# Coding Challenge SQL

**Crime Management Shema DDL and DML**

-- Create tables 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),

ContactInfo VARCHAR(255), Injuries VARCHAR(255),

FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)

);

CREATE TABLE Suspect ( SuspectID INT PRIMARY KEY,

CrimeID INT,

Name VARCHAR(255),

Description TEXT, CriminalHistory TEXT,

FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)

);

-- Insert sample data

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, 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');

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');

# Solve the below queries:

1. Select all open incidents.
2. Find the total number of incidents.
3. List all unique incident types.
4. Retrieve incidents that occurred between '2023-09-01' and '2023-09-10'.
5. List persons involved in incidents in descending order of age.
6. Find the average age of persons involved in incidents.
7. List incident types and their counts, only for open cases.
8. Find persons with names containing 'Doe'.
9. Retrieve the names of persons involved in open cases and closed cases.
10. List incident types where there are persons aged 30 or 35 involved.
11. Find persons involved in incidents of the same type as 'Robbery'.
12. List incident types with more than one open case.
13. List all incidents with suspects whose names also appear as victims in other incidents.
14. Retrieve all incidents along with victim and suspect details.
15. Find incidents where the suspect is older than any victim.
16. Find suspects involved in multiple incidents:
17. List incidents with no suspects involved.
18. List all cases where at least one incident is of type 'Homicide' and all other incidents are of type 'Robbery'.
19. Retrieve a list of all incidents and the associated suspects, showing suspects for each incident, or 'No Suspect' if there are none.
20. List all suspects who have been involved in incidents with incident types 'Robbery' or 'Assault'

ANSWERS

1.

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

2.

SELECT COUNT(\*) AS Total FROM Crime;

3.

SELECT DISTINCT IncidentType FROM Crime;

4.

SELECT \*

FROM Crime

WHERE IncidentDate >= '2023-09-01'

AND IncidentDate <= '2023-09-10';

5.

ALTER TABLE Victim

ADD COLUMN Age INT;

UPDATE Victim SET Age = 30 WHERE VictimID = 1;

UPDATE Victim SET Age = 45 WHERE VictimID = 2;

UPDATE Victim SET Age = 28 WHERE VictimID = 3;

ALTER TABLE Suspect

ADD COLUMN Age INT;

UPDATE Suspect SET Age = 30 WHERE SuspectID = 1;

UPDATE Suspect SET Age = 25 WHERE SuspectID = 2;

UPDATE Suspect SET Age = 35 WHERE SuspectID = 3;

SELECT VictimID, CrimeID, Name, ContactInfo, Injuries, Age, NULL AS Description, NULL AS CriminalHistory FROM Victim

UNION ALL

SELECT SuspectID, CrimeID, Name, NULL AS ContactInfo, NULL AS Injuries, Age, Description, CriminalHistory FROM Suspect

ORDER BY Age DESC;

6.

SELECT AVG(Age) AS AvgAge

FROM (

SELECT Age FROM Victim

UNION ALL

SELECT Age FROM Suspect

) AS CombinedAges;

7.

SELECT IncidentType, COUNT(\*) AS IncidentCount

FROM Crime

WHERE Status = 'Open'

GROUP BY IncidentType;

8.

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

9.

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');

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');

10.

SELECT DISTINCT c.IncidentType

FROM Crime c

JOIN (

SELECT CrimeID FROM Victim WHERE Age IN (30, 35)

UNION

SELECT CrimeID FROM Suspect WHERE Age IN (30, 35)

) AS PersonsInvolved ON c.CrimeID = PersonsInvolved.CrimeID;

11.

SELECT Name, TypeOfInvolvement

FROM (

SELECT Name, 'Victim' AS TypeOfInvolvement

FROM Victim v

INNER JOIN Crime c ON v.CrimeID = c.CrimeID

WHERE c.IncidentType = 'Robbery'

UNION ALL

SELECT Name, 'Suspect' AS TypeOfInvolvement

FROM Suspect s

INNER JOIN Crime c ON s.CrimeID = c.CrimeID

WHERE c.IncidentType = 'Robbery'

) AS PersonsInvolved;

12.

SELECT IncidentType

FROM Crime

WHERE Status = 'Open'

GROUP BY IncidentType

HAVING COUNT(\*) > 1;

13.

SELECT DISTINCT C.\*, S.Name AS SuspectName FROM Crime C

JOIN Suspect S ON C.CrimeID = S.CrimeID

JOIN Victim V ON C.CrimeID = V.CrimeID AND S.Name = V.Name;

14.

SELECT DISTINCT C.\*, S.Name AS SuspectName FROM Crime C

JOIN Suspect S ON C.CrimeID = S.CrimeID

JOIN Victim V ON C.CrimeID = V.CrimeID AND S.Name = V.Name;

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;

15.

SELECT C.IncidentType FROM Crime C

JOIN (SELECT CrimeID FROM Suspect WHERE Age > ANY (SELECT Age FROM Victim WHERE Victim.CrimeID = Suspect.CrimeID))

AS IncidentsWithOlderSuspects ON C.CrimeID = IncidentsWithOlderSuspects.CrimeID;

16.

SELECT Name, COUNT(\*) AS Incident\_Count FROM Suspect

INNER JOIN Crime ON Suspect.CrimeID = Crime.CrimeID GROUP BY Name HAVING COUNT(\*) > 1;

17.

SELECT \* FROM Crime

LEFT JOIN Suspect ON Crime.CrimeID = Suspect.CrimeID WHERE Suspect.SuspectID IS NULL;

18.

SELECT \*

FROM Crime

WHERE IncidentType = 'Homicide'

OR (

IncidentType = 'Robbery'

AND CrimeID NOT IN (

SELECT CrimeID

FROM Crime

WHERE IncidentType = 'Homicide'

)

);

19.

SELECT

c.CrimeID,

c.IncidentType,

c.IncidentDate,

c.Location,

c.Description,

c.Status,

COALESCE(s.Name, 'No Suspect') AS SuspectName,

COALESCE(s.Description, 'No Suspect Description') AS SuspectDescription,

COALESCE(s.CriminalHistory, 'No Suspect Criminal History') AS SuspectCriminalHistory

FROM

Crime c

LEFT JOIN

Suspect s ON c.CrimeID = s.CrimeID;’

20.

SELECT DISTINCT s.\*

FROM Suspect s

JOIN Crime c ON s.CrimeID = c.CrimeID

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