SQL Coding Challenge – Crime Management

1. Create Database.

```
mysql> CREATE DATABASE CrimeManagement;
Query OK, 1 row affected (0.02 sec)
```

```
mysql> USE CrimeManagement;
Database changed
```

2. Create Tables.

Crime:

```
mysql> CREATE TABLE Crime (
    ->  CrimeID INT PRIMARY KEY,
    ->  IncidentType VARCHAR(255),
    ->  IncidentDate DATE,
    ->  Location VARCHAR(255),
    ->  Description TEXT,
    ->  Status VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

Victim:

```
mysql> CREATE TABLE Victim (
    ->  VictimID INT PRIMARY KEY,
    ->  CrimeID INT,
    ->  Name VARCHAR(255),
    ->  ContactInfo VARCHAR(255),
    ->  Injuries VARCHAR(255),
    ->  FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

Suspect:

```
mysql> CREATE TABLE Suspect (
    ->  SuspectID INT PRIMARY KEY,
    ->  CrimeID INT,
    ->  Name VARCHAR(255),
    ->  Description TEXT,
    ->  CriminalHistory TEXT,
    ->  FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

3. Insert Sample Records into tables.

Crime:

```
mysql> INSERT INTO Crime (CrimeID, IncidentType, IncidentDate, Location, Description, Status) VALUES
    -> (1, 'Robbery', '2023-09-15', '123 Main St, Cityville', 'Armed robbery at a convenience store', 'Open'),
    -> (2, 'Homicide', '2023-09-20', '456 Elm St, Townsville', 'Investigation into a murder case', 'Under Investigation'),
    -> (3, 'Theft', '2023-09-10', '789 Oak St, Villagetown', 'Shoplifting incident at a mall', 'Closed');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Victim:

```
mysql> INSERT INTO Victim (VictimID, CrimeID, Name, ContactInfo, Injuries) VALUES
    -> (1, 1, 'John Doe', 'johndoe@example.com', 'Minor injuries'),
    -> (2, 2, 'Jane Smith', 'janesmith@example.com', 'Deceased'),
    -> (3, 3, 'Alice Johnson', 'alicejohnson@example.com', 'None');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Suspect:

```
mysql> INSERT INTO Suspect (SuspectID, CrimeID, Name, Description, CriminalHistory) VALUES
    -> (1, 1, 'Robber 1', 'Armed and masked robber', 'Previous robbery convictions'),
    -> (2, 2, 'Unknown', 'Investigation ongoing', NULL),
    -> (3, 3, 'Suspect 1', 'Shoplifting suspect', 'Prior shoplifting arrests');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Tasks:

1. Select all open incidents.

```
mysql> SELECT * FROM Crime WHERE Status = 'Open';
+---------+--------------+--------------+----------------------+-----------------------------------+--------+
| CrimeID | IncidentType | IncidentDate | Location             | Description                       | Status |
+---------+--------------+--------------+----------------------+-----------------------------------+--------+
|       1 | Robbery      | 2023-09-15   | 123 Main St, Cityville | Armed robbery at a convenience store | Open   |
+---------+--------------+--------------+----------------------+-----------------------------------+--------+
1 row in set (0.00 sec)
```

2. Find the total number of incidents.

```
mysql> SELECT COUNT(*) AS TotalIncidents FROM Crime;
+----------------+
| TotalIncidents |
+----------------+
|              3 |
+----------------+
1 row in set (0.02 sec)
```

3. List all unique incident types.

```
mysql> SELECT DISTINCT IncidentType FROM Crime;
+--------------+
| IncidentType |
+--------------+
| Robbery      |
| Homicide     |
| Theft        |
+--------------+
3 rows in set (0.00 sec)
```

4. Retrieve incidents that occurred between '2023-09-01' and '2023-09-10'.

```
mysql> SELECT * FROM Crime WHERE IncidentDate BETWEEN '2023-09-01' AND '2023-09-10';
+---------+--------------+--------------+----------------------+-------------------------------+--------+
| CrimeID | IncidentType | IncidentDate | Location             | Description                   | Status |
+---------+--------------+--------------+----------------------+-------------------------------+--------+
|       3 | Theft        | 2023-09-10   | 789 Oak St, Villagetown | Shoplifting incident at a mall | Closed |
+---------+--------------+--------------+----------------------+-------------------------------+--------+
1 row in set (0.00 sec)
```

5. List persons involved in incidents in descending order of age.

```
mysql> SELECT Name, Age FROM Victim
    -> UNION
    -> SELECT Name, Age FROM Suspect
    -> ORDER BY Age DESC;
+---------------+------+
| Name          | Age  |
+---------------+------+
| Alice Johnson |   55 |
| Suspect 1     |   50 |
| Jane Smith    |   45 |
| Unknown       |   40 |
| John Doe      |   35 |
| Robber 1      |   30 |
+---------------+------+
6 rows in set (0.00 sec)
```

6. Find the average age of persons involved in incidents.

```
mysql> SELECT AVG(Age) AS AverageAge FROM (
    ->     SELECT Age FROM Victim
    ->     UNION
    ->     SELECT Age FROM Suspect
    -> ) AS AllAges;
+------------+
| AverageAge |
+------------+
|    42.5000 |
+------------+
1 row in set (0.00 sec)
```

7. List incident types and their counts, only for open cases.

```
mysql> SELECT IncidentType, COUNT(*) AS IncidentCount FROM Crime WHERE Status = 'Open' GROUP BY IncidentType;
+--------------+---------------+
| IncidentType | IncidentCount |
+--------------+---------------+
| Robbery      |             1 |
+--------------+---------------+
1 row in set (0.00 sec)
```

8. Find persons with names containing 'Doe'.

```
mysql> SELECT Name FROM Victim WHERE Name LIKE '%Doe%'
    -> UNION
    -> SELECT Name FROM Suspect WHERE Name LIKE '%Doe%';
+----------+
| Name     |
+----------+
| John Doe |
+----------+
1 row in set (0.00 sec)
```

9. Retrieve the names of persons involved in open cases and closed cases.

```
mysql> SELECT Name FROM Victim WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE Status = 'Open')
    -> UNION
    -> SELECT Name FROM Suspect WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE Status = 'Open')
    -> UNION
    -> SELECT Name FROM Victim WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE Status = 'Closed')
    -> UNION
    -> SELECT Name FROM Suspect WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE Status = 'Closed');
+---------------+
| Name          |
+---------------+
| John Doe      |
| Robber 1      |
| Alice Johnson |
| Suspect 1     |
+---------------+
4 rows in set (0.00 sec)
```

10. List incident types where there are persons aged 30 or 35 involved.

```
mysql> SELECT DISTINCT C.IncidentType from Crime C
    -> LEFT JOIN Victim V on V.CrimeID=C.CrimeID
    -> where V.age=30 or V.age=35 ;
+--------------+
| IncidentType |
+--------------+
| Robbery      |
+--------------+
1 row in set (0.00 sec)
```

11. Find persons involved in incidents of the same type as 'Robbery'.

```
mysql> SELECT Name
    -> FROM Victim
    -> WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE IncidentType = 'Robbery')
    -> UNION
    -> SELECT Name
    -> FROM Suspect
    -> WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE IncidentType = 'Robbery');
+----------+
| Name     |
+----------+
| John Doe |
| Robber 1 |
+----------+
2 rows in set (0.00 sec)
```

12. List incident types with more than one open case.

```
mysql> SELECT IncidentType, COUNT(*) AS OpenCases
    -> FROM Crime
    -> WHERE Status = 'Open'
    -> GROUP BY IncidentType
    -> HAVING COUNT(*) > 1;
Empty set (0.00 sec)
```

13. List all incidents with suspects whose names also appear as victims in other incidents.

```
mysql> SELECT C.*, V.Name AS VictimName, S.Name AS SuspectName
    -> FROM Crime C
    -> JOIN Victim V ON C.CrimeID = V.CrimeID
    -> JOIN Suspect S ON C.CrimeID = S.CrimeID AND V.Name = S.Name;
Empty set (0.00 sec)
```

14. Retrieve all incidents along with victim and suspect details.

```
mysql> SELECT C.*, V.Name AS VictimName, V.ContactInfo, V.Injuries, S.Name AS SuspectName, S.Description AS SuspectDescription, S.CriminalHistory
    -> FROM Crime C
    -> LEFT JOIN Victim V ON C.CrimeID = V.CrimeID
    -> LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID;
+---------+--------------+--------------+-------------------------+-----------------------------------+---------------------+---------------+---------------------------+-----------------+-------------+-----------------------+-----------------------------+
| CrimeID | IncidentType | IncidentDate | Location                | Description                       | Status              | VictimName    | ContactInfo               | Injuries        | SuspectName | SuspectDescription    | CriminalHistory             |
+---------+--------------+--------------+-------------------------+-----------------------------------+---------------------+---------------+---------------------------+-----------------+-------------+-----------------------+-----------------------------+
|       1 | Robbery      | 2023-09-15   | 123 Main St, Cityville  | Armed robbery at a convenience store | Open             | John Doe      | johndoe@example.com       | Minor injuries  | Robber 1    | Armed and masked robber | Previous robbery convictions |
|       2 | Homicide     | 2023-09-20   | 456 Elm St, Townsville  | Investigation into a murder case  | Under Investigation | Jane Smith    | janesmith@example.com     | Deceased        | Unknown     | Investigation ongoing | NULL                        |
|       3 | Theft        | 2023-09-10   | 789 Oak St, Villagetown | Shoplifting incident at a mall    | Closed              | Alice Johnson | alicejohnson@example.com  | None            | Suspect 1   | Shoplifting suspect   | Prior shoplifting arrests   |
+---------+--------------+--------------+-------------------------+-----------------------------------+---------------------+---------------+---------------------------+-----------------+-------------+-----------------------+-----------------------------+
3 rows in set (0.00 sec)
```

15. Find incidents where the suspect is older than any victim.

```
mysql> SELECT C.*
    -> FROM Crime C
    -> JOIN Suspect S ON C.CrimeID = S.CrimeID
    -> WHERE S.Age > ANY (SELECT Age FROM Victim WHERE CrimeID = C.CrimeID);
Empty set (0.00 sec)
```

16. Find suspects involved in multiple incidents:

```
mysql> SELECT SuspectID, Name, COUNT(CrimeID) AS IncidentCount
    -> FROM Suspect
    -> GROUP BY SuspectID, Name
    -> HAVING COUNT(CrimeID) > 1;
Empty set (0.00 sec)
```

17. List incidents with no suspects involved.

```
mysql> SELECT C.*
    -> FROM Crime C
    -> LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID
    -> WHERE S.Name = 'Unknown';
+---------+--------------+--------------+------------------------+----------------------------------+---------------------+
| CrimeID | IncidentType | IncidentDate | Location               | Description                      | Status              |
+---------+--------------+--------------+------------------------+----------------------------------+---------------------+
|       2 | Homicide     | 2023-09-20   | 456 Elm St, Townsville | Investigation into a murder case | Under Investigation |
+---------+--------------+--------------+------------------------+----------------------------------+---------------------+
1 row in set (0.00 sec)
```

18. List all cases where at least one incident is of type 'Homicide' and all other incidents are of type 'Robbery'.

```
mysql> SELECT C.*
    -> FROM Crime C
    -> WHERE IncidentType = 'Homicide'
    ->   AND NOT EXISTS (
    ->     SELECT 1
    ->     FROM Crime C2
    ->     WHERE C.CrimeID <> C2.CrimeID AND C2.IncidentType <> 'Robbery'
    -> );
Empty set (0.00 sec)
```

19. Retrieve a list of all incidents and the associated suspects, showing suspects for each incident, or 'No Suspect' if there are none.

```
mysql> SELECT c.CrimeID, c.IncidentType, COALESCE(s.Name, 'No Suspect') AS SuspectName
    -> FROM Crime c JOIN Suspect s ON c.CrimeID = s.CrimeID;
+---------+--------------+-------------+
| CrimeID | IncidentType | SuspectName |
+---------+--------------+-------------+
|       1 | Robbery      | Robber 1    |
|       2 | Homicide     | Unknown     |
|       3 | Theft        | Suspect 1   |
+---------+--------------+-------------+
3 rows in set (0.00 sec)
```

20. List all suspects who have been involved in incidents with incident types 'Robbery' or 'Assault'.

```
mysql> SELECT s.SuspectID, s.Name FROM Suspect s JOIN Crime c ON s.CrimeID = c.CrimeID
    -> WHERE c.IncidentType IN ('Robbery', 'Assault');
+-----------+----------+
| SuspectID | Name     |
+-----------+----------+
|         1 | Robber 1 |
+-----------+----------+
1 row in set (0.00 sec)
```