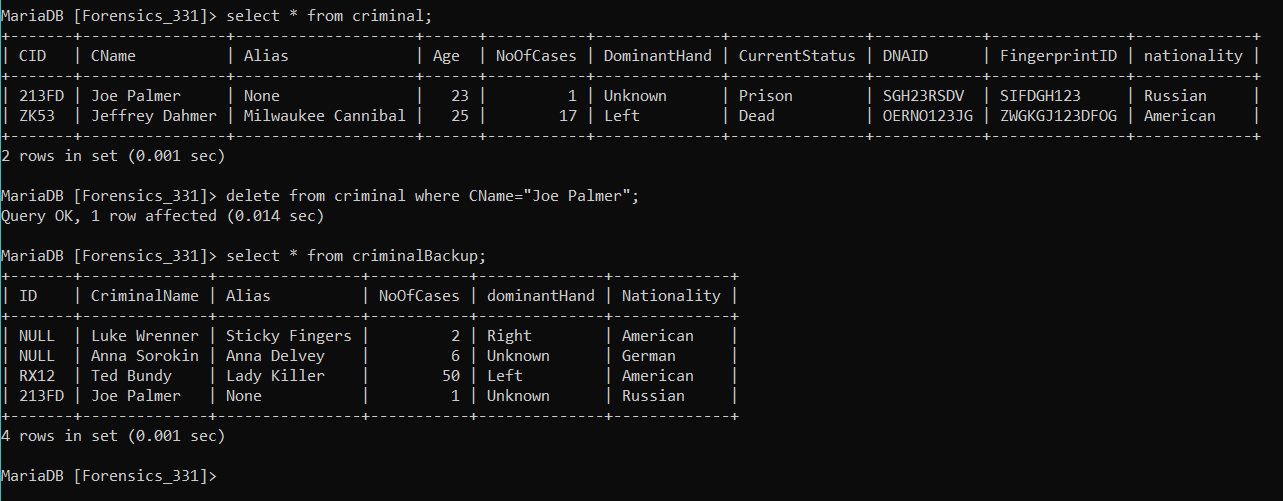
delete from CriminalCase where CriminalID = old.CID;

END $$

DELIMITER ;



Result for above trigger

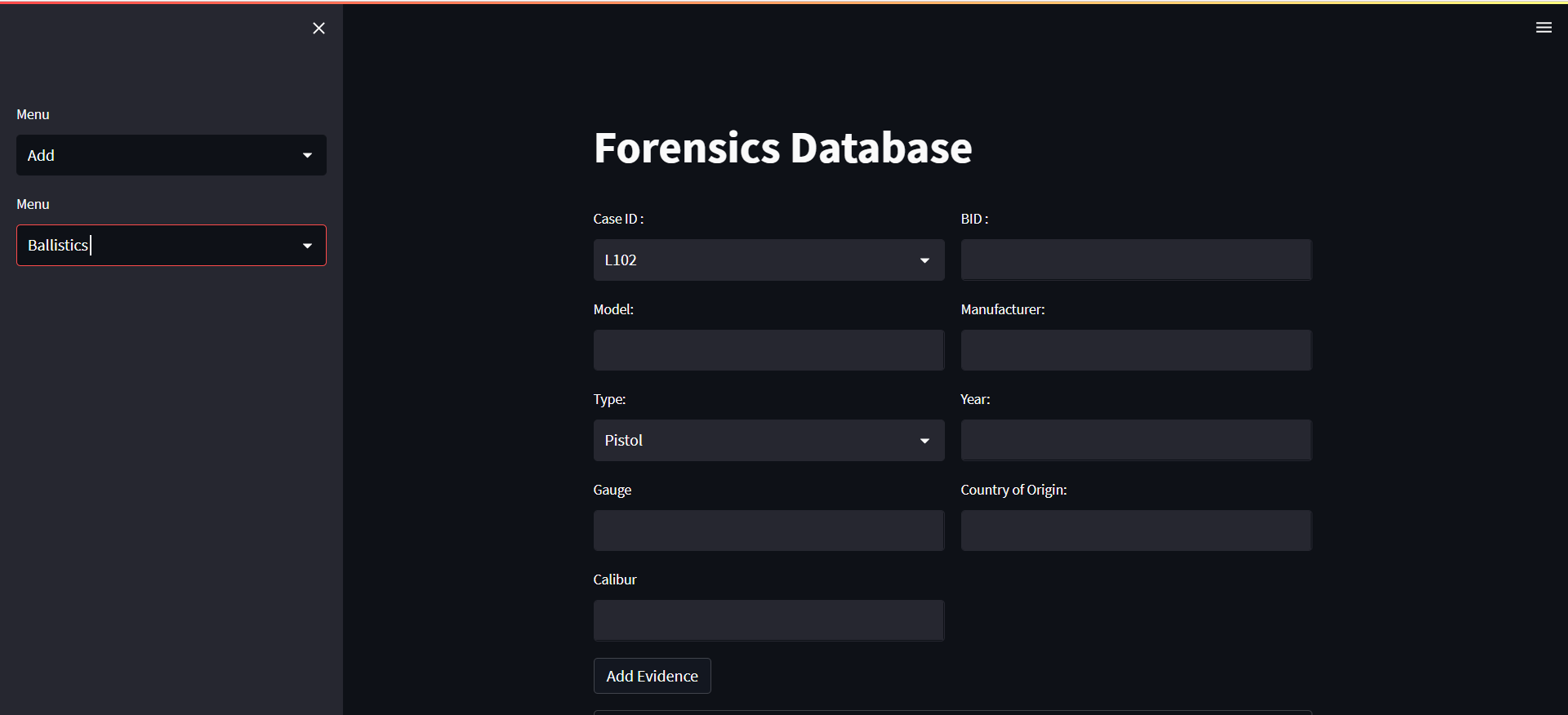


### Frontend –

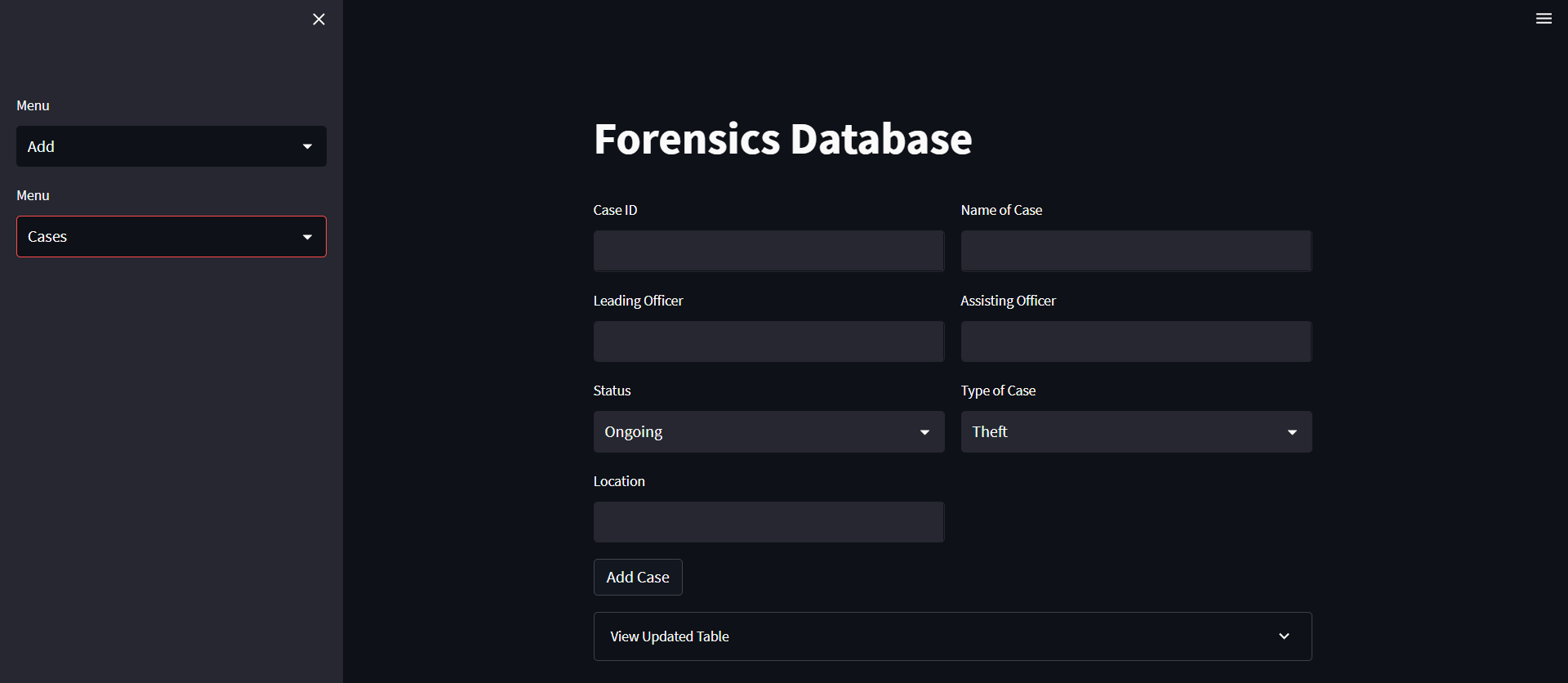
The application allows users to perform CRUD operations, run predefined queries and also provides a terminal to run custom queries. The user is presented with a sidebar and the main page on opening the website. The sidebar allows the user to choose between, add, view, edit, remove, predefined queries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

Create Operations

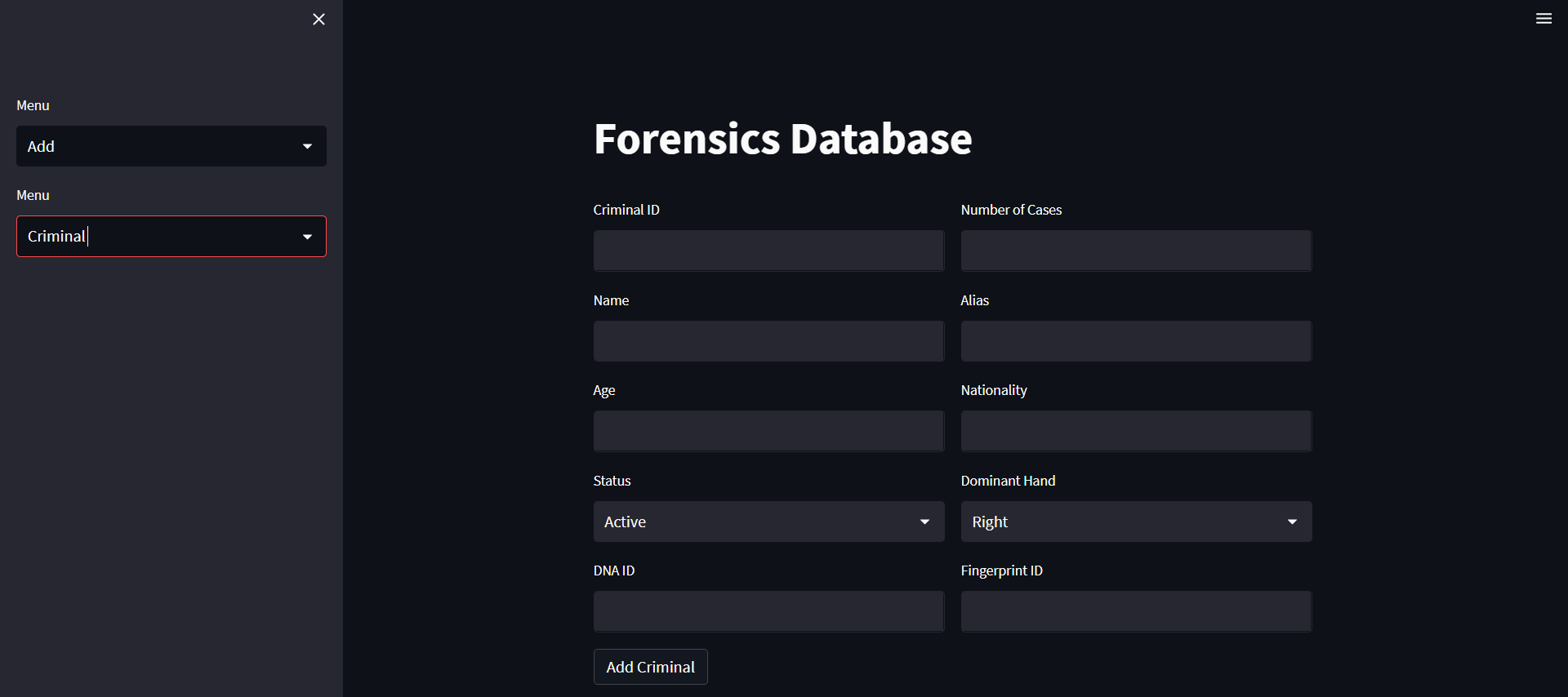
On selecting the add option, the user is presented with a second drop down list to choose a table for which new record is to be added. The user is then provided with the relevant input boxes to add a new record.



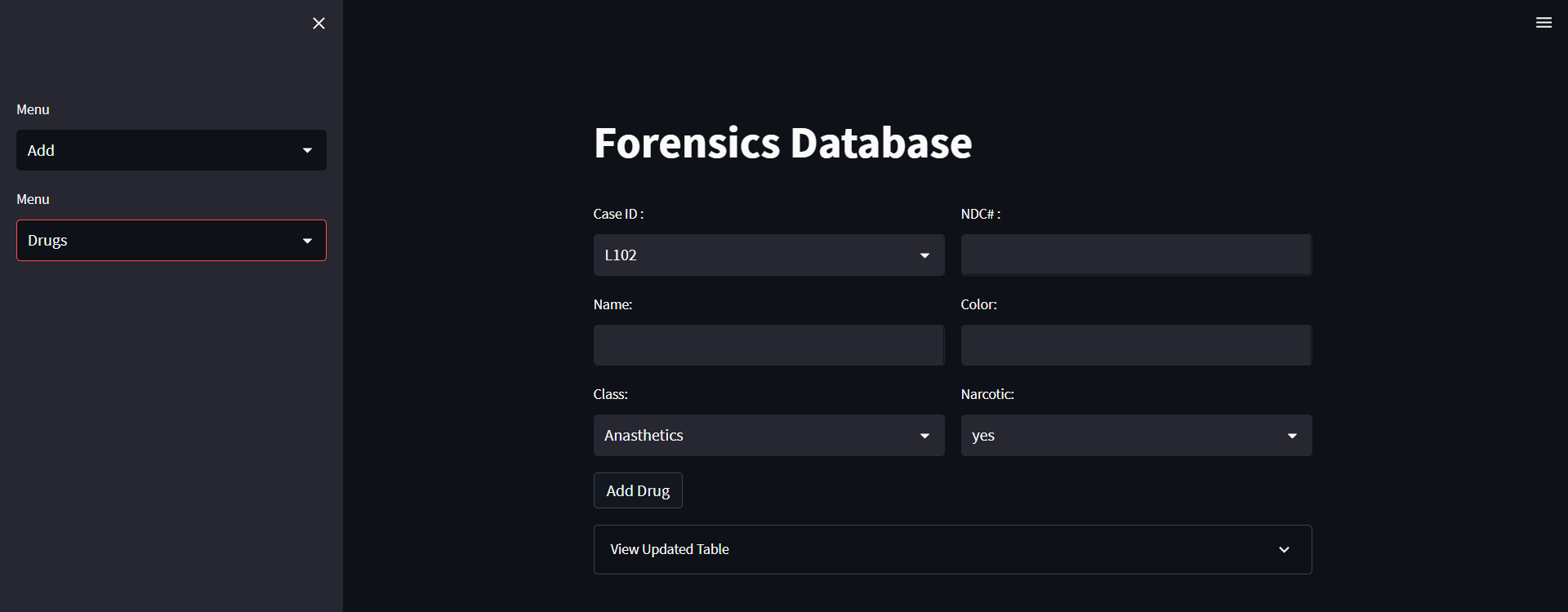
The above picture shows the UI to insert values in the Ballistics tables



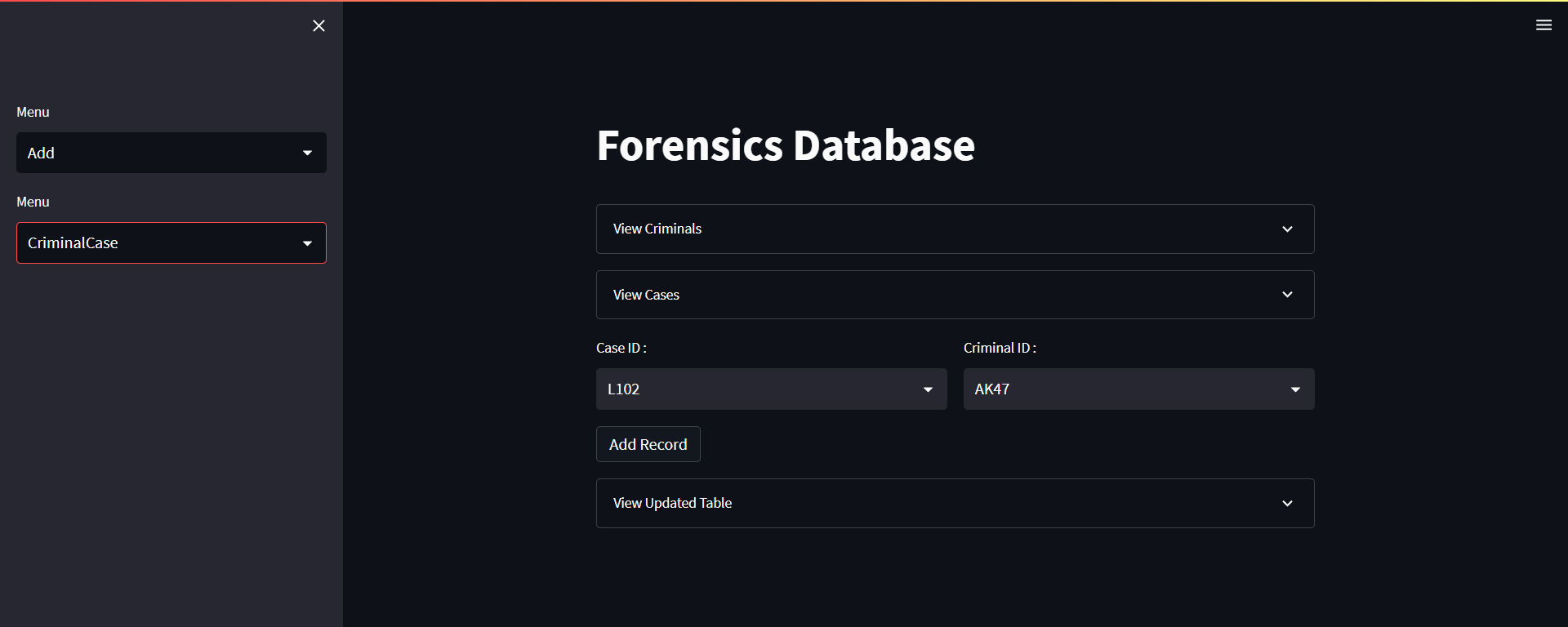
UI to insert a new case



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.



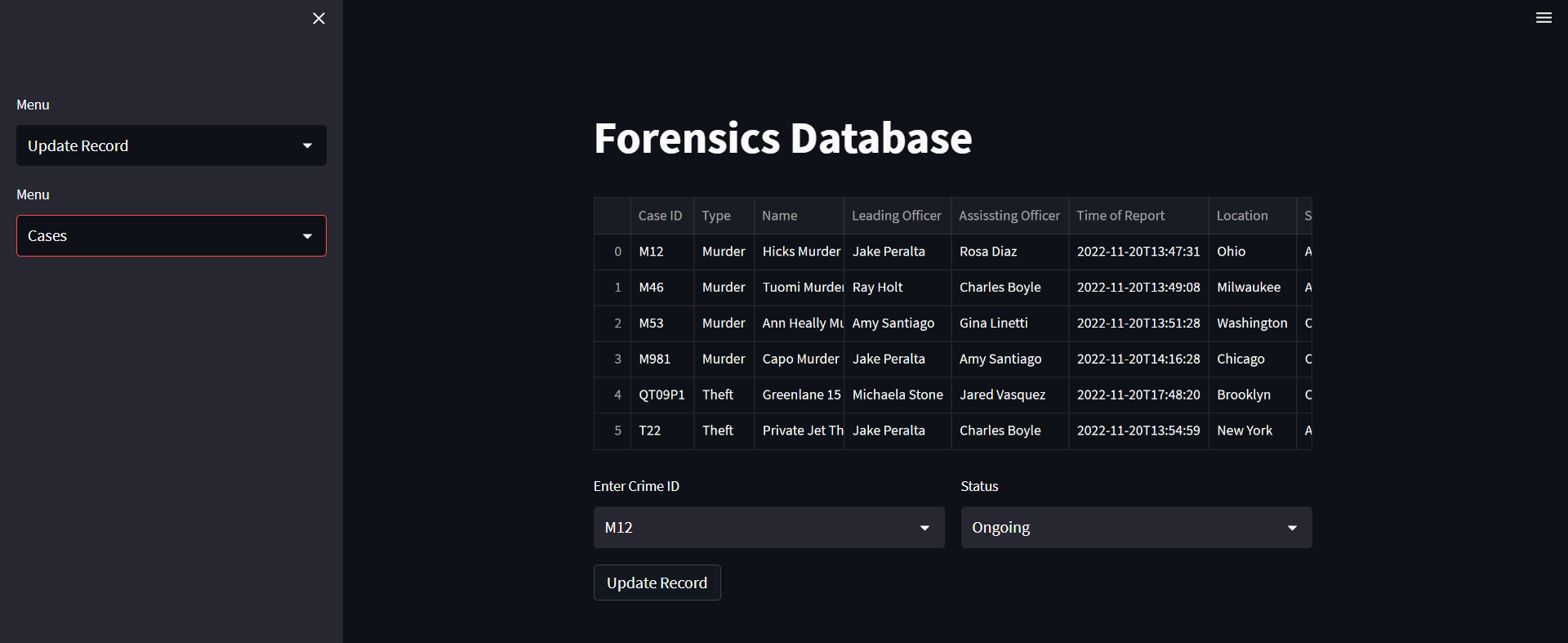
UI to view Drugs table



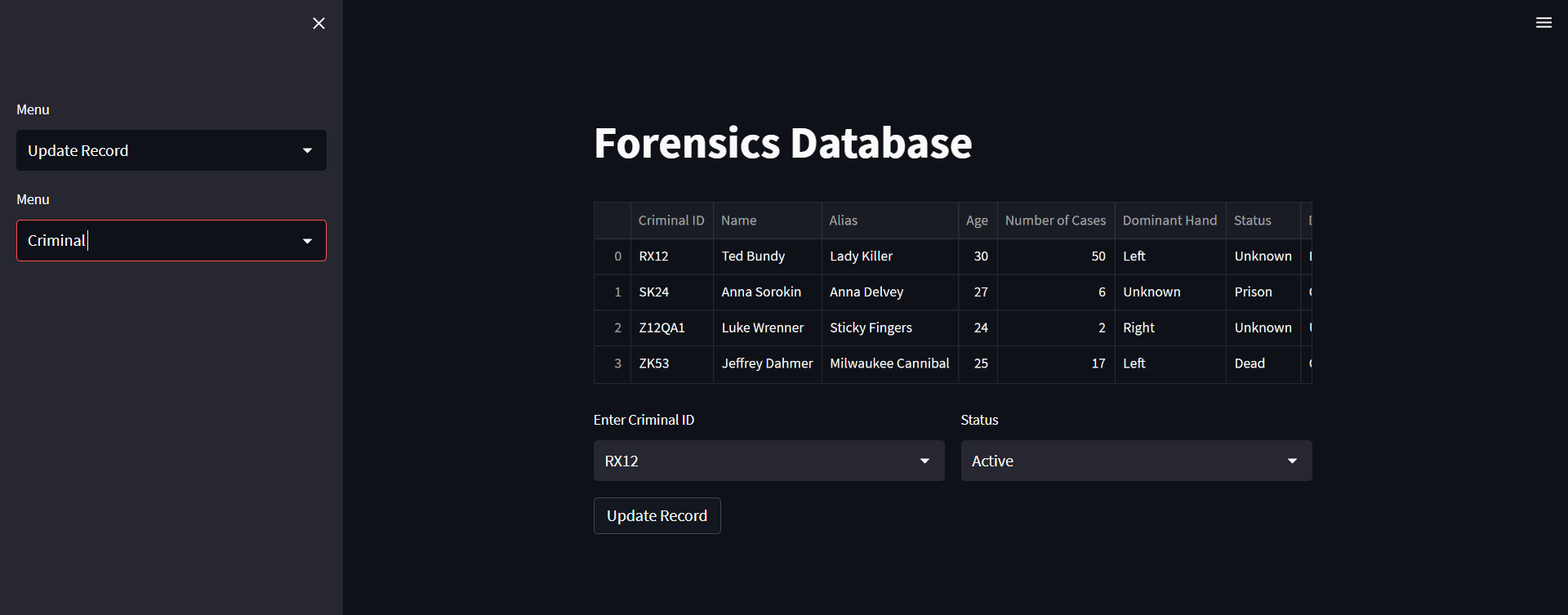
UI to view Criminal table

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.



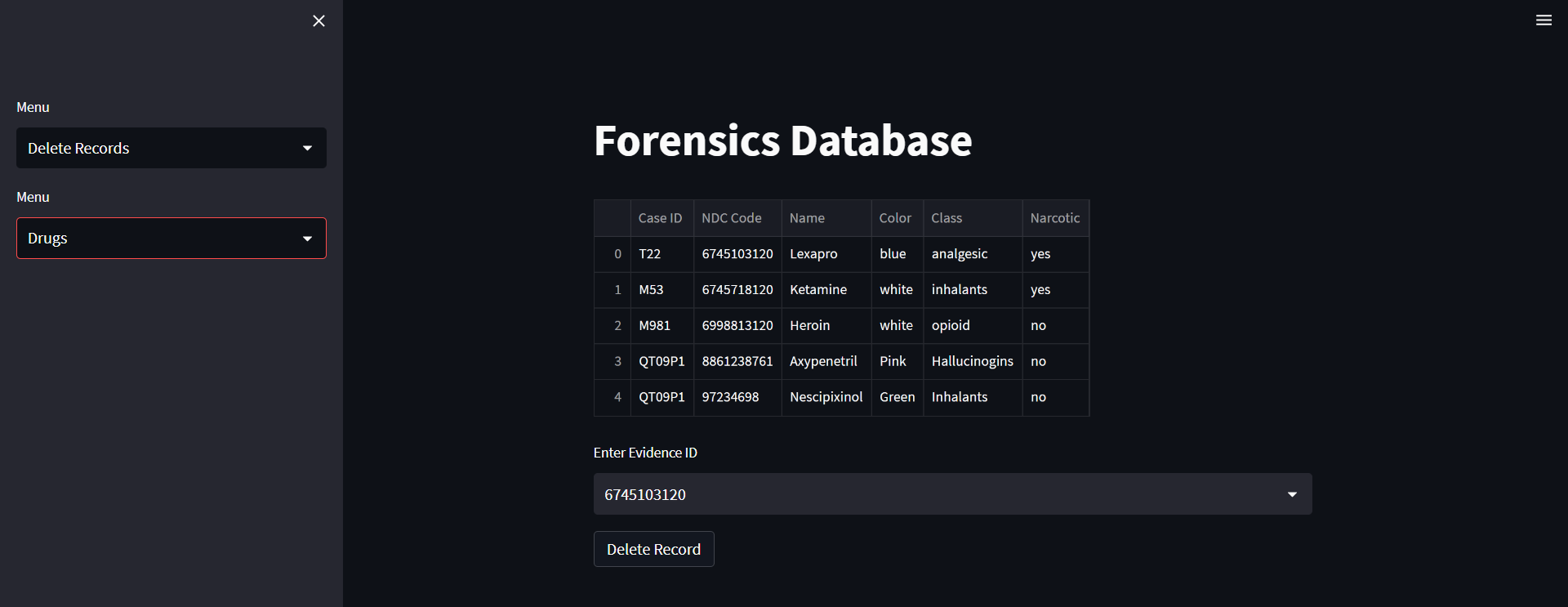
UI to Edit case table



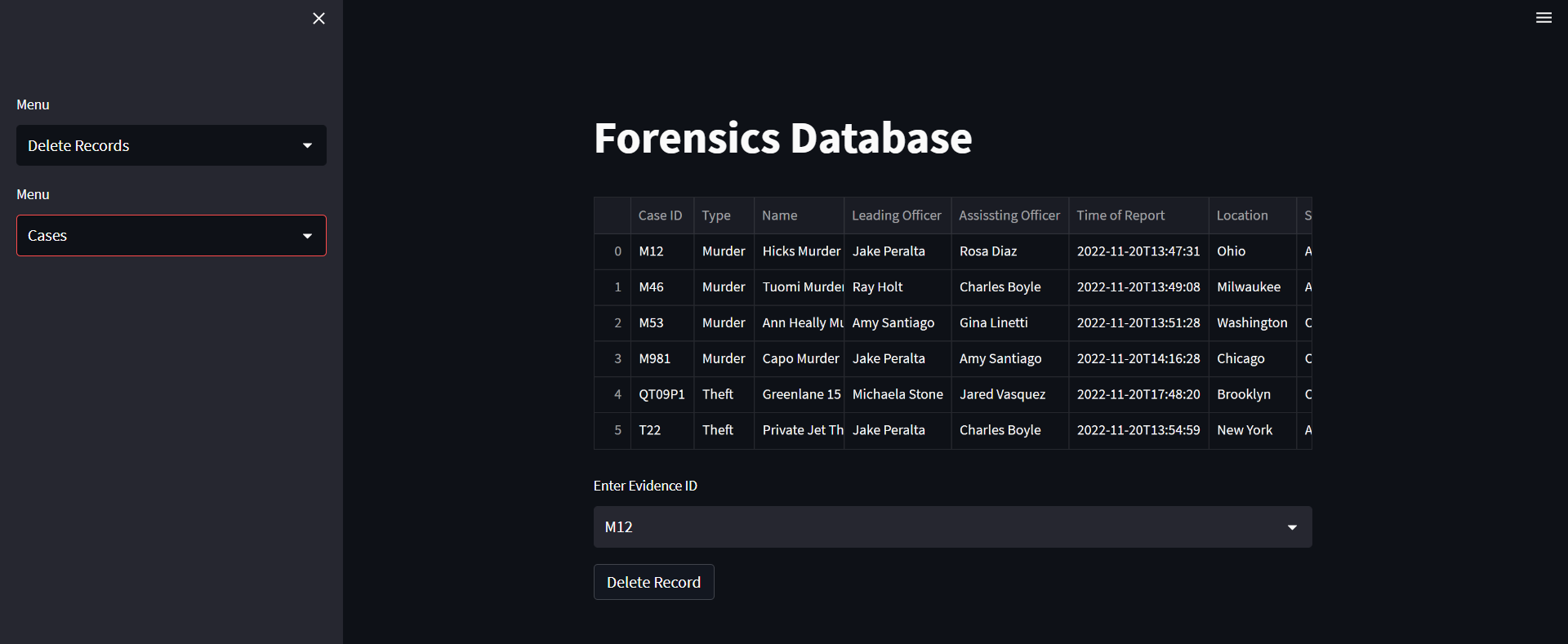
UI to edit criminal table

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.



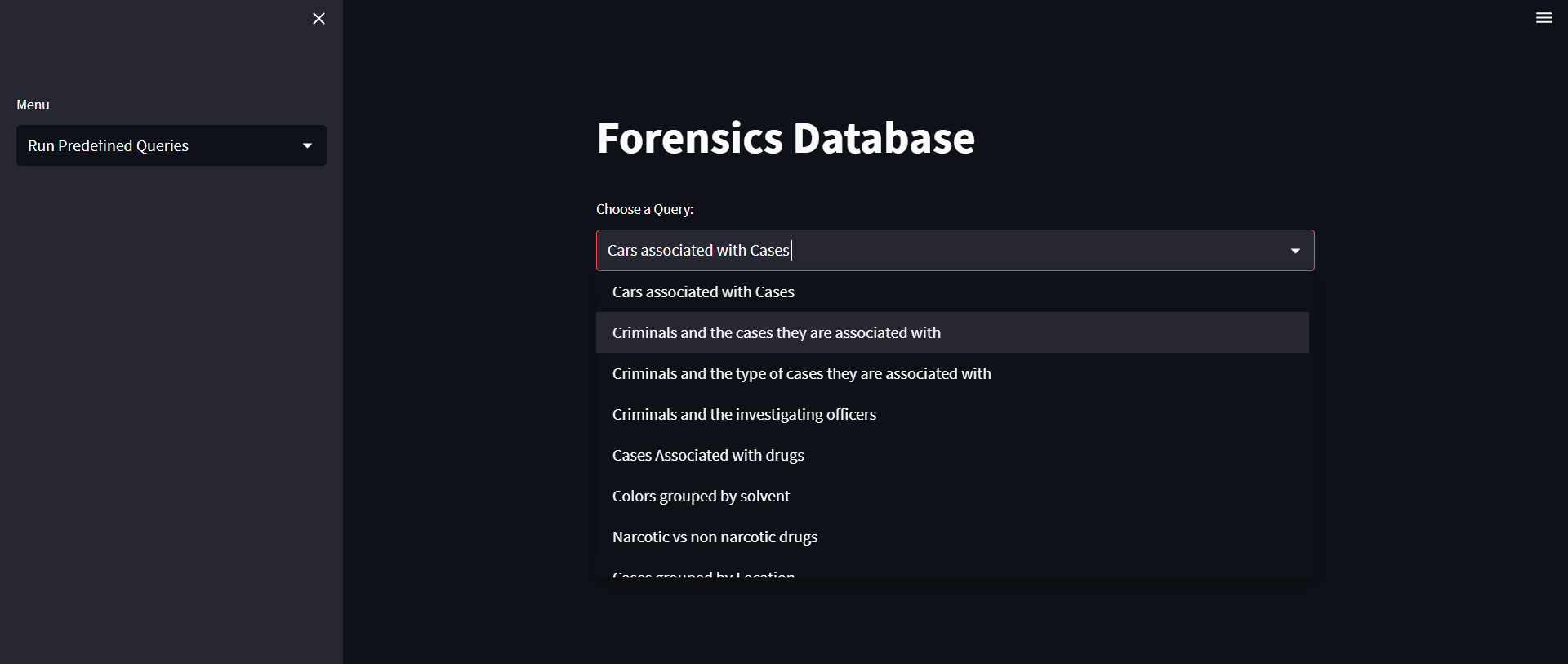
UI to delete drug evidence instance from the table



UI to delete case from the table

Run Predefined Queries

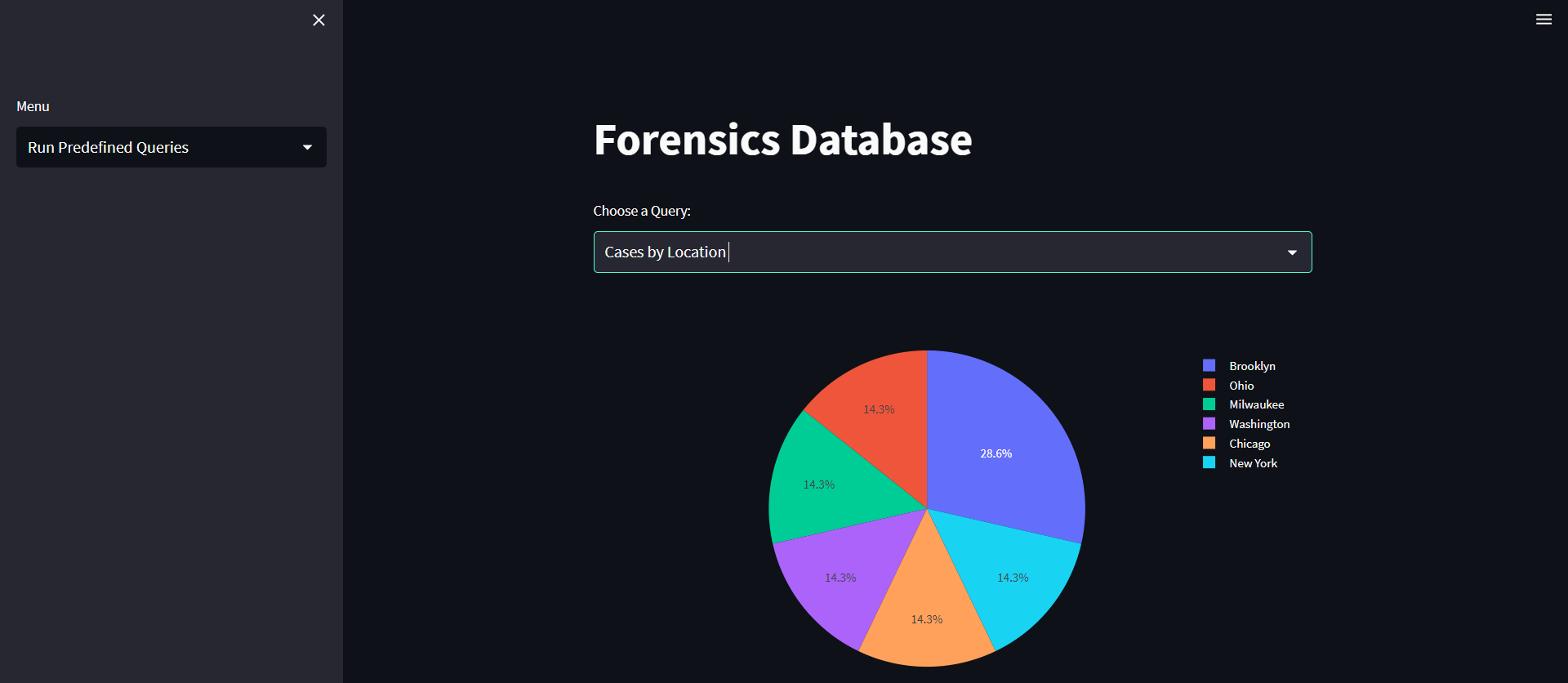
The user can also run queries from a list. These queries include criminals and the name of cases they’re accused of, criminal and the types of crimes they’ve committed, etc.



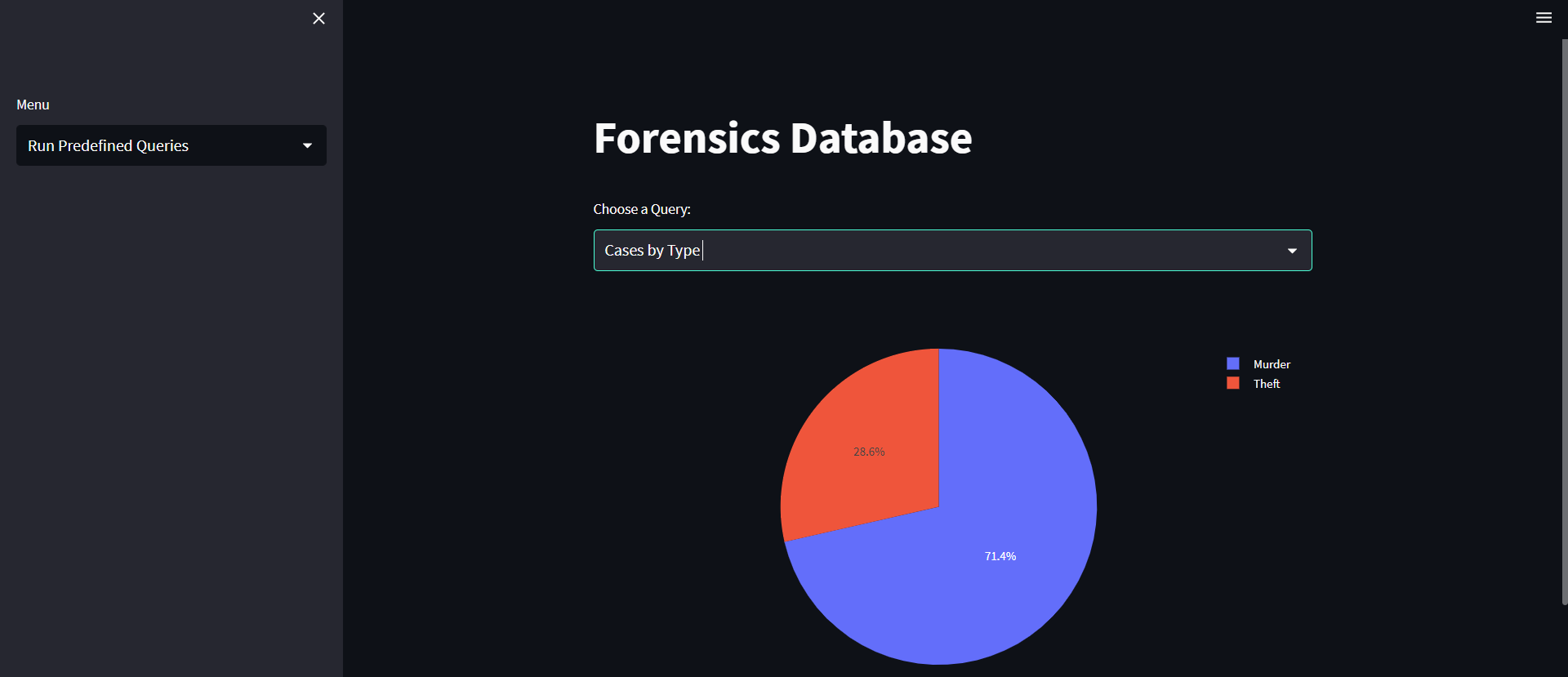
UI displaying the queries available



UI displaying the result of a selected query



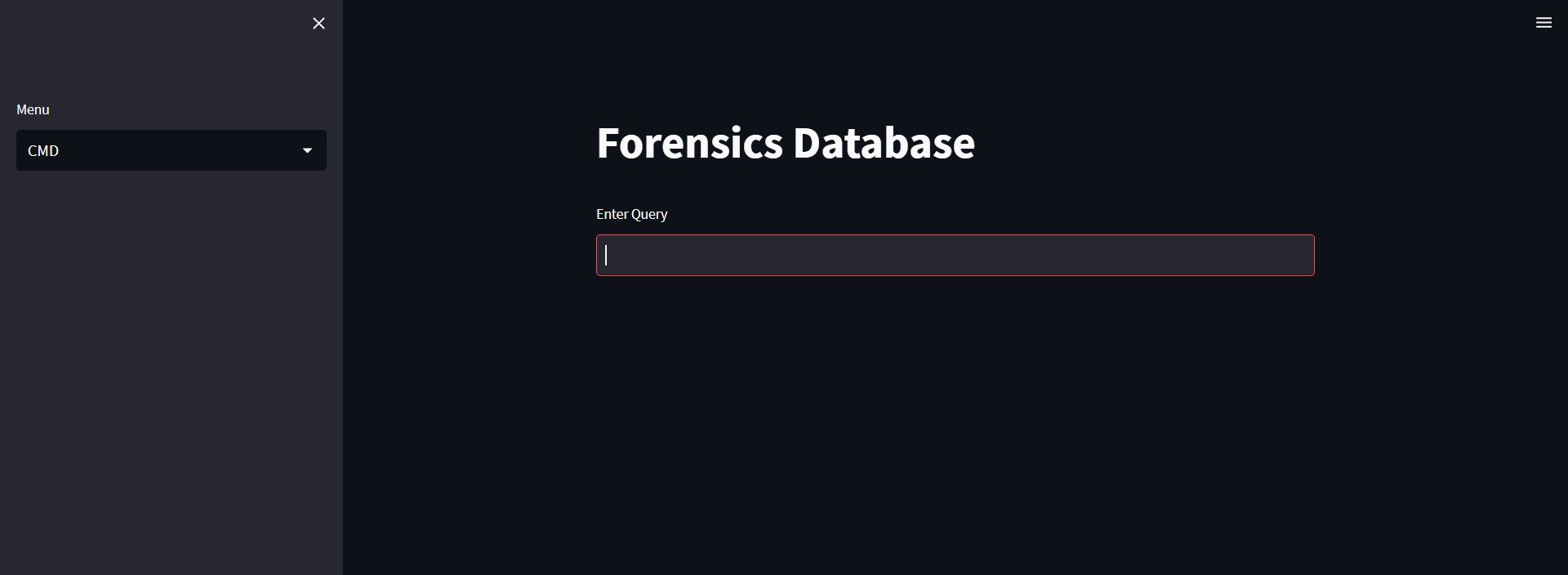
Cases grouped by location



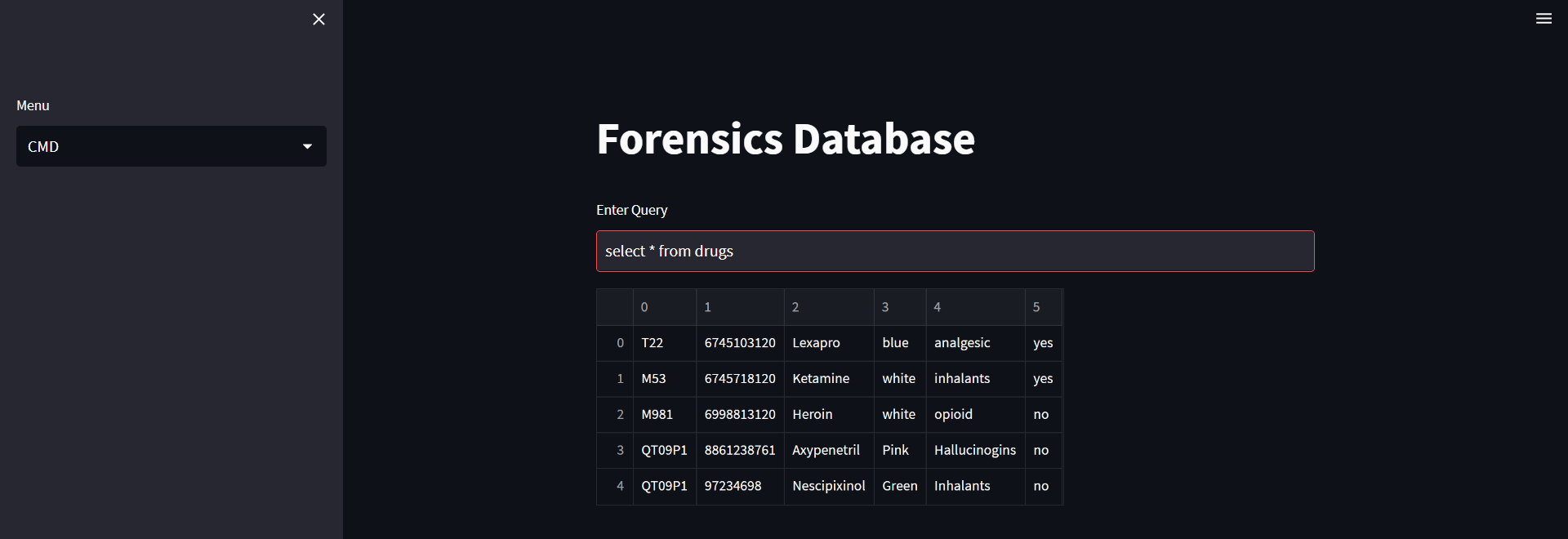
Cases grouped by type

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