**SQL CODING CHALLENGE**

**TOPIC : CRIME MANAGEMENT SYSTEM**

**CREATING 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)

);

**INSERTING RECORDS :**

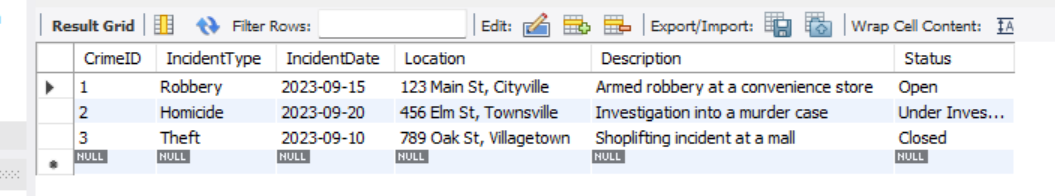
INSERT INTO Crime (CrimeID, IncidentType, IncidentDate, Location, Description, Status)

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');



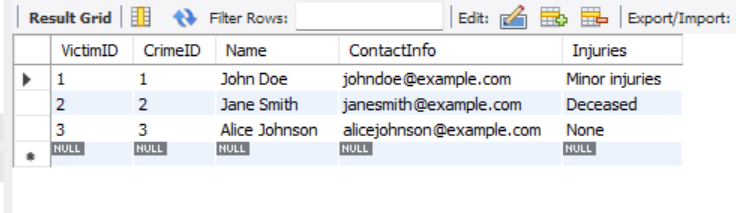
INSERT INTO Victim (VictimID, CrimeID, Name, ContactInfo, Injuries)

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)

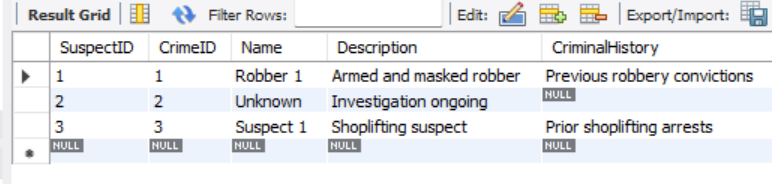
VALUES

(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');

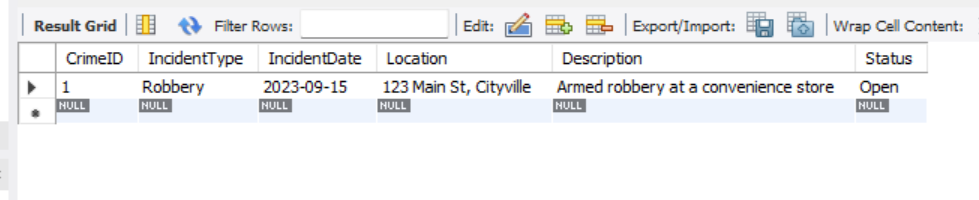
SELECT\*FROM Suspect;



QUERY 1:

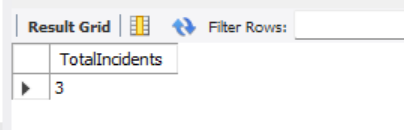
1. Select all open incidents.

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



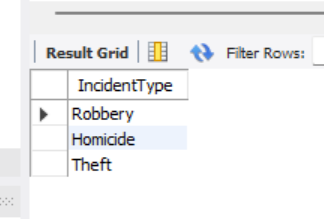
2. Find the total number of incidents.

SELECT COUNT(\*) AS TotalIncidents FROM Crime;



3. List all unique incident types.

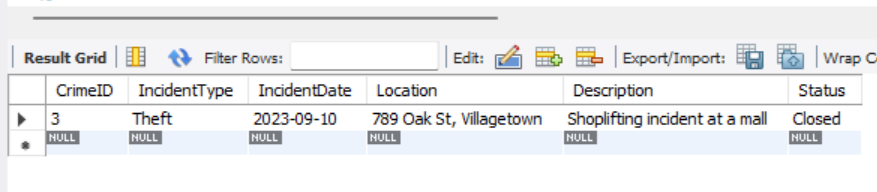
SELECT DISTINCT IncidentType FROM Crime;



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 VictimID AS PersonID, Name, Age, 'Victim' AS Role FROM Victim

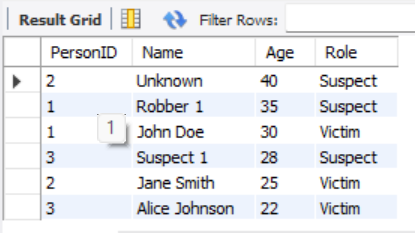
WHERE Age IS NOT NULL

UNION

SELECT SuspectID AS PersonID, Name, Age, 'Suspect' AS Role FROM Suspect

WHERE Age IS NOT NULL

ORDER BY Age DESC;



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

SELECT AVG(Age) AS Average\_Age

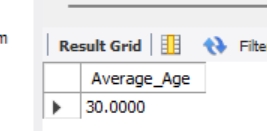
FROM (

SELECT Age FROM Victim WHERE Age IS NOT NULL

UNION ALL

SELECT Age FROM Suspect WHERE Age IS NOT NULL

) AS Combined;



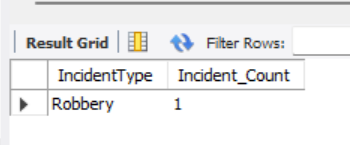
7. List incident types and their counts, only for open cases.

SELECT IncidentType, COUNT(\*) AS Incident\_Count

FROM Crime

WHERE Status = 'Open'

GROUP BY IncidentType;

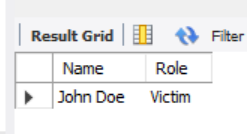


8. Find persons with names containing 'Doe'.

SELECT Name, 'Victim' AS Role FROM Victim WHERE Name LIKE '%Doe%'

UNION

SELECT Name, 'Suspect' AS Role FROM Suspect WHERE Name LIKE '%Doe%';



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

SELECT V.Name, 'Victim' AS Role, C.Status

FROM Victim V

JOIN Crime C ON V.CrimeID = C.CrimeID

WHERE C.Status IN ('Open', 'Closed')

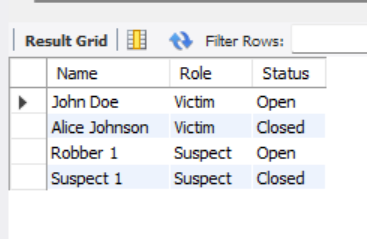
UNION

SELECT S.Name, 'Suspect' AS Role, 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)

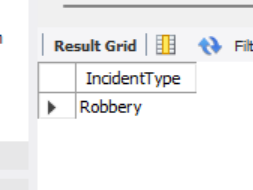
UNION

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 V.Name, 'Victim' AS Role, C.IncidentType

FROM Victim V

JOIN Crime C ON V.CrimeID = C.CrimeID

WHERE C.IncidentType = (SELECT IncidentType FROM Crime WHERE IncidentType = 'Robbery')

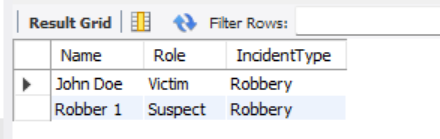
UNION

SELECT S.Name, 'Suspect' AS Role, C.IncidentType

FROM Suspect S

JOIN Crime C ON S.CrimeID = C.CrimeID

WHERE C.IncidentType = (SELECT IncidentType FROM Crime WHERE IncidentType = 'Robbery');



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

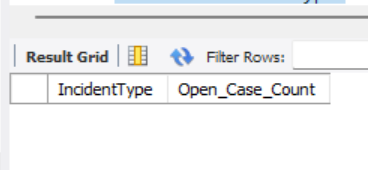
SELECT IncidentType, COUNT(\*) AS Open\_Case\_Count

FROM Crime

WHERE Status = 'Open'

GROUP BY IncidentType

HAVING COUNT(\*) > 1;



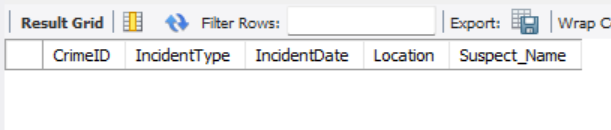
13. List all incidents with suspects whose names also appear as victims in other incidents.

SELECT DISTINCT C.CrimeID, C.IncidentType, C.IncidentDate, C.Location, S.Name AS Suspect\_Name

FROM Crime C

JOIN Suspect S ON C.CrimeID = S.CrimeID

WHERE S.Name IN (SELECT V.Name FROM Victim V);



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

SELECT C.CrimeID, C.IncidentType, C.IncidentDate, C.Location, C.Status,

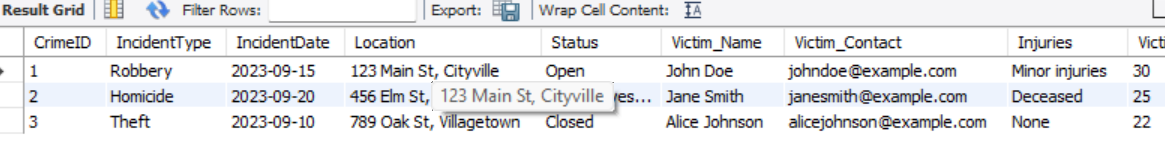
V.Name AS Victim\_Name, V.ContactInfo AS Victim\_Contact, V.Injuries, V.Age AS Victim\_Age,

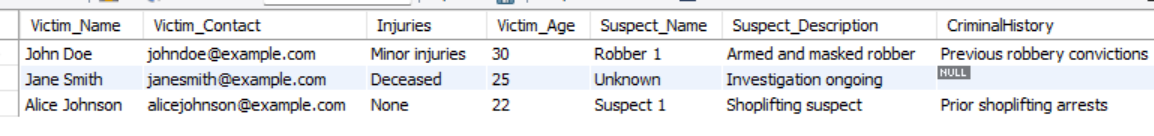
S.Name AS Suspect\_Name, S.Description AS Suspect\_Description, S.CriminalHistory, S.Age AS Suspect\_Age

FROM Crime C

LEFT JOIN Victim V ON C.CrimeID = V.CrimeID

LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID;





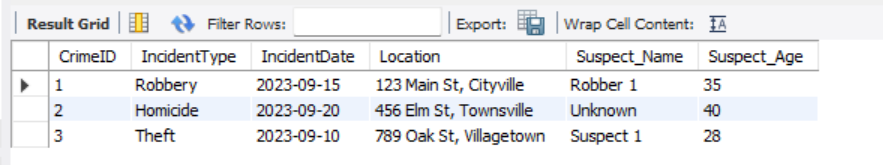
15. Find incidents where the suspect is older than any victim.

SELECT DISTINCT C.CrimeID, C.IncidentType, C.IncidentDate, C.Location, S.Name AS Suspect\_Name, S.Age AS Suspect\_Age

FROM Crime C

JOIN Suspect S ON C.CrimeID = S.CrimeID

WHERE S.Age > ALL (SELECT V.Age FROM Victim V WHERE V.CrimeID = C.CrimeID AND V.Age IS NOT NULL);



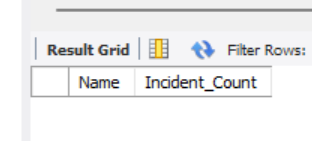
16. Find suspects involved in multiple incidents:

SELECT Name, COUNT(CrimeID) AS Incident\_Count

FROM Suspect

GROUP BY Name

HAVING COUNT(CrimeID) > 1;



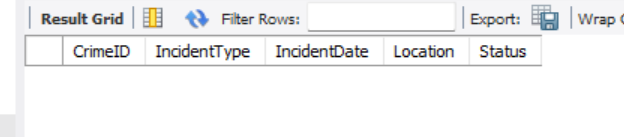
17. List incidents with no suspects involved.

SELECT C.CrimeID, C.IncidentType, C.IncidentDate, C.Location, C.Status

FROM Crime C

LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID

WHERE S.SuspectID IS NULL;



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

SELECT CrimeID, IncidentType, IncidentDate, Location, Status

FROM Crime

WHERE CrimeID IN (

SELECT CrimeID

FROM Crime

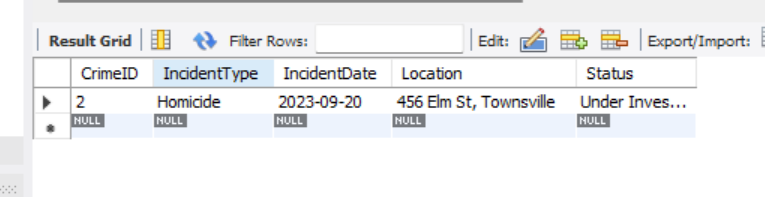
GROUP BY CrimeID

HAVING

SUM(CASE WHEN IncidentType = 'Homicide' THEN 1 ELSE 0 END) >= 1

AND SUM(CASE WHEN IncidentType NOT IN ('Homicide', 'Robbery') THEN 1 ELSE 0 END) = 0

);



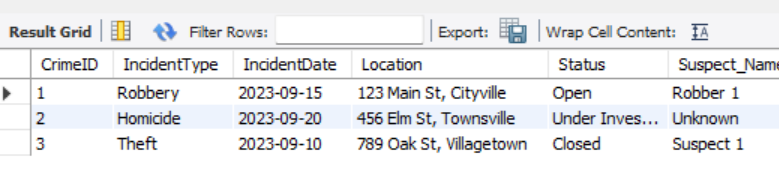
19. Retrieve a list of all incidents and the associated suspects, showing suspects for each incident, or 'No Suspect' if there are none.

SELECT C.CrimeID, C.IncidentType, C.IncidentDate, C.Location, C.Status,

COALESCE(S.Name, 'No Suspect') AS Suspect\_Name

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.SuspectID, S.Name, S.Description, S.CriminalHistory

FROM Suspect S

JOIN Crime C ON S.CrimeID = C.CrimeID

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

