



Mini project report on
Traffic Violation Detection and Penalty System

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering
UE23CS351A – DBMS Project

Submitted by:

Navya G N	PES2UG23CS372
Nikita Kolathaya	PES2UG23CS387

under the guidance of

Prof. Shilpa S
Assistant Professor
PES University

AUG - DEC 2025
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

Traffic Violation Detection and Penalty System

is a bonafide work carried out by

Navya G N

PES2UG23CS372

Nikita Kolathaya

PES2UG23CS387

In partial fulfilment for the completion of fifth semester DBMS Project (UE23CS351A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2025 – DEC. 2025. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature

Prof. Shilpa S Assistant Professor

DECLARATION

We hereby declare that the DBMS Project entitled Traffic Violation Detection and Penalty System has been carried out by us under the guidance of **Prof. Shilpa S, Assistant Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2025.

Navya G N

PES2UG23CS372

Nikita Kolathaya

PES2UG23CS387

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Shilpa S, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE23CS351A - DBMS Project.

I take this opportunity to thank Dr. Sandesh B J, C, Professor, ChairPerson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this DBMS Project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

The Traffic Violation Management System digitizes and automates traffic law enforcement and record management. Traditional paper-based systems are inefficient, prone to errors, and lack real-time access, causing inconsistent penalty enforcement.

This project implements a comprehensive database management system built on MySQL, designed to centralize management of traffic violations from initial recording to final payment. The system maintains detailed records of drivers, vehicles, traffic officers, violations, penalties, payments, and appeals, ensuring data integrity through a well-normalized relational schema adhering to Third Normal Form (3NF).

A user-friendly web-based interface developed using Python's Streamlit framework provides administrative personnel with access to complete CRUD (Create, Read, Update, Delete) operations, and data-entry users to add details, and others to appeal and pay fines.

It incorporates dependency checking mechanisms to prevent data integrity violations during deletion operations, provides dropdown selections to minimize data entry errors, and displays real-time feedback for all operations. The interface includes modules that execute advanced SQL queries to generate insights.

Key features of the system include automated penalty calculation through triggers, systematic appeal processing with automatic status updates, identification of repeat offenders through nested queries, revenue analytics through aggregate functions, and graceful handling of cascading deletions while maintaining referential integrity.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
1.	INTRODUCTION	6
2.	PROBLEM DEFINITION	7
3.	ER MODEL	8
4.	ER TO RELATIONAL MAPPING	10
5.	DDL STATEMENTS	12
6.	DML STATEMENTS	17
7.	QUERIES (SIMPLE QUERY AND UPDATE AND DELETE OPERATION, CORRELATED QUERY AND NESTED QUERY)	22
8.	STORED PROCEDURE, FUNCTIONS AND TRIGGERS	27
9.	FRONT END DEVELOPMENT	35
	CONCLUSION	45
	REFERENCES	46

1. INTRODUCTION

The Traffic Violation Management System is a comprehensive database application designed to digitize and streamline the management of traffic violations, penalties, and related administrative tasks. Traditional paper-based systems for tracking traffic violations are inefficient, prone to errors, and lack real-time accessibility. This project addresses these challenges by implementing a centralized, digital database solution.

The system provides a structured repository for tracking driver information, vehicle registrations, traffic violations, penalty payments, and appeals. Built on MySQL database backend and interfaced through a Python-based Streamlit web application, the system demonstrates complete CRUD (Create, Read, Update, Delete) functionality, advanced query capabilities, and automated business logic through triggers and stored procedures.

Key Objectives:

- Maintain accurate records of drivers, vehicles, and officers
- Track violations with associated penalties automatically
- Process penalty payments and handle appeals systematically
- Generate analytical reports for traffic management insights
- Provide a user-friendly interface for administrative operations

Scope:

The system manages the complete lifecycle of traffic violations from recording the incident to payment collection or appeal resolution, ensuring data integrity through foreign key constraints and automated triggers.

Software Requirements

- Database: MySQL Server 8.0 or higher
- Programming Language: Python 3.8+
- Frontend Framework: Streamlit
- Database Connector: mysql-connector-python
- Additional Python Libraries: pandas, datetime

2. PROBLEM DEFINITION

The Traffic Violation Management System addresses the following critical challenges in traffic enforcement and administration:

Core Problems

1. Manual Record Keeping

- Paper-based violation records are difficult to maintain and retrieve
- High risk of data loss or damage
- Time-consuming search and retrieval process

2. Penalty Calculation Errors

- Manual calculation of fines prone to human error
- Inconsistent penalty amounts for similar violations
- Difficulty in tracking payment due dates and overdue penalties

3. Driver History Tracking

- No centralized system to track repeat offenders
- Unable to identify patterns in driver violations
- Difficulty in implementing progressive penalties

4. Payment and Appeal Management

- Inefficient tracking of paid vs unpaid penalties
- Complex appeal process with no systematic workflow
- No automated status updates based on appeal decisions

5. Reporting and Analytics

- Lack of real-time insights into violation trends
- Inability to generate revenue reports efficiently
- No data-driven decision making for traffic management

Proposed Solution

The Traffic Violation Management System provides:

- Centralized Database: Single source of truth for all traffic violation data
- Automated Workflows: Triggers automatically calculate penalties, update statuses, and handle appeals
- Data Integrity: Foreign key constraints ensure referential integrity across all tables
- Real-time Access: Web-based interface accessible from any location
- Advanced Analytics: SQL queries provide insights into violation patterns and revenue trends

3. ER MODEL

ENTITIES:

DRIVER

- Stores driver personal information
- Attributes: Driver_ID (PK), Name, Address, Contact_no, License_no

VEHICLE

- Stores vehicle registration details
- Attributes: Vehicle_ID (PK), Registration_year, Model, Color, License_plate, Driver_ID (FK)

OFFICER

- Stores traffic officer information
- Attributes: Officer_ID (PK), Name, Officer_Rank, Badge_no, Contact_no

VIOLATION

- Records traffic violation incidents
- Attributes: Violation_ID (PK), Date_Time, Type, Location, Vehicle_ID (FK), Officer_ID (FK), ViolationType_ID (FK)

VIOLATION_TYPE

- Defines violation categories and default penalties
- Attributes: ViolationType_ID (PK), Type_Name, Default_Amount, Default_Demerit_Points, Default_Duedays, Description

PENALTY

- Stores penalty information for violations
- Attributes: Penalty_ID (PK), Amount, Duedate, Status, Violation_ID (FK)

PAYMENT

- Records payment transactions
- Attributes: Payment_ID (PK), Date, Amount, ModeofPayment, Penalty_ID (FK)

APPEAL

- Manages violation appeals
- Attributes: Appeal_ID (PK), Datefiled, Status, Reason, Violation_ID (FK), Driver_ID (FK)

RELATIONS:

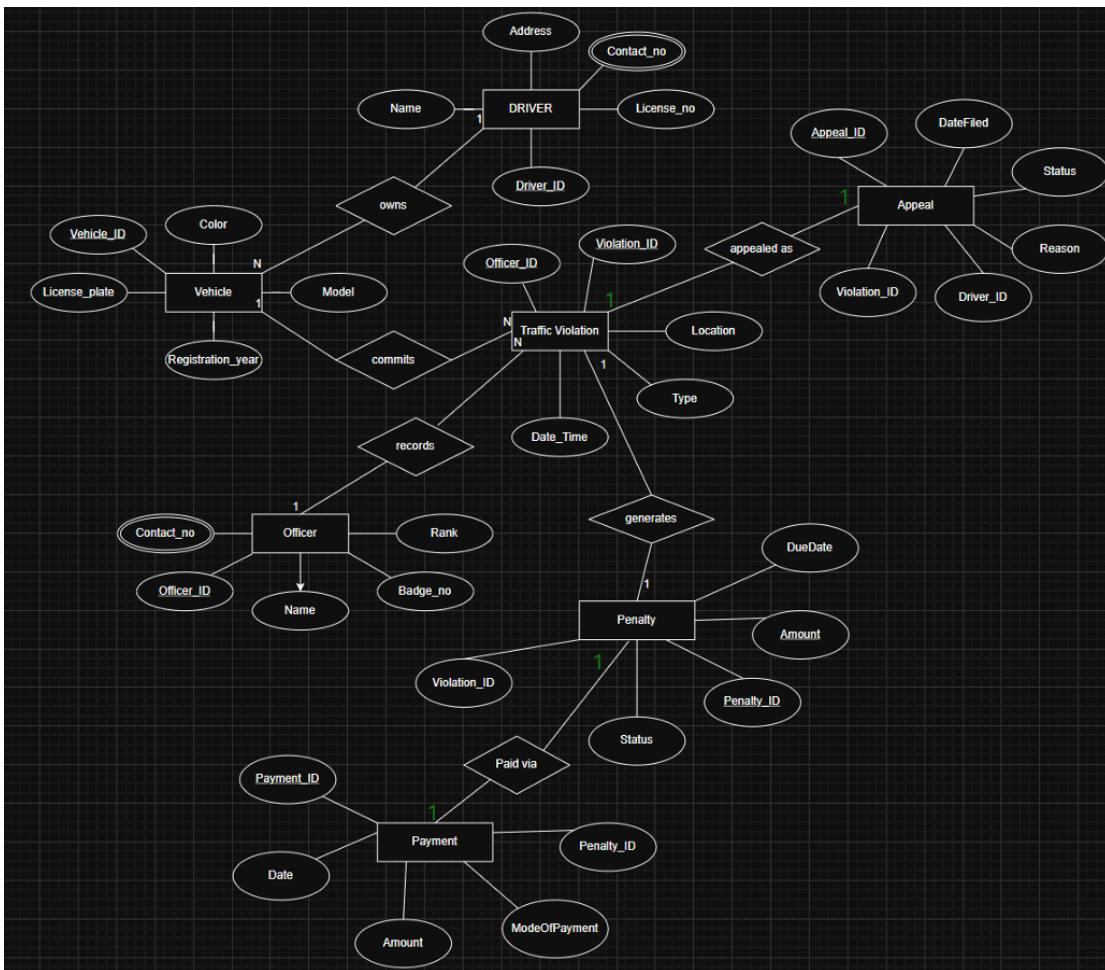
1:M (One-to-Many)

- One DRIVER can own multiple VEHICLES
- One VEHICLE can have multiple VIOLATIONS
- One OFFICER can record multiple VIOLATIONS
- One VIOLATION can have one PENALTY (1:1 relationship)
- One PENALTY can have one PAYMENT (1:1 relationship)

M:N (Many-to-Many)

None directly, but VIOLATION acts as a bridge between VEHICLE and OFFICER

ER DIAGRAM:



4. ER TO RELATIONAL MAPPING

4.1 STEPS OF ALGORITHM FOR CHOSEN PROBLEM

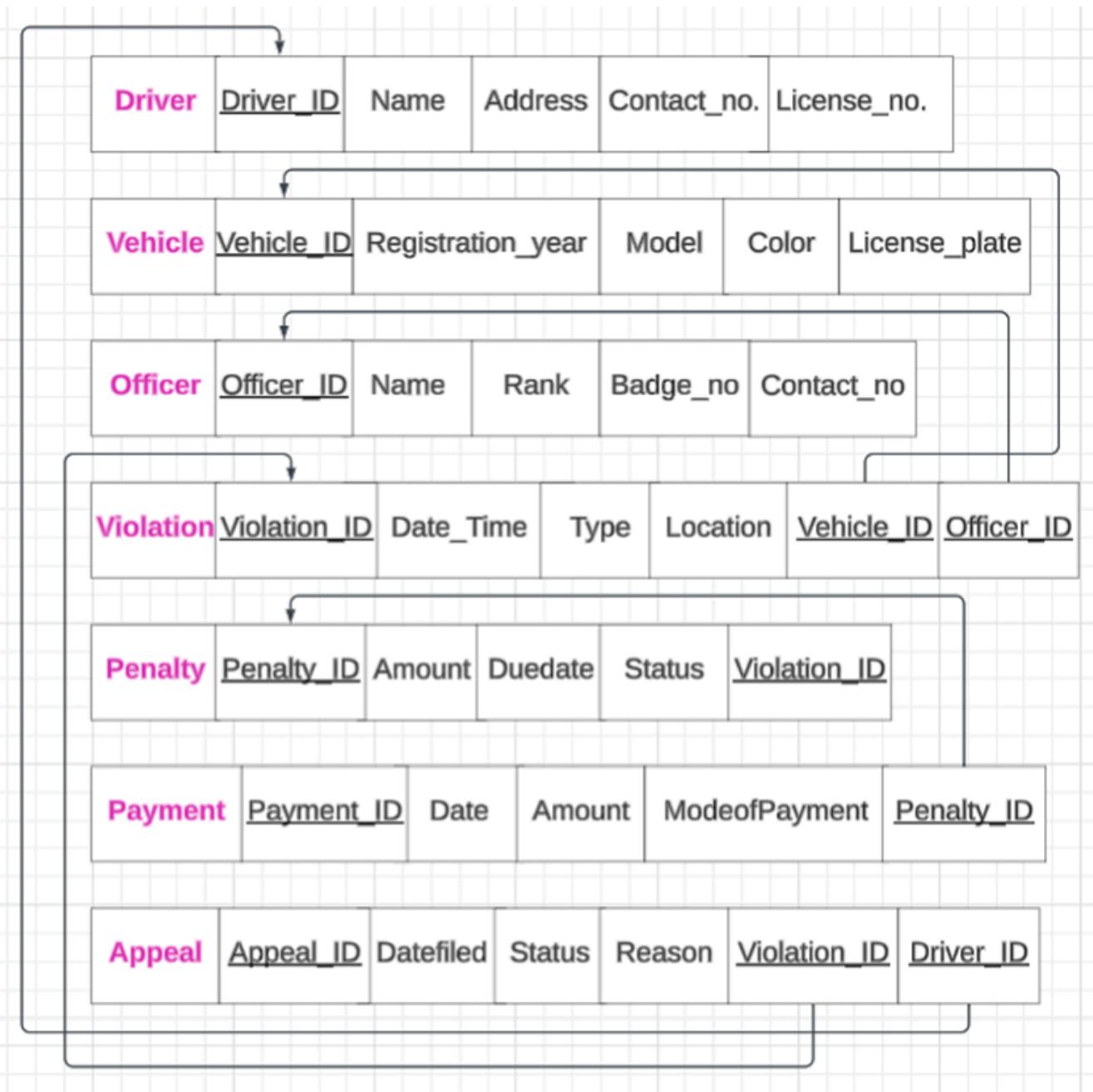
Step 1: Mapping Strong Entities

- Create tables for Driver, Officer, Violation_Type
- Each entity becomes a table with its attributes as columns
- Primary keys are defined

Step 2: Mapping 1:M Relationships

- Vehicle table includes Driver_ID as foreign key
- Violation table includes Vehicle_ID and Officer_ID as foreign keys
- Penalty table includes Violation_ID as foreign key
- Payment table includes Penalty_ID as foreign key
- Appeal table includes Violation_ID and Driver_ID as foreign keys

4.2 COMPLETE DIAGRAM OF RELATIONAL MAPPING



5. DDL STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE CREATION

-- Create and use the database

```
CREATE DATABASE IF NOT EXISTS TrafficVioDB;
```

```
USE TrafficVioDB;
```

-- 1. Driver Table

```
CREATE TABLE Driver (
    Driver_ID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100) NOT NULL,
    Address VARCHAR(255),
    Contact_no VARCHAR(15),
    License_no VARCHAR(50) UNIQUE NOT NULL
);
```

-- 2. Vehicle Table

```
CREATE TABLE Vehicle (
    Vehicle_ID INT PRIMARY KEY AUTO_INCREMENT,
    Registration_year INT,
    Model VARCHAR(50),
    Color VARCHAR(30),
    License_plate VARCHAR(20) UNIQUE NOT NULL,
    Driver_ID INT,
    FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID)
);
```

-- 3. Officer Table

```
CREATE TABLE Officer (
    Officer_ID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100) NOT NULL,
    Officer_Rank VARCHAR(50),
    Badge_no VARCHAR(30) UNIQUE NOT NULL,
```

Contact_no VARCHAR(15)

);

-- 4. Violation Table

CREATE TABLE Violation (

Violation_ID INT PRIMARY KEY AUTO_INCREMENT,
Date_Time TIMESTAMP NOT NULL,
Type VARCHAR(50),
Location VARCHAR(100),
Vehicle_ID INT NOT NULL,
Officer_ID INT,
FOREIGN KEY (Vehicle_ID) REFERENCES Vehicle(Vehicle_ID),
FOREIGN KEY (Officer_ID) REFERENCES Officer(Officer_ID)

);

-- 5. Penalty Table

CREATE TABLE Penalty (

Penalty_ID INT PRIMARY KEY AUTO_INCREMENT,
Amount DECIMAL(10,2) NOT NULL,
Duedate DATE,
Status VARCHAR(20) CHECK (Status IN ('Unpaid','Paid','Appealed')),
Violation_ID INT UNIQUE,
FOREIGN KEY (Violation_ID) REFERENCES Violation(Violation_ID)

);

-- 6. Payment Table

CREATE TABLE Payment (

Payment_ID INT PRIMARY KEY AUTO_INCREMENT,
Date DATE NOT NULL,
Amount DECIMAL(10,2) NOT NULL,
ModeofPayment VARCHAR(20) CHECK (ModeofPayment IN ('Cash','Card','Online','UPI')),
Penalty_ID INT UNIQUE,
FOREIGN KEY (Penalty_ID) REFERENCES Penalty(Penalty_ID)

);

-- 7. Appeal Table

```
CREATE TABLE Appeal (
    Appeal_ID INT PRIMARY KEY AUTO_INCREMENT,
    Datefiled DATE NOT NULL,
    Status VARCHAR(20) CHECK (Status IN ('Pending','Accepted','Rejected')),
    Reason VARCHAR(255),
    Violation_ID INT UNIQUE,
    Driver_ID INT,
    FOREIGN KEY (Violation_ID) REFERENCES Violation(Violation_ID),
    FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID)
);
```

-- 8. Violation Type Table

```
CREATE TABLE IF NOT EXISTS Violation_Type (
    ViolationType_ID INT PRIMARY KEY AUTO_INCREMENT,
    Type_Name VARCHAR(50) UNIQUE NOT NULL,
    Default_Amount DECIMAL(10,2) NOT NULL,
    Default_Demerit_Points INT DEFAULT 0,
    Default_Duedays INT DEFAULT 30,
    Description VARCHAR(255)
);
```

-- 9. Audit Log Table

```
CREATE TABLE IF NOT EXISTS Audit_Log (
    Audit_ID INT PRIMARY KEY AUTO_INCREMENT,
    Action VARCHAR(20),
    Table_Name VARCHAR(50),
    Record_ID INT,
    Action_By VARCHAR(100),
    Action_Time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Details TEXT
);
```

```

mysql> CREATE TABLE Driver (
->     Driver_ID INT PRIMARY KEY AUTO_INCREMENT,
->     Name VARCHAR(100) NOT NULL,
->     Address VARCHAR(255),
->     Contact_no VARCHAR(15),
->     License_no VARCHAR(50) UNIQUE NOT NULL
-> );
Query OK, 0 rows affected (0.10 sec)

mysql> CREATE TABLE Vehicle (
->     Vehicle_ID INT PRIMARY KEY AUTO_INCREMENT,
->     Registration_year INT,
->     Model VARCHAR(50),
->     Color VARCHAR(30),
->     License_plate VARCHAR(20) UNIQUE NOT NULL,
->     Driver_ID INT,
->     FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID)
-> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Officer (
->     Officer_ID INT PRIMARY KEY AUTO_INCREMENT,
->     Name VARCHAR(100) NOT NULL,
->     Officer_Rank VARCHAR(50),
->     Badge_no VARCHAR(30) UNIQUE NOT NULL,
->     Contact_no VARCHAR(15)
-> );
Query OK, 0 rows affected (0.04 sec)

```

```

mysql> CREATE TABLE IF NOT EXISTS Violation_Type (
->     ViolationType_ID INT PRIMARY KEY AUTO_INCREMENT,
->     Type_Name VARCHAR(50) UNIQUE NOT NULL,
->     Default_Amount DECIMAL(10,2) NOT NULL,
->     Default_Demerit_Points INT DEFAULT 0,
->     Default_Duedays INT DEFAULT 30,
->     Description VARCHAR(255)
-> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE IF NOT EXISTS Audit_Log (
->     Audit_ID INT PRIMARY KEY AUTO_INCREMENT,
->     Action VARCHAR(20),
->     Table_Name VARCHAR(50),
->     Record_ID INT,
->     Action_By VARCHAR(100),
->     Action_Time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
->     Details TEXT
-> );
Query OK, 0 rows affected (0.03 sec)

```

```

mysql> CREATE TABLE Violation (
    ->     Violation_ID INT PRIMARY KEY AUTO_INCREMENT,
    ->     Date_Time TIMESTAMP NOT NULL,
    ->     Type VARCHAR(50),
    ->     Location VARCHAR(100),
    ->     Vehicle_ID INT NOT NULL,
    ->     Officer_ID INT,
    ->     FOREIGN KEY (Vehicle_ID) REFERENCES Vehicle(Vehicle_ID),
    ->     FOREIGN KEY (Officer_ID) REFERENCES Officer(Officer_ID)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Penalty (
    ->     Penalty_ID INT PRIMARY KEY AUTO_INCREMENT,
    ->     Amount DECIMAL(10,2) NOT NULL,
    ->     DueDate DATE,
    ->     Status VARCHAR(20) CHECK (Status IN ('Unpaid','Paid','Appealed')),
    ->     Violation_ID INT UNIQUE,
    ->     FOREIGN KEY (Violation_ID) REFERENCES Violation(Violation_ID)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Payment (
    ->     Payment_ID INT PRIMARY KEY AUTO_INCREMENT,
    ->     Date DATE NOT NULL,
    ->     Amount DECIMAL(10,2) NOT NULL,
    ->     ModeofPayment VARCHAR(20) CHECK (ModeofPayment IN ('Cash','Card','Online','UPI')),
    ->     Penalty_ID INT UNIQUE,
    ->     FOREIGN KEY (Penalty_ID) REFERENCES Penalty(Penalty_ID)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Appeal (
    ->     Appeal_ID INT PRIMARY KEY AUTO_INCREMENT,
    ->     Datefiled DATE NOT NULL,
    ->     Status VARCHAR(20) CHECK (Status IN ('Pending','Accepted','Rejected')),
    ->     Reason VARCHAR(255),
    ->     Violation_ID INT UNIQUE,
    ->     Driver_ID INT,
    ->     FOREIGN KEY (Violation_ID) REFERENCES Violation(Violation_ID),
    ->     FOREIGN KEY (Driver_ID) REFERENCES Driver(Driver_ID)
    -> );
Query OK, 0 rows affected (0.05 sec)

```

6. DML STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE WITH INSERTED VALUES

-- Driver Table Inserts

```
INSERT INTO Driver (Name, Address, Contact_no, License_no) VALUES
('Arun Kumar', 'Bangalore', '9876543210', 'KA05AB1234'),
('Meera Nair', 'Mysore', '9988776655', 'KA09CD5678'),
('Rohit Sharma', 'Mangalore', '9123456780', 'KA19EF9101'),
('Sneha Patil', 'Hubli', '9345678910', 'KA25GH1122');
```

-- Vehicle Table Inserts

```
INSERT INTO Vehicle (Registration_year, Model, Color, License_plate, Driver_ID) VALUES
(2020, 'Hyundai i20', 'White', 'KA01XY9999', 1),
(2018, 'Honda City', 'Black', 'KA05AB4321', 2),
(2019, 'Suzuki Baleno', 'Blue', 'KA09MN8765', 3),
(2021, 'Toyota Innova', 'Silver', 'KA19PQ1122', 4);
```

-- Officer Table Inserts

```
INSERT INTO Officer (Name, Officer_Rank, Badge_no, Contact_no) VALUES
('Ravi Shankar', 'Inspector', 'B123', '9123456789'),
('Lakshmi Rao', 'Sub-Inspector', 'B124', '9876501234'),
('Manoj Kumar', 'Head Constable', 'B125', '9765432109');
```

-- Violation Table Inserts

```
INSERT INTO Violation (Date_Time, Type, Location, Vehicle_ID, Officer_ID) VALUES
('2025-09-15 10:30:00', 'Speeding', 'MG Road, Bangalore', 1, 1),
('2025-09-16 11:00:00', 'Signal Jump', 'Brigade Road, Bangalore', 1, 2),
('2025-09-17 09:15:00', 'Parking Violation', 'Mysore Palace Road', 2, 2),
('2025-09-18 14:45:00', 'Drunk Driving', 'Mangalore Port Road', 3, 3),
('2025-09-19 19:00:00', 'Overspeeding', 'Hubli NH Road', 4, 1);
```

-- Penalty Table Inserts

```
INSERT INTO Penalty (Amount, Duedate, Status, Violation_ID) VALUES
(1500, '2025-09-20', 'Unpaid', 1),
(500, '2025-09-25', 'Paid', 2),
(1000, '2025-09-22', 'Unpaid', 3),
(750, '2025-09-20', 'Appealed', 4),
(1200, '2025-09-18', 'Unpaid', 5);
```

-- Payment Table Inserts

```
INSERT INTO Payment (Date, Amount, ModeofPayment, Penalty_ID) VALUES
('2025-09-25', 500, 'Online', 2),
('2025-09-21', 1500, 'Cash', 1);
```

```
-- Appeal Table Inserts
INSERT INTO Appeal (Datefiled, Status, Reason, Violation_ID, Driver_ID) VALUES
('2025-09-15', 'Pending', 'Emergency situation', 4, 3),
('2025-09-19', 'Rejected', 'Wrong parking sign', 3, 2);
```

```
-- Violation Type Inserts
INSERT INTO Violation_Type (Type_Name, Default_Amount, Default_Demerit_Points,
Default_Duedays, Description)
VALUES
('Speeding', 1000, 2, 30, 'Exceeding speed limit'),
('Signal Jump', 1500, 3, 30, 'Jumping red traffic signal'),
('Parking Violation', 500, 1, 30, 'Illegal parking or wrong parking'),
('Drunk Driving', 2500, 6, 30, 'Driving under the influence of alcohol'),
('Overspeeding', 1200, 2, 30, 'Driving above permitted speed');
```

```
mysql> INSERT INTO Payment (Date, Amount, ModeofPayment, Penalty_ID) VALUES
-> ('2025-09-25', 500, 'Online', 2),
-> ('2025-09-21', 1500, 'Cash', 1);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Appeal (Datefiled, Status, Reason, Violation_ID, Driver_ID) VALUES
-> ('2025-09-15', 'Pending', 'Emergency situation', 4, 3),
-> ('2025-09-19', 'Rejected', 'Wrong parking sign', 3, 2);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Violation_Type (Type_Name, Default_Amount, Default_Demerit_Points,
Default_Duedays, Description)
-> VALUES
-> ('Speeding', 1000, 2, 30, 'Exceeding speed limit'),
-> ('Signal Jump', 1500, 3, 30, 'Jumping red traffic signal'),
-> ('Parking Violation', 500, 1, 30, 'Illegal parking or wrong parking')
,
-> ('Drunk Driving', 2500, 6, 30, 'Driving under the influence of alcohol'),
-> ('Overspeeding', 1200, 2, 30, 'Driving above permitted speed');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```

mysql> INSERT INTO Driver (Name, Address, Contact_no, License_no) VALUES
-> ('Arun Kumar', 'Bangalore', '9876543210', 'KA05AB1234'),
-> ('Meera Nair', 'Mysore', '9988776655', 'KA09CD5678'),
-> ('Rohit Sharma', 'Mangalore', '9123456780', 'KA19EF9101'),
-> ('Sneha Patil', 'Hubli', '9345678910', 'KA25GH1122');
Query OK, 4 rows affected (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Vehicle (Registration_year, Model, Color, License_plate,
Driver_ID) VALUES
-> (2020, 'Hyundai i20', 'White', 'KA01XY9999', 1),
-> (2018, 'Honda City', 'Black', 'KA05AB4321', 2),
-> (2019, 'Suzuki Baleno', 'Blue', 'KA09MN8765', 3),
-> (2021, 'Toyota Innova', 'Silver', 'KA19PQ1122', 4);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Officer (Name, Officer_Rank, Badge_no, Contact_no) VALUES
-> ('Ravi Shankar', 'Inspector', 'B123', '9123456789'),
-> ('Lakshmi Rao', 'Sub-Inspector', 'B124', '9876501234'),
-> ('Manoj Kumar', 'Head Constable', 'B125', '9765432109');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Violation (Date_Time, Type, Location, Vehicle_ID, Officer
_ID) VALUES
-> ('2025-09-15 10:30:00', 'Speeding', 'MG Road, Bangalore', 1, 1),
-> ('2025-09-16 11:00:00', 'Signal Jump', 'Brigade Road, Bangalore', 1,
2),
-> ('2025-09-17 09:15:00', 'Parking Violation', 'Mysore Palace Road', 2,
2),
-> ('2025-09-18 14:45:00', 'Drunk Driving', 'Mangalore Port Road', 3, 3)
'
-> ('2025-09-19 19:00:00', 'Overspeeding', 'Hubli NH Road', 4, 1);
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql>
mysql> INSERT INTO Penalty (Amount, DueDate, Status, Violation_ID) VALUES
-> (1500, '2025-09-20', 'Unpaid', 1),
-> (500, '2025-09-25', 'Paid', 2),
-> (1000, '2025-09-22', 'Unpaid', 3),
-> (750, '2025-09-20', 'Appealed', 4),
-> (1200, '2025-09-18', 'Unpaid', 5);

```

```

mysql> select * from appeal;
+-----+-----+-----+-----+-----+-----+
| Appeal_ID | Datefiled | Status | Reason | Violation_ID | Driver_ID |
+-----+-----+-----+-----+-----+-----+
| 1 | 2025-09-15 | Pending | Emergency situation | 4 | 3 |
| 2 | 2025-09-19 | Rejected | Wrong parking sign | 3 | 2 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from audit_log;
Empty set (0.01 sec)

mysql> select * from driver;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'slect * from driver' at line 1
mysql> select * from driver;
+-----+-----+-----+-----+
| Driver_ID | Name | Address | Contact_no | License_no |
+-----+-----+-----+-----+
| 1 | Arun Kumar | Bangalore | 9876543210 | KA05AB1234 |
| 2 | Meera Nair | Mysore | 9988776655 | KA09CD5678 |
| 3 | Rohit Sharma | Mangalore | 9123456780 | KA19EF9101 |
| 4 | Sneha Patil | Hubli | 9345678910 | KA25GH1122 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

```

mysql> select * from officer;
+-----+-----+-----+-----+
| Officer_ID | Name | Officer_Rank | Badge_no | Contact_no |
+-----+-----+-----+-----+
| 1 | Ravi Shankar | Inspector | B123 | 9123456789 |
| 2 | Lakshmi Rao | Sub-Inspector | B124 | 9876501234 |
| 3 | Manoj Kumar | Head Constable | B125 | 9765432109 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from payment;
+-----+-----+-----+-----+
| Payment_ID | Date | Amount | ModeofPayment | Penalty_ID |
+-----+-----+-----+-----+
| 1 | 2025-09-25 | 500.00 | Online | 2 |
| 2 | 2025-09-21 | 1500.00 | Cash | 1 |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> select * from penalty;
+-----+-----+-----+-----+
| Penalty_ID | Amount | Duedate | Status | Violation_ID |
+-----+-----+-----+-----+
| 1 | 1500.00 | 2025-09-20 | Unpaid | 1 |
| 2 | 500.00 | 2025-09-25 | Paid | 2 |
| 3 | 1000.00 | 2025-09-22 | Unpaid | 3 |
| 4 | 750.00 | 2025-09-20 |Appealed | 4 |
| 5 | 1200.00 | 2025-09-18 | Unpaid | 5 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from vehicle;
+-----+-----+-----+-----+-----+
| Vehicle_ID | Registration_year | Model | Color | License_plate | Driver_ID |
+-----+-----+-----+-----+-----+
| 1 | 2020 | Hyundai i20 | White | KA01XY9999 | 1 |
| 2 | 2018 | Honda City | Black | KA05AB4321 | 2 |
| 3 | 2019 | Suzuki Baleno | Blue | KA09MN8765 | 3 |
| 4 | 2021 | Toyota Innova | Silver | KA19PQ1122 | 4 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

```

mysql> select * from violation;
+-----+-----+-----+-----+-----+
| Violation_ID | Date_Time | Type | Location | Vehicle_ID | Officer_ID |
+-----+-----+-----+-----+-----+
| 1 | 2025-09-15 10:30:00 | Speeding | MG Road, Bangalore | 1 | 1 |
| 2 | 2025-09-16 11:00:00 | Signal Jump | Brigade Road, Bangalore | 1 | 2 |
| 3 | 2025-09-17 09:15:00 | Parking Violation | Mysore Palace Road | 2 | 2 |
| 4 | 2025-09-18 14:45:00 | Drunk Driving | Mangalore Port Road | 3 | 3 |
| 5 | 2025-09-19 19:00:00 | Overspeeding | Hubli NH Road | 4 | 1 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from violation_type;
+-----+-----+-----+-----+-----+
| ViolationType_ID | Type_Name | Default_Amount | Default_Demerit_Points | Default_Duedays | Description |
+-----+-----+-----+-----+-----+
| 1 | Speeding | 1000.00 | 2 | 30 | Exceeding speed limit |
| 2 | Signal Jump | 1500.00 | 3 | 30 | Jumping red traffic signal |
| 3 | Parking Violation | 500.00 | 1 | 30 | Illegal parking or wrong parking |
| 4 | Drunk Driving | 2500.00 | 6 | 30 | Driving under the influence of alcohol |
| 5 | Overspeeding | 1200.00 | 2 | 30 | Driving above permitted speed |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

7. QUERIES

7.1 SIMPLE QUERY WITH GROUP BY, AGRREGATE

SELECT

```
COUNT(DISTINCT d.Driver_ID) AS Total_Drivers,  
COUNT(v.Violation_ID) AS Total_Violations,  
SUM(p.Amount) AS Total_Revenue,  
AVG(p.Amount) AS Average_Fine,  
MAX(p.Amount) AS Highest_Fine,  
MIN(p.Amount) AS Lowest_Fine,  
SUM(CASE WHEN p.Status = 'Paid' THEN p.Amount ELSE 0 END) AS  
Revenue_Collected,  
SUM(CASE WHEN p.Status = 'Unpaid' THEN p.Amount ELSE 0 END) AS  
Revenue_Pending  
FROM Driver d  
JOIN Vehicle ve ON d.Driver_ID = ve.Driver_ID  
JOIN Violation v ON ve.Vehicle_ID = v.Vehicle_ID  
JOIN Penalty p ON v.Violation_ID = p.Violation_ID;
```

```
mysql> SELECT  
->     COUNT(DISTINCT d.Driver_ID) AS Total_Drivers,  
->     COUNT(v.Violation_ID) AS Total_Violations,  
->     SUM(p.Amount) AS Total_Revenue,  
->     AVG(p.Amount) AS Average_Fine,  
->     MAX(p.Amount) AS Highest_Fine,  
->     MIN(p.Amount) AS Lowest_Fine,  
->     SUM(CASE WHEN p.Status = 'Paid' THEN p.Amount ELSE 0 END) AS Revenue_Collected,  
->     SUM(CASE WHEN p.Status = 'Unpaid' THEN p.Amount ELSE 0 END) AS Revenue_Pending  
->   FROM Driver d  
->   JOIN Vehicle ve ON d.Driver_ID = ve.Driver_ID  
->   JOIN Violation v ON ve.Vehicle_ID = v.Vehicle_ID  
->   JOIN Penalty p ON v.Violation_ID = p.Violation_ID;  
+-----+-----+-----+-----+-----+-----+-----+  
| Total_Drivers | Total_Violations | Total_Revenue | Average_Fine | Highest_Fine | Lowest_Fine | Revenue_Collected | Revenue_Pending |  
+-----+-----+-----+-----+-----+-----+-----+  
|         9 |          12 |      14250.00 |    1187.500000 |       2500.00 |        500.00 |        2000.00 |       4700.00 |  
+-----+-----+-----+-----+-----+-----+-----+  
1 row in set (0.43 sec)
```

7.2 UPDATE OPERATION

UPDATE Driver

```
SET Contact_no = '9999999999',  
Address = 'New Residential Area, Bangalore'  
WHERE Driver_ID = 1;
```

```

mysql> SELECT Driver_ID, Name, Address, Contact_no
-> FROM Driver
-> WHERE Driver_ID = 1;
+-----+-----+-----+
| Driver_ID | Name      | Address    | Contact_no |
+-----+-----+-----+
|       1 | Arun Kumar | Bangalore | 9876543210 |
+-----+-----+-----+
1 row in set (0.12 sec)

mysql> UPDATE Driver
-> SET Contact_no = '9999999999',
->      Address = 'New Residential Area, Bangalore'
-> WHERE Driver_ID = 1;
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT Driver_ID, Name, Address, Contact_no
-> FROM Driver
-> WHERE Driver_ID = 1;
+-----+-----+-----+-----+
| Driver_ID | Name      | Address          | Contact_no |
+-----+-----+-----+-----+
|       1 | Arun Kumar | New Residential Area, Bangalore | 9999999999 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

7.3 DELETE OPERATION

Delete a vehicle or driver with dependency checking (CASCADE delete).

[DELETE FROM Vehicle WHERE Vehicle_ID = 10;](#)

```

mysql> select * from vehicle;
+-----+-----+-----+-----+-----+-----+
| Vehicle_ID | Registration_year | Model | Color | License_plate | Driver_ID |
+-----+-----+-----+-----+-----+-----+
| 1 | 2020 | Hyundai i20 | White | KA01XY9999 | 1 |
| 2 | 2018 | Honda City | Black | KA05AB4321 | 2 |
| 3 | 2019 | Suzuki Baleno | Blue | KA09MN8765 | 3 |
| 4 | 2021 | Toyota Innova | Silver | KA19PQ1122 | 4 |
| 5 | 2017 | Ford Figo | Red | DL02CL4321 | 5 |
| 6 | 2023 | Volkswagen Polo | White | GA09NM7654 | 6 |
| 7 | 2022 | Honda Amaze | Silver | GJ06AB9912 | 7 |
| 8 | 2020 | Hyundai Creta | Black | KL09ZZ9876 | 8 |
| 9 | 1996 | Maruti 800 | Violet | KA03FH7474 | 9 |
| 10 | 2023 | Toyota Fortuner | Silver | KA20XY9999 | 1 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> DELETE FROM Vehicle WHERE Vehicle_ID = 10;
Query OK, 1 row affected (0.14 sec)

mysql> select * from vehicle;
+-----+-----+-----+-----+-----+-----+
| Vehicle_ID | Registration_year | Model | Color | License_plate | Driver_ID |
+-----+-----+-----+-----+-----+-----+
| 1 | 2020 | Hyundai i20 | White | KA01XY9999 | 1 |
| 2 | 2018 | Honda City | Black | KA05AB4321 | 2 |
| 3 | 2019 | Suzuki Baleno | Blue | KA09MN8765 | 3 |
| 4 | 2021 | Toyota Innova | Silver | KA19PQ1122 | 4 |
| 5 | 2017 | Ford Figo | Red | DL02CL4321 | 5 |
| 6 | 2023 | Volkswagen Polo | White | GA09NM7654 | 6 |
| 7 | 2022 | Honda Amaze | Silver | GJ06AB9912 | 7 |
| 8 | 2020 | Hyundai Creta | Black | KL09ZZ9876 | 8 |
| 9 | 1996 | Maruti 800 | Violet | KA03FH7474 | 9 |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

```

7.4 CORRELATED QUERY

Find drivers with above-average total fines

```
SELECT d.Name, d.License_no,
       (SELECT SUM(p.Amount)
        FROM Penalty p
        JOIN Violation v ON p.Violation_ID = v.Violation_ID
        JOIN Vehicle ve ON v.Vehicle_ID = ve.Vehicle_ID
        WHERE ve.Driver_ID = d.Driver_ID) AS Total_Fines
  FROM Driver d
 WHERE (SELECT SUM(p.Amount)
        FROM Penalty p
        JOIN Violation v ON p.Violation_ID = v.Violation_ID
        JOIN Vehicle ve ON v.Vehicle_ID = ve.Vehicle_ID)
```

```

        WHERE ve.Driver_ID = d.Driver_ID) >
        (SELECT AVG(Amount) FROM Penalty)
ORDER BY Total_Fines DESC;
```

```

mysql> SELECT d.Name, d.License_no,
->           (SELECT SUM(p.Amount)
->             FROM Penalty p
->             JOIN Violation v ON p.Violation_ID = v.Violation_ID
->             JOIN Vehicle ve ON v.Vehicle_ID = ve.Vehicle_ID
->             WHERE ve.Driver_ID = d.Driver_ID) AS Total_Fines
->   FROM Driver d
-> WHERE (SELECT SUM(p.Amount)
->             FROM Penalty p
->             JOIN Violation v ON p.Violation_ID = v.Violation_ID
->             JOIN Vehicle ve ON v.Vehicle_ID = ve.Vehicle_ID
->             WHERE ve.Driver_ID = d.Driver_ID) >
->           (SELECT AVG(Amount) FROM Penalty)
-> ORDER BY Total_Fines DESC;
+-----+-----+-----+
| Name      | License_no | Total_Fines |
+-----+-----+-----+
| Arun Kumar | KA05AB1234 | 4000.00 |
| Nandana Mathur | KA04FG83933 | 2500.00 |
| Priya Menon | KL07ZE9982 | 2300.00 |
| Sneha Patil | KA25GH1122 | 1200.00 |
+-----+-----+-----+
4 rows in set (0.10 sec)
```

7.5 NESTED QUERY

Drivers with more than one violation

DELIMITER \$\$

```

CREATE VIEW DriversWithMultipleViolations AS
SELECT d.Driver_ID, d.Name, COUNT(v.Violation_ID) AS Total_Violations
FROM Driver d
JOIN Vehicle ve ON d.Driver_ID = ve.Driver_ID
JOIN Violation v ON ve.Vehicle_ID = v.Vehicle_ID
GROUP BY d.Driver_ID, d.Name
HAVING COUNT(v.Violation_ID) > 1;
$$
DELIMITER ;
```

```
mysql> SELECT * FROM DriversWithMultipleViolations;
-> $$
```

Driver_ID	Name	Total_Violations
1	Arun Kumar	3
5	Vishal Singh	2

```
2 rows in set (0.34 sec)
```

8. STORED PROCEDURES, FUNCTIONS AND TRIGGERS

8.1 STORED PROCEDURES OR FUNCTIONS

1. GetMostFrequentViolationType()

This function finds and returns the most commonly occurring violation type in the database. It helps traffic authorities understand common offenses and improve enforcement policies.

```
mysql> DELIMITER $$  
mysql> CREATE FUNCTION GetMostFrequentViolationType()  
-> RETURNS VARCHAR(50)  
-> DETERMINISTIC  
-> BEGIN  
->     DECLARE mostFrequentType VARCHAR(50);  
->  
->     SELECT vi.Type  
->         INTO mostFrequentType  
->         FROM Violation vi  
->         GROUP BY vi.Type  
->         ORDER BY COUNT(*) DESC  
->         LIMIT 1;  
->  
->     RETURN mostFrequentType;  
-> END $$  
Query OK, 0 rows affected (0.03 sec)  
  
mysql>  
mysql> DELIMITER ;  
mysql> SELECT GetMostFrequentViolationType() AS MostCommonViolation;  
+-----+  
| MostCommonViolation |  
+-----+  
| Underage Driving |  
+-----+  
1 row in set (0.01 sec)
```

1. CalculateTotalUnpaidFines

This procedure returns the total unpaid fine amount for a specific driver. It helps track outstanding penalties and supports enforcement authorities in monitoring violators.

```
mysql> DELIMITER $$  
mysql>  
mysql> CREATE PROCEDURE CalculateTotalUnpaidFines(IN p_DriverID INT)  
    -> BEGIN  
    ->     SELECT  
    ->         d.Driver_ID,  
    ->         d.Name,  
    ->         SUM(p.Amount) AS Total_Unpaid_Fines  
    ->     FROM Driver d  
    ->     JOIN Vehicle v ON d.Driver_ID = v.Driver_ID  
    ->     JOIN Violation vi ON v.Vehicle_ID = vi.Vehicle_ID  
    ->     JOIN Penalty p ON vi.Violation_ID = p.Violation_ID  
    ->     WHERE d.Driver_ID = p_DriverID  
    ->         AND p.Status = 'Unpaid'  
    ->     GROUP BY d.Driver_ID, d.Name;  
    -> END $$  
Query OK, 0 rows affected (0.01 sec)  
  
mysql>  
mysql> DELIMITER ;  
mysql> CALL CalculateTotalUnpaidFines(2);  
+-----+-----+  
| Driver_ID | Name      | Total_Unpaid_Fines |  
+-----+-----+  
|          2 | Meera Nair |           1000.00 |  
+-----+-----+  
1 row in set (0.01 sec)  
  
Query OK, 0 rows affected (0.02 sec)
```

2. GetDriverViolationHistory

This procedure retrieves a complete violation and penalty history for a given driver. It is useful for generating driver profiles, legal review, and reporting purposes.

```
mysql> DELIMITER $$  
mysql>  
mysql> CREATE PROCEDURE GetDriverViolationHistory(IN p_DriverID INT)  
-> BEGIN  
->     SELECT  
->         d.Driver_ID,  
->         d.Name,  
->         v.Vehicle_ID,  
->         vi.Violation_ID,  
->         vi.Type AS Violation_Type,  
->         vi.Date_Time,  
->         vi.Location,  
->         p.Amount AS Penalty_Amount,  
->         p.Status AS Penalty_Status  
->     FROM Driver d  
->     JOIN Vehicle v ON d.Driver_ID = v.Driver_ID  
->     JOIN Violation vi ON v.Vehicle_ID = vi.Vehicle_ID  
->     LEFT JOIN Penalty p ON vi.Violation_ID = p.Violation_ID  
->     WHERE d.Driver_ID = p_DriverID  
->     ORDER BY vi.Date_Time DESC;  
-> END $$  
Query OK, 0 rows affected (0.01 sec)  
  
mysql>  
mysql> DELIMITER ;  
mysql> CALL GetDriverViolationHistory(2);  
+-----+-----+-----+-----+-----+  
| Driver_ID | Name      | Vehicle_ID | Violation_ID | Violation_Type | D  
ate_Time          | Location      | Penalty_Amount | Penalty_Status |  
+-----+-----+-----+-----+-----+  
|        2 | Meera Nair |          2 |            3 | Parking Violation | 2  
025-09-17 09:15:00 | Mysore Palace Road |       1000.00 | Unpaid |  
+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)  
  
Query OK, 0 rows affected (0.04 sec)
```

8.2 TRIGGERS

1. trg_setViolationType_id

Before inserting a violation, this trigger automatically maps the violation type to the correct violation type ID. It maintains referential integrity and avoids data entry mistakes.

```
mysql> CREATE TRIGGER trg_setViolationType_id
-> BEFORE INSERT ON Violation
-> FOR EACH ROW
-> BEGIN
->     DECLARE type_id INT;
->
->     -- Get ViolationType_ID from Violation_Type table
->     SELECT ViolationType_ID INTO type_id
->     FROM Violation_Type
->     WHERE Type_Name = NEW.Type;
->
->     -- Set the ViolationType_ID
->     IF type_id IS NOT NULL THEN
->         SET NEW.ViolationType_ID = type_id;
->     END IF;
-> END$$
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> INSERT INTO Violation (Date_Time, Type, Location, Vehicle_ID, Officer_ID)
-> VALUES (NOW(), 'No Insurance', 'Highway Checkpoint', 5, 1);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT *
-> FROM Violation
-> ORDER BY Violation_ID DESC
-> LIMIT 1;
+-----+-----+-----+-----+-----+
| Violation_ID | Date_Time | Type | Location | Vehicle_ID | Officer_ID | ViolationType_ID |
+-----+-----+-----+-----+-----+
|      15 | 2025-11-07 11:59:12 | No Insurance | Highway Checkpoint |      5 |      1 |          10 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

2. trg_auto_create_penalty

This trigger automatically calculates a penalty amount based on the type of violation recorded. After a new violation is inserted, it then creates a corresponding 'Unpaid' penalty record with the calculated fine and a due date set for 30 days later.

```
mysql> DELIMITER $$  
mysql> CREATE TRIGGER trg_auto_create_penalty  
-> AFTER INSERT ON Violation  
-> FOR EACH ROW  
-> BEGIN  
->     DECLARE violation_amount DECIMAL(10,2);  
->  
->     SET violation_amount = CASE NEW.Type  
->         WHEN 'Speeding' THEN 1000.00  
->         WHEN 'Signal Jump' THEN 1500.00  
->         WHEN 'Parking Violation' THEN 500.00  
->         WHEN 'Drunk Driving' THEN 2500.00  
->         WHEN 'Underage Driving' THEN 2000.00  
->         WHEN 'Seatbelt Violation' THEN 500.00  
->         WHEN 'Mobile Usage' THEN 1000.00  
->         WHEN 'No Insurance' THEN 2300.00  
->         ELSE 500.00  
->     END;  
->  
->     INSERT INTO Penalty (Amount, Duedate, Status, Violation_ID)  
->     VALUES (violation_amount, DATE_ADD(CURRENT_DATE, INTERVAL 30 DAY)  
, 'Unpaid', NEW.Violation_ID);  
-> END$$  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SELECT *  
-> FROM Penalty  
-> ORDER BY Penalty_ID DESC  
-> LIMIT 1;  
+-----+-----+-----+-----+-----+  
| Penalty_ID | Amount | Duedate | Status | Violation_ID |  
+-----+-----+-----+-----+-----+  
| 12 | 2300.00 | 2025-12-07 | Unpaid | 15 |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

3. trg_update_penalty_status

When a payment entry is made, this trigger updates the corresponding penalty status to "Paid". It guarantees real-time synchronization between payment and penalty status, improving data consistency.

```
mysql> DELIMITER $$  
mysql>  
mysql> DROP TRIGGER IF EXISTS trg_update_penalty_status$$  
Query OK, 0 rows affected, 1 warning (0.01 sec)  
  
mysql>  
mysql> CREATE TRIGGER trg_update_penalty_status  
    -> AFTER INSERT ON Payment  
    -> FOR EACH ROW  
    -> BEGIN  
    ->     -- Update the penalty status to 'Paid' when payment is made  
    ->     UPDATE Penalty  
    ->     SET Status = 'Paid'  
    ->     WHERE Penalty_ID = NEW.Penalty_ID;  
    -> END$$  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> DELIMITER ;
```

```
mysql> INSERT INTO Payment (Date, Amount, ModeofPayment, Penalty_ID)  
    -> VALUES (CURDATE(), 1200, 'UPI', 5);  
Query OK, 1 row affected (0.01 sec)  
  
mysql> SELECT Penalty_ID, Status FROM Penalty WHERE Penalty_ID = 5;  
+-----+-----+  
| Penalty_ID | Status |  
+-----+-----+  
|          5 | Paid   |  
+-----+-----+  
1 row in set (0.00 sec)
```

4. trg_appeal_filed

This trigger runs when an appeal is submitted and updates the penalty status to "Appealed". It also logs the action into the audit log table for transparency and traceability.

```
mysql> CREATE TRIGGER trg_appeal_filed
-> AFTER INSERT ON Appeal
-> FOR EACH ROW
-> BEGIN
->     -- As soon as appeal is filed, mark penalty as 'Appealed' (pending review)
->     UPDATE Penalty
->     SET Status = 'Appealed'
->     WHERE Violation_ID = NEW.Violation_ID;
->
->     -- Log it
->     INSERT INTO Audit_Log (Action, Table_Name, Record_ID, Action_By, Details)
->         VALUES ('INSERT', 'Appeal', NEW.Appeal_ID, USER(),
->                 CONCAT('Appeal filed for Violation #', NEW.Violation_ID,
' - Penalty status → Appealed'));
-> END$$
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
```

```
mysql> SELECT * FROM Penalty WHERE Status = 'Unpaid' LIMIT 1;
+-----+-----+-----+-----+
| Penalty_ID | Amount | Duedate | Status | Violation_ID |
+-----+-----+-----+-----+
| 1 | 1500.00 | 2025-09-20 | Unpaid | 1 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> INSERT INTO Appeal (Datefiled, Status, Reason, Violation_ID, Driver_ID)
-> VALUES (CURDATE(), 'Pending', 'Requesting reconsideration', 1, 1);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT Violation_ID, Status FROM Penalty WHERE Violation_ID = 1;
+-----+-----+
| Violation_ID | Status |
+-----+-----+
| 1 | Appealed |
+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM Audit_Log ORDER BY Action_Time DESC LIMIT 1;
+-----+-----+-----+-----+-----+-----+
| Audit_ID | Action | Table_Name | Record_ID | Action_By | Action_Time | Details |
+-----+-----+-----+-----+-----+-----+
| 3 | INSERT | Appeal | 5 | root@localhost | 2025-11-07 12:20:59 | Appeal filed for Violation #1 - Penalty status ? Appealed |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

5. trg_update_penalty_on_appeal

This trigger activates when an appeal status changes and updates the penalty to "Paid" or "Unpaid" based on the decision. It ensures penalties reflect correct legal outcomes and creates a log entry for documentation.

```
mysql> DELIMITER $$  
mysql>  
mysql> CREATE TRIGGER trg_update_penalty_on_appeal  
    --> AFTER UPDATE ON Appeal  
    --> FOR EACH ROW  
    --> BEGIN  
    -->     -- Only act if status changed  
    -->     IF OLD.Status != NEW.Status THEN  
    -->  
    -->         -- If appeal is ACCEPTED, mark penalty as 'Paid' (waived/forgiven)  
    -->         IF NEW.Status = 'Accepted' THEN  
    -->             UPDATE Penalty  
    -->             SET Status = 'Paid'  
    -->             WHERE Violation_ID = NEW.Violation_ID;  
    -->  
    -->             INSERT INTO Audit_Log (Action, Table_Name, Record_ID, Action_By, Details)  
    -->             VALUES ('UPDATE', 'Penalty', NEW.Violation_ID, USER(),  
    -->                         CONCAT('Appeal #', NEW.Appeal_ID, ' ACCEPTED - Penalty waived (Status > Paid)'));  
    -->  
    -->         -- If appeal is REJECTED, mark penalty as 'Unpaid' (must pay)  
    -->         ELSEIF NEW.Status = 'Rejected' THEN  
    -->             UPDATE Penalty  
    -->             SET Status = 'Unpaid'  
    -->             WHERE Violation_ID = NEW.Violation_ID;  
    -->  
    -->             INSERT INTO Audit_Log (Action, Table_Name, Record_ID, Action_By, Details)  
    -->             VALUES ('UPDATE', 'Penalty', NEW.Violation_ID, USER(),  
    -->                         CONCAT('Appeal #', NEW.Appeal_ID, ' REJECTED - Penalty must be paid (Status > Unpaid)'));  
    -->  
    -->         -- If changed back to PENDING (rare case)  
    -->         ELSEIF NEW.Status = 'Pending' THEN  
    -->             UPDATE Penalty  
    -->             SET Status = 'Appealed'  
    -->             WHERE Violation_ID = NEW.Violation_ID;  
    -->         END IF;  
    -->  
    -->     END IF;  
    --> END$$  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SELECT * FROM Appeal ORDER BY Appeal_ID DESC LIMIT 1;  
+-----+-----+-----+-----+-----+-----+  
| Appeal_ID | Datefiled | Status | Reason | Violation_ID | Driver_ID |  
+-----+-----+-----+-----+-----+-----+  
| 5 | 2025-11-07 | Pending | Requesting reconsideration | 1 | 1 |  
+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> UPDATE Appeal  
    -->     SET Status = 'Accepted'  
    -->     WHERE Appeal_ID = 5;  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> SELECT Violation_ID, Status FROM Penalty WHERE Violation_ID = 1;  
+-----+-----+  
| Violation_ID | Status |  
+-----+-----+  
| 1 | Paid |  
+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> SELECT * FROM Audit_Log ORDER BY Action_Time DESC LIMIT 1;  
+-----+-----+-----+-----+-----+-----+  
| Audit_ID | Action | Table_Name | Record_ID | Action_By | Action_Time | Details |  
+-----+-----+-----+-----+-----+-----+  
| 4 | UPDATE | Penalty | 1 | root@localhost | 2025-11-07 12:26:21 | Appeal #5 ACCEPTED - Penalty waived (Status ? Paid) |  
+-----+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

9. FRONT END DEVELOPMENT

```
PS C:\Users\Lenovo\OneDrive\Desktop\Sem V\DBMS\Mini_Project\traffic-guard> streamlit run Home.py
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://10.1.4.19:8501
```

run the streamlit command to open the GUI.

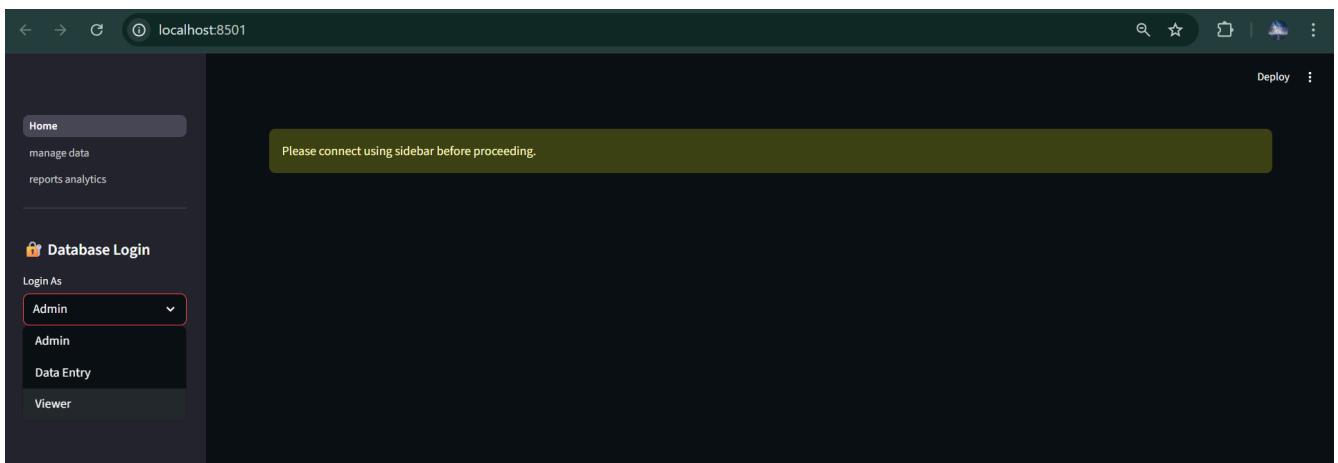


Figure 7.1: User Interface - 3 user roles as options to log in with different privileges.

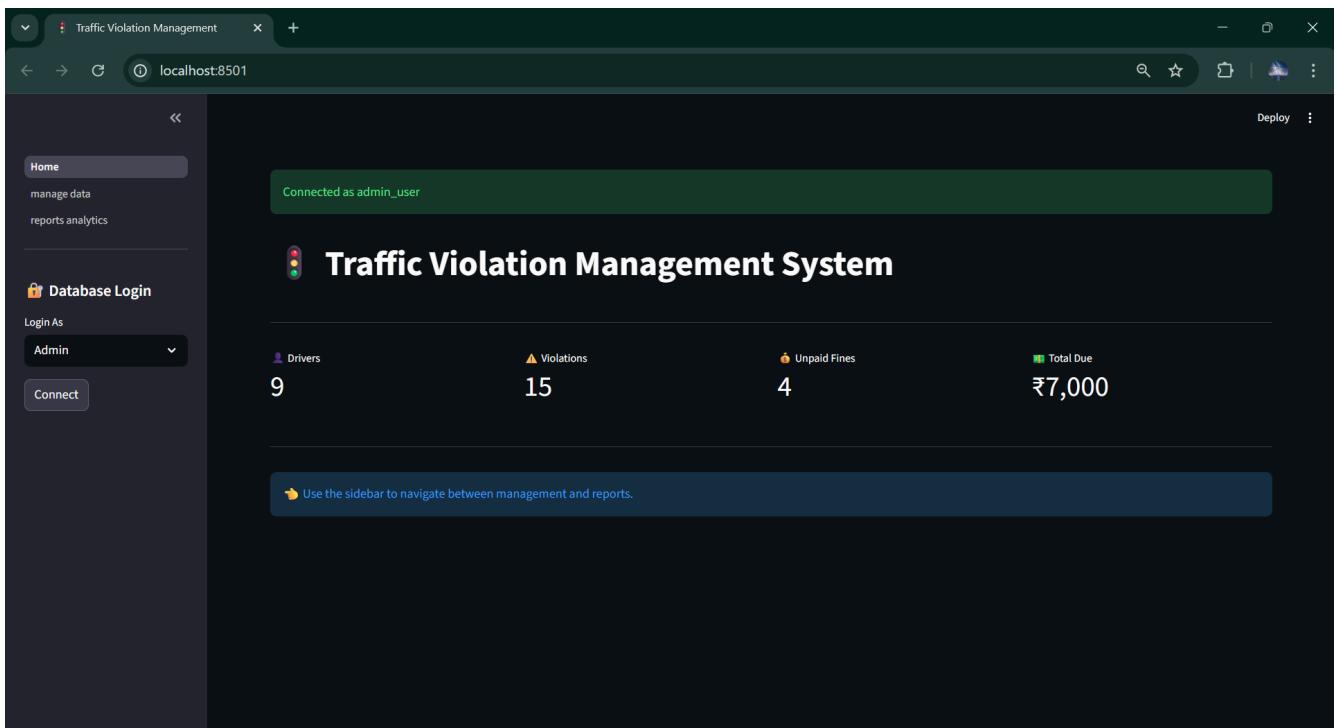


Figure 7.2: User Interface: Click on connect and get connected to the database as admin.

The UI is clearly designed to provide the administrator with a quick, high-level overview of the entire system's current statistics, including the number of registered drivers, total violations, and critical financial data (unpaid fines and total revenue pending).

The screenshot shows a web-based application titled "Data Management" under the "Manage Data" tab. The URL is "localhost:8501/manage_data". The left sidebar has links for "Home", "manage data" (which is selected), and "reports analytics". The main content area is titled "Drivers" and displays a table of driver data. The table has columns: Driver_ID, Name, Address, Contact_no, and License_no. The data is as follows:

Driver_ID	Name	Address	Contact_no	License_no
0	Arun Kumar	Mangalore	9876543210	KA05AB1234
1	Meera Nair	Mysore	9988776555	KA09CD5678
2	Rohit Sharma	Mangalore	9123456780	KA19EF9101
3	Sneha Patil	Hubli	9345678910	KA25GH1122
4	Vishal Singh	Delhi	9845123456	DL01AK5123
5	Sara Fernandes	Goa	9938123076	GA08Y1547
6	Ankit Bhatt	Ahmedabad	9823032133	GJ05RT7721
7	Priya Menon	Kochi	9891123567	KL07ZE9982
8	Himani Singh	Delhi	9449469592	DL02RY7853

Below the table are three buttons: "Add Driver", "Update Driver", and "Delete Driver".

Figure 7.3: Data Management - Drivers Tab

The screenshot shows a web-based administrative interface for managing data. On the left sidebar, there are links for Home, manage data (which is currently selected), and reports analytics. The main content area has two forms:

- Add Driver**: Fields include Name (Suresh Kumar), Address (Hassan), Contact Number (8745123698), and License Number (KA09RS7456). A button labeled "Add Driver" is at the bottom.
- Update Driver**: Shows a dropdown for "Select Driver" with "ID 1: Arun Kumar" selected. It also includes fields for Name (Arun Kumar) and Address (Mangalore).

Figure 7.4 : In admin and data entry mode - enter details to add a new driver into the database.

The screenshot shows a table titled "Drivers" with the following data:

	Driver_ID	Name	Address	Contact_no	License_no
0	1	Arun Kumar	Mangalore	9876543210	KA05AB1234
1	2	Meera Nair	Mysore	9988776655	KA09CD5678
2	3	Rohit Sharma	Mangalore	9123456780	KA19EF9101
3	4	Sneha Patil	Hubli	9345678910	KA25GH1122
4	5	Vishal Singh	Delhi	9845123456	DL01AK123
5	6	Sara Fernandes	Goa	9938123076	GA08XY1547
6	7	Ankit Bhatt	Ahmedabad	9823032133	GJ05RT7721
7	8	Priya Menon	Kochi	9891123567	KL07ZE9982
8	12	Himani Singh	Delhi	9449469592	DL02RY7853
9	15	Suresh Kumar	Hassan	9785691236	KA09CD1234

Figure 7.5 : Driver details added at the end of the table.

Update Driver

Select Driver
ID 15: Suresh Kumar

Name
Suresh Kumar

Address
Bangalore

Contact
9785691236

License
KA09CD1234

Update

Press Ctrl+Enter to submit form

Figure 7.6 : Updated Address from Hassan to Bangalore.Driver table gets updated instantly.

Manage Data

localhost:8501/manage_data

Home

manage data

reports analytics

Data Management

Drivers

Driver_ID	Name	Address	Contact_no	License_no
0	1 Arun Kumar	Mangalore	9876543210	KA05AB1234
1	2 Meera Nair	Mysore	9988766555	KA09CD1234
2	3 Rohit Sharma	Mangalore	9123456780	KA19EF9101
3	4 Sneha Patil	Hubli	9345678910	KA25GH1122
4	5 Vishal Singh	Delhi	9845123456	DL01AK5123
5	6 Sara Fernandes	Goa	9938123076	GA08XY1547
6	7 Ankit Bhatt	Ahmedabad	9823032133	GU05RT7721
7	8 Priya Menon	Kochi	9801123567	KL07ZE9982
8	12 Himani Singh	Delhi	9449469592	DL02RY7853
9	15 Suresh Kumar	Bangalore	9785691236	KA09CD1234

Add Driver

Name*

Figure 7.7 : Updated table .

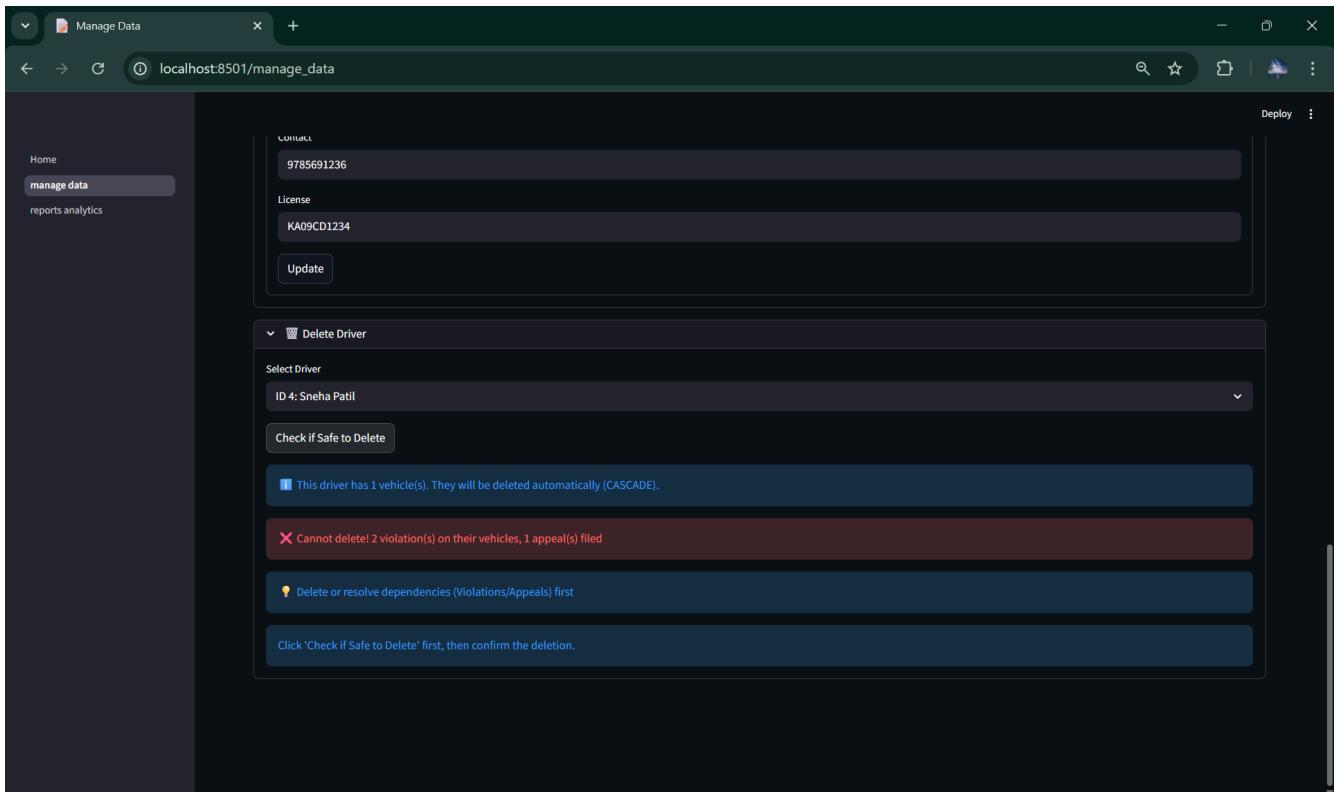


Figure 7.8 : Safety check before deleting driver. Not allowed to delete if there are pending violations against that driver.

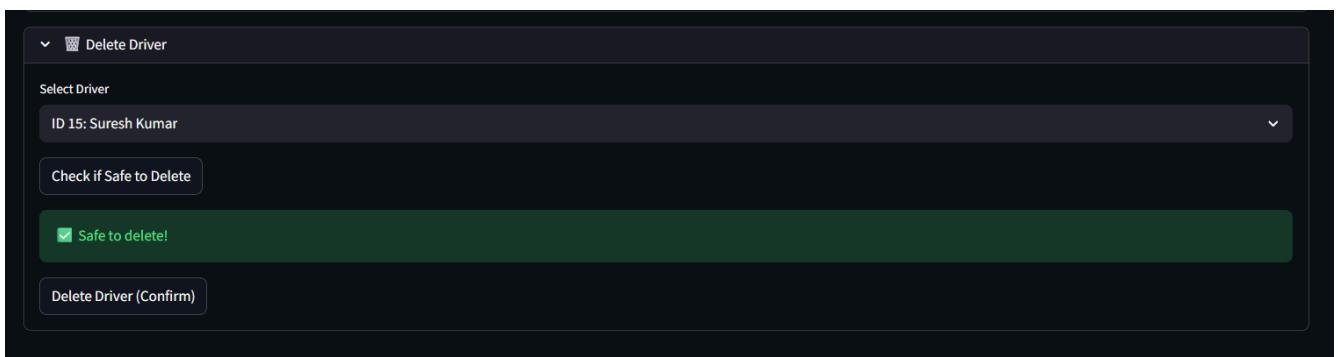


Figure 7.9 : Delete driver successful. Removed driver details from table (database).

The screenshot shows a web-based application titled "Data Management" with a dark theme. The top navigation bar includes links for Home, manage data (which is selected), and reports analytics. Below the navigation is a horizontal menu with icons for Drivers, Vehicles (highlighted in red), Officers, Violations, Payments, and Appeals. The main content area is titled "Vehicles" and displays a table with 9 rows of vehicle data. The columns are: Vehicle_ID, Registration_year, Model, Color, License_plate, Driver_ID, and Owner. The data includes various car models like Hyundai i20, Honda City, Suzuki Baleno, Toyota Innova, Ford Figo, Volkswagen Polo, Honda Amaze, and Hyundai Creta, with drivers like Arun Kumar, Meera Nair, Rohit Sharma, Sneha Patil, Vishal Singh, Sara Fernandes, Ankit Bhatt, and Priya Menon. Below the table are three buttons: "+ Add Vehicle", "Update Vehicle", and "Delete Vehicle".

Vehicle_ID	Registration_year	Model	Color	License_plate	Driver_ID	Owner
0	1	2020 Hyundai i20	White	KA01XY9999		Arun Kumar
1	2	2018 Honda City	Black	KA05AB4321		Meera Nair
2	3	2019 Suzuki Baleno	Blue	KA09MN8765		Rohit Sharma
3	4	2021 Toyota Innova	Silver	KA19PQ1122		Sneha Patil
4	5	2017 Ford Figo	Red	DL02CL4321		Vishal Singh
5	6	2023 Volkswagen Polo	White	GA09NM7654		Sara Fernandes
6	7	2022 Honda Amaze	Silver	GJ06AB9912		Ankit Bhatt
7	8	2020 Hyundai Creta	Black	KL09ZZ9876		Priya Menon

Figure 7.10 : List of vehicles in the database.

The screenshot shows the "Add Vehicle" form within the application. The form fields include: Year* (2020), Model* (Toyota Glanza), Color (White), License Plate* (KA19CS3456), and Owner* (ID 8: Priya Menon). A "Add Vehicle" button is at the bottom. Below this, there is a partially visible "Update Vehicle" form with a "Select Vehicle" dropdown containing "ID 1: KA01XY9999" and other fields for Year and Model.

Figure 7.11 : Add vehicle

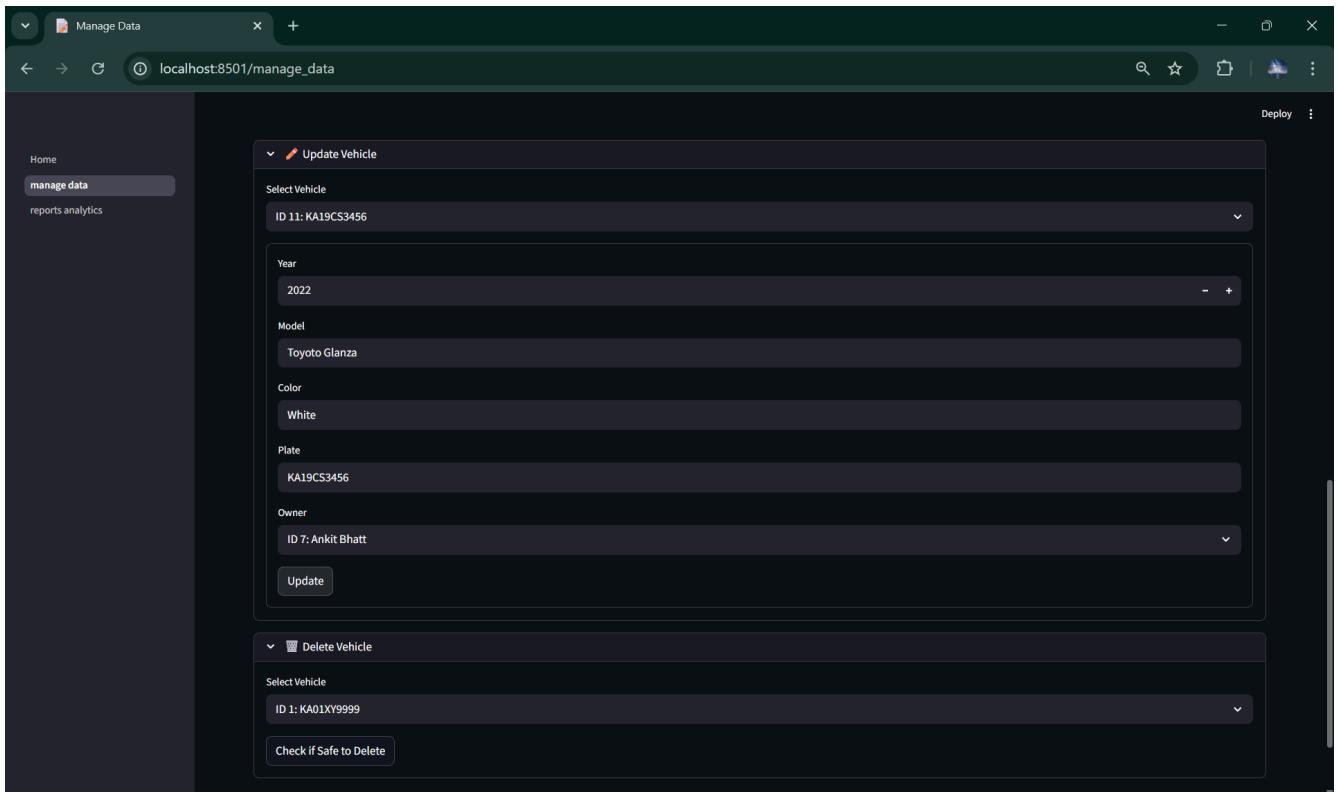


Figure 7.12 : Update Vehicle details

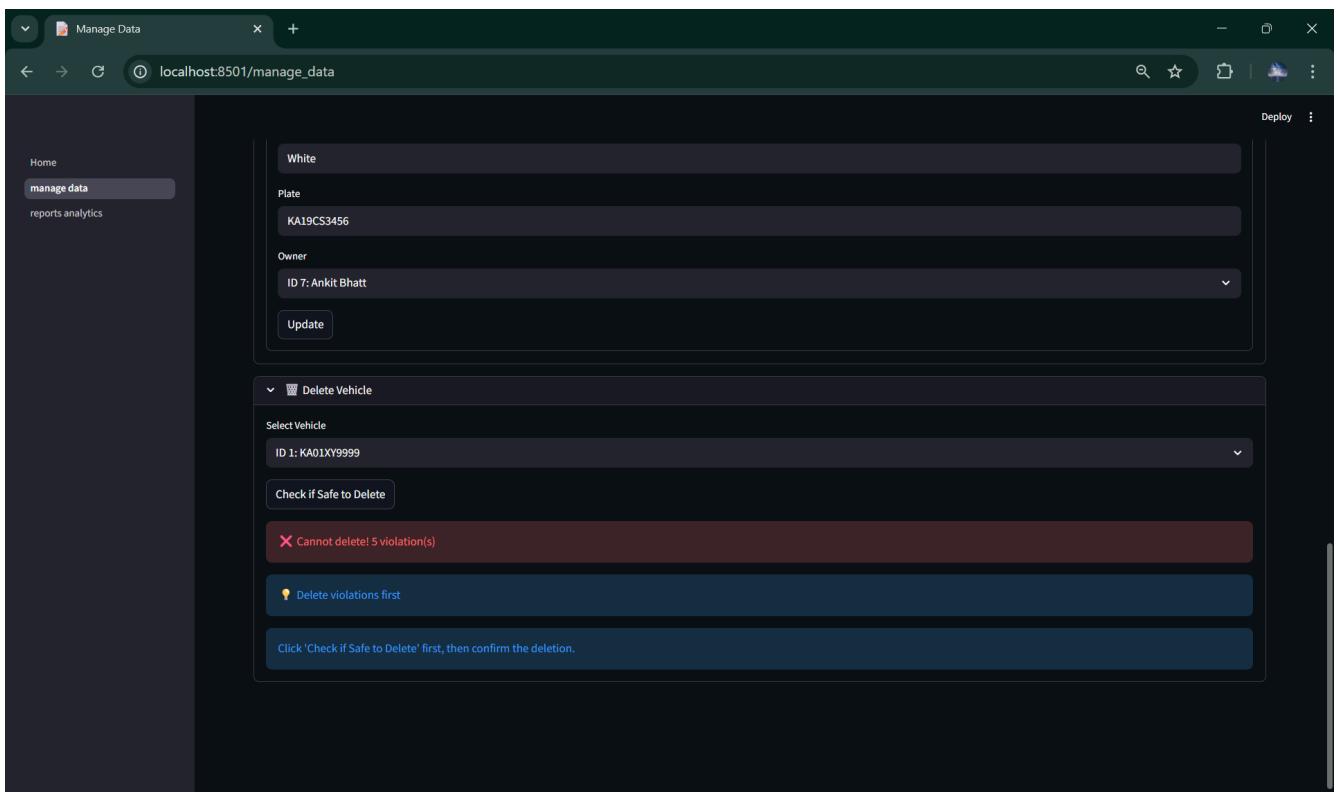


Figure 7.13 : Delete Vehicle option has a safety check feature.

The screenshot shows a web application interface titled "Manage Data" at the top. Below the title, there are navigation links: Drivers, Vehicles, Officers (which is underlined, indicating it's the active page), Violations, Payments, and Appeals. On the far right of the header, there are icons for Deploy and three dots.

The main content area has a sidebar on the left with "Home", "manage data" (which is highlighted in a dark grey box), and "reports analytics".

The main content area displays a table titled "Officers" with the following data:

Officer_ID	Name	Officer_Rank	Badge_no	Contact_no
0	Ravi Shankar	Inspector	B123	9123456789
1	Lakshmi Rao	Sub-Inspector	B124	986501234
2	Manoj Kumar	Head Constable	B125	9765432109
3	Sahil Gupta	Inspector	B126	9678123456
4	Fatima Noor	Sub-Inspector	B134	9182736451
5	Sara Patel	Head Constable	B139	8956231465

Below the table, there is a modal window titled "+ Add Officer" with the following fields:

- Name*: Sara Patel
- Rank: Head Constable (selected)
- Inspector
- Sub-Inspector
- Head Constable
- Constable

At the bottom of the modal is a "Add Officer" button.

Figure 7.14 : List of officers in the data.base

This screenshot is identical to Figure 7.14, showing the list of officers and the "Add Officer" modal. However, the data in the modal has been updated:

- Name*: Sara Patel
- Rank: Head Constable (selected)
- Badge Number*: B139
- Contact: 8956231465

At the bottom of the modal is a "Add Officer" button.

Figure 7.14 : Added new officer.

Data Management

Violations

	Violation_ID	Date_Time	Type	Location	Vehicle_ID	Officer_ID	ViolationType_ID	License_plate	Driver	Officer	Amount	Status	
0	15	2025-11-04 12:37:00	Drunk Driving	Sarjapur	8	1	4	KL09ZZ9876	Priya Menon	Ravi Shankar	2500	Unpaid	
1	14	2025-11-04 12:33:00	Speeding	HSR		1	1	KA01XY9999	Arun Kumar	Ravi Shankar	1000	Paid	
2	13	2025-11-04 12:14:00	Seatbelt Violation	HSR		1	3	KA01XY9999	Arun Kumar	Manoj Kumar	500	Paid	
3	12	2025-11-04 10:56:00	Mobile Usage	Sarjapur,Bangalore	4	1	9	KA19PQ1122	Sneha Patil	Ravi Shankar	1000	Paid	
4	11	2025-10-03 11:40:00	Underage Driving	NH66, Kochi		1	2	None	KA01XY9999	Arun Kumar	Lakshmi Rao	2000	Appealed
5	10	2025-10-02 19:15:00	No Insurance	Marine Drive, Kochi	8	5	None	KL09ZZ9876	Priya Menon	Fatima Noor	2300	Unpaid	
6	9	2025-10-02 17:50:00	Seatbelt Violation	Riverfront Road, Ahmedabad		7	3	None	GJ06AB9912	Ankit Bhatt	Manoj Kumar	500	Paid
7	8	2025-10-01 15:20:00	Mobile Usage	Calangute Beach Road, Goa	6	1	None	GA09NM7654	Sara Fernandes	Ravi Shankar	1000	Paid	
8	7	2025-10-01 09:00:00	Red Light Jump	Ashram Chowk, Delhi		5	2	None	DL02CL4321	Vishal Singh	Lakshmi Rao	500	Paid
9	6	2025-10-01 08:15:00	Seatbelt Violation	NH8, Delhi		5	1	None	DL02CL4321	Vishal Singh	Ravi Shankar	500	Paid

+ Record Violation

Figure 7.15 : List of violations in the database.

Penalty & ViolationType_ID auto-created by triggers!

Vehicle*	Officer*
GJ06AB9912 (Ankit Bhatt)	ID 2: Lakshmi Rao
Type*	Date
Signal Jump	2025/11/07
Location*	Time
Marathahalli,Bangalore	14:00

Record Violation

Figure 7.16 : Record violation and add into Violation database.

The screenshot shows a web application interface for managing data. At the top, there are navigation links for Drivers, Vehicles, Officers, Violations, Payments, and Appeals. The 'Payments' link is currently active, indicated by a red underline. Below the navigation, there is a heading 'Payments' and a table displaying a list of penalties. The table columns include Penalty_ID, Amount, Status, DueDate, Type, Date_Time, Driver, and License_plate. The data in the table is as follows:

Penalty_ID	Amount	Status	DueDate	Type	Date_Time	Driver	License_plate	
0	24	1500	Unpaid	2025-12-07	Signal Jump	2025-11-07 14:00:00	Ankit Bhatt	GJ06AB9912
1	23	1500	Unpaid	2025-12-07	Signal Jump	2025-11-07 14:00:00	Ankit Bhatt	GJ06AB9912
2	22	1500	Unpaid	2025-12-07	Signal Jump	2025-11-07 14:00:00	Ankit Bhatt	GJ06AB9912
3	21	2500	Unpaid	2025-12-04	Drunk Driving	2025-11-04 12:37:00	Priya Menon	KL09ZZ876
4	20	1000	Paid	2025-12-04	Speeding	2025-11-04 12:33:00	Arun Kumar	KA01XY9999
5	19	500	Paid	2025-12-04	Seatbelt Violation	2025-11-04 12:14:00	Arun Kumar	KA01XY9999
6	18	1000	Paid	2025-12-04	Mobile Usage	2025-11-04 10:56:00	Sneha Patil	KA19PQ1122
7	11	2000	Appealed	2025-12-04	Underage Driving	2025-10-03 11:40:00	Arun Kumar	KA01XY9999
8	10	2300	Unpaid	2025-12-04	No Insurance	2025-10-02 19:15:00	Priya Menon	KL09ZZ876
9	9	500	Paid	2025-12-04	Seatbelt Violation	2025-10-02 17:50:00	Ankit Bhatt	GJ06AB9912

Below the table, there is a form titled 'Pay Penalty'. It includes fields for 'Select Penalty*' (showing 'ID 24: Ankit Bhatt - Signal Jump - ₹1500.00'), 'Payment Mode' (set to 'Cash'), and a 'Pay Now' button.

Figure 7.17 : Payments Table with option to pay when signed as the Viewer.

The screenshot shows a web application interface for managing data. At the top, there are navigation links for Drivers, Vehicles, Officers, Violations, Payments, and Appeals. The 'Appeals' link is currently active, indicated by a red underline. Below the navigation, there is a heading 'Appeals' and a table displaying a list of appeals. The table columns include Appeal_ID, Datefiled, Status, Reason, Driver, Type, and Penalty_Status. The data in the table is as follows:

Appeal_ID	Datefiled	Status	Reason	Driver	Type	Penalty_Status
0	2025-11-04	Accepted	error in camera detection	Arun Kumar	Speeding	Paid
1	2025-11-04	Rejected	I didnt do this	Sneha Patil	Underage Driving	Unpaid
2	2025-11-04	Rejected	sorry I was lazy	Arun Kumar	Seatbelt Violation	Paid
3	2025-11-04	Rejected	I didnt do this	Priya Menon	Drunk Driving	Unpaid
4	2025-10-21	Accepted	Age incorrect in records	Priya Menon	Underage Driving	Appealed
5	2025-10-18	Rejected	Insurance in process	Priya Menon	No Insurance	Unpaid
6	2025-09-19	Rejected	Wrong parking sign	Meera Nair	Parking Violation	Unpaid
7	2025-09-15	Pending	Emergency situation	Rohit Sharma	Drunk Driving	Appealed

Below the table, there is a form titled 'File Appeal (Admin Only)'. It includes fields for 'Select Violation*' (showing 'ID 16: Ankit Bhatt - Signal Jump on 2025-11-07 - ₹1500.00') and 'Reason* (min 10 chars)', with a 'Submit Appeal' button.

Figure 7.18 : Uses have an option to file an appeal with reason.

The screenshot shows a web application interface for managing appeals. At the top, there's a navigation bar with links like Home, manage data, and reports analytics. The main title is "Appeals". Below it, a sub-section titled "File appeal → Penalty: Appealed (by trigger)" is shown. A table lists 8 appeals with columns: Appeal_ID, Datefiled, Status, Reason, Driver, Type, and Penalty_Status. The rows show various reasons like "error in camera detection", "I didn't do this", etc., and drivers like Arun Kumar, Priya Menon, etc. Below this, a modal window titled "Process Appeal (Admin Only)" is open, showing a dropdown "Select Pending Appeal" with "ID 1: Rohit Sharma - Drunk Driving - Reason: Emergency situation..." selected. It also has a "Full Reason: Emergency situation" input field, a "Decision" section with radio buttons for "Accept" (selected) and "Reject", and a "Finalize Decision" button.

Figure 7.19 : Admin can reject/accept the appeal request.
The status in Penalty table gets automatically updated - to Paid if appeal is accepted .To unpaid if appeal is rejected by Admin.

The screenshot shows the "Reports & Analytics Dashboard". The main title is "Reports & Analytics Dashboard". Below it, a sub-section titled "Driver Violation History" is shown. It has a search bar "Enter Driver ID" with value "1" and a button "Get Violation History". Below this, a table lists driver violations with columns: Driver_ID, Name, Vehicle_ID, Violation_ID, Violation_Type, Date_Time, Location, and Penalty_Amount. The rows show multiple violations for Driver ID 1, Arun Kumar, at various locations like HSR, NH66, Kochi, Brigade Road, Bangalore, MG Road, Bangalore, etc., with penalties ranging from 500 to 1500.

Figure 7.20: The admin can view the analytics dashboard.

City-wise Violation Summary (JOIN + Aggregate)

	City	Total_Violations	Total_Penalties	Unpaid_Count
0	Bangalore	4	5500	3
1	Bangalore	2	2000	0
2	Delhi	2	1000	0
3	Kochi	2	4300	1
4	HSR	2	1500	0
5	Mysore Palace Road	1	1000	1
6	Mangalore Port Road	1	750	0
7	Hubli NH Road	1	1200	1
8	Goa	1	1000	0
9	Ahmedabad	1	500	0

Figure 7.21: City wise violation summary.

Total Unpaid Fines per Driver

Enter Driver ID for Total Fines

2

Calculate Fines

	Driver_ID	Name	Total_Unpaid_Fines
0	2	Meera Nair	1000

Figure 7.22: Total unpaid fines per driver.

The screenshot shows a dark-themed web application interface titled "Reports & Analytics Dashboard". On the left sidebar, there are links for "Home", "manage data", and "reports analytics" (which is highlighted). The main content area has a header "Most Frequent Violation Type (Function)" with a sub-section "Most Common Violation: Signal Jump". Below this is a table titled "Nested Query – Drivers with Multiple Violations" showing the following data:

	Driver_ID	Name	Total_Violations
0	1	Arun Kumar	5
1	4	Sneha Patil	2
2	5	Vishal Singh	2
3	7	Ankit Bhatt	4
4	8	Priya Menon	3

Figure 7.23: Drivers with multiple violations

The screenshot shows the same dark-themed web application interface as Figure 7.23. The main content area has a header "Custom SQL Reports" with a sub-section "Detailed Violation Report (JOIN Query)". Below this is a table showing a detailed violation report using a JOIN query:

	Violation_ID	Date_Time	Violation_Type	Location	Driver_Name	License_no	License_plate	Model
0	16	2025-11-07 14:00:00	Signal Jump	Marathahalli,Bangalore	Ankit Bhatt	GJ05RT7721	GJ06AB9912	Honda Amaze
1	17	2025-11-07 14:00:00	Signal Jump	Marathahalli,Bangalore	Ankit Bhatt	GJ05RT7721	GJ06AB9912	Honda Amaze
2	18	2025-11-07 14:00:00	Signal Jump	Marathahalli,Bangalore	Ankit Bhatt	GJ05RT7721	GJ06AB9912	Honda Amaze
3	15	2025-11-04 12:37:00	Drunk Driving	Sarjapur	Priya Menon	KL07ZE9982	KL09ZZ9876	Hyundai Creta
4	14	2025-11-04 12:33:00	Speeding	HSR	Arun Kumar	KA05AB1234	KA01XY9999	Hyundai i20
5	13	2025-11-04 12:14:00	Seatbelt Violation	HSR	Arun Kumar	KA05AB1234	KA01XY9999	Hyundai i20
6	12	2025-11-04 10:56:00	Mobile Usage	Sarjapur,Bangalore	Sneha Patil	KA25GH1122	KA19PQ1122	Toyota Innova
7	11	2025-10-03 11:40:00	Underage Driving	NH66, Kochi	Arun Kumar	KA05AB1234	KA01XY9999	Hyundai i20
8	10	2025-10-02 19:15:00	No Insurance	Marine Drive, Kochi	Priya Menon	KL07ZE9982	KL09ZZ9876	Hyundai Creta
9	9	2025-10-02 17:50:00	Seatbelt Violation	Riverfront Road, Ahmeda	Ankit Bhatt	GJ05RT7721	GJ06AB9912	Honda Amaze

Figure 7.24 :A detailed report using Join query

10. CONCLUSION

The Traffic Violation Management System successfully addresses the complex requirements of modern traffic enforcement administration. The project demonstrates proficiency in:

Database Design:

- Well-normalized schema (3NF compliance)
- Proper use of constraints and foreign keys
- Efficient indexing on frequently queried columns

Implementation:

- Comprehensive DDL and DML statements
- Complex queries demonstrating JOIN, AGGREGATE, and NESTED operations
- Robust stored procedures for business logic
- Automated triggers for data consistency

Business Value:

- Centralized traffic violation tracking
- Automated penalty calculation and status updates
- Efficient payment and appeal processing
- Data-driven insights through analytics

FUTURE SCOPE

- **AI-Powered Automation:** Automated violation detection via AI cameras, OCR, and real-time license plate recognition for contactless enforcement, eliminating manual entry.
- **Mobile Application:** Android/iOS apps for drivers to check history, make digital payments, file online appeals, and receive instant notifications (SMS/email).
- **Government Database Integration:** Connects with national vehicle and driver databases for automated verification, cross-referencing, and comprehensive driver history tracking across jurisdictions.
- **Predictive Analytics:** Machine learning models identify high-risk violation hotspots and peak times, enabling proactive, data-driven traffic management.

REFERENCES

1. MySQL Documentation (Version 8.0). <https://dev.mysql.com/doc/>
2. Streamlit Documentation. <https://docs.streamlit.io/>
3. Elmasri, Ramez, and Navathe, Shamkant B. Fundamentals of Database Systems. Pearson Education

GITHUB LINK:

<https://github.com/Neonishh/traffic-guard>