**Coding Challenge - MySQL**

**Crime Management**

**SaiPrabath Chowdary S**

**Creating DataBase :**

CREATE DATABASE IF NOT EXISTS CrimeManagement;

Screenshot 2024-04-18 at 10.24.03 AM

**Using DataBase :**

USE CrimeManagement;

**Screenshot 2024-04-18 at 10.24.53 AM**

**Creating Tables:**

1. Crime

CREATE TABLE IF NOT EXISTS Crime (

CrimeID INT PRIMARY KEY,

IncidentType VARCHAR(255),

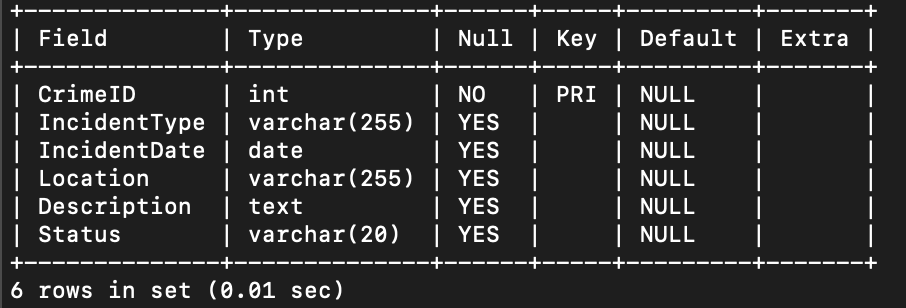
IncidentDate DATE,

Location VARCHAR(255),

Description TEXT,

Status VARCHAR(20)

);



1. Victim

CREATE TABLE IF NOT EXISTS Victim (

VictimID INT PRIMARY KEY,

CrimeID INT,

Name VARCHAR(255),

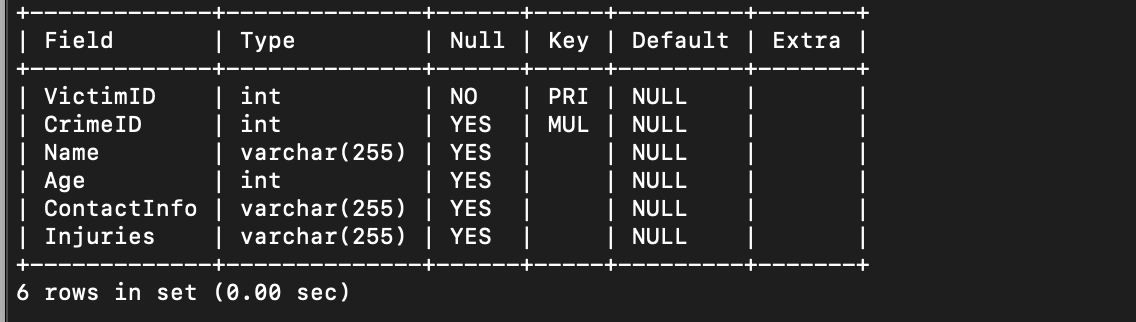
age INT,

ContactInfo VARCHAR(255),

Injuries VARCHAR(255),

FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)

);



1. Suspect

CREATE TABLE IF NOT EXISTS Suspect (

SuspectID INT PRIMARY KEY,

CrimeID INT,

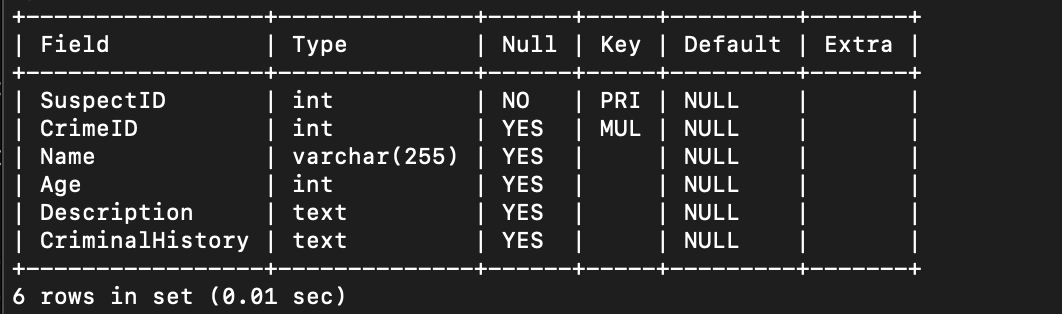
Name VARCHAR(255),

Description TEXT,

CriminalHistory TEXT,

FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)

);

****

**Inserting Values Into Tables:**

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'),

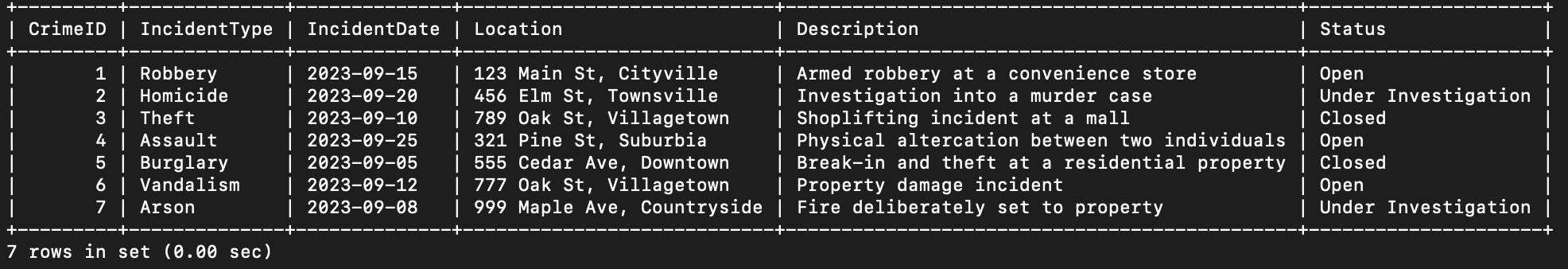
(3, 'Theft', '2023-09-10', '789 Oak St, Villagetown', 'Shoplifting incident at a mall', 'Closed'),

(4, 'Assault', '2023-09-25', '321 Pine St, Suburbia', 'Physical altercation between two individuals', 'Open'),

(5, 'Burglary', '2023-09-05', '555 Cedar Ave, Downtown', 'Break-in and theft at a residential property', 'Closed'),

(6, 'Vandalism', '2023-09-12', '777 Oak St, Villagetown', 'Property damage incident', 'Open'),

(7, 'Arson', '2023-09-08', '999 Maple Ave, Countryside', 'Fire deliberately set to property', 'Under Investigation');



INSERT INTO Victim (VictimID, CrimeID, Name,age, ContactInfo, Injuries) VALUES

(1, 1, 'John Doe', 25 , 'johndoe@example.com', 'Minor injuries'),

(2, 2, 'Jane Smith', 32 , 'janesmith@example.com', 'Deceased'),

(3, 3, 'Alice Johnson', 40 , 'alicejohnson@example.com', 'None'),

(4, 4, 'Michael Brown', 35 , 'michaelbrown@example.com', 'Bruises'),

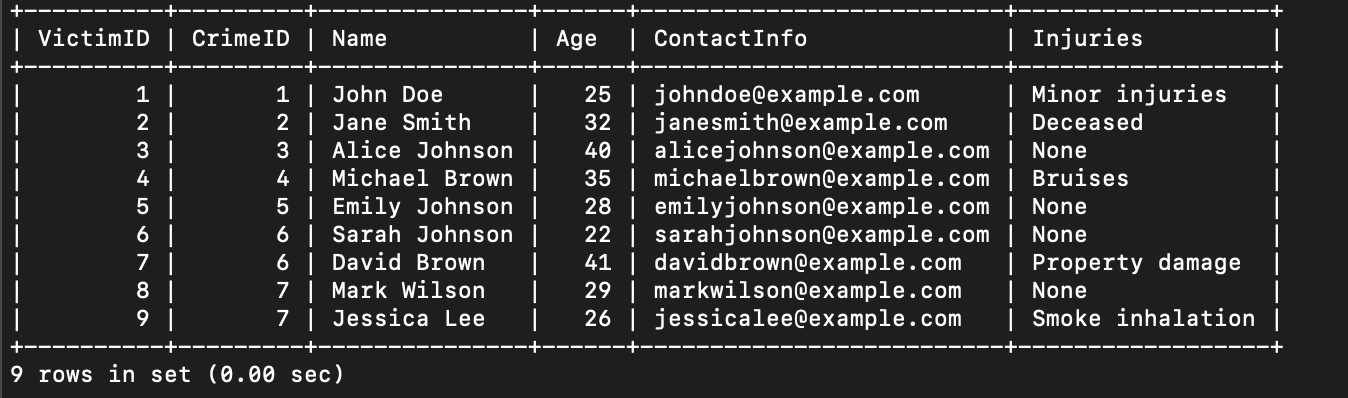
(5, 5, 'Emily Johnson', 28 , 'emilyjohnson@example.com', 'None'),

(6, 6, 'Sarah Johnson', 22 , 'sarahjohnson@example.com', 'None'),

(7, 6, 'David Brown', 41 , 'davidbrown@example.com', 'Property damage'),

(8, 7, 'Mark Wilson', 29 , 'markwilson@example.com', 'None'),

(9, 7, 'Jessica Lee', 26 , 'jessicalee@example.com', 'Smoke inhalation');



INSERT INTO Suspect (SuspectID, CrimeID, Name,age, Description, CriminalHistory) VALUES

(1, 1, 'Robber 1',30 , 'Armed and masked robber', 'Previous robbery convictions'),

(2, 2, 'Unknown',, 'Investigation ongoing', NULL),

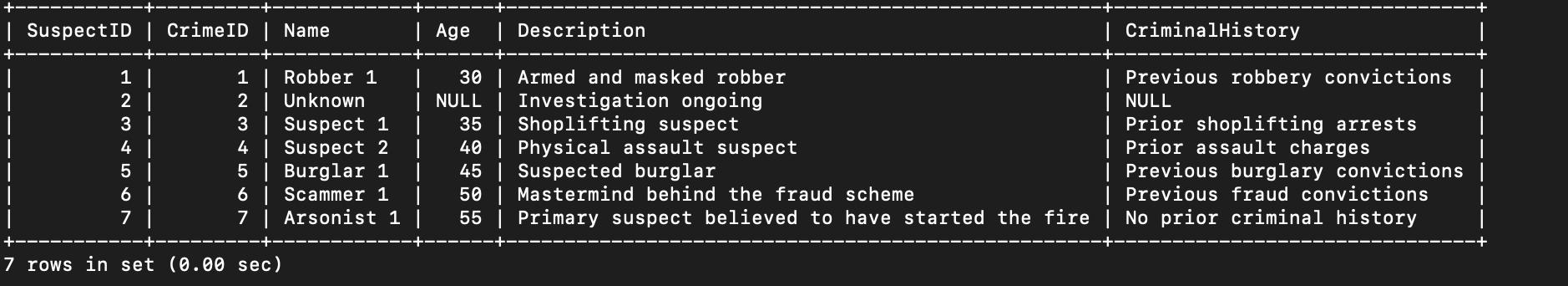
(3, 3, 'Suspect 1',35 , 'Shoplifting suspect', 'Prior shoplifting arrests'),

(4, 4, 'Suspect 2',40 , 'Physical assault suspect', 'Prior assault charges'),

(5, 5, 'Burglar 1',45 , 'Suspected burglar', 'Previous burglary convictions'),

(6, 6, 'Scammer 1',50 , 'Mastermind behind the fraud scheme', 'Previous fraud convictions'),

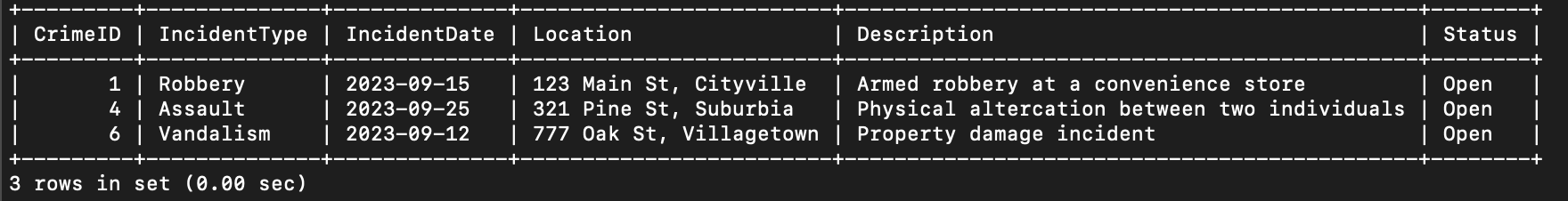
(7, 7, 'Arsonist 1',55 , 'Primary suspect believed to have started the fire', 'No prior criminal history');

****

**Queries:**

1. Select all open incidents.

SELECT \* FROM Crime WHERE Status = 'Open';



1. Find the total number of incidents.

SELECT COUNT(crime\_id) AS TotalIncidents FROM Crime;



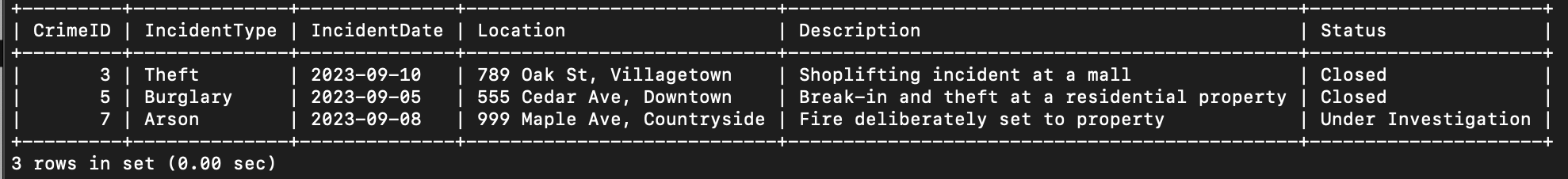
1. List all unique incident types.

SELECT DISTINCT IncidentType FROM Crime;



1. 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';



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

SELECT Name, Age

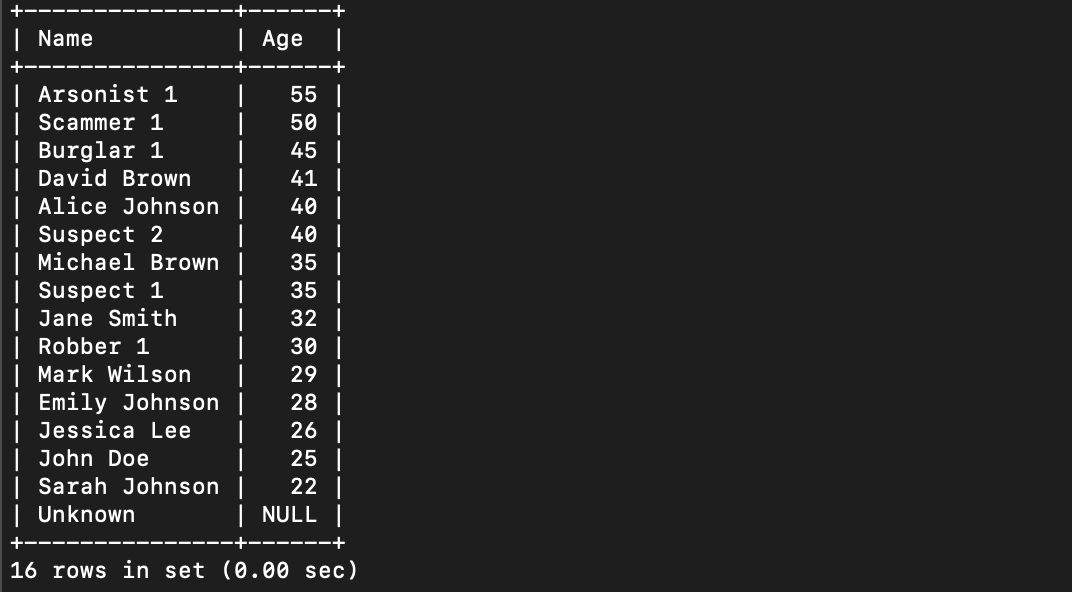
FROM (SELECT Name, Age FROM Victim

UNION ALL

SELECT Name, Age FROM Suspect

) AS temp

ORDER BY Age DESC;



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

SELECT ROUND(AVG(Age),0) AS AverageAge

FROM (SELECT Name, Age FROM Victim

UNION ALL

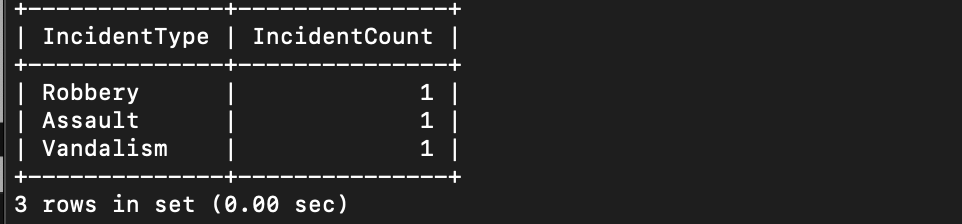
SELECT Name, Age FROM Suspect

) AS temp;



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

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



1. Find persons with names containing 'Doe'.

SELECT name FROM Victim WHERE Name LIKE '%Doe%'

UNION ALL

SELECT name FROM Suspect WHERE Name LIKE '%Doe%';



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

SELECT Name FROM (

SELECT Name FROM Victim

WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE Status = 'Open' or Status = 'Closed' )

UNION

SELECT Name FROM Suspect

WHERE CrimeID IN (SELECT CrimeID FROM Crime WHERE Status = 'Open' or Status = 'Closed' )

) AS temp;



1. 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

WHERE v.age IN (30, 35);

SELECT c.IncidentType

FROM Crime c

WHERE c.CrimeID in

(SELECT v.CrimeID FROM Victim v WHERE age=30 or age=35)



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

SELECT Name FROM (

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')

) AS temp;



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

SELECT IncidentType, COUNT(\*)

FROM Crime

WHERE Status = 'Open'

GROUP BY IncidentType

HAVING COUNT(\*) > 1;

Screenshot 2024-04-18 at 12.06.03 PM

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

SELECT c.\*, v.Name AS VictimName, s.Name AS SuspectName

FROM Crime c

JOIN Victim v ON c.CrimeID = v.CrimeID

JOIN Suspect s ON c.CrimeID = s.CrimeID AND v.Name = s.Name;

Screenshot 2024-04-18 at 12.06.03 PM

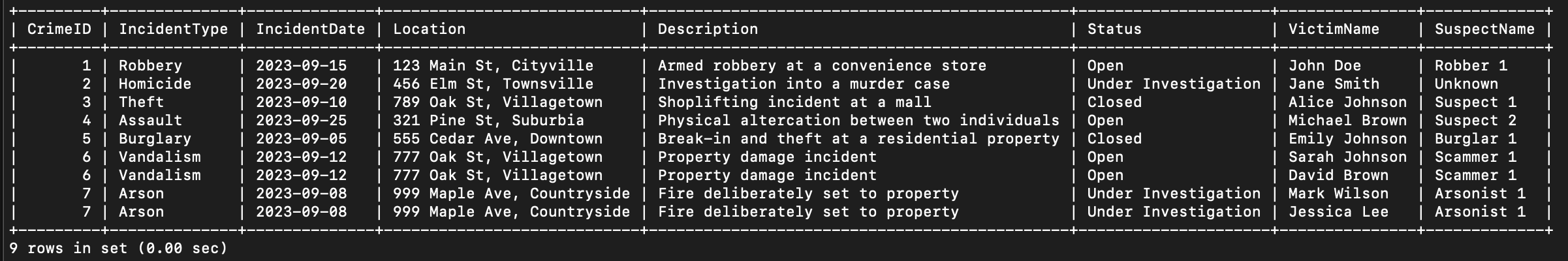
1. Retrieve all incidents along with victim and suspect details.

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;

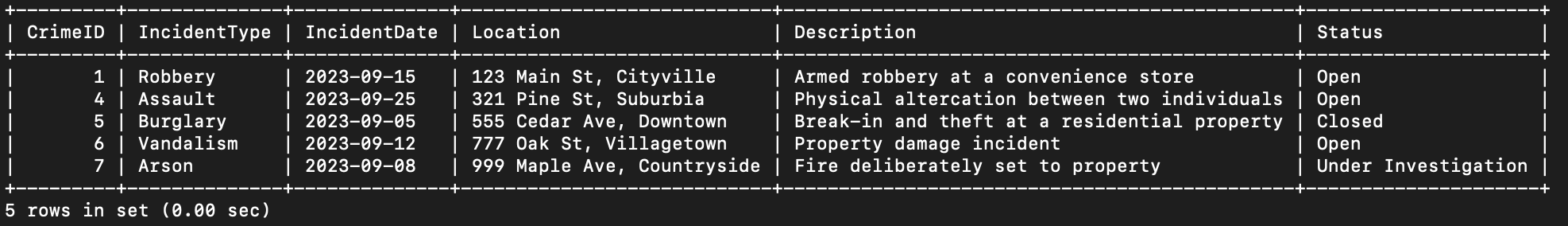


1. Find incidents where the suspect is older than any victim.

SELECT c.\* FROM Crime c

INNER JOIN Suspect s ON c.CrimeID = s.CrimeID

WHERE s.Age > ANY (SELECT v.Age FROM Victim v WHERE v.CrimeID = c.CrimeID);



1. Find suspects involved in multiple incidents:

SELECT SuspectID, Name

FROM Suspect

GROUP BY SuspectID, Name

HAVING COUNT(DISTINCT CrimeID) > 1;

Screenshot 2024-04-18 at 12.06.03 PM

1. List incidents with no suspects involved.

SELECT c.\*

FROM Crime c

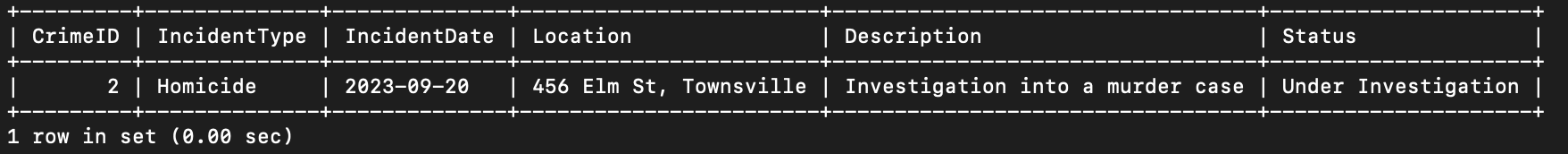
LEFT JOIN Suspect s ON c.CrimeID = s.CrimeID

WHERE s.SuspectID IS NULL;

Screenshot 2024-04-18 at 12.06.03 PM

1. 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';



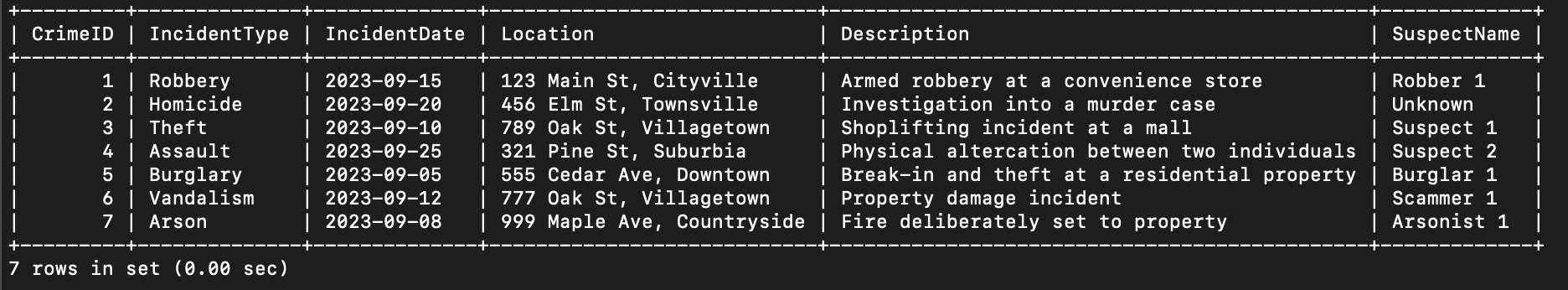
1. Retrieve a list of all incidents and the associated suspects, showing suspects for each incident, or 'No Suspect' if there are none.

SELECT c.CrimeID, c.IncidentType, c.IncidentDate, c.Location, c.Description,

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

FROM Crime c

LEFT JOIN Suspect s ON c.CrimeID = s.CrimeID;



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

SELECT DISTINCT s.SuspectID, s.Name

FROM Suspect s

INNER JOIN Crime c ON s.CrimeID = c.CrimeID

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

****