PUNJAB UNIVERSITY COLLEGE OF INFORMATION TECHNOLOGY

Sir Asif Sohail

# Database System

**Project:** Crime Reporting System

## Group Members


- Asadullah Ahmad (BCSF23M020)
- Shahram Munsaf (BCSF23M023)
- Muazzam Ali (BCSF23M047)

## 1. Introduction to the Working of the Crime Reporting System

The Crime Reporting System is designed to make it easier for citizens to report crimes and for police officers to investigate and manage those cases. It ensures that all data related to crime reports, cases, and investigations is stored in one place, making the process efficient and transparent for everyone involved.

Here's how the system works:

1. **Users and Roles**
    - There are three types of users in the system: **Citizens**, **Police Officers**, and **Admins**.
    - **Citizens** can report crimes, track the status of their reports, and give feedback.
    - **Police Officers** are responsible for handling cases, gathering evidence, and managing suspects.
    - **Admins** oversee the system and ensure that everything runs smoothly.

    **Table Used:**

    - **Users Table:** Stores information about users, including their names, roles, contact details, and addresses.
2. **Reporting Crimes**
    - Citizens can report crimes by providing details such as the type of crime, where it happened, and a description of what occurred.
    - Each report is assigned a unique ID and tracked as it moves through different stages, like "Pending," "In Progress," or "Resolved."

    **Table Used:**

    - **CrimeReport Table:** Stores details of each crime report, including its description, location, and status.
3. **Police Stations and Officers**
    - Police officers are assigned to specific police stations. When a case is created, it is linked to an officer for investigation.

    **Tables Used:**

    - **PoliceStation Table:** Keeps records of police stations, their addresses, and contact details.
    - **PoliceOfficer Table:** Stores information about officers, including their station, rank, and badge number.
4. **Managing Cases**
    - Every crime report leads to the creation of a case. Cases are handled by police officers, who investigate and update their progress.
    - Officers can mark cases as "Open," "Under Investigation," or "Closed."

    **Table Used:**

- **Case Table:** Manages details of cases, such as the officer handling them, their status, and resolution dates.

5. **Evidence Collection**
   - During investigations, police officers can collect evidence like photos, videos, documents, or physical items. All evidence is linked to a case for proper tracking.

   **Table Used:**

   - **Evidence Table:** Stores details of evidence collected for cases, including its type and collection date.

6. **Suspect Management**
   - Suspects involved in cases are tracked in the system. Police officers can record their information, roles in the crime (e.g., witness, suspect, or accused), and any statements they provide.

   **Tables Used:**

   - **Suspect Table:** Stores personal details of suspects.
   - **SuspectCase Table:** Links suspects to cases and records their roles in the crime.

7. **Feedback from Citizens**
   - After their reports are handled, citizens can share feedback by rating the service and leaving comments. This helps improve the system and ensures accountability.

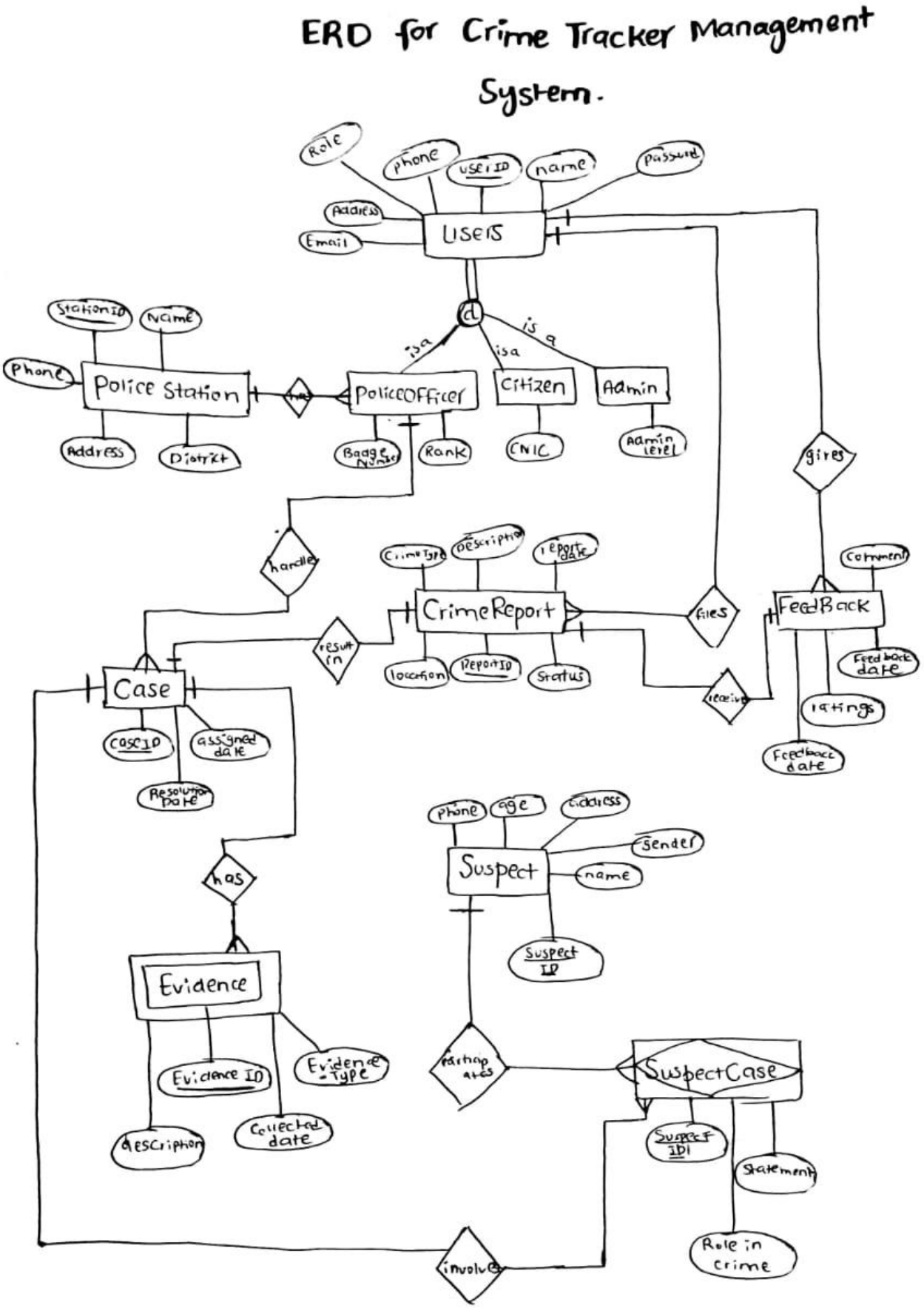
   **Table Used:**

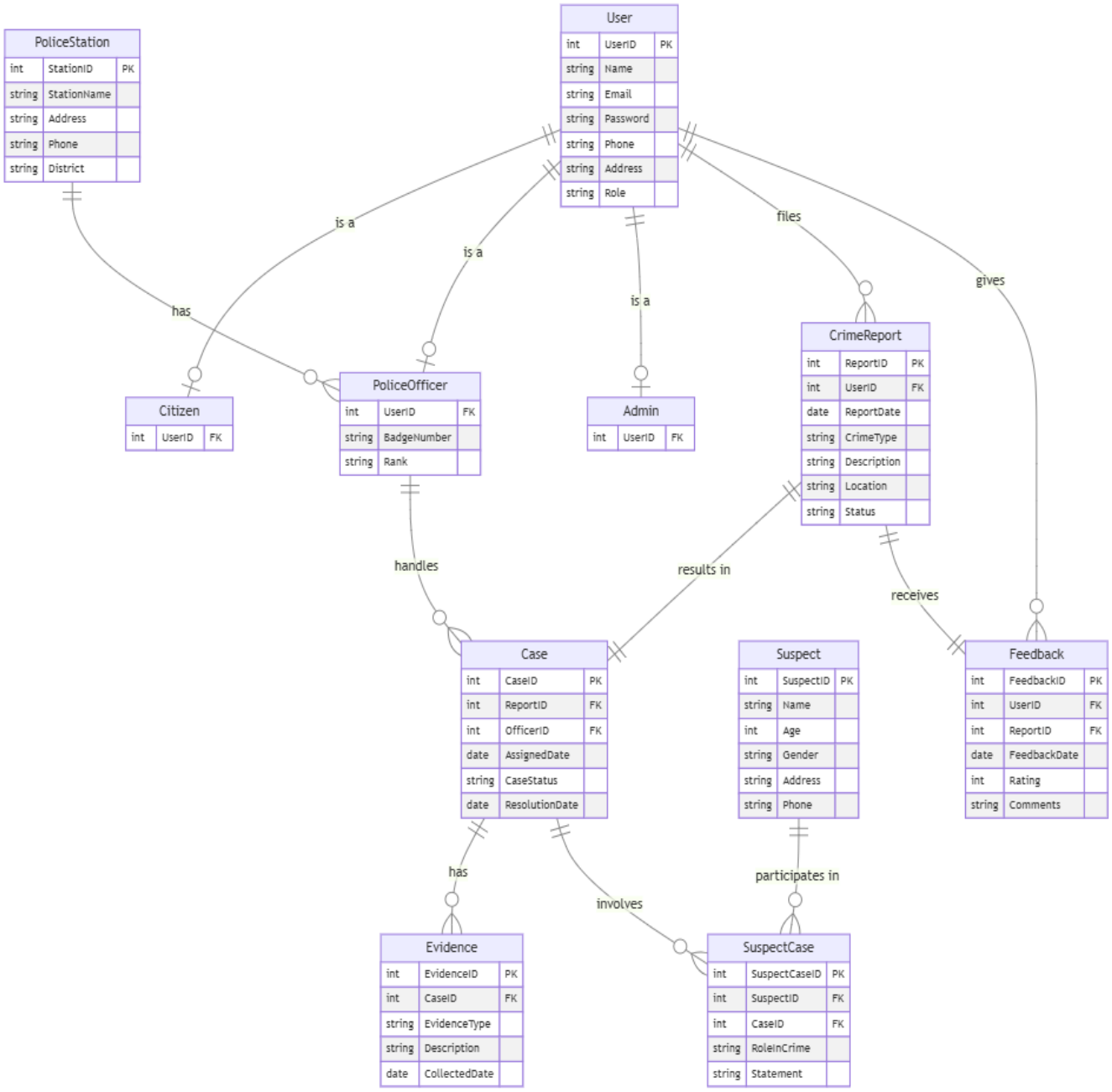
   - **Feedback Table:** Stores feedback from citizens, including their ratings and comments.

**Summary of Tables and Their Uses**

| Table Name | Purpose |
|---|---|
| **Users** | Stores details of all users (citizens, police, admins). |
| **CrimeReport** | Keeps records of all crime reports filed by citizens. |
| **PoliceStation** | Stores information about police stations and their locations. |
| **PoliceOfficer** | Tracks details of police officers, their ranks, and assignments. |
| **Case** | Manages cases created from crime reports and their progress. |
| **Evidence** | Records evidence linked to specific cases. |
| **Suspect** | Stores personal details of suspects involved in cases. |
| **SuspectCase** | Links suspects to cases and records their role in the crime. |
| **Feedback** | Stores feedback from citizens about the handling of their reports. |

## 2. ERD of the system




PoliceStation
int StationID PK
string StationName
string Address
string Phone
string District
User
int UserID PK
string Name
string Email
string Password
string Phone
string Address
string Role
is a
is a
is a
files
gives
has
Citizen
int UserID FK
PoliceOfficer
int UserID FK
string BadgeNumber
string Rank
Admin
int UserID FK
CrimeReport
int ReportID PK
int UserID FK
date ReportDate
string CrimeType
string Description
string Location
string Status
handles
results in
receives
Case
int CaseID PK
int ReportID FK
int OfficerID FK
date AssignedDate
string CaseStatus
date ResolutionDate
Suspect
int SuspectID PK
string Name
int Age
string Gender
string Address
string Phone
Feedback
int FeedbackID PK
int UserID FK
int ReportID FK
date FeedbackDate
int Rating
string Comments
has
involves
participates in
Evidence
int EvidenceID PK
int CaseID FK
string EvidenceType
string Description
date CollectedDate
SuspectCase
int SuspectCaseID PK
int SuspectID FK
int CaseID FK
string RoleInCrime
string Statement

## 3. Construction of the Relational Schema by using both bottom-up approach and top-down approach. The schema must clearly mention the relational keys.

**Relational Schema Construction Using Top-Down Approach**

- Users table, common attributes for all user types

  **Users** (**UserID**, Name, Email, Password, Phone, Address, Role)

- Subtypes with specific attributes

  **Citizen** (UserID, CNIC)

  **PoliceOfficer** (UserID, BadgeNumber, Rank, StationID)

  **Admin** (UserID, AdminLevel)

- Other tables

  **CrimeReport** (**ReportID**, UserID, ReportDate, CrimeType, Description, Location, Status)

  **PoliceStation** (**StationID**, StationName, Address, Phone, District)

  **Case** (**CaseID**, ReportID, OfficerID, AssignedDate, CaseStatus, ResolutionDate)

  **Evidence** (**EvidenceID**, CaseID, EvidenceType, Description, CollectedDate)

  **Suspect** (**SuspectID**, Name, Age, Gender, Address, Phone)

  **SuspectCase** (**SuspectCaseID**, SuspectID, CaseID, RoleInCrime, Statement)

  **Feedback** (**FeedbackID**, UserID, ReportID, FeedbackDate, Rating, Comments)

**Relational Schema Construction Using Bottom-Up Approach**

**Step 1: Initial Relation (Before Normalization)**

AllAttributes (UserID, Name, Email, Password, Phone, Address, Role, CNIC, BadgeNumber, Rank, StationID, AdminLevel, ReportID, ReportDate, CrimeType, Description, Location, Status, StationName, StationAddress, StationPhone, District, CaseID, OfficerID, AssignedDate, CaseStatus, ResolutionDate, EvidenceID, EvidenceType, EvidenceDescription, CollectedDate, SuspectID, SuspectName, SuspectAge, SuspectGender, SuspectAddress, SuspectPhone, SuspectCaseID, RoleInCrime, Statement, FeedbackID, FeedbackDate, Rating, Comments)

**Step 2: Convert to 1NF (First Normal Form)**

All the attributes seem to be atomic, so there is no issue with non-atomic data.We need to ensure that there are no repeating groups or multi-valued attributes. In this schema, no attribute appears to have multiple values, so this relation is already in **1NF**.

**Step 3: Convert to 2NF (Second Normal Form)**

- **Users** (UserID, Name, Email, Password, Phone, Address, Role)
- **Citizen** (UserID, CNIC)
- **PoliceOfficer** (UserID, BadgeNumber, Rank, StationID)
- **Admin** (UserID, AdminLevel)
- **CrimeReport** (ReportID, UserID, ReportDate, CrimeType, Description, Location, Status)
- **PoliceStation** (StationID, StationName, StationAddress, StationPhone, District)
- **Case** (CaseID, ReportID, OfficerID, AssignedDate, CaseStatus, ResolutionDate)
- **Evidence** (EvidenceID, CaseID, EvidenceType, EvidenceDescription, CollectedDate)
- **Suspect** (SuspectID, SuspectName, SuspectAge, SuspectGender, SuspectAddress, SuspectPhone)
- **SuspectCase** (SuspectCaseID, SuspectID, CaseID, RoleInCrime, Statement)
- **Feedback** (FeedbackID, UserID, ReportID, FeedbackDate, Rating, Comments)

**Primary Keys**:

- Users (UserID)
- Citizen (UserID)
- PoliceOfficer (UserID)
- Admin (UserID)
- CrimeReport (ReportID)
- PoliceStation (StationID)
- Case (CaseID)
- Evidence (EvidenceID)
- Suspect (SuspectID)
- SuspectCase (SuspectCaseID)
- Feedback (FeedbackID)

**Step 4: Convert to 3NF (Third Normal Form)**

Our relations already seem to be free of transitive dependencies based on the breakdown above.

**Final Schema After Bottom-Up Approach (Normalized):**

1. **Users** (**UserID**, Name, Email, Password, Phone, Address, Role)
2. **Citizen** (**UserID**, CNIC)
3. **PoliceOfficer** (**UserID**, BadgeNumber, Rank, StationID)
4. **Admin** (**UserID**, AdminLevel)
5. **CrimeReport** (**ReportID**, UserID, ReportDate, CrimeType, Description, Location, Status)
6. **PoliceStation** (**StationID**, StationName, StationAddress, StationPhone, District)
7. **Case** (**CaseID**, ReportID, OfficerID, AssignedDate, CaseStatus, ResolutionDate)
8. **Evidence** (**EvidenceID**, CaseID, EvidenceType, EvidenceDescription, CollectedDate)
9. **Suspect** (**SuspectID**, SuspectName, SuspectAge, SuspectGender, SuspectAddress, SuspectPhone)
10. **SuspectCase** (**SuspectCaseID**, SuspectID, CaseID, RoleInCrime, Statement)
11. **Feedback** (**FeedbackID**, UserID, ReportID, FeedbackDate, Rating, Comments)

## 4. Description of the relations in the following format:

**Table: Users**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| UserID | Varchar2 | 10 | Primary Key, Not Null |
| Name | Varchar2 | 50 | Not Null |
| Email | Varchar2 | 100 | Not Null, Unique |
| Password | Varchar2 | 50 | Not Null |
| Phone | Varchar2 | 15 | Not Null |
| Address | Varchar2 | 255 | Not Null |
| Role | Varchar2 | 10 | Not Null, Check ('Citizen', 'PoliceOfficer', 'Admin') |

**Table: Citizen (Subtype of Users)**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| UserID | Varchar2 | 10 | Primary Key, Foreign Key (Users), Not Null |
| CNIC | Varchar2 | 15 | Not Null, Unique |

**Table: PoliceOfficer (Subtype of Users)**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| UserID | Varchar2 | 10 | Primary Key, Foreign Key (Users), Not Null |
| BadgeNumber | Varchar2 | 20 | Not Null |
| Rank | Varchar2 | 20 | Not Null |
| StationID | Varchar2 | 10 | Foreign Key (PoliceStation), Not Null |

**Table: Admin (Subtype of Users)**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| UserID | Varchar2 | 10 | Primary Key, Foreign Key (Users), Not Null |
| AdminLevel | Varchar2 | 10 | Not Null |

**Table: CrimeReport**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| ReportID | Varchar2 | 10 | Primary Key, Not Null |
| UserID | Varchar2 | 10 | Foreign Key (Users), Not Null |
| ReportDate | Date | - | Not Null |
| CrimeType | Varchar2 | 50 | Not Null |
| Description | Varchar2 | 255 | Not Null |
| Location | Varchar2 | 100 | Not Null |
| Status | Varchar2 | 15 | Not Null, Check ('Pending', 'InProgress', 'Resolved') |

**Table: PoliceStation**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| StationID | Varchar2 | 10 | Primary Key, Not Null |
| StationName | Varchar2 | 100 | Not Null |
| Address | Varchar2 | 255 | Not Null |
| Phone | Varchar2 | 15 | Not Null |
| District | Varchar2 | 50 | Not Null |

**Table: Case**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| CaseID | Varchar2 | 10 | Primary Key, Not Null |
| ReportID | Varchar2 | 10 | Foreign Key (CrimeReport), Not Null |
| OfficerID | Varchar2 | 10 | Foreign Key (PoliceOfficer), Not Null |
| AssignedDate | Date | - | Not Null |
| CaseStatus | Varchar2 | 20 | Not Null, Check ('Open', 'Under Investigation', 'Closed') |
| ResolutionDate | Date | - | - |

**Table: Evidence**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| EvidenceID | Varchar2 | 10 | Primary Key, Not Null |
| CaseID | Varchar2 | 10 | Foreign Key (Case), Not Null |
| EvidenceType | Varchar2 | 50 | Not Null |
| Description | Varchar2 | 255 | Not Null |
| CollectedDate | Date | - | Not Null |

**Table: Suspect**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| SuspectID | Varchar2 | 10 | Primary Key, Not Null |
| Name | Varchar2 | 50 | Not Null |
| Age | Integer | - | Not Null |
| Gender | Varchar2 | 10 | Not Null |
| Address | Varchar2 | 255 | Not Null |
| Phone | Varchar2 | 15 | Not Null |

**Table: SuspectCase**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| SuspectCaseID | Varchar2 | 10 | Primary Key, Not Null |
| SuspectID | Varchar2 | 10 | Foreign Key (Suspect), Not Null |
| CaseID | Varchar2 | 10 | Foreign Key (Case), Not Null |

| | | | |
|---|---|---|---|
| RoleInCrime | Varchar2 | 50 | Not Null |
| Statement | Varchar2 | 500 | Not Null |

**Table: Feedback**

| Attribute | Data Type | Size | Constraints |
|---|---|---|---|
| FeedbackID | Varchar2 | 10 | Primary Key, Not Null |
| UserID | Varchar2 | 10 | Foreign Key (Users), Not Null |
| ReportID | Varchar2 | 10 | Foreign Key (CrimeReport), Not Null |
| FeedbackDate | Date | - | Not Null |
| Rating | Integer | - | Not Null, Check (1 to 5) |
| Comments | Varchar2 | 255 | Not Null |

## 5. CREATE TABLE statements for all the relations of your system.

**Create the Users Table**

```
CREATE TABLE Users (
    UserID VARCHAR2(10) PRIMARY KEY,
    Name VARCHAR2(50) NOT NULL,
    Email VARCHAR2(100) NOT NULL UNIQUE,
    Password VARCHAR2(50) NOT NULL,
    Phone VARCHAR2(15) NOT NULL,
    Address VARCHAR2(255) NOT NULL,
    Role VARCHAR2(10) NOT NULL CHECK (Role IN ('Citizen', 'PoliceOfficer', 'Admin'))
);
```

**Create the Citizen Table (Subtype of Users)**

```
CREATE TABLE Citizen (
    UserID VARCHAR2(10) PRIMARY KEY,
    CNIC VARCHAR2(15) NOT NULL UNIQUE,
    FOREIGN KEY (UserID) REFERENCES Users (UserID)
);
```

**Create the PoliceOfficer Table (Subtype of Users)**

```
CREATE TABLE PoliceOfficer (
    UserID VARCHAR2(10) PRIMARY KEY,
    BadgeNumber VARCHAR2(20) NOT NULL,
    Rank VARCHAR2(20) NOT NULL,
    StationID VARCHAR2(10) NOT NULL,
    FOREIGN KEY (UserID) REFERENCES Users (UserID),
    FOREIGN KEY (StationID) REFERENCES PoliceStation (StationID)
);
```

**Create the Admin Table (Subtype of Users)**

```
CREATE TABLE Admin (
   UserID VARCHAR2(10) PRIMARY KEY,
   AdminLevel VARCHAR2(10) NOT NULL,
   FOREIGN KEY (UserID) REFERENCES Users (UserID)
);
```

**Create the CrimeReport Table**

```
CREATE TABLE CrimeReport (
   ReportID VARCHAR2(10) PRIMARY KEY,
   UserID VARCHAR2(10) NOT NULL,
   ReportDate DATE NOT NULL,
   CrimeType VARCHAR2(50) NOT NULL,
   Description VARCHAR2(255) NOT NULL,
   Location VARCHAR2(100) NOT NULL,
   Status VARCHAR2(15) NOT NULL CHECK (Status IN ('Pending', 'InProgress', 'Resolved')),
   FOREIGN KEY (UserID) REFERENCES Users (UserID)
);
```

**Create the PoliceStation Table**

```
CREATE TABLE PoliceStation (
   StationID VARCHAR2(10) PRIMARY KEY,
   StationName VARCHAR2(100) NOT NULL,
   Address VARCHAR2(255) NOT NULL,
   Phone VARCHAR2(15) NOT NULL,
   District VARCHAR2(50) NOT NULL
);
```

**Create the Case Table**

```
CREATE TABLE Case (
   CaseID VARCHAR2(10) PRIMARY KEY,
   ReportID VARCHAR2(10) NOT NULL,
   OfficerID VARCHAR2(10) NOT NULL,
   AssignedDate DATE NOT NULL,
   CaseStatus VARCHAR2(20) NOT NULL CHECK (CaseStatus IN ('Open', 'Under Investigation',
'Closed')),
   ResolutionDate DATE,
   FOREIGN KEY (ReportID) REFERENCES CrimeReport (ReportID),
   FOREIGN KEY (OfficerID) REFERENCES PoliceOfficer (UserID)
);
```

**Create the Evidence Table**

```
CREATE TABLE Evidence (
   EvidenceID VARCHAR2(10) PRIMARY KEY,
   CaseID VARCHAR2(10) NOT NULL,
   EvidenceType VARCHAR2(50) NOT NULL,
```

```
    Description VARCHAR2(255) NOT NULL,
    CollectedDate DATE NOT NULL,
    FOREIGN KEY (CaseID) REFERENCES Case (CaseID)
);
```

**Create the Suspect Table**

```
CREATE TABLE Suspect (
    SuspectID VARCHAR2(10) PRIMARY KEY,
    Name VARCHAR2(50) NOT NULL,
    Age INTEGER NOT NULL,
    Gender VARCHAR2(10) NOT NULL,
    Address VARCHAR2(255) NOT NULL,
    Phone VARCHAR2(15) NOT NULL
);
```

**Create the SuspectCase Table**

```
CREATE TABLE SuspectCase (
    SuspectCaseID VARCHAR2(10) PRIMARY KEY,
    SuspectID VARCHAR2(10) NOT NULL,
    CaseID VARCHAR2(10) NOT NULL,
    RoleInCrime VARCHAR2(50) NOT NULL,
    Statement VARCHAR2(500) NOT NULL,
    FOREIGN KEY (SuspectID) REFERENCES Suspect (SuspectID),
    FOREIGN KEY (CaseID) REFERENCES Case (CaseID)
);
```

**Create the Feedback Table**

```
CREATE TABLE Feedback (
    FeedbackID VARCHAR2(10) PRIMARY KEY,
    UserID VARCHAR2(10) NOT NULL,
    ReportID VARCHAR2(10) NOT NULL,
    FeedbackDate DATE NOT NULL,
    Rating INTEGER NOT NULL CHECK (Rating BETWEEN 1 AND 5),
    Comments VARCHAR2(255) NOT NULL,
    FOREIGN KEY (UserID) REFERENCES User (UsersID),
    FOREIGN KEY (ReportID) REFERENCES CrimeReport (ReportID)
);
```

## 6. Desing of at least two VIEWS, that you feel are the most important.

**1. View for Citizen's Crime Reports and Feedback**

```
CREATE VIEW CitizenCrimeFeedback AS

SELECT C.UserID, C.Name AS CitizenName, R.ReportID,R.ReportDate, R.CrimeType,R.Description AS CrimeDescription, R.Location, R.Status AS ReportStatus, F.FeedbackDate, F.Rating, F.Comments AS FeedbackComments FROM  Users C
```

JOIN  CrimeReport R ON C.UserID = R.UserID

LEFT JOIN  Feedback F ON R.ReportID = F.ReportID

WHERE C.Role = 'Citizen';

**2. View for Officer's Case Evidence**

CREATE VIEW OfficerCaseEvidenceView AS

SELECT  U.Name AS OfficerName, C.CaseID, C.ReportID, C.AssignedDate, C.CaseStatus, E.EvidenceID, E.EvidenceType, E.Description AS EvidenceDescription, E.CollectedDate

FROM Users U

JOIN PoliceOfficer P ON U.UserID = P.UserID

JOIN Case C ON P.UserID = C.OfficerID

LEFT JOIN  Evidence E ON C.CaseID = E.CaseID;

## 7. SELECT statement for at least five common reports to be generated by the system. The queries should include joins, subqueries, and other appropriate clauses as per requirement.

1. **Report of All Crime Reports with Citizen Feedback:**

SELECT R.ReportID, R.ReportDate, R.CrimeType, R.Description AS CrimeDescription, R.Status AS ReportStatus, F.FeedbackDate, F.Rating, F.Comments AS FeedbackComments FROM CrimeReport R

LEFT JOIN Feedback F ON R.ReportID = F.ReportID

WHERE R.Status = 'Resolved';

2. **Report of Evidence Collected for Each Case:**

SELECT C.CaseID, E.EvidenceID, E.EvidenceType, E.Description AS EvidenceDescription, E.CollectedDate FROM Case C

JOIN Evidence E ON C.CaseID = E.CaseID

WHERE C.CaseStatus = 'Under Investigation';

3. **Report of Citizens Who Have Provided Feedback for Their Crime Reports:**

SELECT U.Name AS CitizenName, R.CrimeType, R.Description AS CrimeDescription, F.FeedbackDate, F.Rating, F.Comments AS FeedbackComments FROM Users U

JOIN CrimeReport R ON U.UserID = R.UserID

LEFT JOIN Feedback F ON R.ReportID = F.ReportID

WHERE U.Role = 'Citizen' AND F.FeedbackDate IS NOT NULL;

4. **Report of Suspects Associated with a Specific Case:**

SELECT S.Name AS SuspectName, S.Age, S.Gender, SC.RoleInCrime,  SC.Statement FROM Suspect S

JOIN SuspectCase SC ON S.SuspectID = SC.SuspectID

WHERE SC.CaseID = (SELECT CaseID FROM Case WHERE ReportID = '001')

ORDER BY S.Name;

**5. Report of Citizens Who Gave Feedback on Their Crime Reports**

SELECT C.UserID, C.Name AS CitizenName, R.ReportID, R.CrimeType, R.Status AS ReportStatus, F.FeedbackDate, F.Rating, F.Comments AS FeedbackComments FROM Users C

JOIN CrimeReport R ON C.UserID = R.UserID

LEFT JOIN Feedback F ON R.ReportID = F.ReportID

WHERE C.Role = 'Citizen' AND R.ReportID IN (SELECT ReportID FROM Feedback WHERE FeedbackDate IS NOT NULL)

ORDER BY R.ReportDate DESC;

## 8. PL/SQL: Implementation of at least two functions, two stored procedures, and two database triggers.