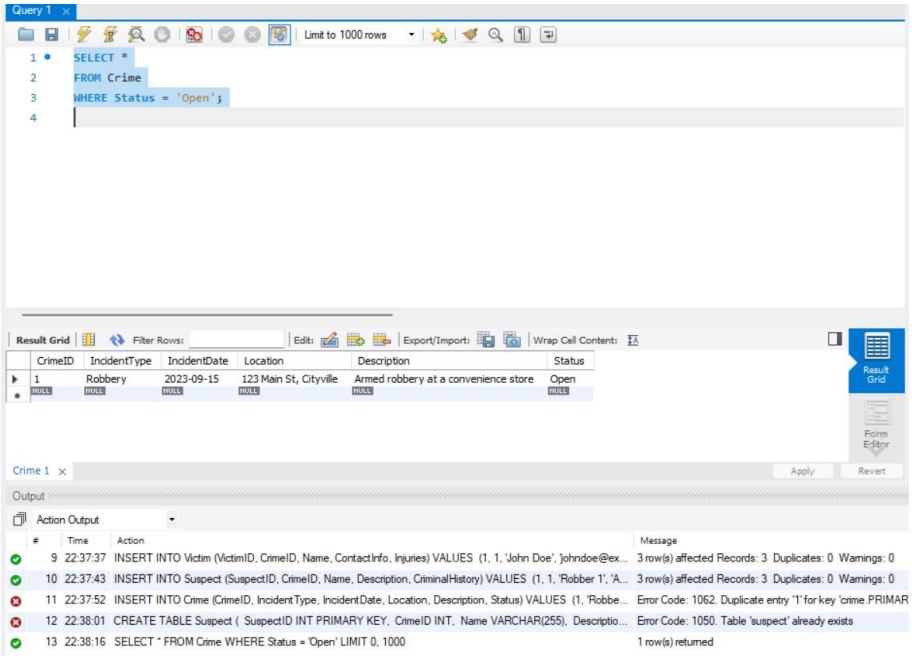
CODING CHALLENGES 3

1. Select all open incidents.



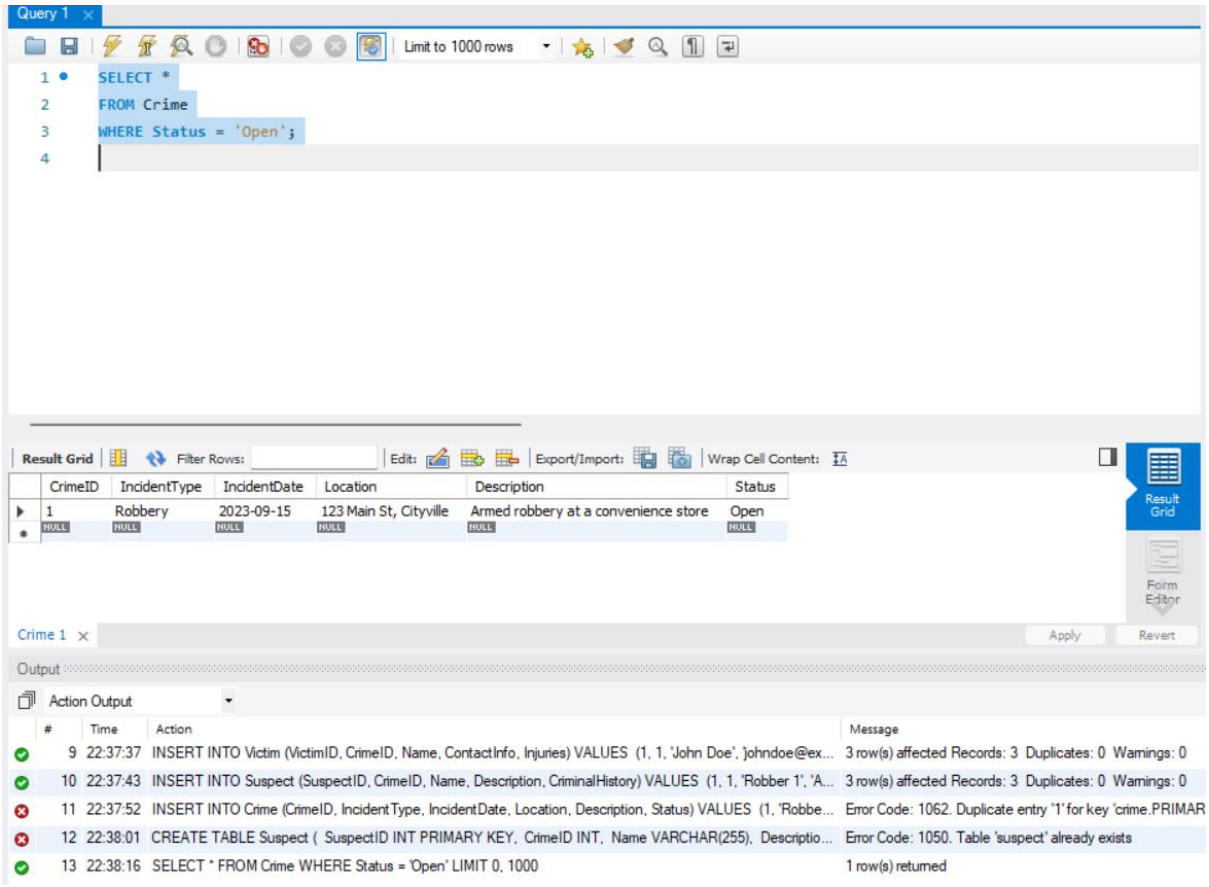
2. Find the total number of incidents.

```
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'.



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

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

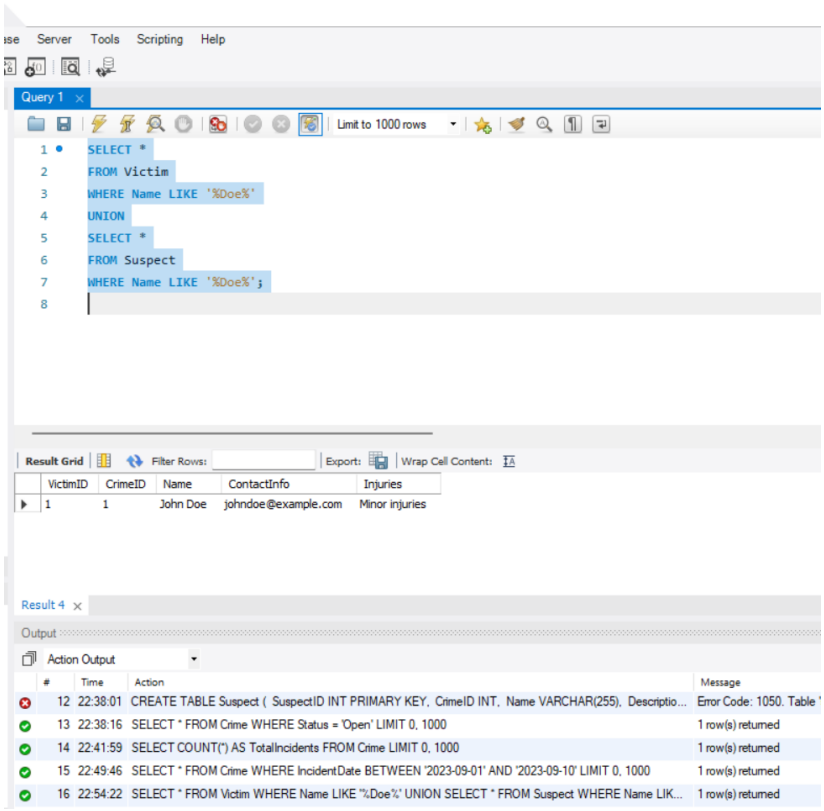
5. Find the average age of persons involved in incidents.

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

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

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

8. Find persons with names containing 'Doe'.



9. 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');
```

10. 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;
```

11. 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');
```

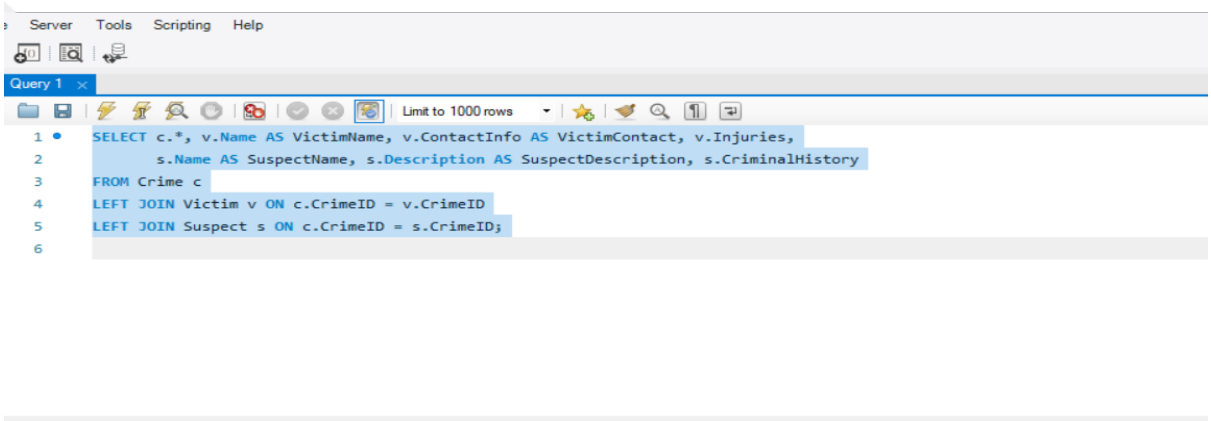
12. List incident types with more than one open case.

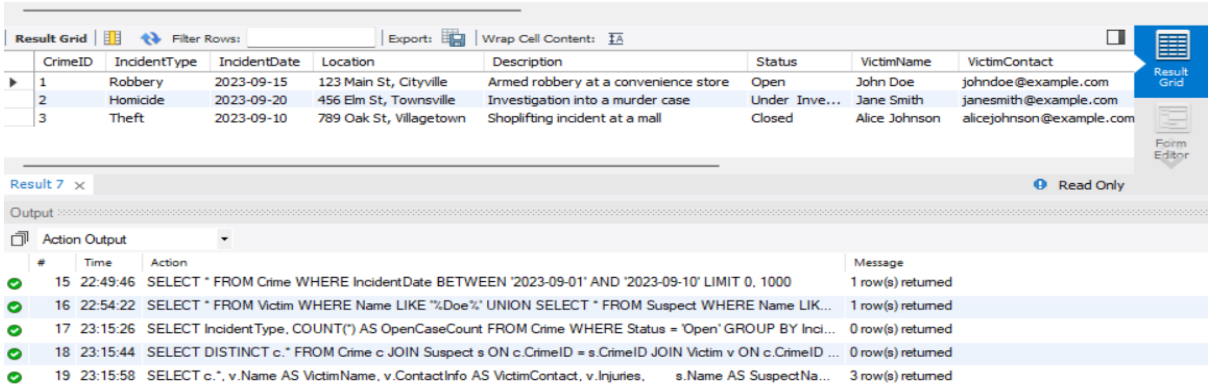
```
SELECT IncidentType, COUNT(*) AS OpenCaseCount
FROM Crime
WHERE Status = 'Open'
GROUP BY IncidentType
HAVING COUNT(*) > 1;
```

13. 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;
```

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





15. 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);
```

16. Find suspects involved in multiple incidents:

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

17. List incidents with no suspects involved.

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

18. 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'));
```

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

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