

## Coding Challenge 3 - Answers

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

```
SELECT Name, Age  
FROM Victim  
ORDER BY Age DESC;
```

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

```
SELECT AVG(Age) AS AverageAge  
FROM (  
  SELECT Age FROM Victim  
  UNION  
  SELECT Age FROM Suspect  
) AS CombinedAges;
```

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

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

7. Find persons with names containing 'Doe'.

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

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

```
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');
```

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

```
SELECT DISTINCT IncidentType  
FROM (  
SELECT IncidentType FROM Crime  
WHERE CrimeID IN (SELECT CrimeID FROM Victim WHERE Age IN (30, 35))  
UNION  
SELECT IncidentType FROM Crime  
WHERE CrimeID IN (SELECT CrimeID FROM Suspect WHERE Age IN (30, 35))  
) AS IncidentsWithAge;
```

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

```
SELECT Name
FROM Victim
WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE IncidentType = 'Robbery')
UNION
SELECT Name
FROM Suspect
WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE IncidentType = 'Robbery');
```

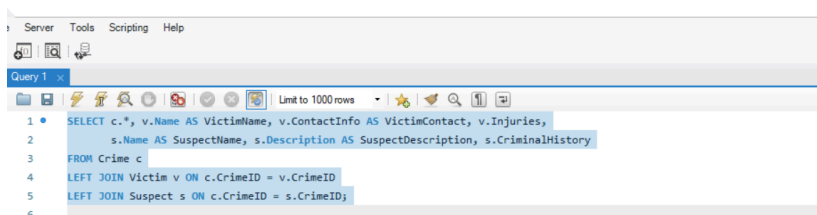
11. List incident types with more than one open case.

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

12. 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 c.CrimeID = v.CrimeID
WHERE s.Name = v.Name;
```

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



The screenshot shows a database application interface. The top part displays a 'Result Grid' with the following data:

| CrimeID | IncidentType | IncidentDate | Location                | Description                          | Status        | VictimName    | VictimContact            |
|---------|--------------|--------------|-------------------------|--------------------------------------|---------------|---------------|--------------------------|
| 1       | Robbery      | 2023-09-15   | 123 Main St, Cityville  | Armed robbery at a convenience store | Open          | John Doe      | johndoe@example.com      |
| 2       | Homicide     | 2023-09-20   | 456 Elm St, Townsville  | Investigation into a murder case     | Under Inve... | Jane Smith    | janesmith@example.com    |
| 3       | Theft        | 2023-09-10   | 789 Oak St, Villagetown | Shoplifting incident at a mall       | Closed        | Alice Johnson | alicejohnson@example.com |

The bottom part shows an 'Action Output' log with the following entries:

| #  | Time     | Action  | Message           |
|----|----------|---|-------------------|
| 15 | 22:49:46 | SELECT * FROM Crime WHERE IncidentDate BETWEEN '2023-09-01' AND '2023-09-10' LIMIT 0, 1000              | 1 row(s) returned |
| 16 | 22:54:22 | SELECT * FROM Victim WHERE Name LIKE '%Doe%' UNION SELECT * FROM Suspect WHERE Name LIK...              | 1 row(s) returned |
| 17 | 23:15:26 | SELECT IncidentType, COUNT(*) AS OpenCaseCount FROM Crime WHERE Status = 'Open' GROUP BY Inci...        | 0 row(s) returned |
| 18 | 23:15:44 | SELECT DISTINCT c.* FROM Crime c JOIN Suspect s ON c.CrimeID = s.CrimeID JOIN Victim v ON c.CrimeID ... | 0 row(s) returned |
| 19 | 23:15:58 | SELECT c.*, v.Name AS VictimName, v.ContactInfo AS VictimContact, v.Injuries, s.Name AS SuspectNa...    | 3 row(s) returned |

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

```
SELECT c.*
FROM Crime c
JOIN Suspect s ON c.CrimeID = s.CrimeID
WHERE s.Age > ALL (SELECT Age FROM Victim WHERE CrimeID = c.CrimeID);
```

15. Find suspects involved in multiple incidents:

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

16. List incidents with no suspects involved.

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

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

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

18. 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;
```

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

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