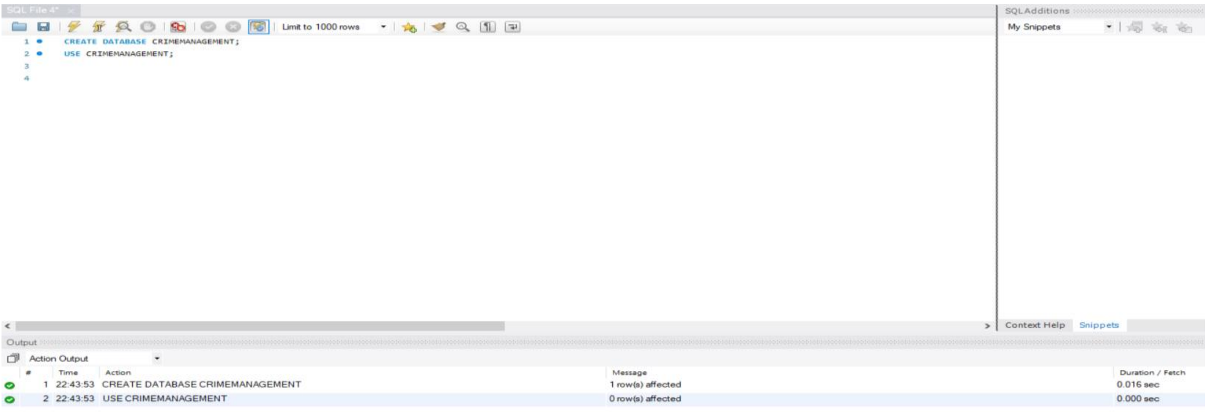
SQL CODING CHALLENGE – CRIME MANAGEMENT – M C Priya Dharsini

➢ Crime Management Database Setup

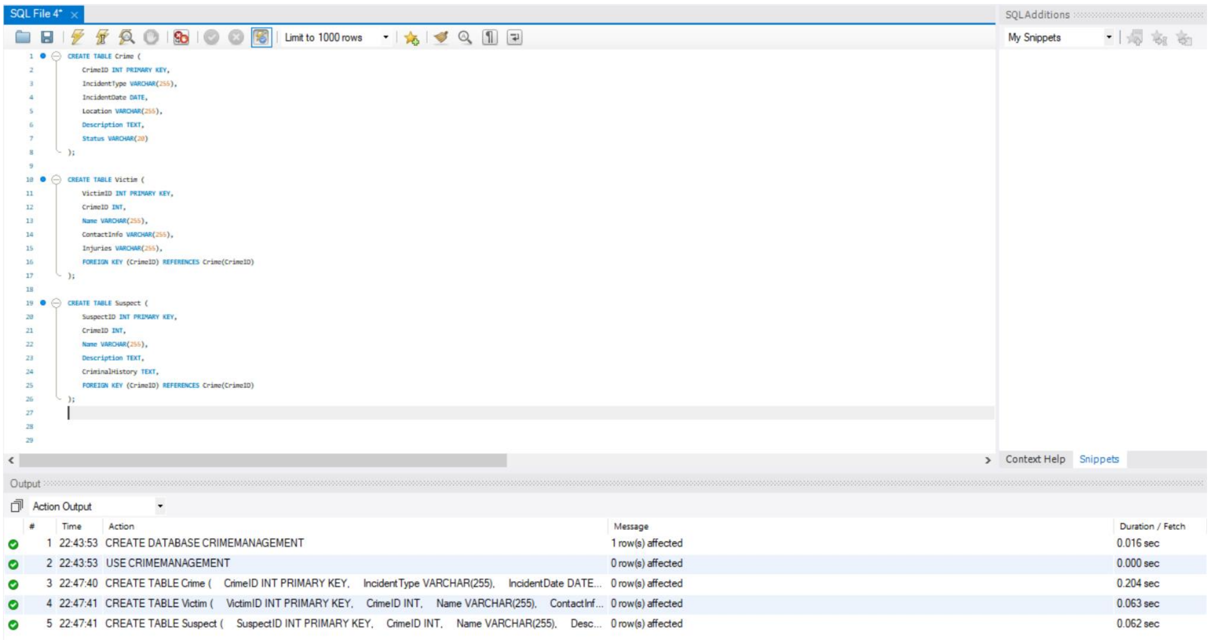
1. Creating a Database



```
CREATE DATABASE CRIMEMANAGEMENT;
USE CRIMEMANAGEMENT;
```

```
#  Time      Action                           Message            Duration / Fetch
1  22:43:53  CREATE DATABASE CRIMEMANAGEMENT  1 row(s) affected  0.016 sec
2  22:43:53  USE CRIMEMANAGEMENT              0 row(s) affected  0.000 sec
```

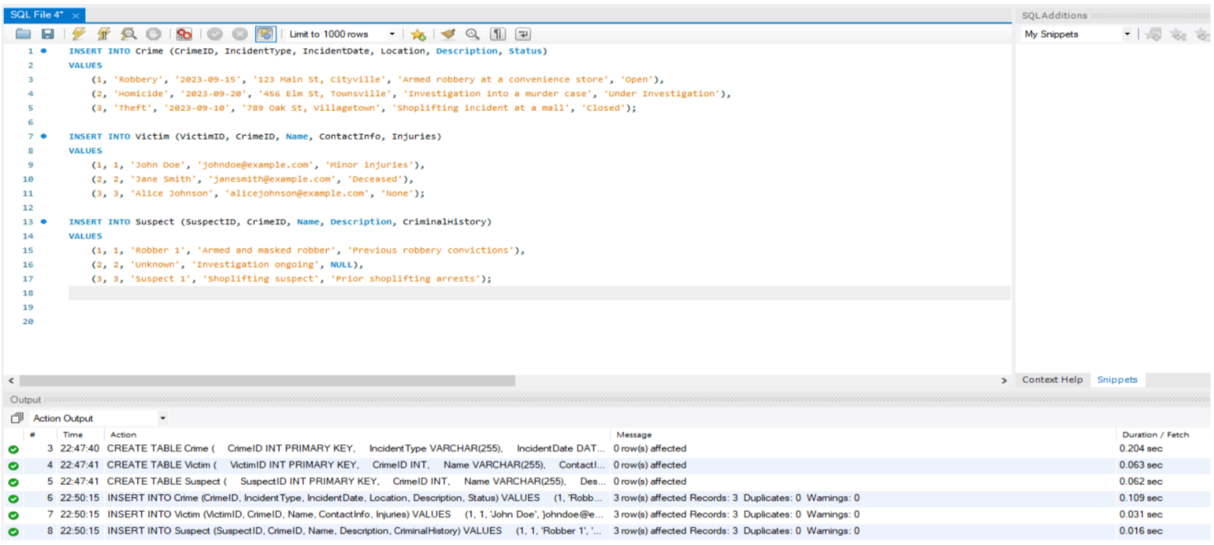
2. Creating Tables for crime management, including the **Crime**, **Victim**, and **Suspect** tables.



```
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)
);
```

```
#  Time      Action                                                                                   Message            Duration / Fetch
1  22:43:53  CREATE DATABASE CRIMEMANAGEMENT                                                          1 row(s) affected  0.016 sec
2  22:43:53  USE CRIMEMANAGEMENT                                                                      0 row(s) affected  0.000 sec
3  22:47:40  CREATE TABLE Crime (   CrimeID INT PRIMARY KEY,   IncidentType VARCHAR(255),   IncidentDate DATE...  0 row(s) affected  0.204 sec
4  22:47:41  CREATE TABLE Victim (   VictimID INT PRIMARY KEY,   CrimeID INT,   Name VARCHAR(255),   ContactInf...  0 row(s) affected  0.063 sec
5  22:47:41  CREATE TABLE Suspect (   SuspectID INT PRIMARY KEY,   CrimeID INT,   Name VARCHAR(255),   Desc...  0 row(s) affected  0.062 sec
```

3. Sample Data Insertion - **Crime**, **Victim**, and **Suspect** tables with sample data.



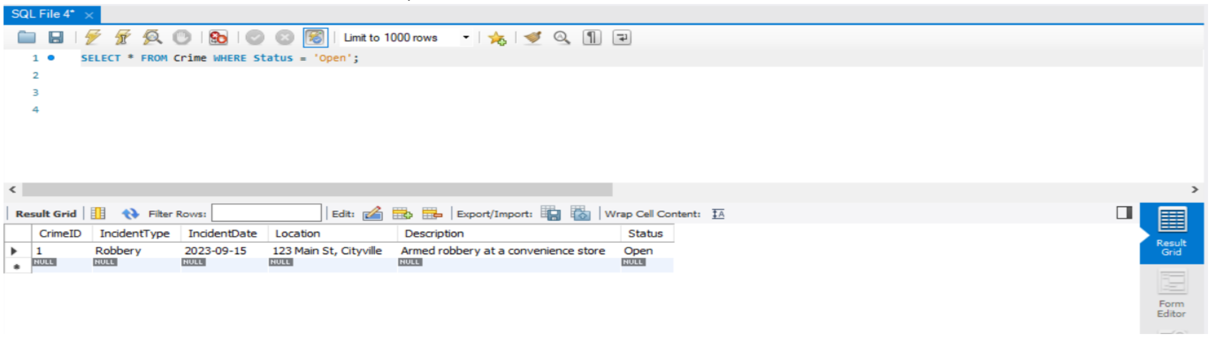
```
INSERT INTO Crime (CrimeID, IncidentType, IncidentDate, Location, Description, Status)
VALUES
    (1, 'Robbery', '2023-09-15', '123 Main St, Cityville', 'Armed robbery at a convenience store', 'Open'),
    (2, 'Homicide', '2023-09-20', '456 Elm St, Townsville', 'Investigation into a murder case', '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');
```

```
#  Time      Action                                                                                                   Message                                                    Duration / Fetch
3  22:47:40  CREATE TABLE Crime (   CrimeID INT PRIMARY KEY,   IncidentType VARCHAR(255),   IncidentDate DAT...             0 row(s) affected                                          0.204 sec
4  22:47:41  CREATE TABLE Victim (   VictimID INT PRIMARY KEY,   CrimeID INT,   Name VARCHAR(255),   ContactI...             0 row(s) affected                                          0.063 sec
5  22:47:41  CREATE TABLE Suspect (   SuspectID INT PRIMARY KEY,   CrimeID INT,   Name VARCHAR(255),   Des...                0 row(s) affected                                          0.062 sec
6  22:50:15  INSERT INTO Crime (CrimeID, IncidentType, IncidentDate, Location, Description, Status) VALUES   (1, 'Robb...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.109 sec
7  22:50:15  INSERT INTO Victim (VictimID, CrimeID, Name, ContactInfo, Injuries) VALUES   (1, 1, 'John Doe', 'johndoe@e...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.031 sec
8  22:50:15  INSERT INTO Suspect (SuspectID, CrimeID, Name, Description, CriminalHistory) VALUES   (1, 1, 'Robber 1', '...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.016 sec
```

➢ **SQL Queries for Analysis**

1. **Select all open incidents**

SELECT * FROM Crime WHERE Status = 'Open';



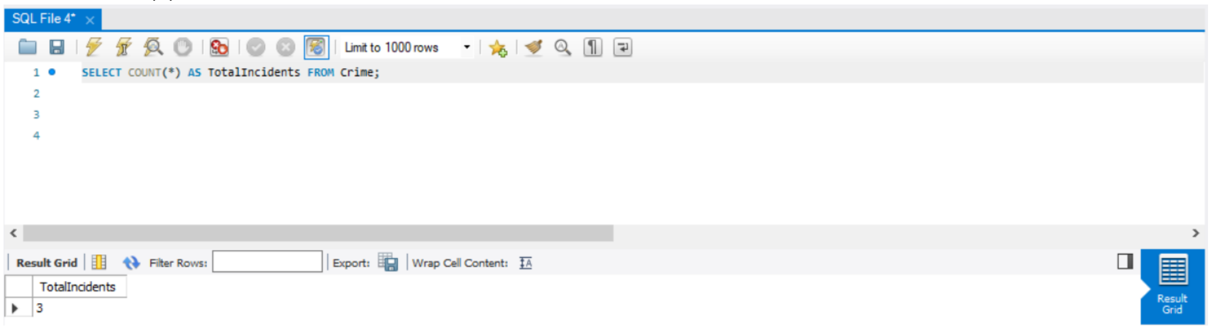
SQL File 4*

```sql
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
NULL	NULL	NULL	NULL	NULL	NULL

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

SELECT COUNT(*) AS TotalIncidents FROM Crime;



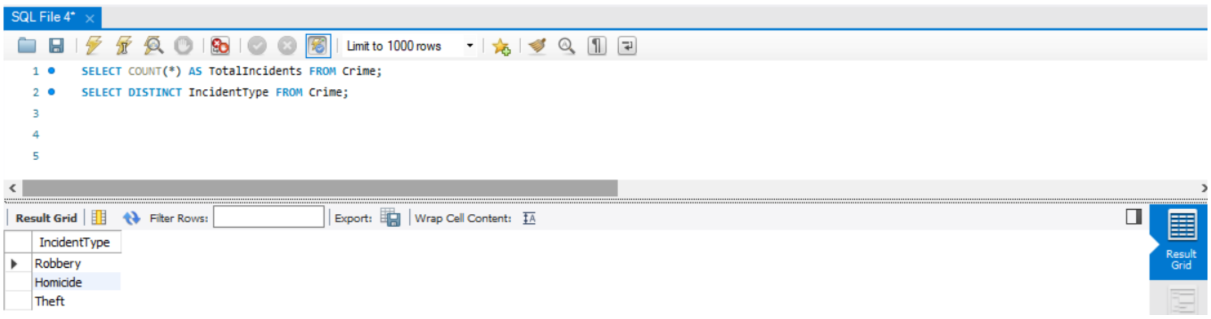
SQL File 4*

```sql
SELECT COUNT(*) AS TotalIncidents FROM Crime;
```

TotalIncidents
3

3. **List all unique incident types**

SELECT DISTINCT IncidentType FROM Crime;



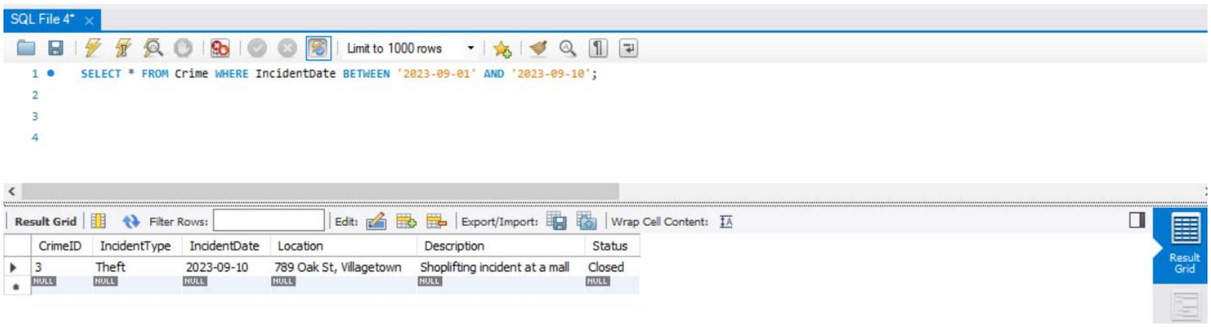
SQL File 4*

```sql
SELECT COUNT(*) AS TotalIncidents FROM Crime;
SELECT DISTINCT IncidentType FROM Crime;
```

IncidentType
Robbery
Homicide
Theft

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';



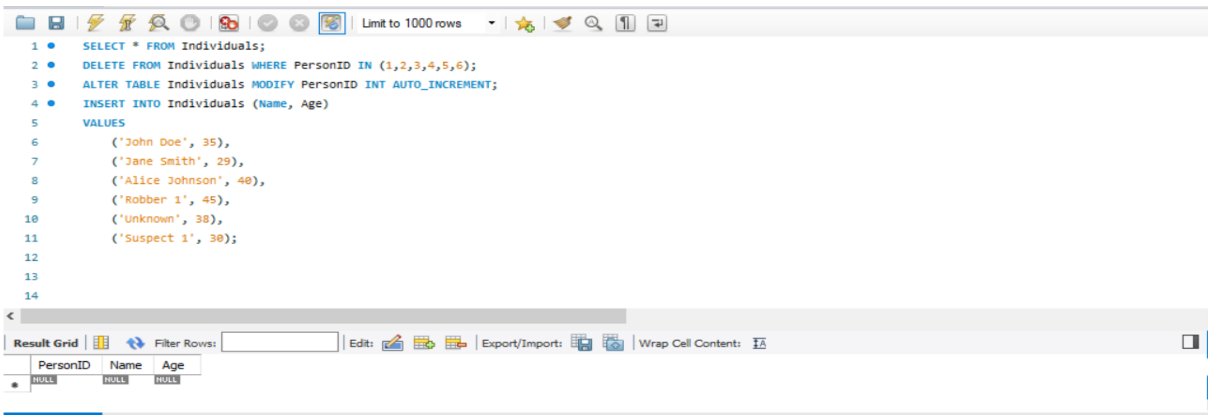
SQL File 4*

```sql
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
NULL	NULL	NULL	NULL	NULL	NULL

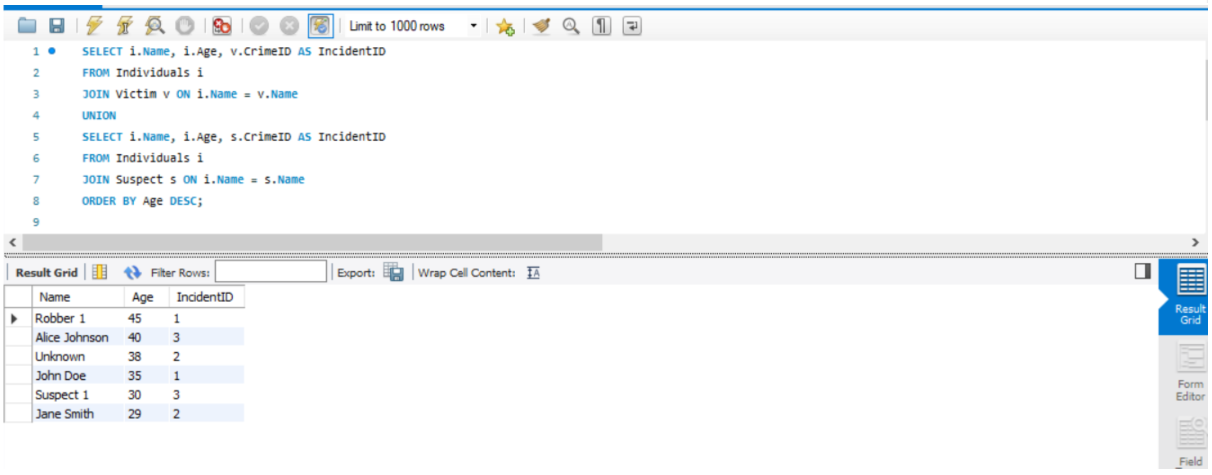
5. **List persons involved in incidents in descending order of age**

I first created an Individuals table to store personal details such as names and ages. This table was then linked to both the Victim and Suspect tables, allowing us to retrieve relevant information based on age.



```
SELECT * FROM Individuals;
DELETE FROM Individuals WHERE PersonID IN (1,2,3,4,5,6);
ALTER TABLE Individuals MODIFY PersonID INT AUTO_INCREMENT;
INSERT INTO Individuals (Name, Age)
VALUES
    ('John Doe', 35),
    ('Jane Smith', 29),
    ('Alice Johnson', 40),
    ('Robber 1', 45),
    ('Unknown', 38),
    ('Suspect 1', 30);
```

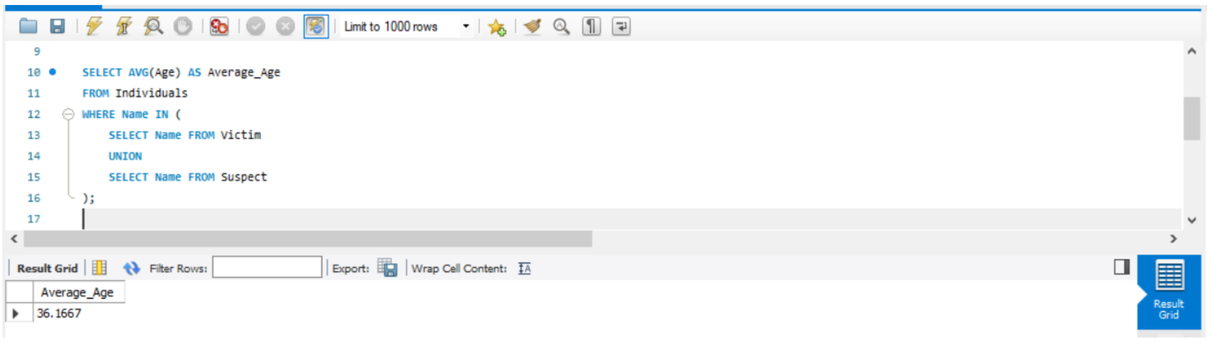
```
  PersonID  Name   Age
* NULL      NULL   NULL
```



```
SELECT i.Name, i.Age, v.CrimeID AS IncidentID
FROM Individuals i
JOIN Victim v ON i.Name = v.Name
UNION
SELECT i.Name, i.Age, s.CrimeID AS IncidentID
FROM Individuals i
JOIN Suspect s ON i.Name = s.Name
ORDER BY Age DESC;
```

```
  Name           Age   IncidentID
▶ Robber 1       45    1
  Alice Johnson  40    3
  Unknown        38    2
  John Doe       35    1
  Suspect 1      30    3
  Jane Smith     29    2
```

6. Find the average age of persons involved in incidents



```
SELECT AVG(Age) AS Average_Age
FROM Individuals
WHERE Name IN (
    SELECT Name FROM Victim
    UNION
    SELECT Name FROM Suspect
);
```

```
  Average_Age
▶ 36.1667
```

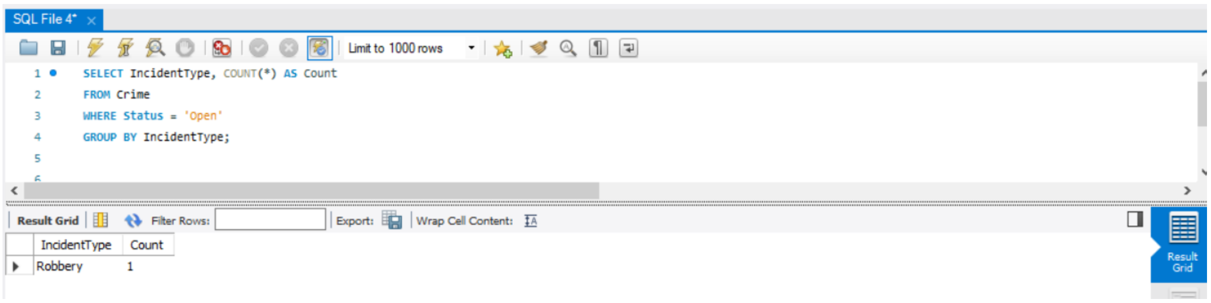
7. List incident types and their counts, only for open cases

SELECT IncidentType, COUNT(*) AS Count

FROM Crime

WHERE Status = 'Open'

GROUP BY IncidentType;

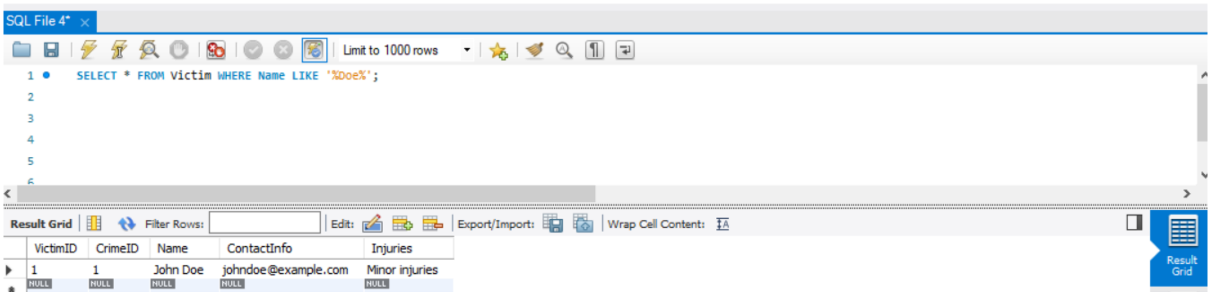


```
SELECT IncidentType, COUNT(*) AS Count
FROM Crime
WHERE Status = 'Open'
GROUP BY IncidentType;
```

```
  IncidentType  Count
▶ Robbery       1
```

8. Find persons with names containing 'Doe'

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



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

VictimID	CrimeID	Name	ContactInfo	Injuries
1	1	John Doe	johndoe@example.com	Minor injuries
NULL	NULL	NULL	NULL	NULL

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

SELECT Name FROM Victim

JOIN Crime ON Victim.CrimeID = Crime.CrimeID

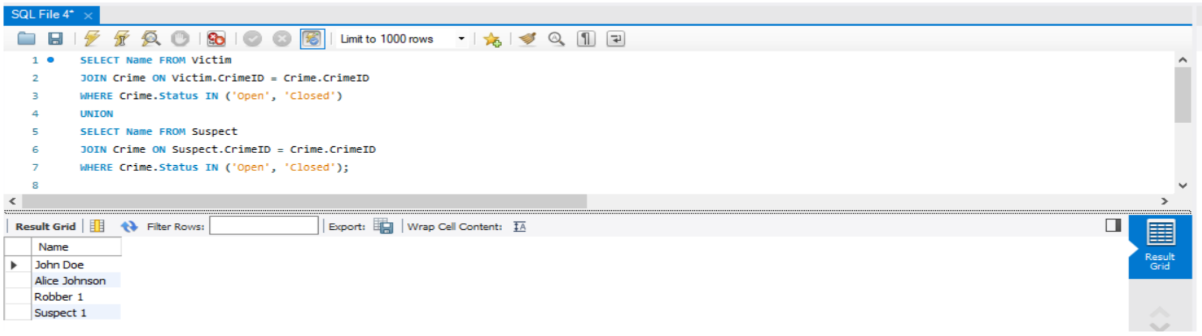
WHERE Crime.Status IN ('Open', 'Closed')

UNION

SELECT Name FROM Suspect

JOIN Crime ON Suspect.CrimeID = Crime.CrimeID

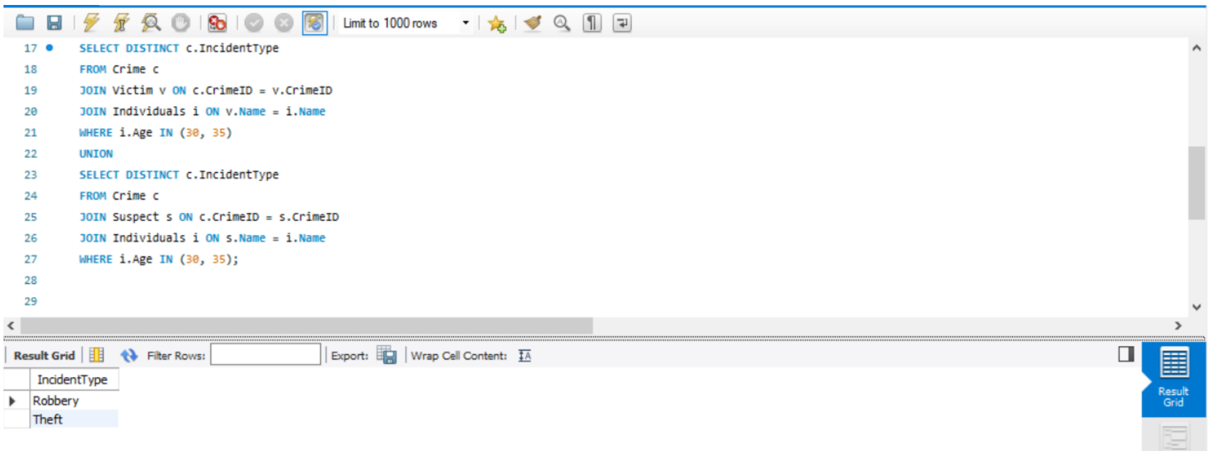
WHERE Crime.Status IN ('Open', 'Closed');



```
SELECT Name FROM Victim
JOIN Crime ON Victim.CrimeID = Crime.CrimeID
WHERE Crime.Status IN ('Open', 'Closed')
UNION
SELECT Name FROM Suspect
JOIN Crime ON Suspect.CrimeID = Crime.CrimeID
WHERE Crime.Status IN ('Open', 'Closed');
```

Name
John Doe
Alice Johnson
Robber 1
Suspect 1

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
JOIN Individuals i ON v.Name = i.Name
WHERE i.Age IN (30, 35)
UNION
SELECT DISTINCT c.IncidentType
FROM Crime c
JOIN Suspect s ON c.CrimeID = s.CrimeID
JOIN Individuals i ON s.Name = i.Name
WHERE i.Age IN (30, 35);
```

IncidentType
Robbery
Theft

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

SELECT Name FROM Victim

JOIN Crime ON Victim.CrimeID = Crime.CrimeID

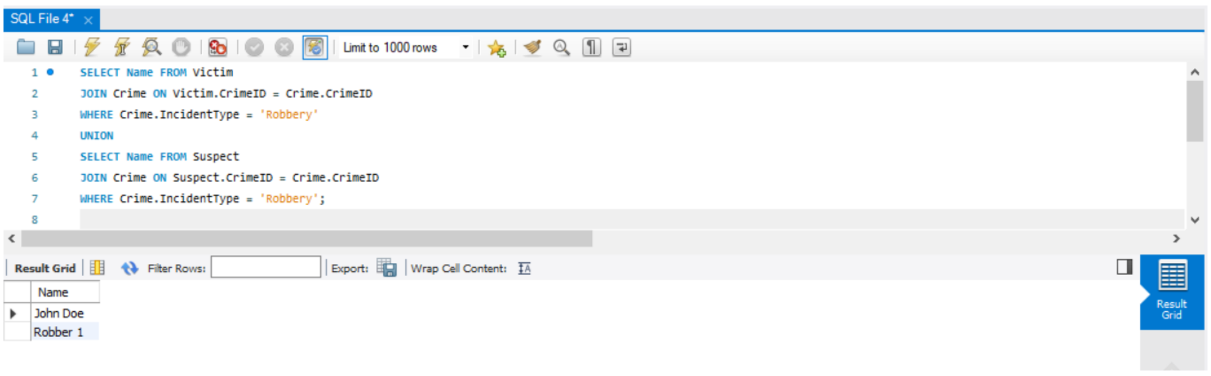
WHERE Crime.IncidentType = 'Robbery'

UNION

SELECT Name FROM Suspect

JOIN Crime ON Suspect.CrimeID = Crime.CrimeID

WHERE Crime.IncidentType = 'Robbery';



```
SELECT Name FROM Victim
JOIN Crime ON Victim.CrimeID = Crime.CrimeID
WHERE Crime.IncidentType = 'Robbery'
UNION
SELECT Name FROM Suspect
JOIN Crime ON Suspect.CrimeID = Crime.CrimeID
WHERE Crime.IncidentType = 'Robbery';
```

Name
John Doe
Robber 1

12. List incident types with more than one open case

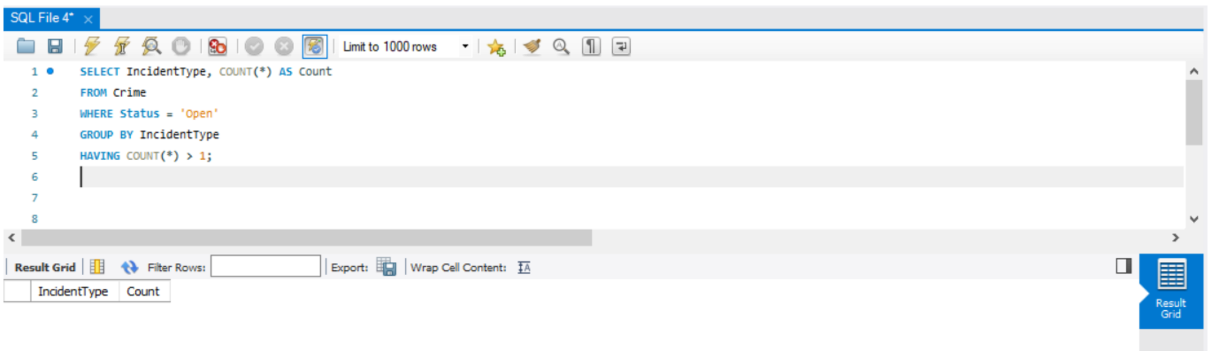
SELECT IncidentType, COUNT(*) AS Count

FROM Crime

WHERE Status = 'Open'

GROUP BY IncidentType

HAVING COUNT(*) > 1;



```
SELECT IncidentType, COUNT(*) AS Count
FROM Crime
WHERE Status = 'Open'
GROUP BY IncidentType
HAVING COUNT(*) > 1;
```

IncidentType	Count

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

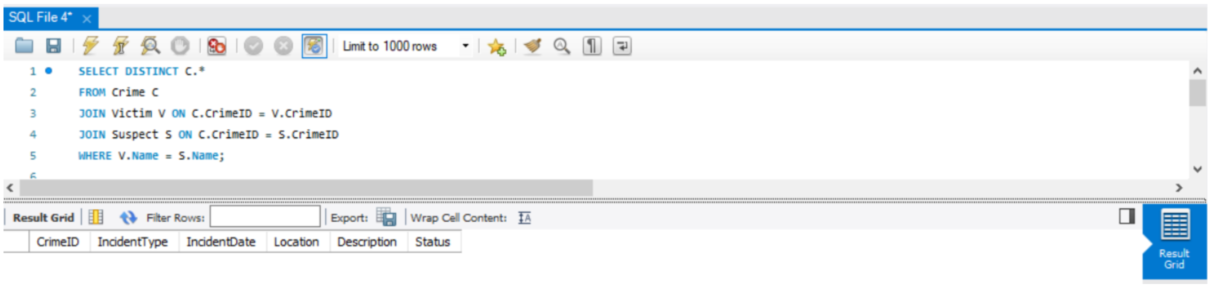
SELECT DISTINCT C.*

FROM Crime C

JOIN Victim V ON C.CrimeID = V.CrimeID

JOIN Suspect S ON C.CrimeID = S.CrimeID

WHERE V.Name = S.Name;



```
SELECT DISTINCT C.*
FROM Crime C
JOIN Victim V ON C.CrimeID = V.CrimeID
JOIN Suspect S ON C.CrimeID = S.CrimeID
WHERE V.Name = S.Name;
```

CrimeID	IncidentType	IncidentDate	Location	Description	Status

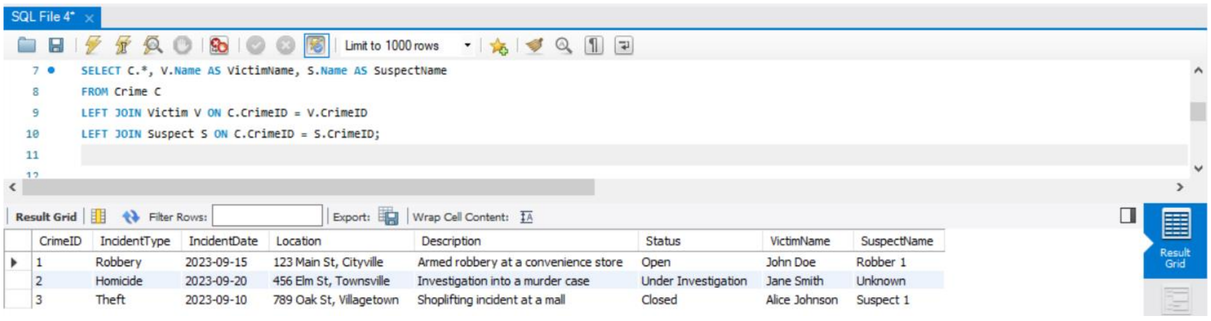
14. Retrieve all incidents along with victim and suspect details

SELECT C.*, V.Name AS VictimName, S.Name AS SuspectName

FROM Crime C

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

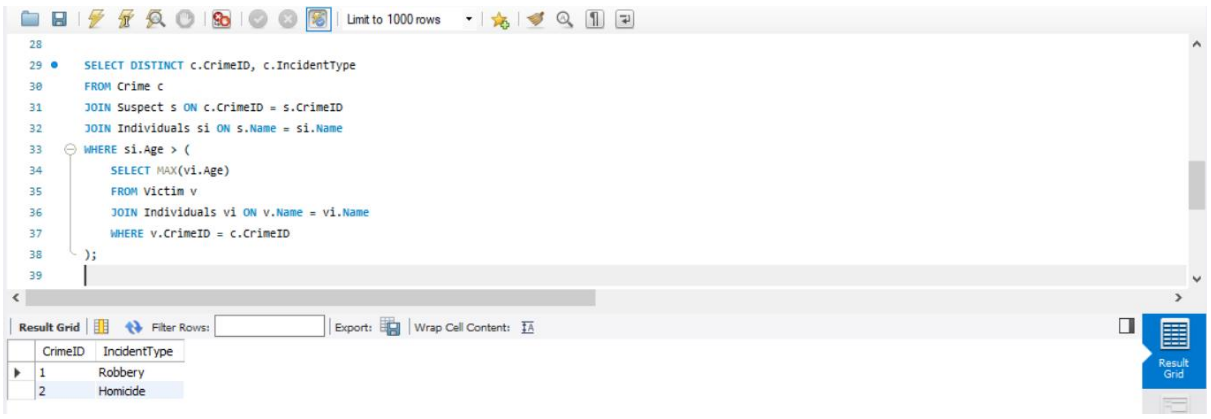
LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID;



```
SELECT C.*, V.Name AS VictimName, S.Name AS SuspectName
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	SuspectName
1	Robbery	2023-09-15	123 Main St, Cityville	Armed robbery at a convenience store	Open	John Doe	Robber 1
2	Homicide	2023-09-20	456 Elm St, Townsville	Investigation into a murder case	Under Investigation	Jane Smith	Unknown
3	Theft	2023-09-10	789 Oak St, Villagetown	Shoplifting incident at a mall	Closed	Alice Johnson	Suspect 1

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



```
SELECT DISTINCT c.CrimeID, c.IncidentType
FROM Crime c
JOIN Suspect s ON c.CrimeID = s.CrimeID
JOIN Individuals si ON s.Name = si.Name
WHERE si.Age > (
    SELECT MAX(vi.Age)
    FROM Victim v
    JOIN Individuals vi ON v.Name = vi.Name
    WHERE v.CrimeID = c.CrimeID
);
```

CrimeID	IncidentType
1	Robbery
2	Homicide

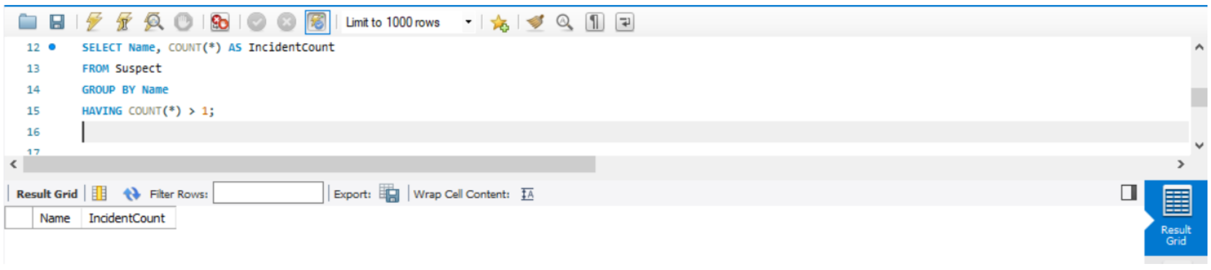
16. Find suspects involved in multiple incidents

SELECT Name, COUNT(*) AS IncidentCount

FROM Suspect

GROUP BY Name

HAVING COUNT(*) > 1;



```
SELECT Name, COUNT(*) AS IncidentCount
FROM Suspect
GROUP BY Name
HAVING COUNT(*) > 1;
```

Name	IncidentCount

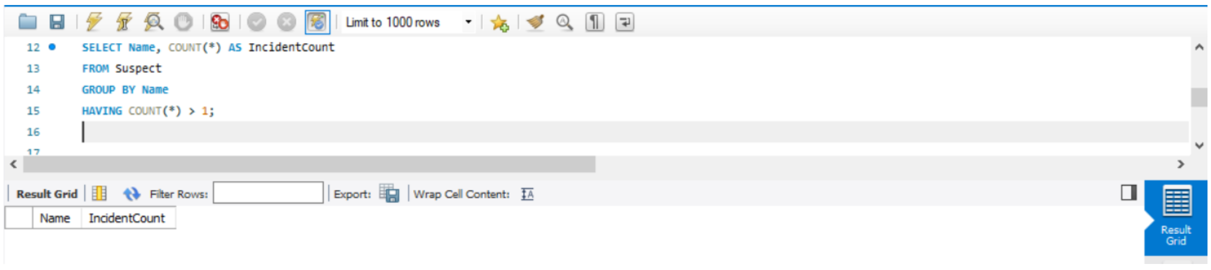
17. List incidents with no suspects involved

SELECT C.*

FROM Crime C

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

WHERE S.SuspectID IS NULL;



```
SELECT Name, COUNT(*) AS IncidentCount
FROM Suspect
GROUP BY Name
HAVING COUNT(*) > 1;
```

Name	IncidentCount

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

SELECT *

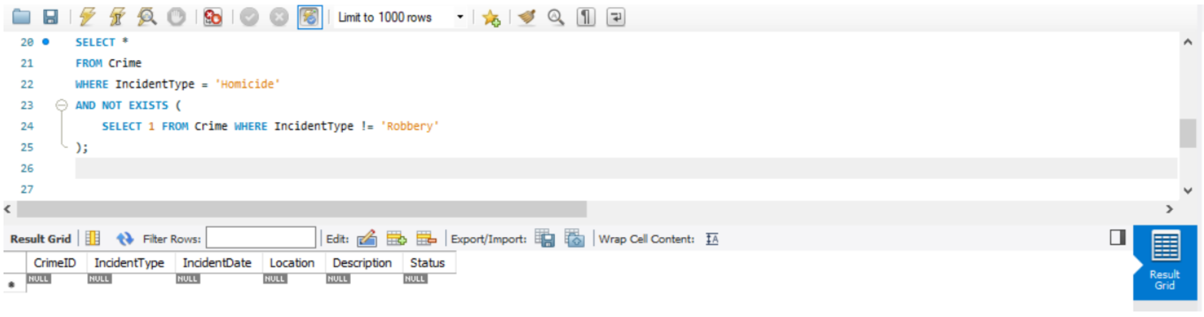
FROM Crime

WHERE IncidentType = 'Homicide'

AND NOT EXISTS (

    SELECT 1 FROM Crime WHERE IncidentType != 'Robbery'

);



```
SELECT *
FROM Crime
WHERE IncidentType = 'Homicide'
AND NOT EXISTS (
    SELECT 1 FROM Crime WHERE IncidentType != 'Robbery'
);
```

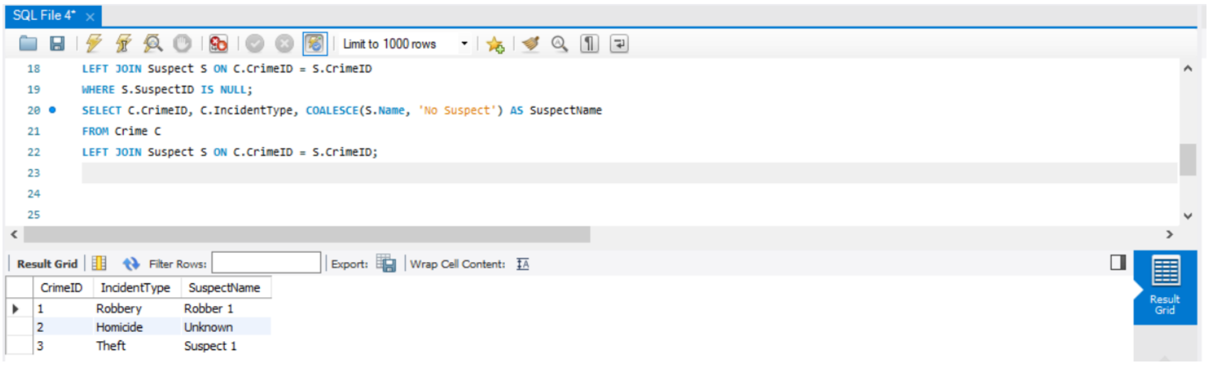
CrimeID	IncidentType	IncidentDate	Location	Description	Status
NULL	NULL	NULL	NULL	NULL	NULL

19. Retrieve all incidents and the associated suspects, showing 'No Suspect' if none

SELECT C.CrimeID, C.IncidentType, COALESCE(S.Name, 'No Suspect') AS SuspectName

FROM Crime C

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



```
LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID
WHERE S.SuspectID IS NULL;
SELECT C.CrimeID, C.IncidentType, COALESCE(S.Name, 'No Suspect') AS SuspectName
FROM Crime C
LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID;
```

CrimeID	IncidentType	SuspectName
1	Robbery	Robber 1
2	Homicide	Unknown
3	Theft	Suspect 1

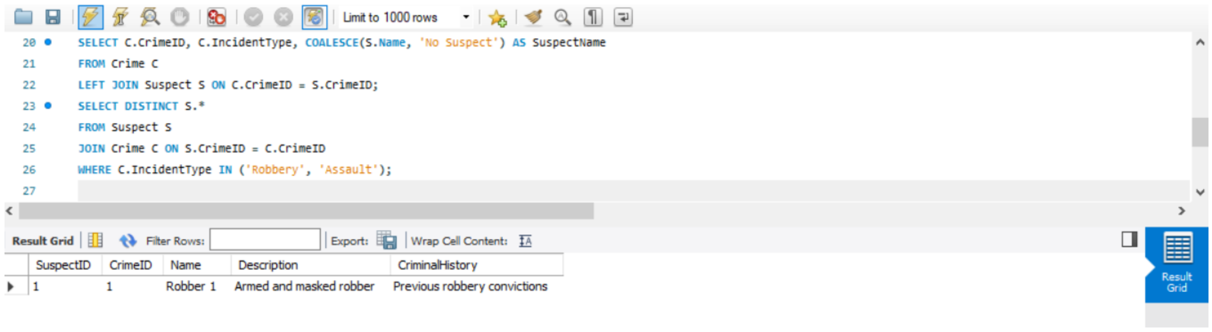
20. List all suspects involved in incidents of type 'Robbery' or 'Assault'

SELECT DISTINCT S.*

FROM Suspect S

JOIN Crime C ON S.CrimeID = C.CrimeID

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



```
SELECT C.CrimeID, C.IncidentType, COALESCE(S.Name, 'No Suspect') AS SuspectName
FROM Crime C
LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID;
SELECT DISTINCT S.*
FROM Suspect S
JOIN Crime C ON S.CrimeID = C.CrimeID
WHERE C.IncidentType IN ('Robbery', 'Assault');
```

SuspectID	CrimeID	Name	Description	CriminalHistory
1	1	Robber 1	Armed and masked robber	Previous robbery convictions