**NAME:ASENATH A**

# CODING CHALLENGE SQL
# CRIME MANAGEMENT

## CREATING DATABASE

```
CREATE DATABASE CrimeManagement;

USE CrimeManagement;
```

## 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)
);
```

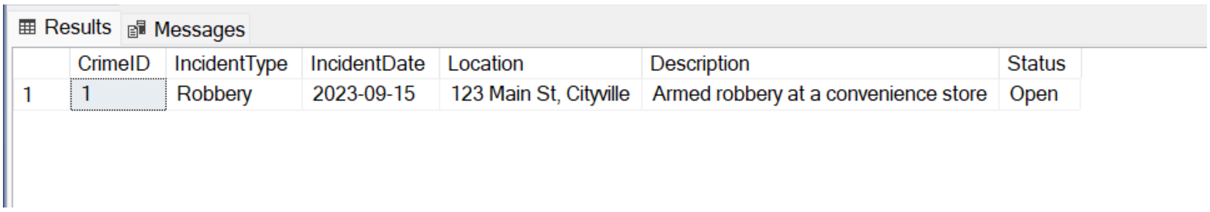
## INSERTING VALUES

```
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');
```

**NAME:ASENATH A**

### 1. Select all open incidents.

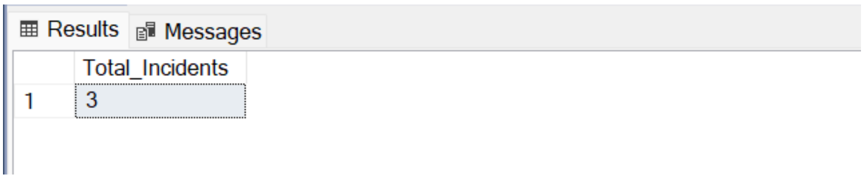
```sql
SELECT * FROM Crime WHERE Status='Open';
```



| | CrimeID | IncidentType | IncidentDate | Location | Description | Status |
|---|---|---|---|---|---|---|
| 1 | 1 | Robbery | 2023-09-15 | 123 Main St, Cityville | Armed robbery at a convenience store | Open |

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

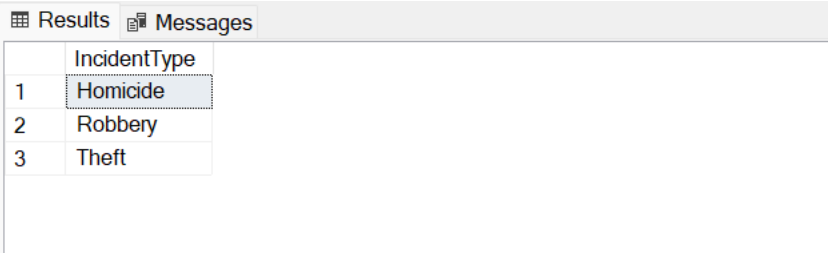
```sql
SELECT COUNT(*) AS Total_Incidents FROM Crime;
```



| | Total_Incidents |
|---|---|
| 1 | 3 |

### 3. List all unique incident types.

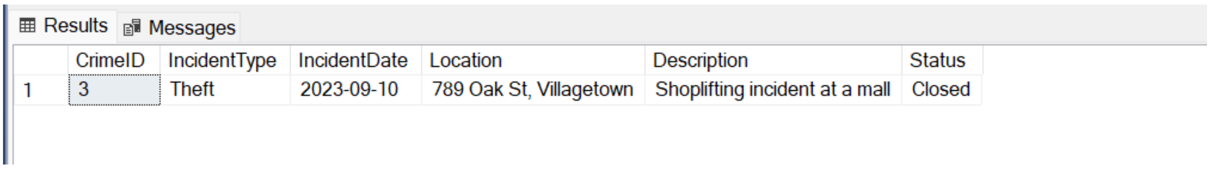
```sql
SELECT DISTINCT IncidentType FROM Crime;
```



| | IncidentType |
|---|---|
| 1 | Homicide |
| 2 | Robbery |
| 3 | Theft |

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

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



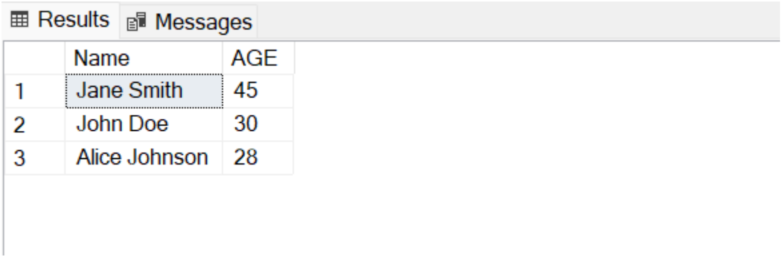
| | CrimeID | IncidentType | IncidentDate | Location | Description | Status |
|---|---|---|---|---|---|---|
| 1 | 3 | Theft | 2023-09-10 | 789 Oak St, Villagetown | Shoplifting incident at a mall | Closed |

### (ADDING THE COLUMN AGE IN Victim Table)

```sql
ALTER TABLE Victim ADD 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;
```

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

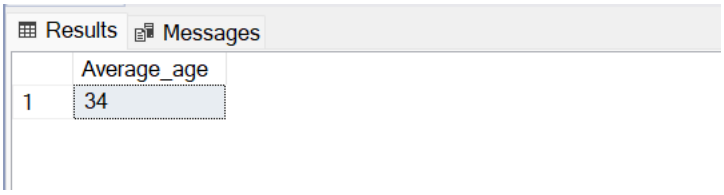
```sql
SELECT Name, AGE FROM Victim ORDER BY AGE DESC;
```



| | Name | AGE |
|---|---|---|
| 1 | Jane Smith | 45 |
| 2 | John Doe | 30 |
| 3 | Alice Johnson | 28 |

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

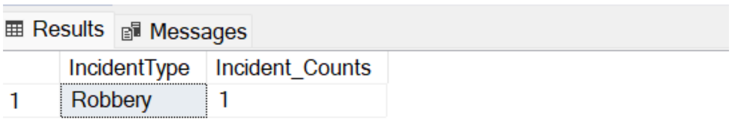
```
SELECT AVG(AGE) AS Average_age FROM Victim;
```



| | Average_age |
|---|---|
| 1 | 34 |

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

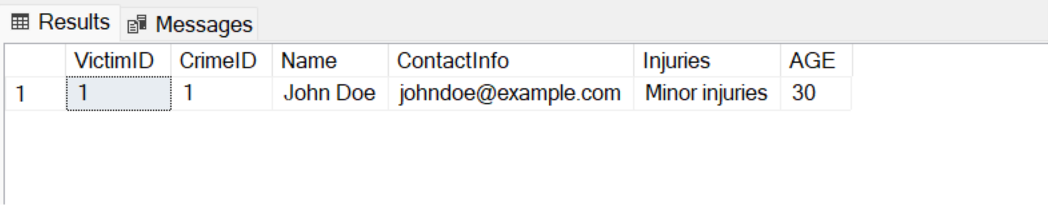
```
SELECT IncidentType, COUNT(*) AS Incident_Counts
FROM Crime WHERE Status='Open'
GROUP BY IncidentType;
```



| | IncidentType | Incident_Counts |
|---|---|---|
| 1 | Robbery | 1 |

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

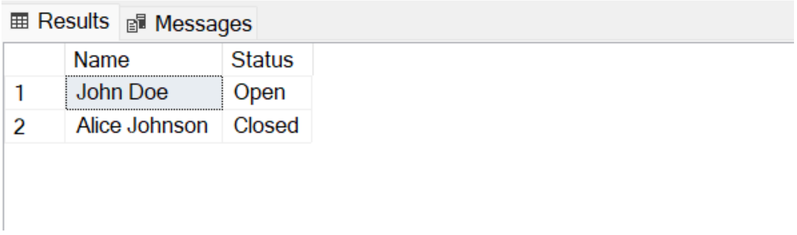
```
SELECT * FROM Victim WHERE Name LIKE '%doe%';
```



| | VictimID | CrimeID | Name | ContactInfo | Injuries | AGE |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | John Doe | johndoe@example.com | Minor injuries | 30 |

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

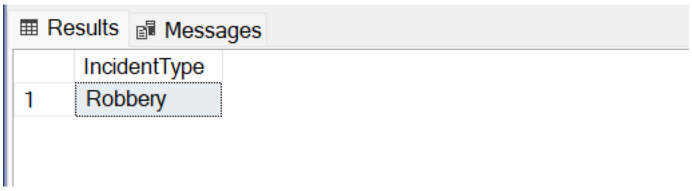
```
SELECT V.Name, C.Status
FROM Crime C
INNER JOIN Victim V
ON C.CrimeID=V.CrimeID
WHERE C.Status='Open' OR C.Status='Closed';
```



| | Name | Status |
|---|---|---|
| 1 | John Doe | Open |
| 2 | Alice Johnson | Closed |

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

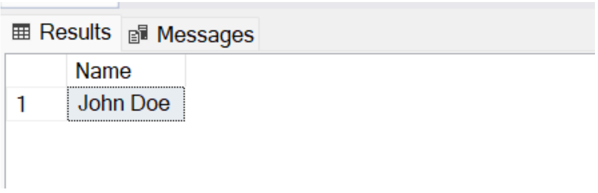
```
SELECT IncidentType From Crime C
JOIN Victim V ON C.CrimeID=V.CrimeID
WHERE AGE='30' OR AGE='35'
GROUP BY IncidentType;
```



|   | IncidentType |
|---|---|
| 1 | Robbery |

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

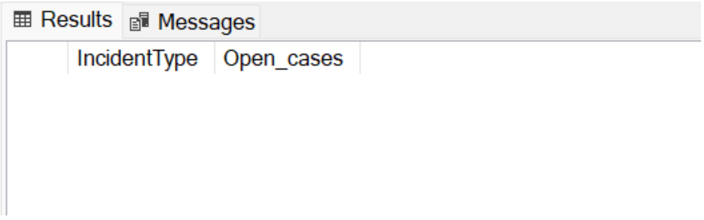
```
SELECT Name FROM Victim V
JOIN Crime C ON C.CrimeID=V.CrimeID
WHERE IncidentType='Robbery'
GROUP BY Name;
```



|   | Name |
|---|---|
| 1 | John Doe |

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

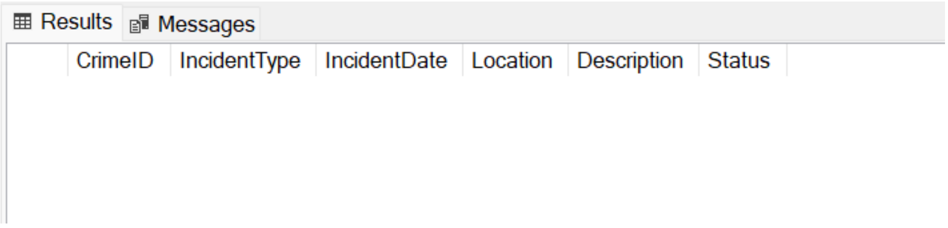
```
SELECT IncidentType, COUNT(*) AS Open_cases
FROM Crime
WHERE Status='Open'
GROUP BY IncidentType
HAVING COUNT(*)>1;
```



|   | IncidentType | Open_cases |
|---|---|---|

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

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

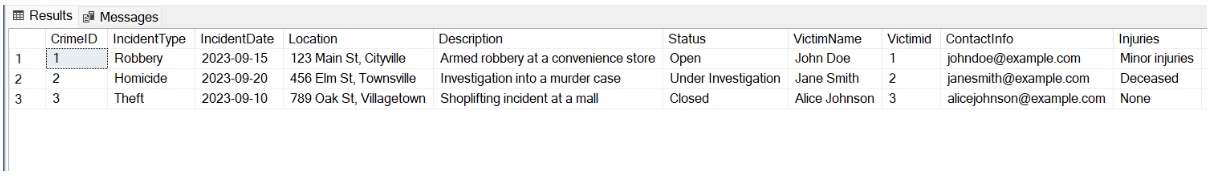


|   | CrimeID | IncidentType | IncidentDate | Location | Description | Status |
|---|---|---|---|---|---|---|

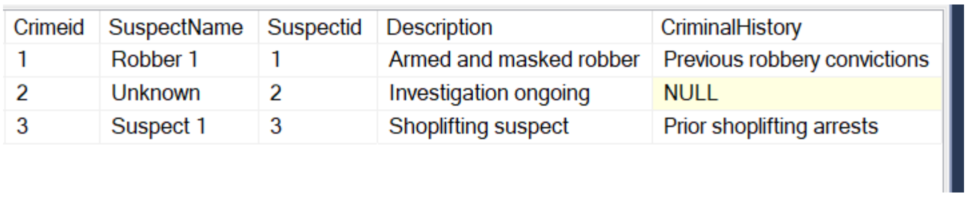
**14. Retrieve all incidents along with victim and suspect details.**

```
SELECT C.*, V.name AS VictimName, V.Victimid, V.ContactInfo, V.Injuries,V.Crimeid,
S.Name as SuspectName,S.Suspectid, 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;
```



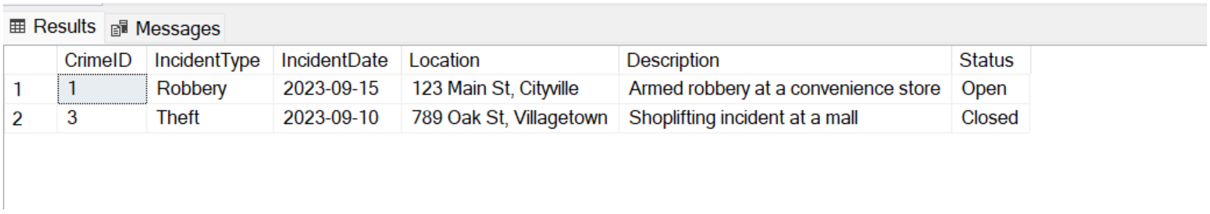
| | CrimeID | IncidentType | IncidentDate | Location | Description | Status | VictimName | Victimid | ContactInfo | Injuries |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Robbery | 2023-09-15 | 123 Main St, Cityville | Armed robbery at a convenience store | Open | John Doe | 1 | johndoe@example.com | Minor injuries |
| 2 | 2 | Homicide | 2023-09-20 | 456 Elm St, Townsville | Investigation into a murder case | Under Investigation | Jane Smith | 2 | janesmith@example.com | Deceased |
| 3 | 3 | Theft | 2023-09-10 | 789 Oak St, Villagetown | Shoplifting incident at a mall | Closed | Alice Johnson | 3 | alicejohnson@example.com | None |



| Crimeid | SuspectName | Suspectid | Description | CriminalHistory |
|---|---|---|---|---|
| 1 | Robber 1 | 1 | Armed and masked robber | Previous robbery convictions |
| 2 | Unknown | 2 | Investigation ongoing | NULL |
| 3 | Suspect 1 | 3 | Shoplifting suspect | Prior shoplifting arrests |

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

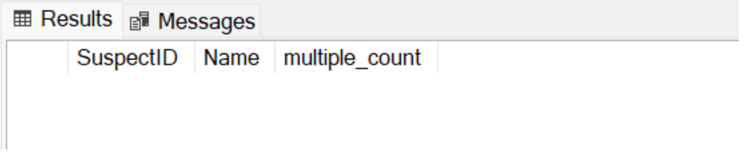
```
SELECT C.* FROM Crime C
JOIN Suspect S ON C.CrimeID = S.CrimeID
JOIN Victim V ON C.CrimeID = V.CrimeID
WHERE S.SAGE > V.Age;
```



| | CrimeID | IncidentType | IncidentDate | Location | Description | Status |
|---|---|---|---|---|---|---|
| 1 | 1 | Robbery | 2023-09-15 | 123 Main St, Cityville | Armed robbery at a convenience store | Open |
| 2 | 3 | Theft | 2023-09-10 | 789 Oak St, Villagetown | Shoplifting incident at a mall | Closed |

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

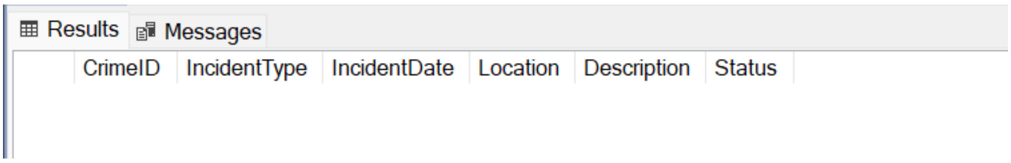
```
SELECT SuspectID, Name, COUNT(DISTINCT CrimeID) AS multiple_count
FROM Suspect
GROUP BY SuspectID, Name
HAVING COUNT(DISTINCT CrimeID) > 1;
```



| SuspectID | Name | multiple_count |
|---|---|---|

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

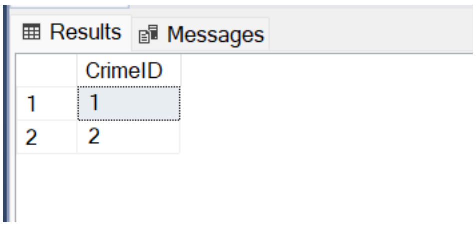
```
SELECT * FROM Crime
WHERE CrimeID NOT IN (SELECT DISTINCT CrimeID FROM Suspect);
```



| CrimeID | IncidentType | IncidentDate | Location | Description | Status |
|---|---|---|---|---|---|

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

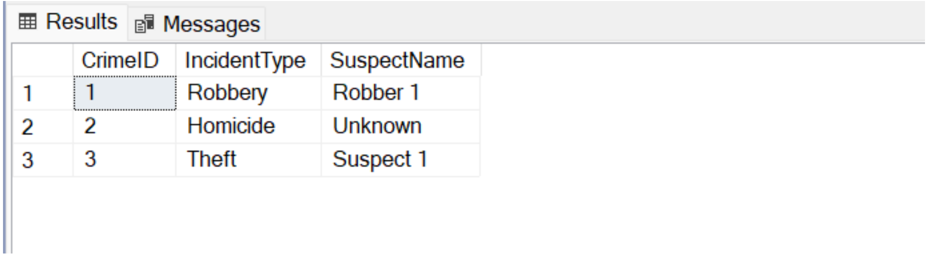
```
SELECT CrimeID
FROM Crime
WHERE IncidentType = 'Homicide'
OR (IncidentType = 'Robbery' AND CrimeID NOT IN (SELECT CrimeID FROM Crime WHERE
IncidentType = 'Homicide'));
```



|  | CrimeID |
|---|---|
| 1 | 1 |
| 2 | 2 |

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

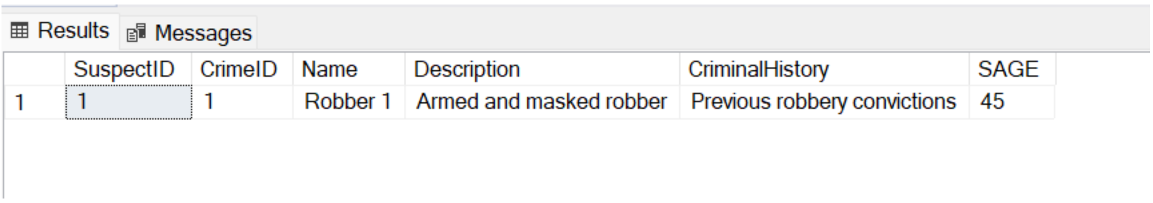
```
SELECT C.CrimeID, C.IncidentType, S.Name AS SuspectName
FROM Crime C
LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID;
```



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

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

```
SELECT S.*
FROM Suspect S
JOIN Crime C ON S.CrimeID = C.CrimeID
WHERE C.IncidentType IN ('Robbery', 'Assault');
```



|  | SuspectID | CrimeID | Name | Description | CriminalHistory | SAGE |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | Robber 1 | Armed and masked robber | Previous robbery convictions | 45 |