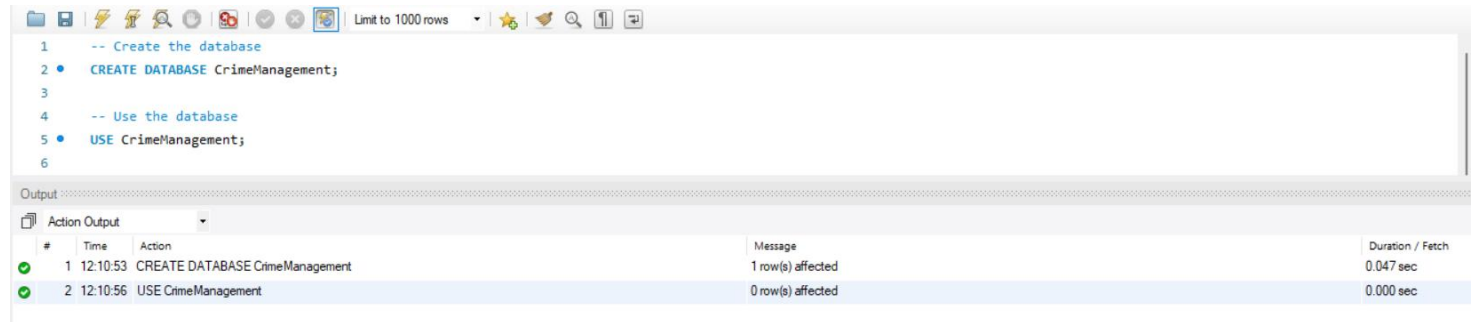


# SQL Coding Challenge

## Crime Management – Query Output

### DATABASE SETUP

#### 1. Creating and Selecting Crime Management Database



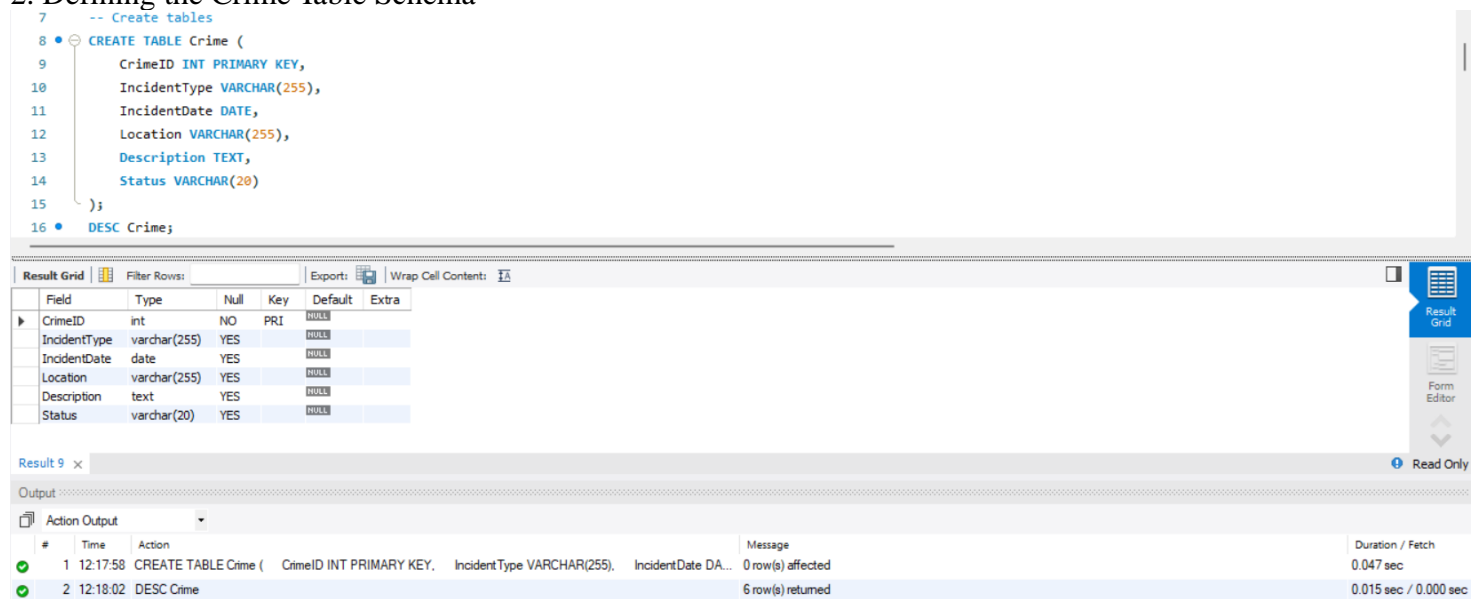
The screenshot shows a SQL IDE window with a toolbar at the top. The main editor contains the following SQL code:

```
1 -- Create the database
2 • CREATE DATABASE CrimeManagement;
3
4 -- Use the database
5 • USE CrimeManagement;
6
```

Below the editor is the 'Output' pane, which is currently showing 'Action Output'. It contains two rows of execution results:

#	Time	Action	Message	Duration / Fetch
1	12-10-53	CREATE DATABASE CrimeManagement	1 row(s) affected	0.047 sec
2	12-10-56	USE CrimeManagement	0 row(s) affected	0.000 sec

#### 2. Defining the Crime Table Schema



The screenshot shows the SQL IDE with the following SQL code for creating the Crime table:

```
7 -- Create tables
8 • CREATE TABLE Crime (
9     CrimeID INT PRIMARY KEY,
10    IncidentType VARCHAR(255),
11    IncidentDate DATE,
12    Location VARCHAR(255),
13    Description TEXT,
14    Status VARCHAR(20)
15 );
16 • DESC Crime;
```

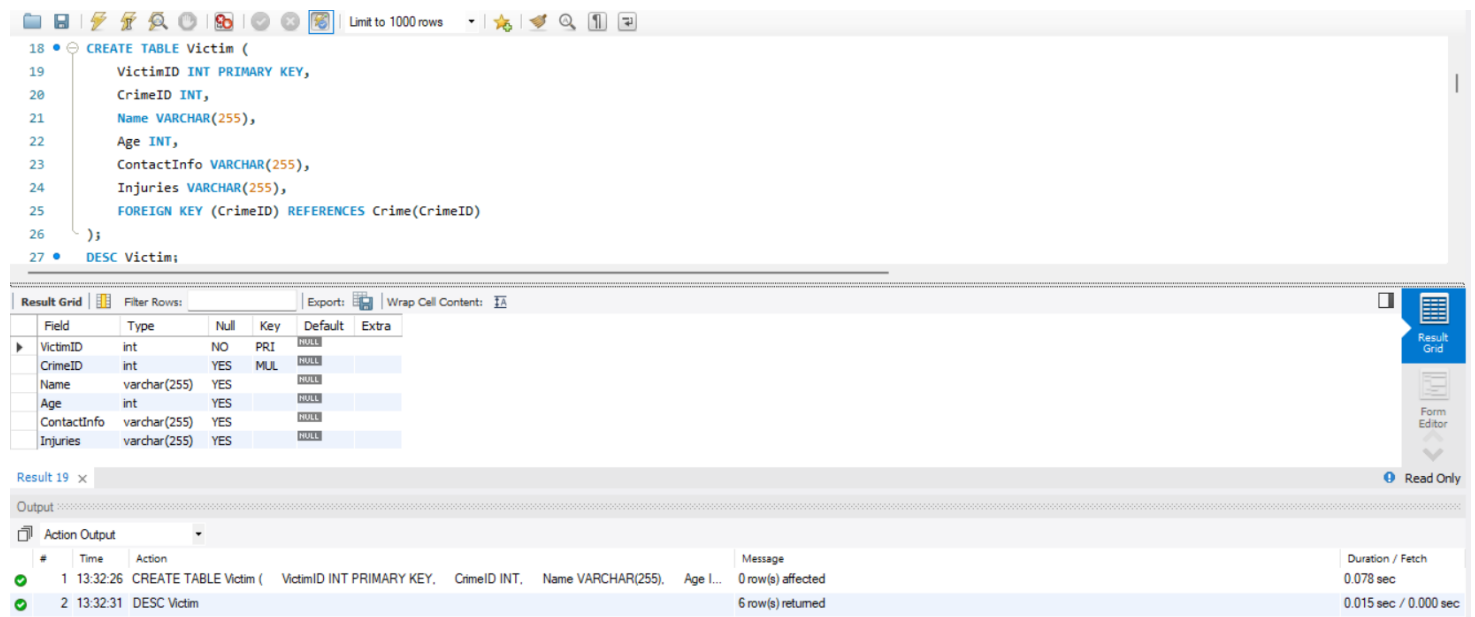
The 'Result Grid' pane shows the schema details for the Crime table:

Field	Type	Null	Key	Default	Extra
CrimeID	int	NO	PRI	NULL	
IncidentType	varchar(255)	YES		NULL	
IncidentDate	date	YES		NULL	
Location	varchar(255)	YES		NULL	
Description	text	YES		NULL	
Status	varchar(20)	YES		NULL	

The 'Output' pane shows the execution results for the CREATE and DESC statements:

#	Time	Action	Message	Duration / Fetch
1	12-17-58	CREATE TABLE Crime ( CrimeID INT PRIMARY KEY, IncidentType VARCHAR(255), IncidentDate DA...	0 row(s) affected	0.047 sec
2	12-18-02	DESC Crime	6 row(s) returned	0.015 sec / 0.000 sec

#### 3. Defining the Victim Table Schema



The screenshot shows the SQL IDE with the following SQL code for creating the Victim table:

```
18 • CREATE TABLE Victim (
19     VictimID INT PRIMARY KEY,
20     CrimeID INT,
21     Name VARCHAR(255),
22     Age INT,
23     ContactInfo VARCHAR(255),
24     Injuries VARCHAR(255),
25     FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)
26 );
27 • DESC Victim;
```

The 'Result Grid' pane shows the schema details for the Victim table:

Field	Type	Null	Key	Default	Extra
VictimID	int	NO	PRI	NULL	
CrimeID	int	YES	MUL	NULL	
Name	varchar(255)	YES		NULL	
Age	int	YES		NULL	
ContactInfo	varchar(255)	YES		NULL	
Injuries	varchar(255)	YES		NULL	

The 'Output' pane shows the execution results for the CREATE and DESC statements:

#	Time	Action	Message	Duration / Fetch
1	13-32-26	CREATE TABLE Victim ( VictimID INT PRIMARY KEY, CrimeID INT, Name VARCHAR(255), Age I...	0 row(s) affected	0.078 sec
2	13-32-31	DESC Victim	6 row(s) returned	0.015 sec / 0.000 sec

## 4. Defining the Suspect Table Schema

The screenshot shows a database IDE with a SQL editor and a result grid. The SQL editor contains the following code:

```
29 • CREATE TABLE Suspect (  
30     SuspectID INT PRIMARY KEY,  
31     CrimeID INT,  
32     Name VARCHAR(255),  
33     Age INT,  
34     Description TEXT,  
35     CriminalHistory TEXT,  
36     FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)  
37 );  
38 • DESC Suspect;
```

The result grid shows the table structure for 'Suspect':

Field	Type	Null	Key	Default	Extra
SuspectID	int	NO	PRI		
CrimeID	int	YES	MUL		
Name	varchar(255)	YES			
Age	int	YES			
Description	text	YES			
CriminalHistory	text	YES			

The output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	13:34:24	CREATE TABLE Suspect ( SuspectID INT PRIMARY KEY, CrimeID INT, Name VARCHAR(255), A...	0 row(s) affected	0.078 sec
2	13:34:28	DESC Suspect	6 row(s) returned	0.000 sec / 0.000 sec

## 5. Inserting Crime Records

The screenshot shows a database IDE with a SQL editor and a result grid. The SQL editor contains the following code:

```
38 -- Insert data  
39 • INSERT INTO Crime (CrimeID, IncidentType, IncidentDate, Location, Description, Status)  
40 VALUES  
41     (1, 'Robbery', '2023-09-15', '123 Main St, Cityville', 'Armed robbery at a convenience store', 'Open'),  
42     (2, 'Homicide', '2023-09-20', '456 Elm St, Townsville', 'Investigation into a murder case', 'Under Investigation'),  
43     (3, 'Theft', '2023-09-10', '789 Oak St, Villagetown', 'Shoplifting incident at a mall', 'Closed');  
44 • SELECT * FROM Crime;
```

The result grid shows the data for 'Crime':

CrimeID	IncidentType	IncidentDate	Location	Description	Status
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

The output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	12:40:10	INSERT INTO Crime (CrimeID, IncidentType, IncidentDate, Location, Description, Status) VALUES (1, 'Robb...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.031 sec
2	12:40:15	SELECT * FROM Crime LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

## 6. Inserting Victim Records

The screenshot shows a database IDE with a SQL editor and a result grid. The SQL editor contains the following code:

```
48 • INSERT INTO Victim (VictimID, CrimeID, Name, Age, ContactInfo, Injuries)  
49 VALUES  
50     (1, 1, 'John Doe', 35, 'johndoe@example.com', 'Minor injuries'),  
51     (2, 2, 'Jane Smith', 42, 'janesmith@example.com', 'Deceased'),  
52     (3, 3, 'Alice Johnson', 28, 'alicejohnson@example.com', 'None');  
53 • SELECT * FROM Victim;  
54
```

The result grid shows the data for 'Victim':

VictimID	CrimeID	Name	Age	ContactInfo	Injuries
1	1	John Doe	35	johndoe@example.com	Minor injuries
2	2	Jane Smith	42	janesmith@example.com	Deceased
3	3	Alice Johnson	28	alicejohnson@example.com	None

## 7. Inserting Suspect Records

Limit to 1000 rows

```
55 • INSERT INTO Suspect (SuspectID, CrimeID, Name, Age, Description, CriminalHistory)
56   VALUES
57     (1, 1, 'Robber 1', 40, 'Armed and masked robber', 'Previous robbery convictions'),
58     (2, 2, 'Unknown', NULL, 'Investigation ongoing', NULL),
59     (3, 3, 'Suspect 1', 30, 'Shoplifting suspect', 'Prior shoplifting arrests');
60 • SELECT * FROM Suspect;
61
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [IA](#)

	SuspectID	CrimeID	Name	Age	Description	CriminalHistory
▶	1	1	Robber 1	40	Armed and masked robber	Previous robbery convictions
	2	2	Unknown	NULL	Investigation ongoing	NULL
	3	3	Suspect 1	30	Shoplifting suspect	Prior shoplifting arrests

## QUERIES

### 1. Select all the open incidents

```
60   -- Q1. Select all open incidents
61 •  SELECT * FROM Crime WHERE Status = 'Open';
62
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [IA](#)

	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

```
63   -- Q2. Find the total number of incidents.
64 •  SELECT COUNT(*) AS TotalIncidents FROM Crime;
65
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	TotalIncidents
▶	3

### 3. List all unique incident types

```
66   -- Q3. List all unique incident types.
67 •  SELECT DISTINCT IncidentType FROM Crime;
68
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	IncidentType
▶	Robbery
	Homicide
	Theft

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

```
69   -- Q4. Retrieve incidents that occurred between '2023-09-01' and '2023-09-10'.
70 •  SELECT * FROM Crime WHERE IncidentDate BETWEEN '2023-09-01' AND '2023-09-10';
71
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [IA](#)

	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

```
74  -- Q5. List persons involved in incidents in descending order of age.
75  •  SELECT Name, Age
76  ◉  FROM (
77      SELECT Name, Age FROM Victim
78      UNION
79      SELECT Name, Age FROM Suspect
80  ) AS Persons
81  ORDER BY Age DESC;
```

Name	Age
Jane Smith	42
Robber 1	40
John Doe	35
Suspect 1	30
Alice Johnson	28
Unknown	NULL

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

```
83  -- Q6. Find the average age of persons involved in incidents.
84  •  SELECT AVG(Age) AS AverageAge FROM (SELECT Age FROM Victim UNION SELECT Age FROM Suspect) AS Persons;
85
```

AverageAge
35.0000

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

```
86  -- Q7. List incident types and their counts, only for open cases.
87  •  SELECT IncidentType, COUNT(*) AS IncidentCount FROM Crime WHERE Status = 'Open' GROUP BY IncidentType;
88
```

IncidentType	IncidentCount
Robbery	1

8. Find persons with names containing 'Doe'.

```
89  -- Q8. Find persons with names containing 'Doe'.
90  •  SELECT Name FROM (SELECT Name FROM Victim UNION SELECT Name FROM Suspect) AS Persons WHERE Name LIKE '%Doe%';
91
```

Name
John Doe

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

```
92  -- Q9. Retrieve the names of persons involved in open cases and closed cases.
93  •  SELECT v.Name AS VictimName, s.Name AS SuspectName, c.Status
94  FROM Crime c
95  LEFT JOIN Victim v ON c.CrimeID = v.CrimeID
96  LEFT JOIN Suspect s ON c.CrimeID = s.CrimeID;
97
```

VictimName	SuspectName	Status
John Doe	Robber 1	Open
Jane Smith	Unknown	Under Investigation
Alice Johnson	Suspect 1	Closed

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

```
98 -- Q10. List incident types where there are persons aged 30 or 35 involved.
99 • SELECT DISTINCT c.IncidentType
100 FROM Crime c
101 JOIN Victim v ON c.CrimeID = v.CrimeID
102 WHERE v.Age IN (30, 35)
103 UNION
104 SELECT DISTINCT c.IncidentType
105 FROM Crime c
106 JOIN Suspect s ON c.CrimeID = s.CrimeID
107 WHERE s.Age IN (30, 35);
108
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

IncidentType
Robbery
Theft

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

```
109 -- Q11. Find persons involved in incidents of the same type as 'Robbery'.
110 • SELECT Name FROM (SELECT v.Name, c.IncidentType FROM Victim v JOIN Crime c ON v.CrimeID = c.CrimeID
111 WHERE c.IncidentType = 'Robbery'
112 UNION
113 SELECT s.Name, c.IncidentType FROM Suspect s JOIN Crime c ON s.CrimeID = c.CrimeID
114 WHERE c.IncidentType = 'Robbery') AS Persons;
115
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Name
John Doe
Robber 1

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

```
116 -- Q12. List incident types with more than one open case.
117 • SELECT IncidentType FROM Crime WHERE Status = 'Open' GROUP BY IncidentType HAVING COUNT(*) > 1;
118
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

IncidentType
--------------

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

```
119 -- Q13. List all incidents with suspects whose names also appear as victims in other incidents.
120 • SELECT DISTINCT c.*
121 FROM Crime c
122 JOIN Suspect s ON c.CrimeID = s.CrimeID
123 JOIN Victim v ON s.Name = v.Name AND s.CrimeID <> v.CrimeID;
124
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

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

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

```
125 -- Q14. Retrieve all incidents along with victim and suspect details.
126 • SELECT c.*, v.Name AS VictimName, v.ContactInfo, v.Injuries, s.Name AS SuspectName, s.Description, s.CriminalHistory
127 FROM Crime c
128 LEFT JOIN Victim v ON c.CrimeID = v.CrimeID
129 LEFT JOIN Suspect s ON c.CrimeID = s.CrimeID;
130
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

CrimeID	IncidentType	IncidentDate	Location	Description	Status	VictimName	ContactInfo	Injuries	SuspectName	Description	CriminalHistory
1	Robbery	2023-09-15	123 Main St, Cityville	Armed robbery at a convenience store	Open	John Doe	john.doe@example.com	Minor injuries	Robber 1	Armed and masked robber	Previous robbery convictions
2	Homicide	2023-09-20	456 Elm St, Townsville	Investigation into a murder case	Under Investigation	Jane Smith	jane.smith@example.com	Deceased	Unknown	Investigation ongoing	
3	Theft	2023-09-10	789 Oak St, Villagetown	Shoplifting incident at a mall	Closed	Alice Johnson	alice.johnson@example.com	None	Suspect 1	Shoplifting suspect	Prior shoplifting arrests

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

```
131 -- Q15. Find incidents where the suspect is older than any victim.
132 • SELECT DISTINCT c.*
133 FROM Crime c
134 JOIN Suspect s ON c.CrimeID = s.CrimeID
135 JOIN Victim v ON c.CrimeID = v.CrimeID
136 WHERE s.Age > v.Age;
137
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

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

16. Find suspects involved in multiple incidents.

```
138 -- Q16. Find suspects involved in multiple incidents.
139 • SELECT Name FROM Suspect GROUP BY Name HAVING COUNT(DISTINCT CrimeID) > 1;
140
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Name
------

17. List incidents with no suspects involved.

```
141 -- Q17. List incidents with no suspects involved.
142 • SELECT * FROM Crime WHERE CrimeID NOT IN (SELECT DISTINCT CrimeID FROM Suspect);
143
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [IA](#)

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

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

```
144 -- Q18. List all cases where at least one incident is of type 'Homicide' and all other incidents are of type 'Robbery'.
145 • SELECT * FROM Crime
146 WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE IncidentType = 'Homicide')
147 AND NOT EXISTS (SELECT 1 FROM Crime WHERE IncidentType NOT IN ('Homicide', 'Robbery'));
148
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [IA](#)

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

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

```
149 -- Q19. Retrieve a list of all incidents and the associated suspects, showing suspects for each incident, or 'No Suspect' if there are none.
150 SELECT c.*, COALESCE(s.Name, 'No Suspect') AS SuspectName
151 FROM Crime c
152 LEFT JOIN Suspect s ON c.CrimeID = s.CrimeID;
153
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

CrimeID	IncidentType	IncidentDate	Location	Description	Status	SuspectName
▶ 1	Robbery	2023-09-15	123 Main St, Cityville	Armed robbery at a convenience store	Open	Robber 1
	Homicide	2023-09-20	456 Elm St, Townsville	Investigation into a murder case	Under Investigation	Unknown
3	Theft	2023-09-10	789 Oak St, Villagetown	Shoplifting incident at a mall	Closed	Suspect 1

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

```
154 -- Q20. List all suspects who have been involved in incidents with incident types 'Robbery' or 'Assault'.
155 • SELECT DISTINCT s.Name
156 FROM Suspect s
157 JOIN Crime c ON s.CrimeID = c.CrimeID
158 WHERE c.IncidentType IN ('Robbery', 'Assault');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Name
▶ Robber 1