**CRIME MANAGEMENT CODING CHALLENGE**

CREATE DATABSE crime;

USE crime;

**Table Creation (Additionally add age column to victim and suspect table to perform query):**

CREATE TABLE Crime (

CrimeID INT PRIMARY KEY,

IncidentType VARCHAR(255),

IncidentDate DATE,

Location VARCHAR(255),

Description TEXT,

Status VARCHAR(20)

);

CREATE TABLE Victim (

VictimID INT PRIMARY KEY,

CrimeID INT,

Name VARCHAR (255),

Age INT,

ContactInfo VARCHAR(255),

Injuries VARCHAR(255),

FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID) ON DELETE CASCADE

);

CREATE TABLE Suspect (

SuspectID INT PRIMARY KEY,

CrimeID INT,

Name VARCHAR(255),

Age INT,

Description TEXT,

CriminalHistory TEXT,

FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID) ON DELETE CASCADE

);

**Insert Values to the Tables:**

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');

INSERT INTO Victim (VictimID, CrimeID, Name, Age, ContactInfo, Injuries)

VALUES

(1, 1, 'John Doe', 35, 'johndoe@example.com', 'Minor injuries'),

(2, 2, 'Jane Smith', 42, 'janesmith@example.com', 'Deceased'),

(3, 3, 'Alice Johnson', 29, 'alicejohnson@example.com', 'None');

INSERT INTO Suspect (SuspectID, CrimeID, Name, Age, Description, CriminalHistory)

VALUES

(1, 1, 'Robber 1', 40, 'Armed and masked robber', 'Previous robbery convictions'),

(2, 2, 'Unknown', NULL, 'Investigation ongoing', NULL),

(3, 3, 'Suspect 1', 31, 'Shoplifting suspect', 'Prior shoplifting arrests');

**1. Select all open incidents.**

SELECT \* FROM Crime WHERE Status = 'Open';

**2. Find the total number of incidents.**

SELECT COUNT(\*) AS TotalIncidents FROM Crime;

**3. List all unique incident types.**

SELECT DISTINCT IncidentType FROM Crime;

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

SELECT \* FROM Crime WHERE IncidentDate BETWEEN '2023-09-01' AND '2023-09-10';

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

SELECT Name, Age FROM Victim UNION

SELECT Name, Age FROM Suspect ORDER BY Age DESC;

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

SELECT AVG(Age) AS AvgAge FROM (SELECT Age FROM Victim WHERE Age IS NOT NULL UNION ALL SELECT Age FROM Suspect WHERE Age IS NOT NULL) AS AllPersons;

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

SELECT IncidentType, COUNT(\*) AS IncidentCount FROM Crime

WHERE Status = 'Open' GROUP BY IncidentType;

**8. Find persons with names containing 'Doe'.**

SELECT Name FROM Victim WHERE Name LIKE '%Doe%' UNION

SELECT Name FROM Suspect WHERE Name LIKE '%Doe%';

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

SELECT DISTINCT v.Name, c.Status FROM Victim v

JOIN Crime c ON v.CrimeID = c.CrimeID WHERE c.Status IN ('Open', 'Closed') UNION

SELECT DISTINCT s.Name, c.Status FROM Suspect s

JOIN Crime c ON s.CrimeID = c.CrimeID WHERE c.Status IN ('Open', 'Closed');

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

SELECT DISTINCT c.IncidentType FROM Crime c

JOIN Victim v ON c.CrimeID = v.CrimeID WHERE v.Age IN (30, 35) UNION

SELECT DISTINCT c.IncidentType FROM Crime c

JOIN Suspect s ON c.CrimeID = s.CrimeID WHERE s.Age IN (30, 35);

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

SELECT DISTINCT v.Name FROM Victim v

JOIN Crime c ON v.CrimeID = c.CrimeID WHERE c.IncidentType = 'Robbery' UNION

SELECT DISTINCT s.Name FROM Suspect s

JOIN Crime c ON s.CrimeID = c.CrimeID WHERE c.IncidentType = 'Robbery';

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

SELECT IncidentType FROM Crime WHERE Status = 'Open'

GROUP BY IncidentType HAVING COUNT(\*) > 1;

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

SELECT DISTINCT c.\* FROM Crime c

JOIN Suspect s ON c.CrimeID = s.CrimeID

JOIN Victim v ON s.Name = v.Name;

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

SELECT c.\*, v.Name AS VictimName, v.Age AS VictimAge, v.ContactInfo, v.Injuries,

s.Name AS SuspectName, s.Age AS SuspectAge, s.Description, s.CriminalHistory

FROM Crime c LEFT JOIN Victim v ON c.CrimeID = v.CrimeID

LEFT JOIN Suspect s ON c.CrimeID = s.CrimeID;

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

SELECT DISTINCT c.\* FROM Crime c

JOIN Suspect s ON c.CrimeID = s.CrimeID

JOIN Victim v ON c.CrimeID = v.CrimeID WHERE s.Age > v.Age;

**16. Find suspects involved in multiple incidents:**

SELECT Name, COUNT(DISTINCT CrimeID) AS IncidentCount

FROM Suspect GROUP BY Name

HAVING COUNT(DISTINCT CrimeID) > 1;

**17. List incidents with no suspects involved.**

SELECT \* FROM Crime WHERE CrimeID NOT IN

(SELECT DISTINCT CrimeID FROM Suspect);

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

SELECT \* FROM Crime WHERE CrimeID IN (SELECT CrimeID

FROM Crime WHERE IncidentType = 'Homicide')

AND NOT EXISTS (SELECT 1 FROM Crime

WHERE IncidentType NOT IN ('Homicide', 'Robbery'));

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

SELECT c.\*, COALESCE(s.Name, 'No Suspect') AS SuspectName FROM Crime c

LEFT JOIN Suspect s ON c.CrimeID = s.CrimeID;

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

SELECT DISTINCT s.Name FROM Suspect s JOIN Crime c

ON s.CrimeID = c.CrimeID

WHERE c.IncidentType IN ('Robbery', 'Assault');