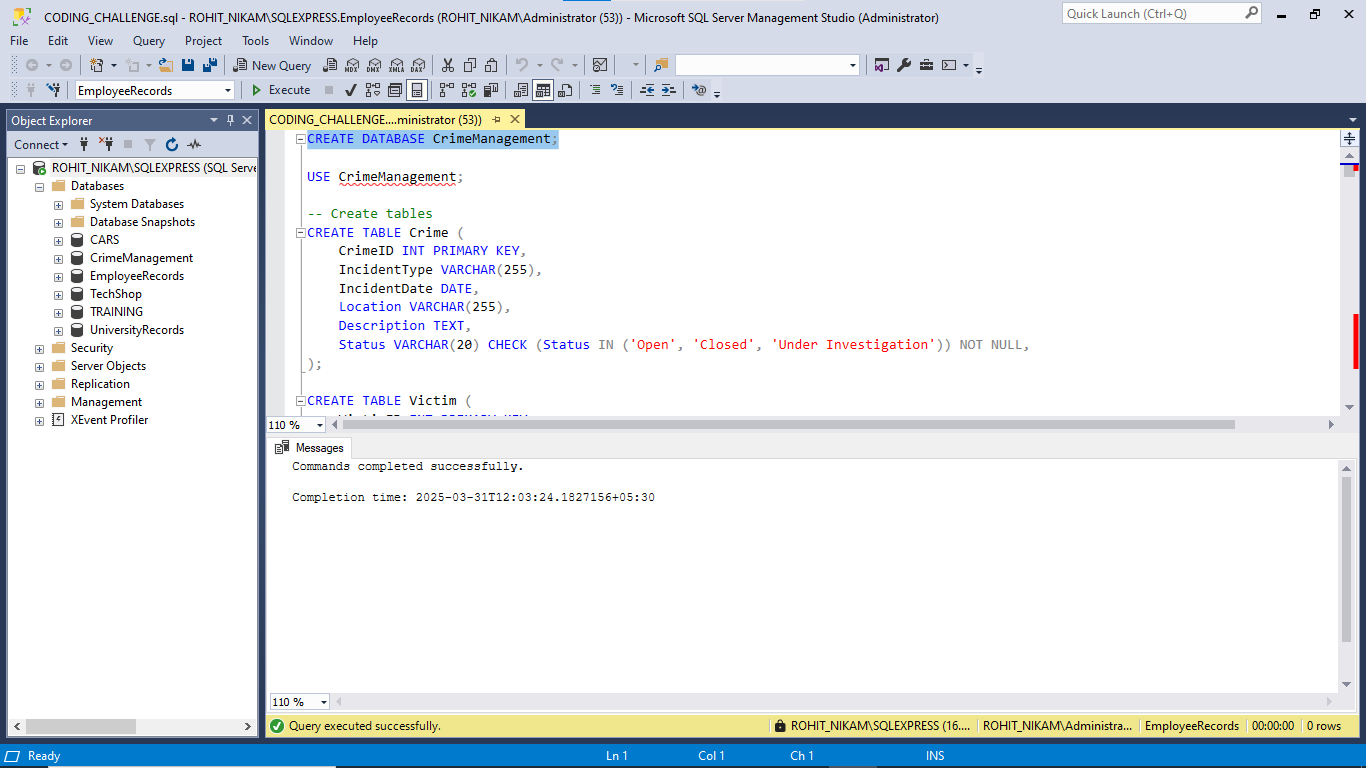
**CODING CHALLENGE – MSSQL**

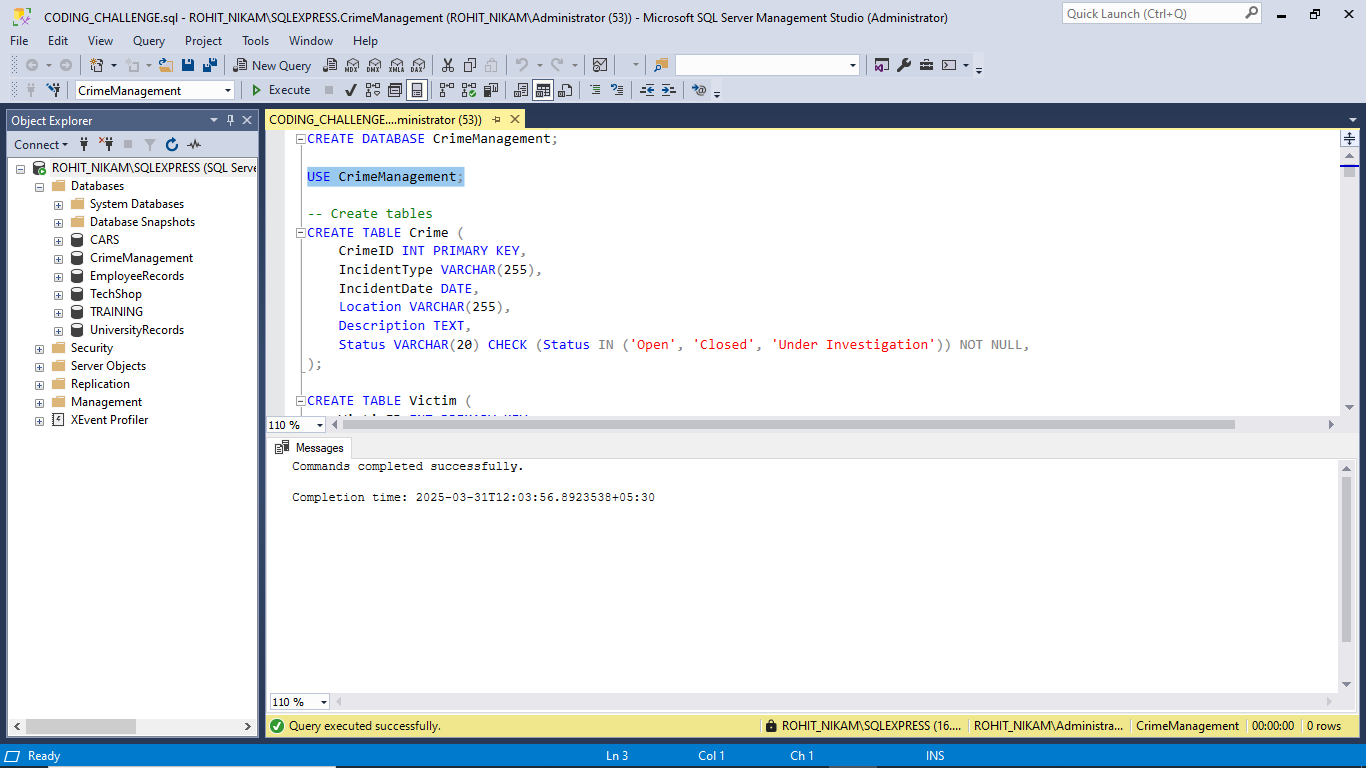
**NAME : ROHIT NIKAM**

**BATCH : PYTHON BATCH 1**

CREATE DATABASE CrimeManagement;



USE CrimeManagement;



-- Create tables

CREATE TABLE Crime (

CrimeID INT PRIMARY KEY,

IncidentType VARCHAR(255),

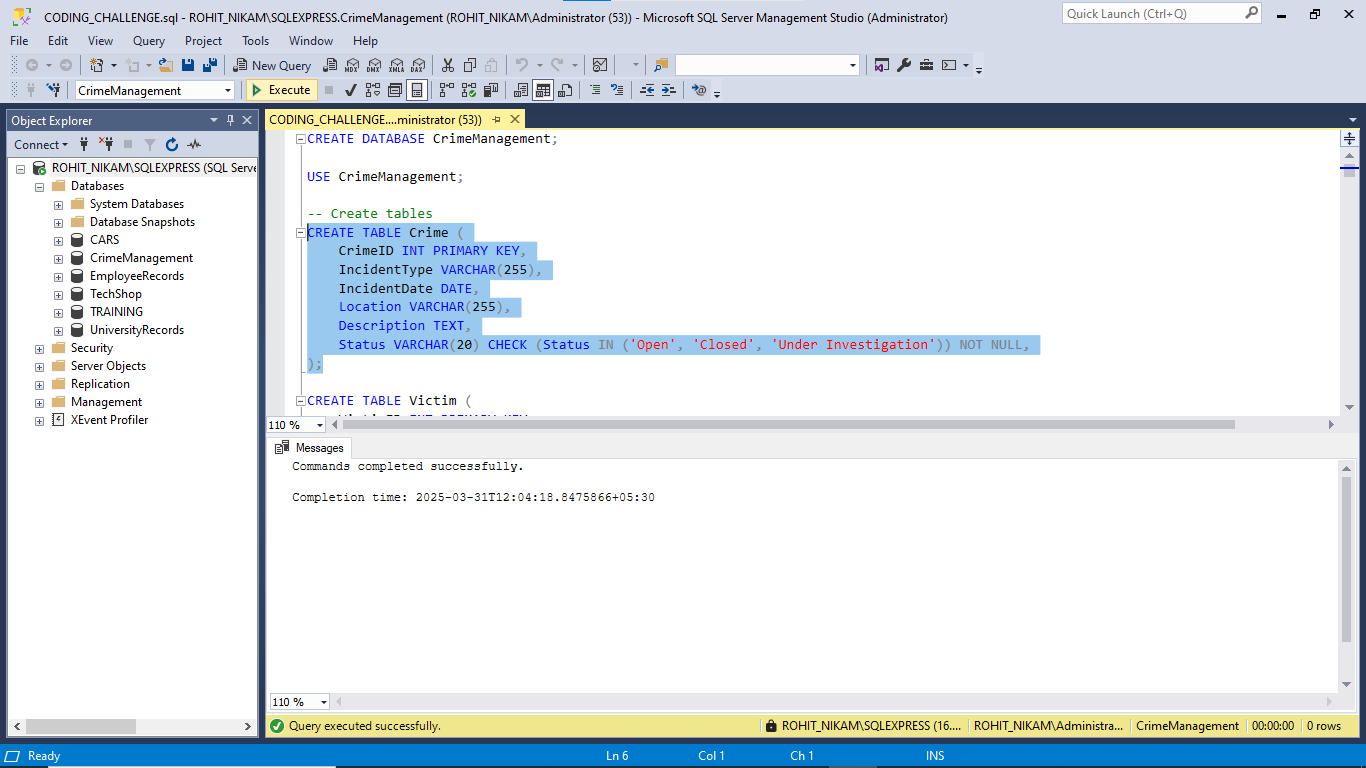
IncidentDate DATE,

Location VARCHAR(255),

Description TEXT,

Status VARCHAR(20) CHECK (Status IN ('Open', 'Closed', 'Under Investigation')) NOT NULL,

);



CREATE TABLE Victim (

VictimID INT PRIMARY KEY,

CrimeID INT,

Name VARCHAR(255),

ContactInfo VARCHAR(255),

Injuries VARCHAR(255),

FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID) ON DELETE CASCADE

);



CREATE TABLE Suspect (

SuspectID INT PRIMARY KEY,

CrimeID INT,

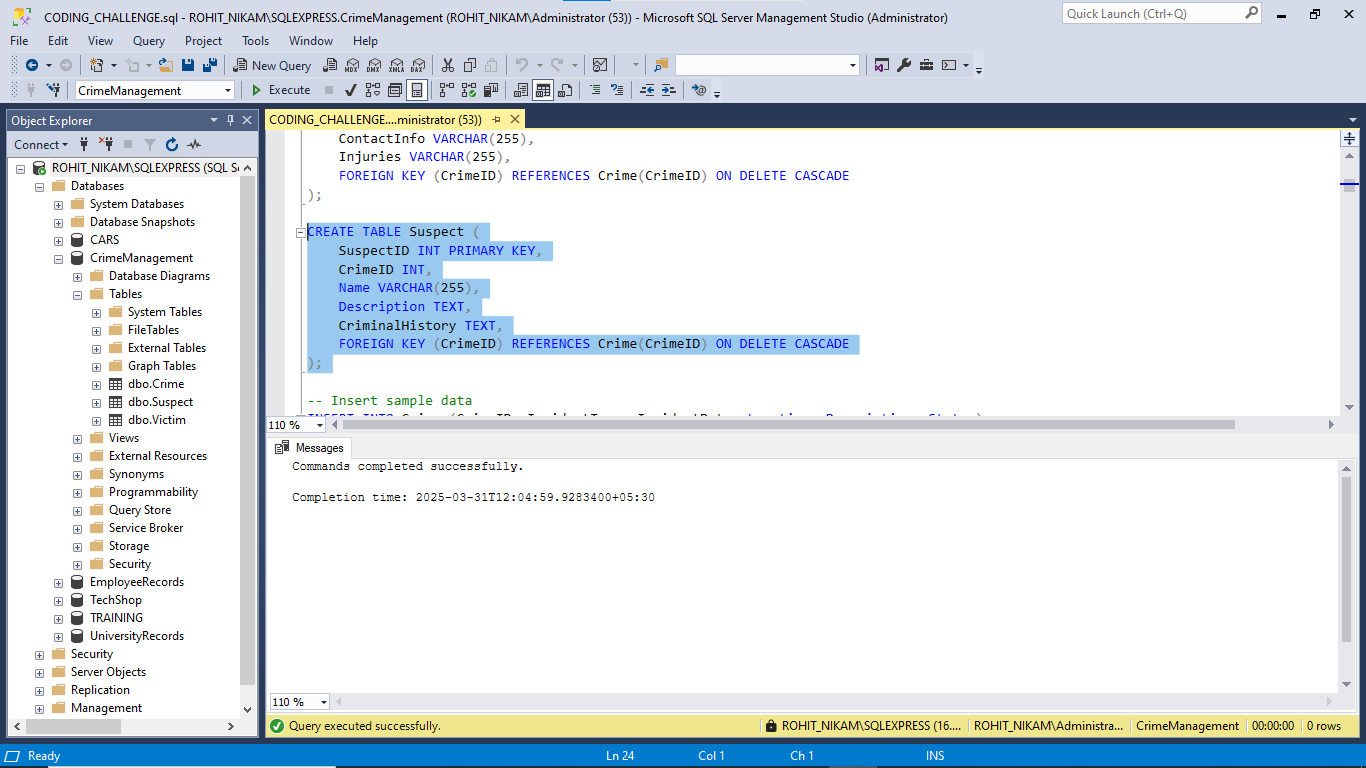
Name VARCHAR(255),

Description TEXT,

CriminalHistory TEXT,

FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID) ON DELETE CASCADE

);



-- Insert sample data

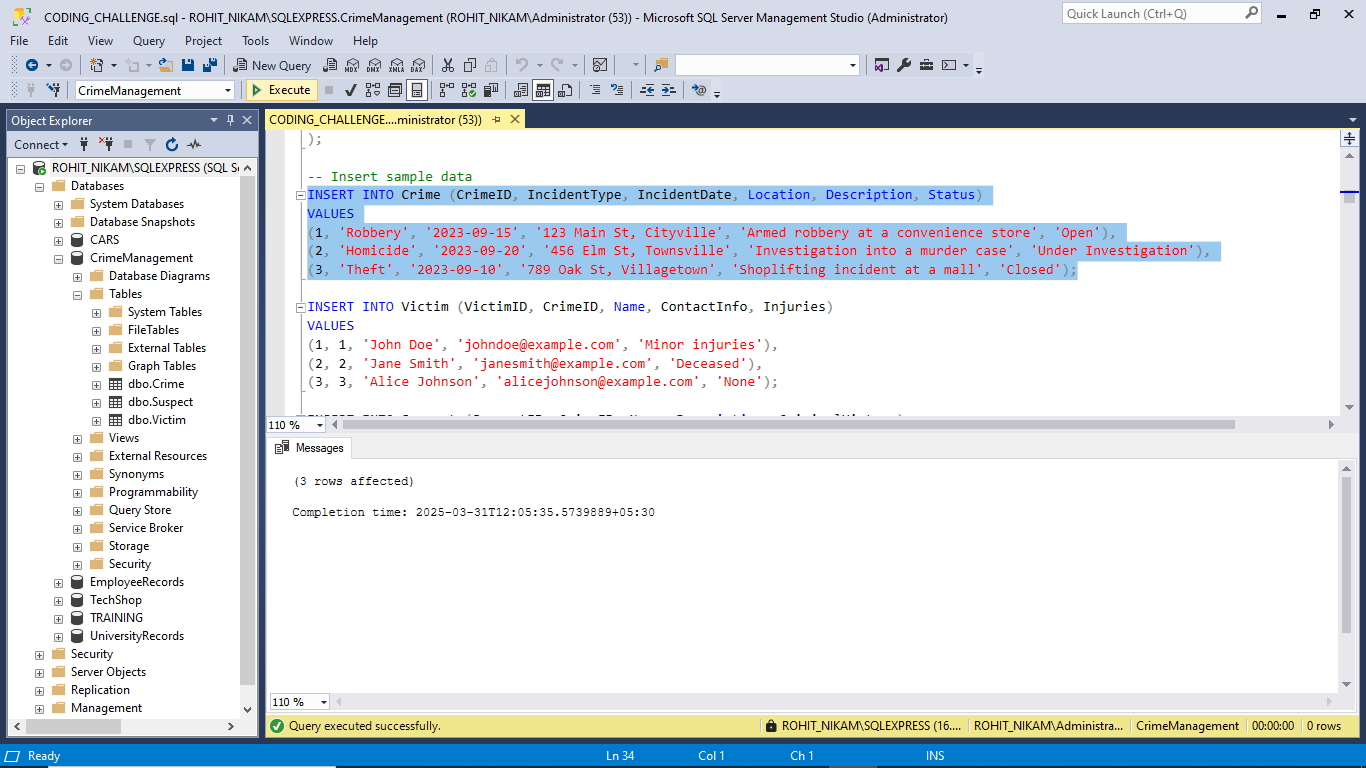
INSERT INTO Crime (CrimeID, IncidentType, IncidentDate, Location, Description, Status)

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');



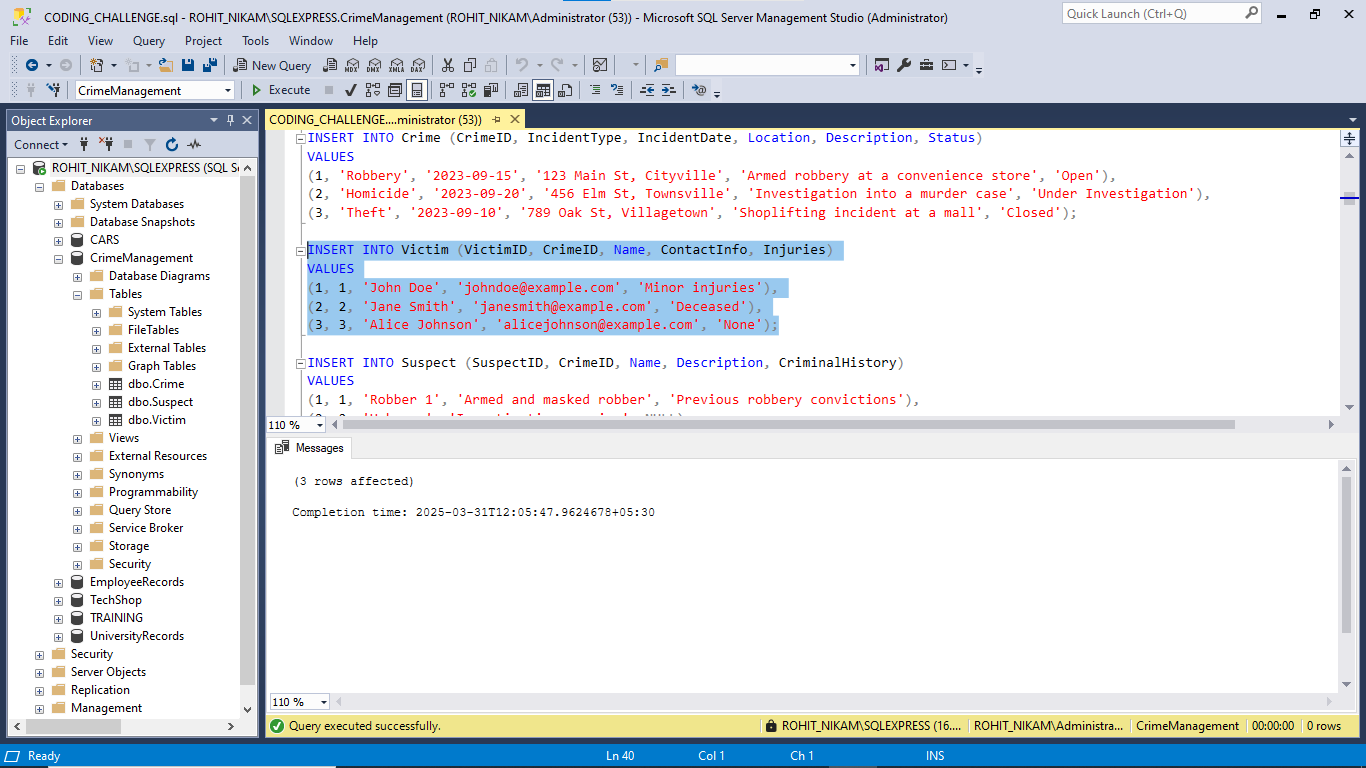
INSERT INTO Victim (VictimID, CrimeID, Name, ContactInfo, Injuries)

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');



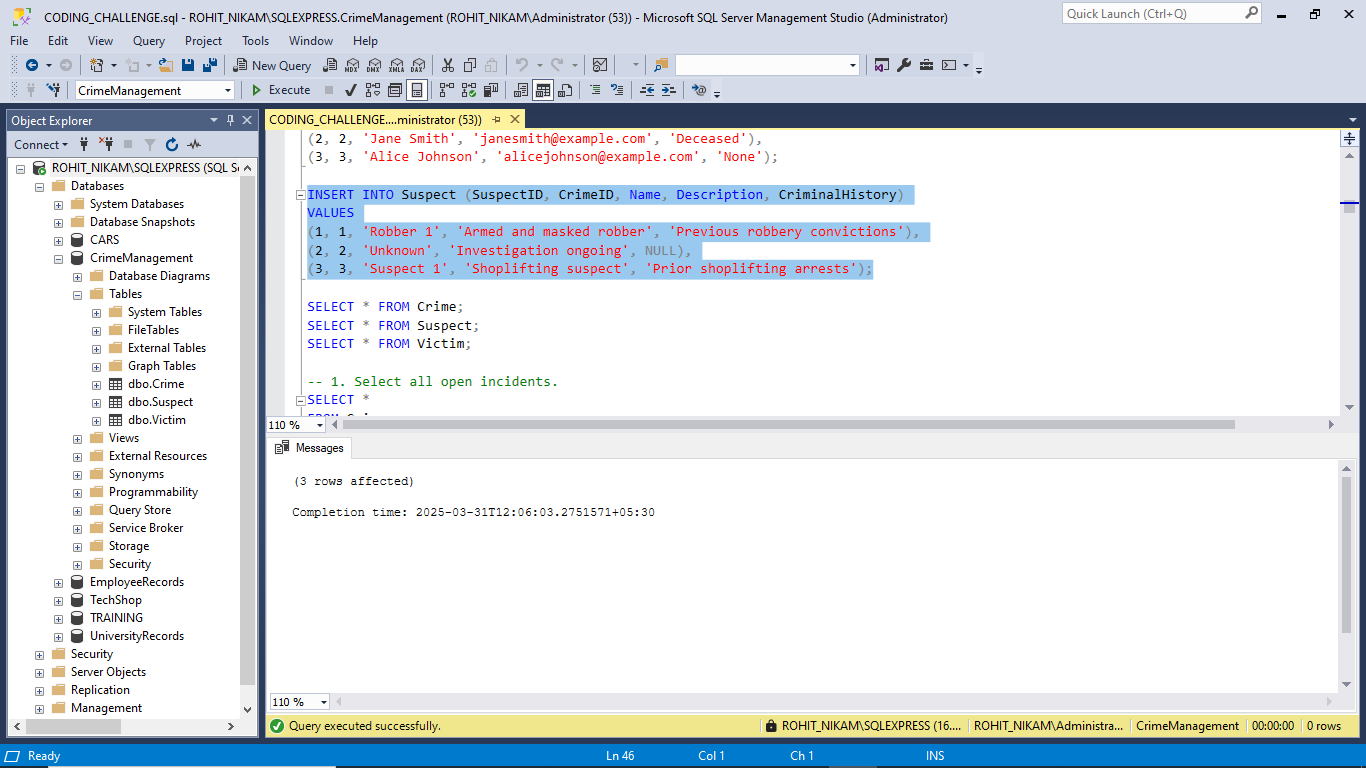
INSERT INTO Suspect (SuspectID, CrimeID, Name, Description, CriminalHistory)

VALUES

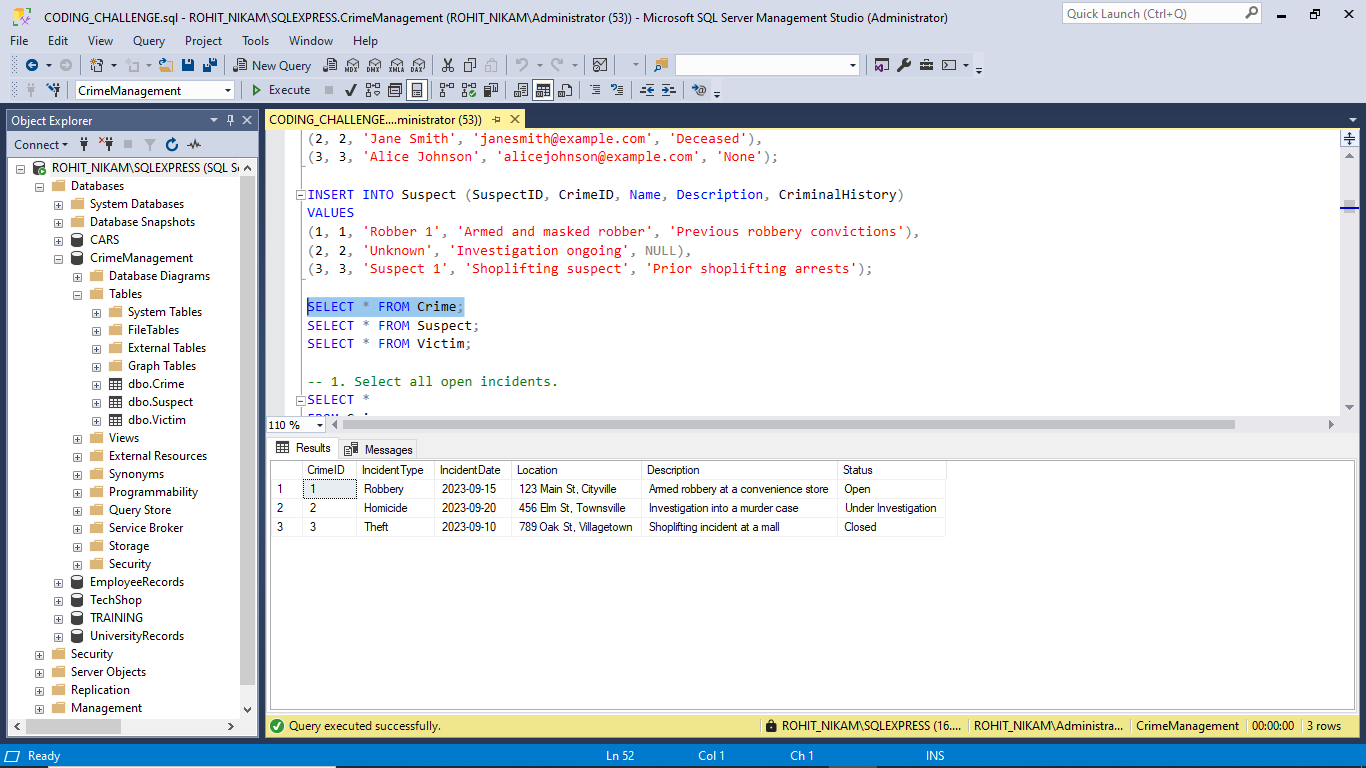
(1, 1, 'Robber 1', 'Armed and masked robber', 'Previous robbery convictions'),

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

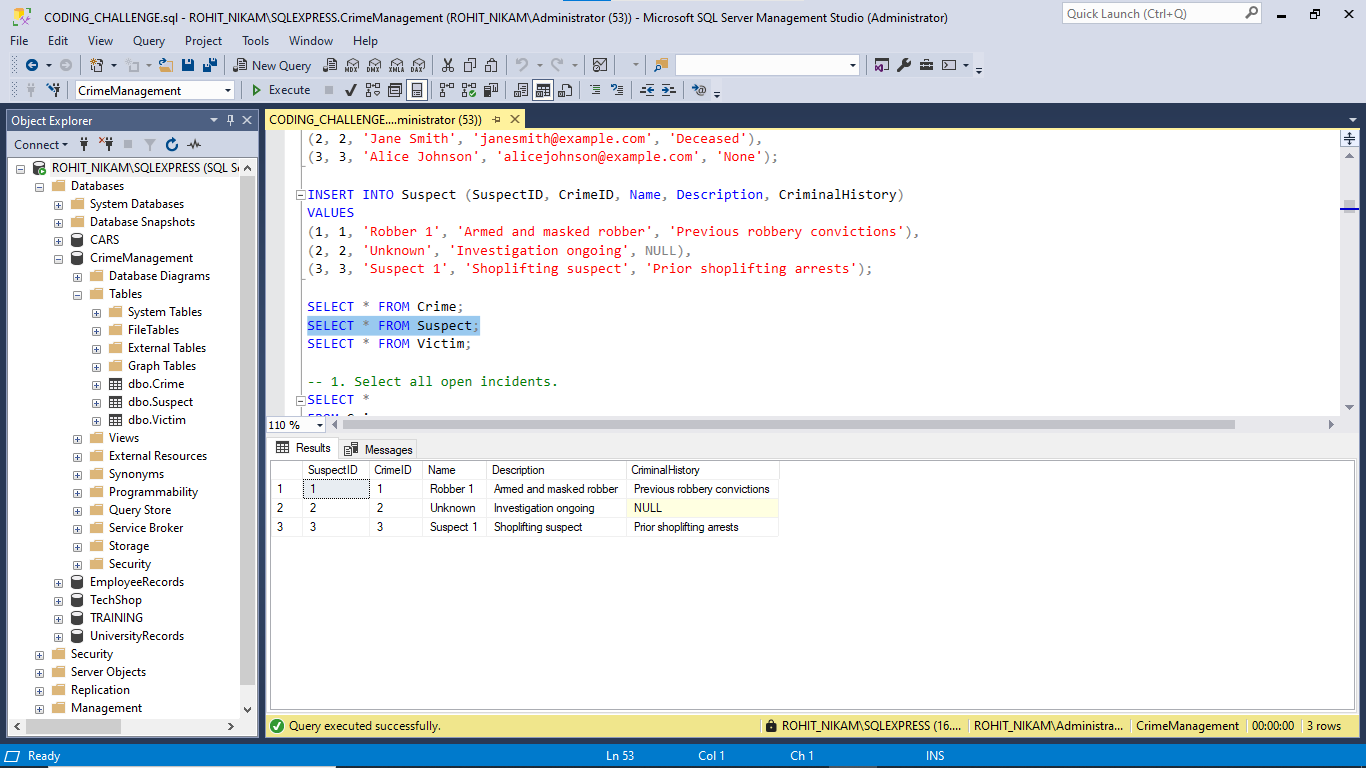
(3, 3, 'Suspect 1', 'Shoplifting suspect', 'Prior shoplifting arrests');



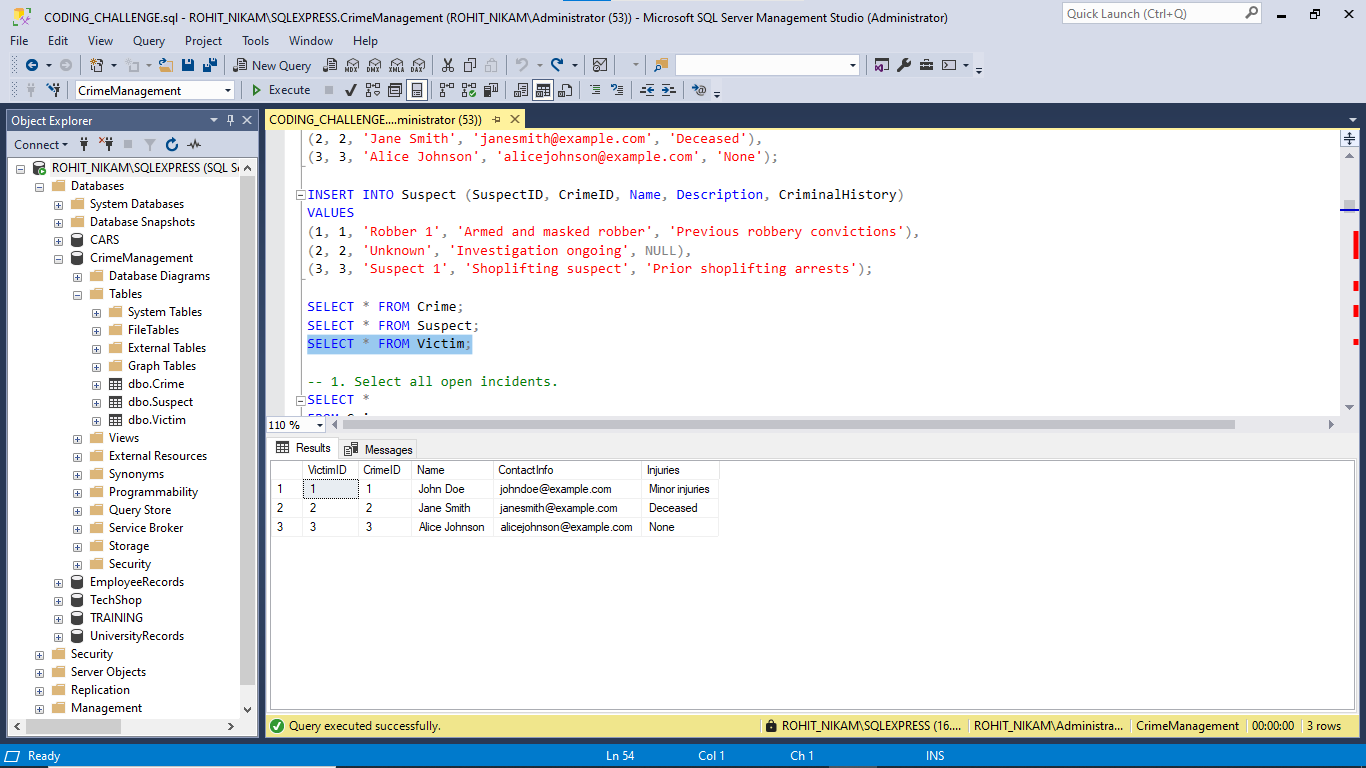
SELECT \* FROM Crime;



SELECT \* FROM Suspect;



SELECT \* FROM Victim;

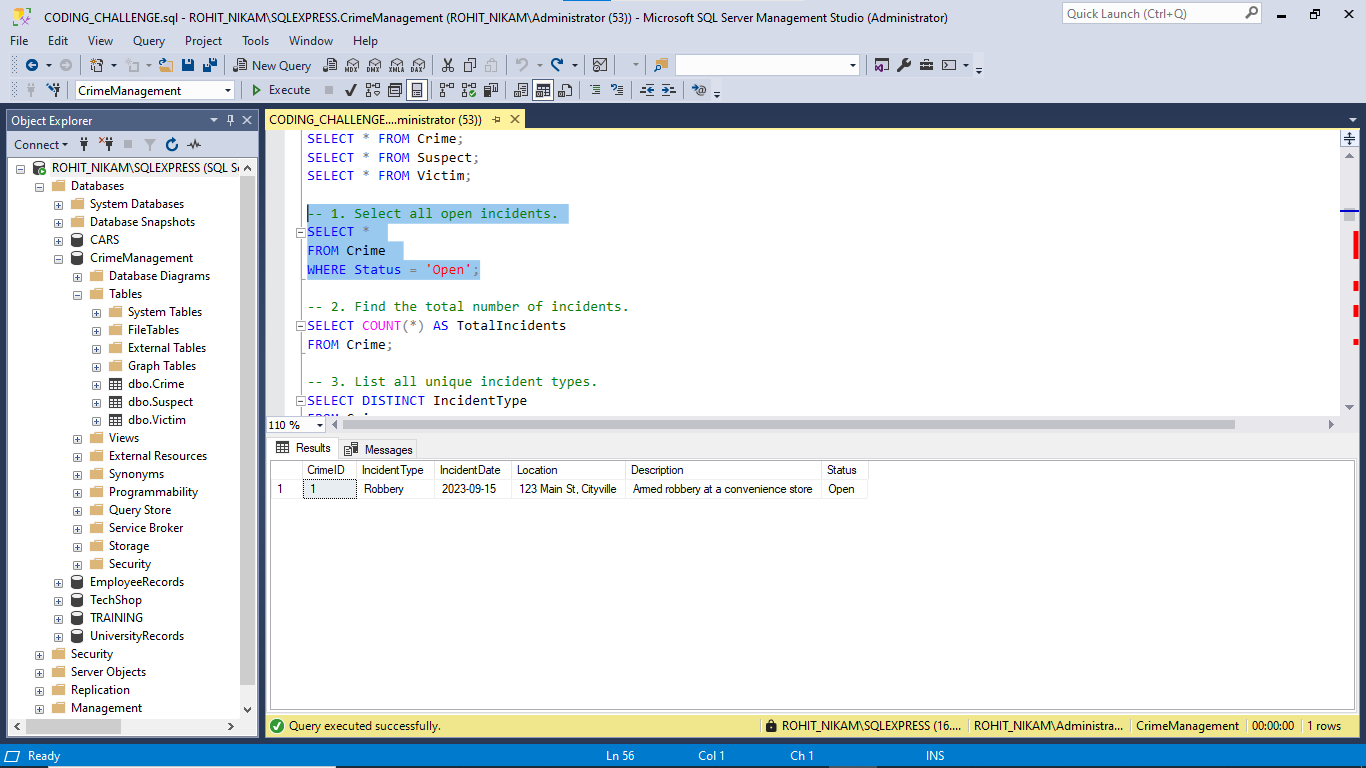


-- 1. Select all open incidents.

SELECT \*

FROM Crime

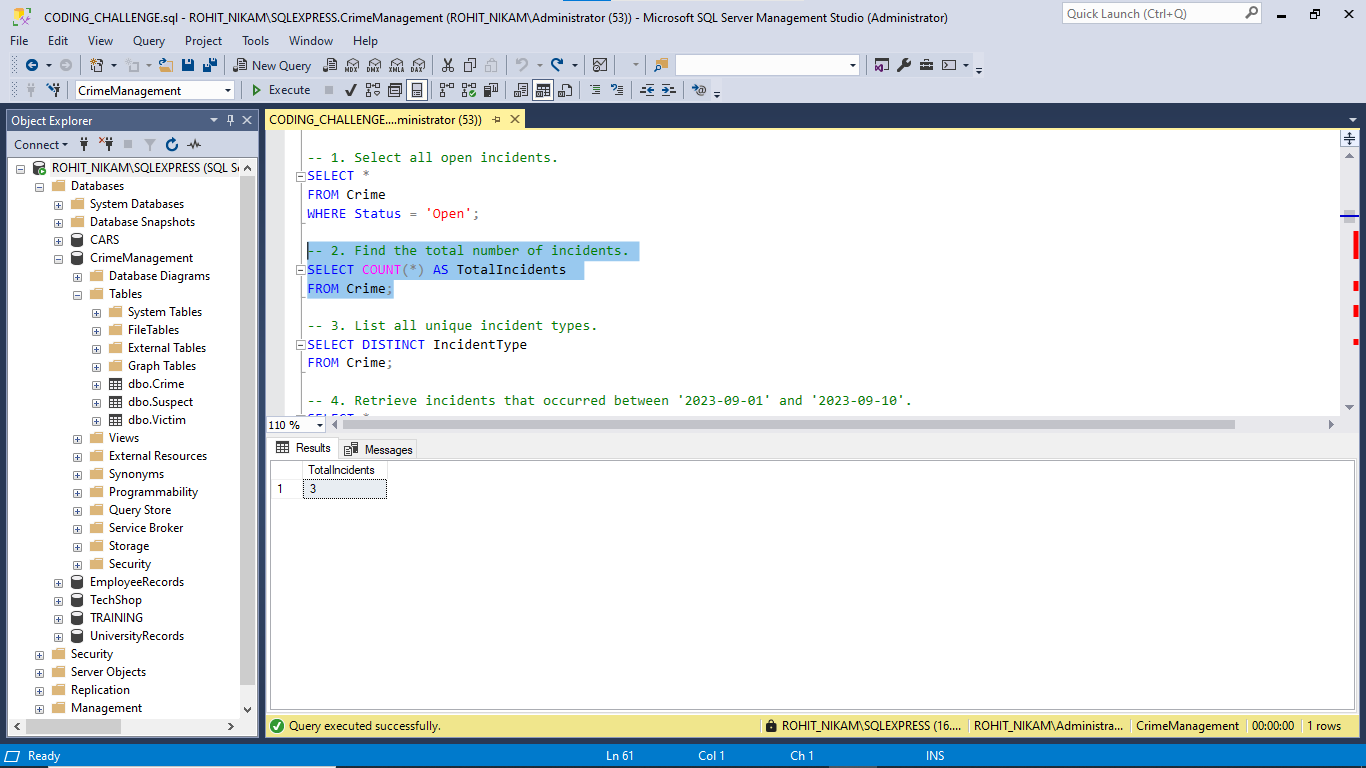
WHERE Status = 'Open';



-- 2. Find the total number of incidents.

SELECT COUNT(\*) AS TotalIncidents

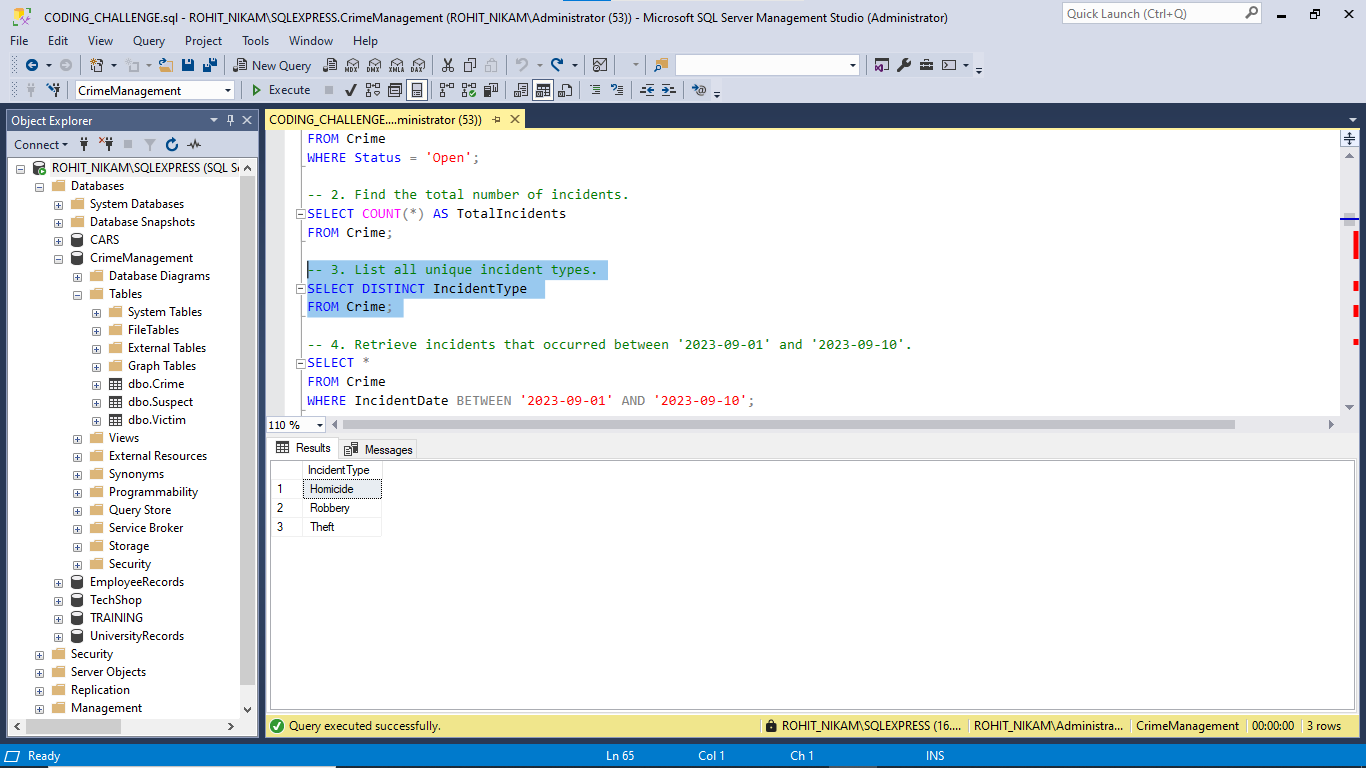
FROM Crime;



-- 3. List all unique incident types.

SELECT DISTINCT IncidentType

FROM Crime;

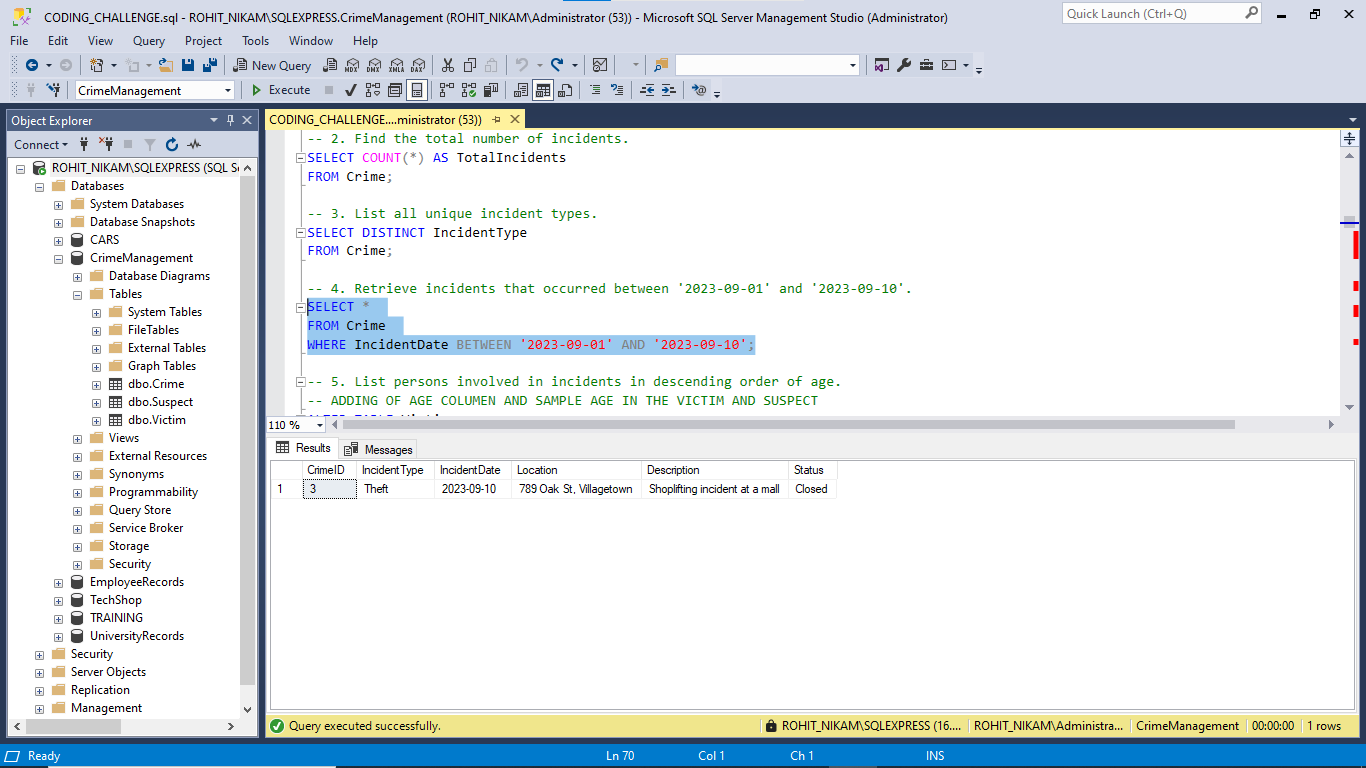


-- 4. Retrieve incidents that occurred between '2023-09-01' and '2023-09-10'.

SELECT \*

FROM Crime

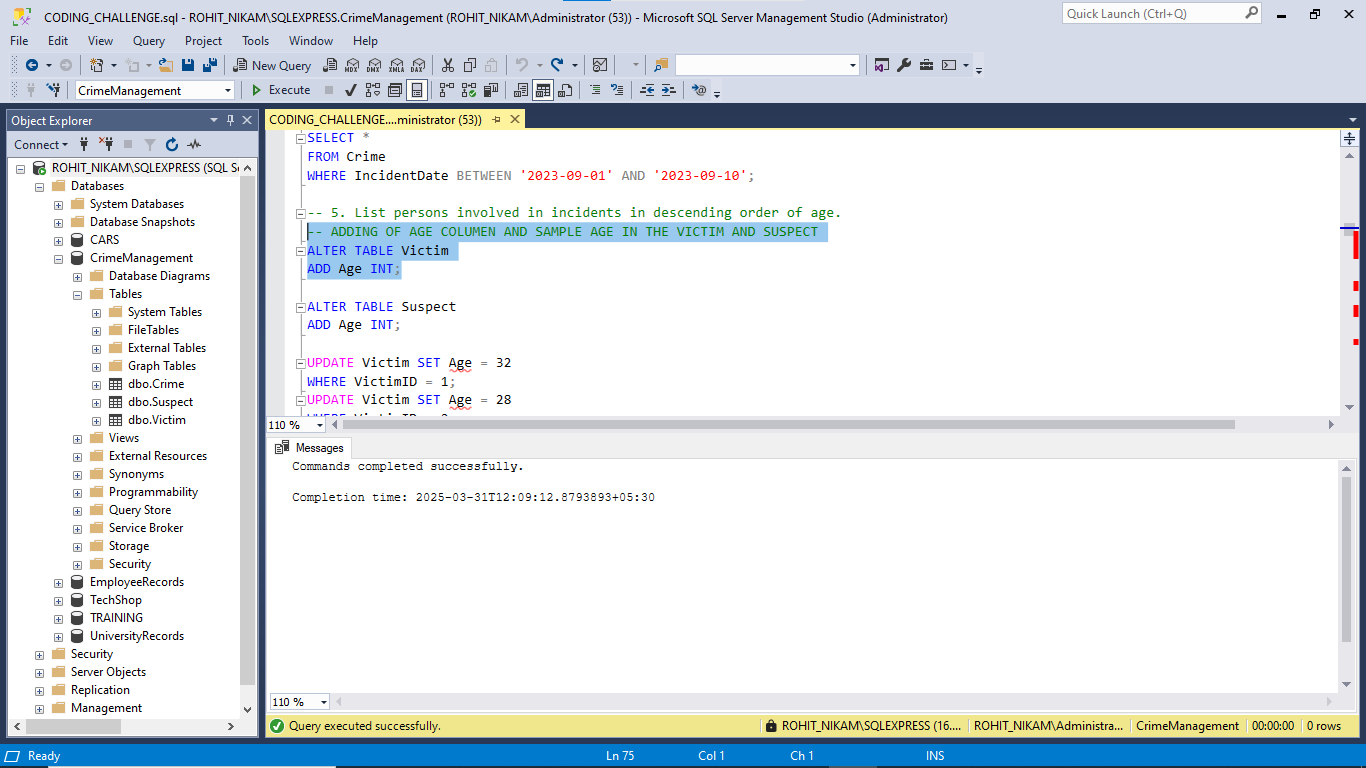
WHERE IncidentDate BETWEEN '2023-09-01' AND '2023-09-10';



-- ADDING OF AGE COLUMEN AND SAMPLE AGE IN THE VICTIM AND SUSPECT

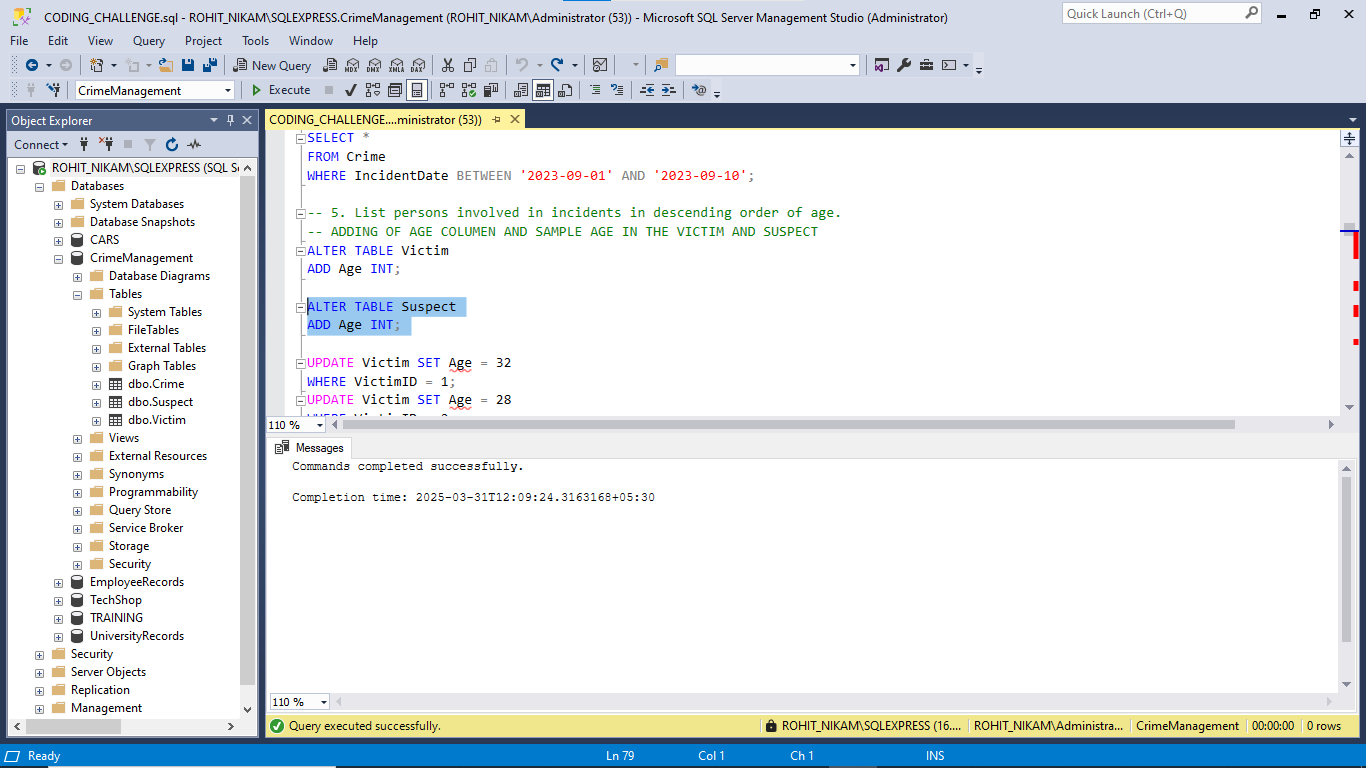
ALTER TABLE Victim

ADD Age INT;



ALTER TABLE Suspect

ADD Age INT;



UPDATE Victim SET Age = 32

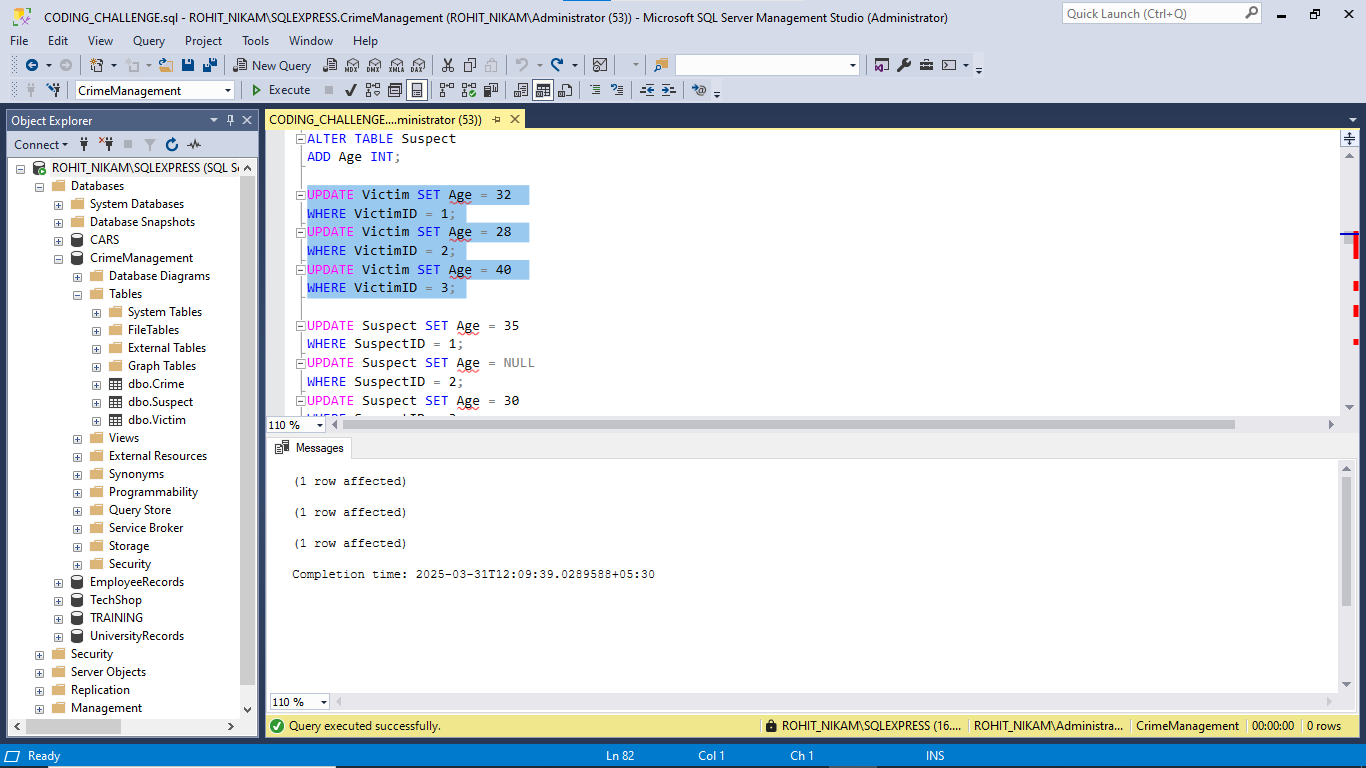
WHERE VictimID = 1;

UPDATE Victim SET Age = 28

WHERE VictimID = 2;

UPDATE Victim SET Age = 40

WHERE VictimID = 3;



UPDATE Suspect SET Age = 35

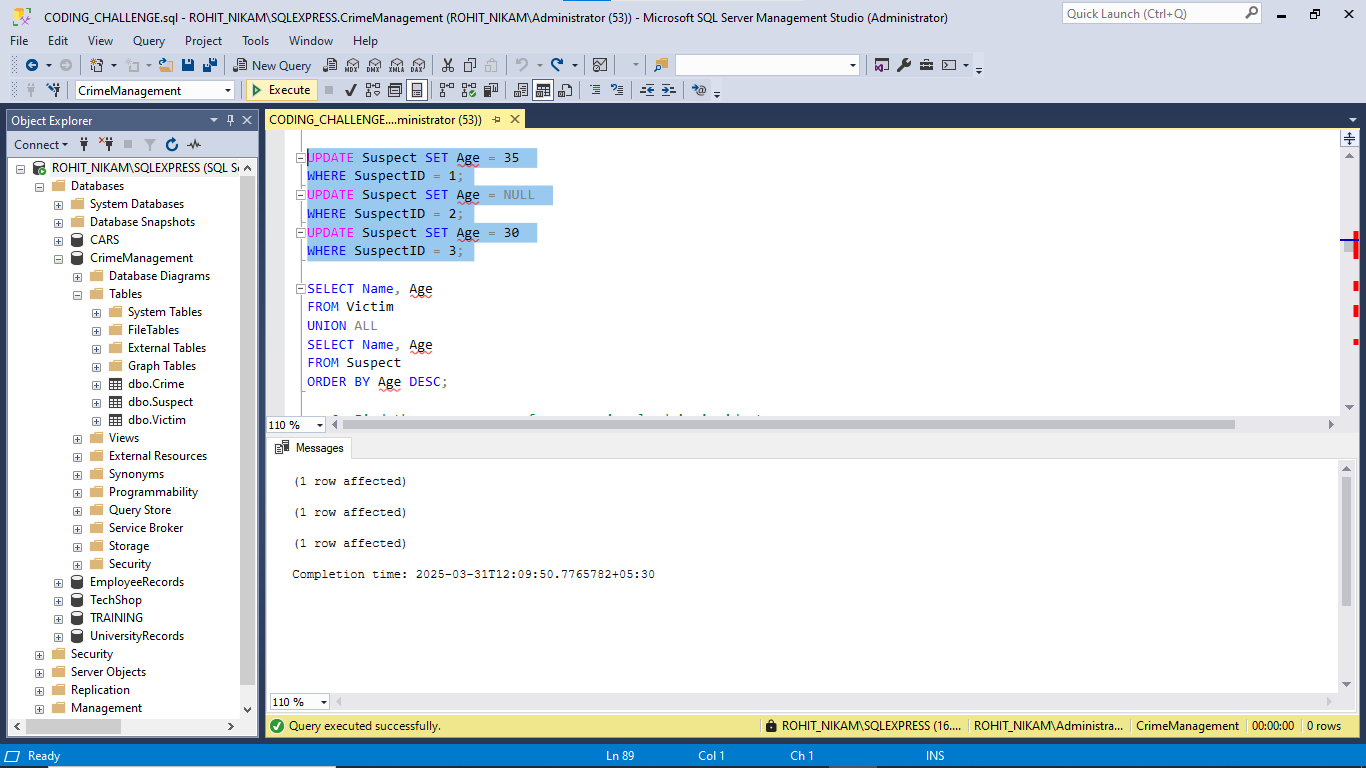
WHERE SuspectID = 1;

UPDATE Suspect SET Age = NULL

WHERE SuspectID = 2;

UPDATE Suspect SET Age = 30

WHERE SuspectID = 3;



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

SELECT Name, Age

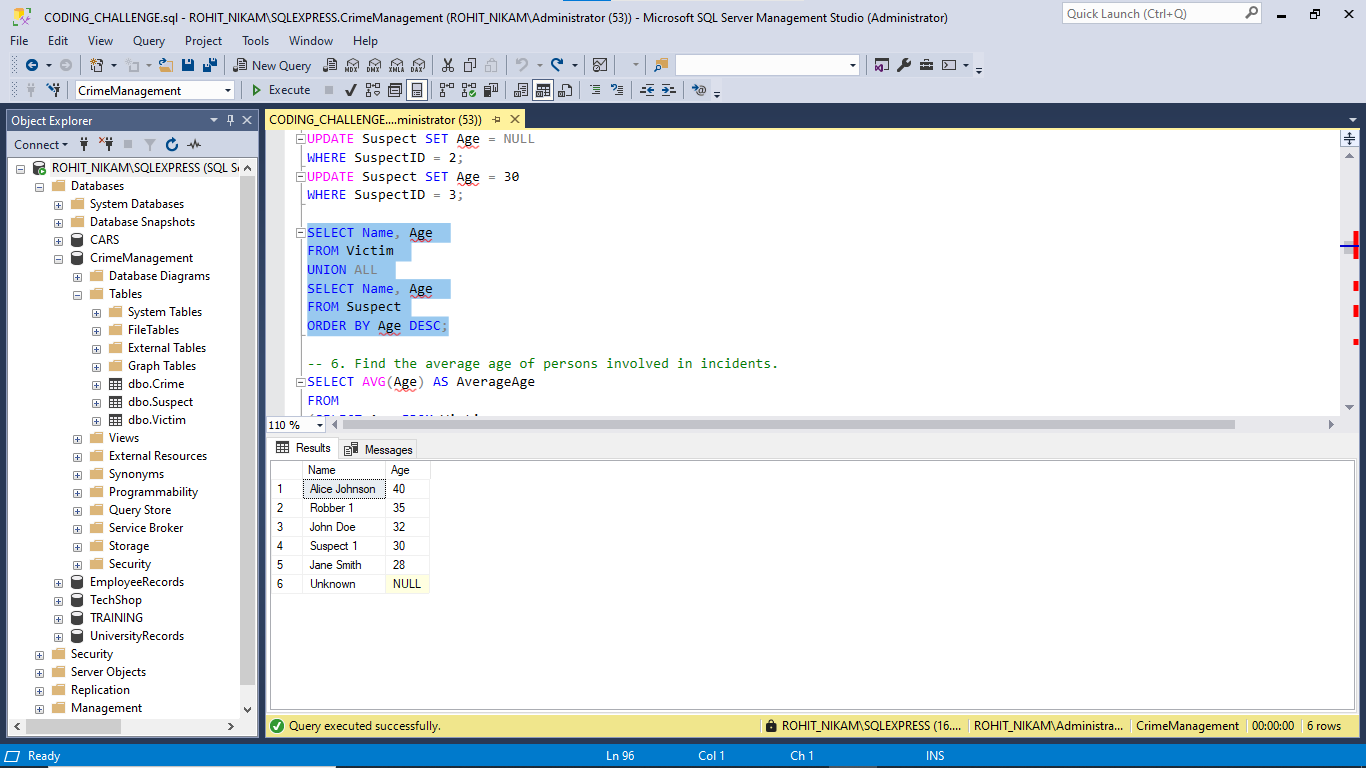
FROM Victim

UNION ALL

SELECT Name, Age

FROM Suspect

ORDER BY Age DESC;



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

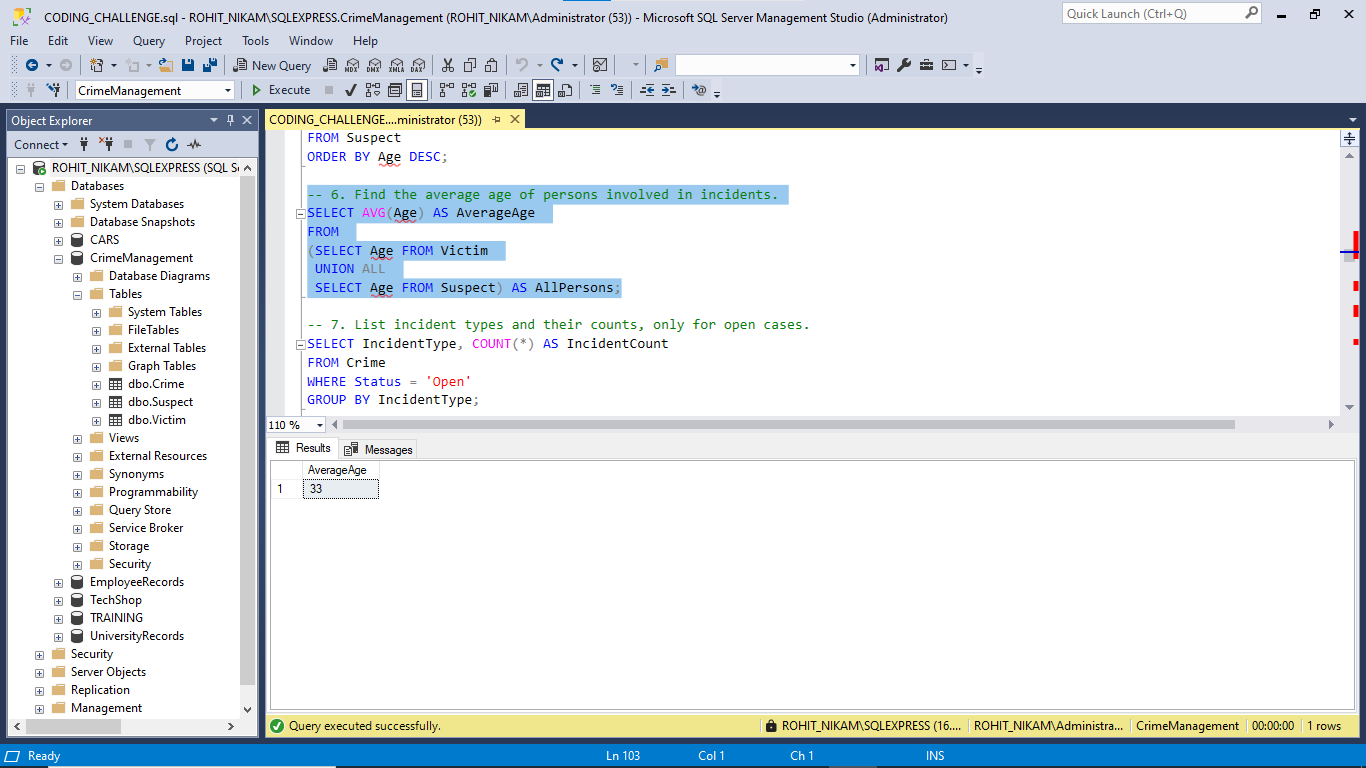
SELECT AVG(Age) AS AverageAge

FROM

(SELECT Age FROM Victim

UNION ALL

SELECT Age FROM Suspect) AS AllPersons;



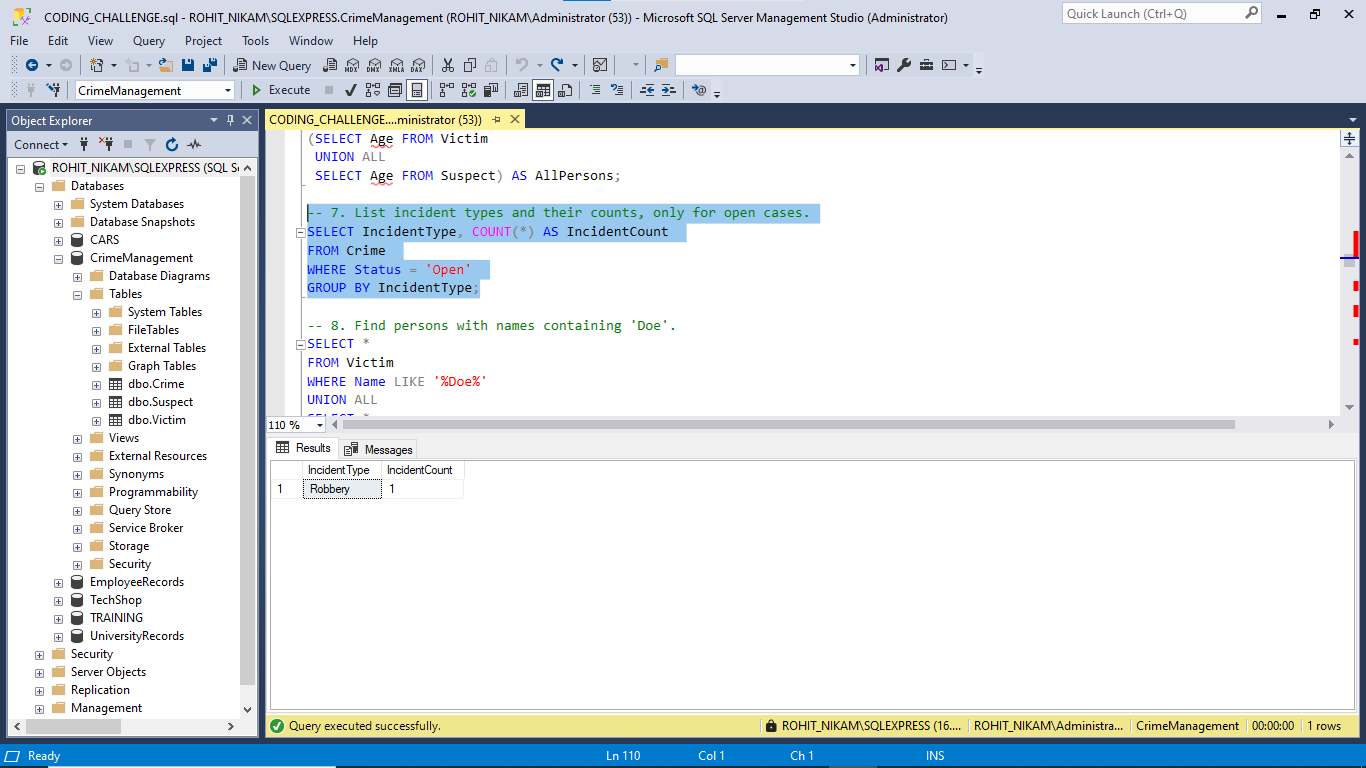
-- 7. List incident types and their counts, only for open cases.

SELECT IncidentType, COUNT(\*) AS IncidentCount

FROM Crime

WHERE Status = 'Open'

GROUP BY IncidentType;



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

SELECT \*

FROM Victim

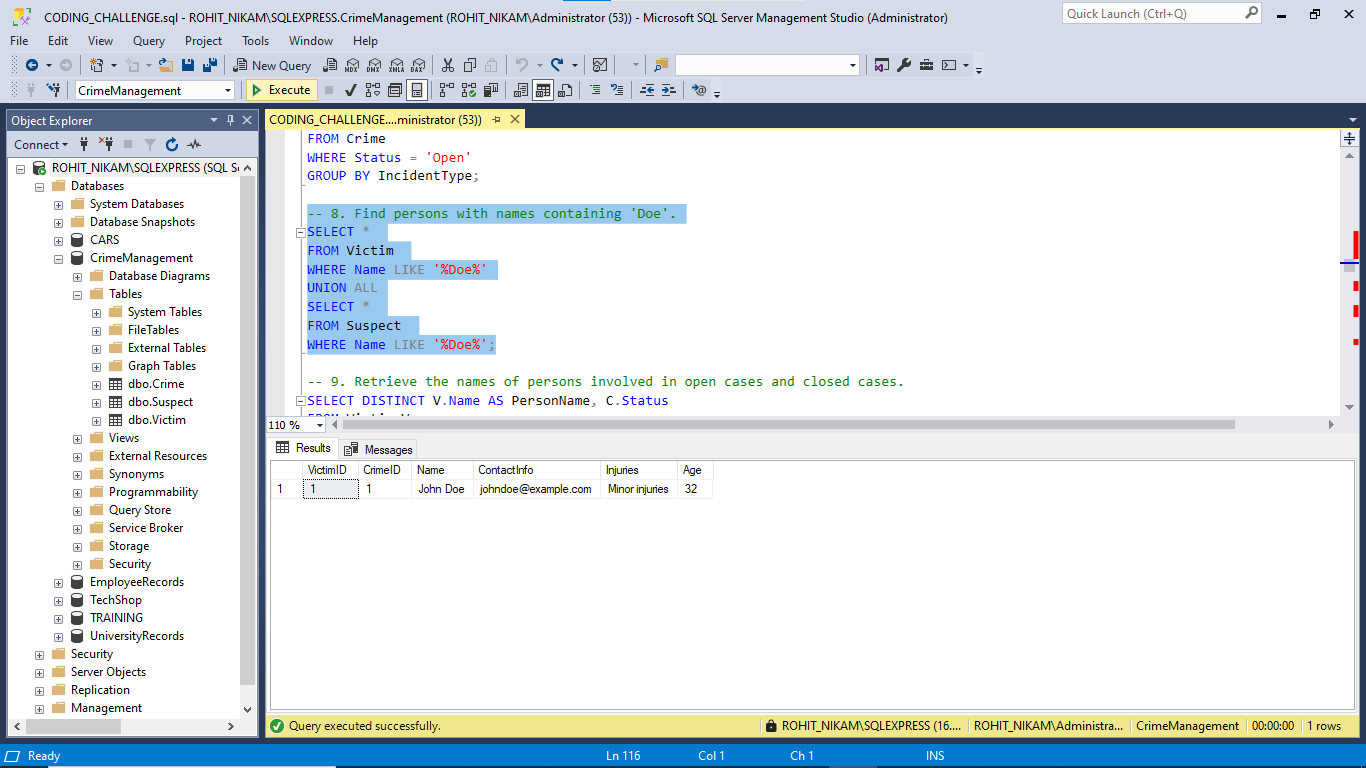
WHERE Name LIKE '%Doe%'

UNION ALL

SELECT \*

FROM Suspect

WHERE Name LIKE '%Doe%';



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

SELECT DISTINCT V.Name AS PersonName, C.Status

FROM Victim V

JOIN Crime C ON V.CrimeID = C.CrimeID

WHERE C.Status IN ('Open', 'Closed')

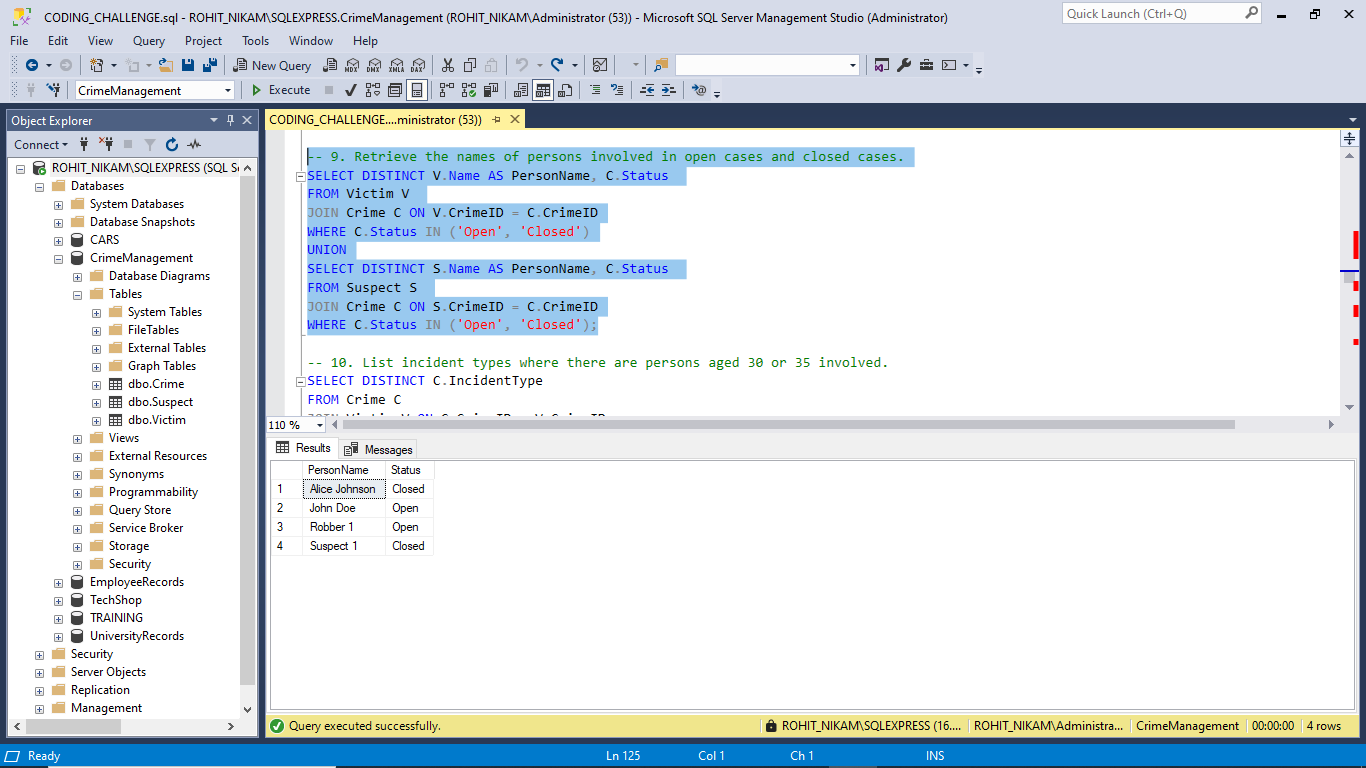
UNION

SELECT DISTINCT S.Name AS PersonName, C.Status

FROM Suspect S

JOIN Crime C ON S.CrimeID = C.CrimeID

WHERE C.Status IN ('Open', 'Closed');



-- 10. 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)

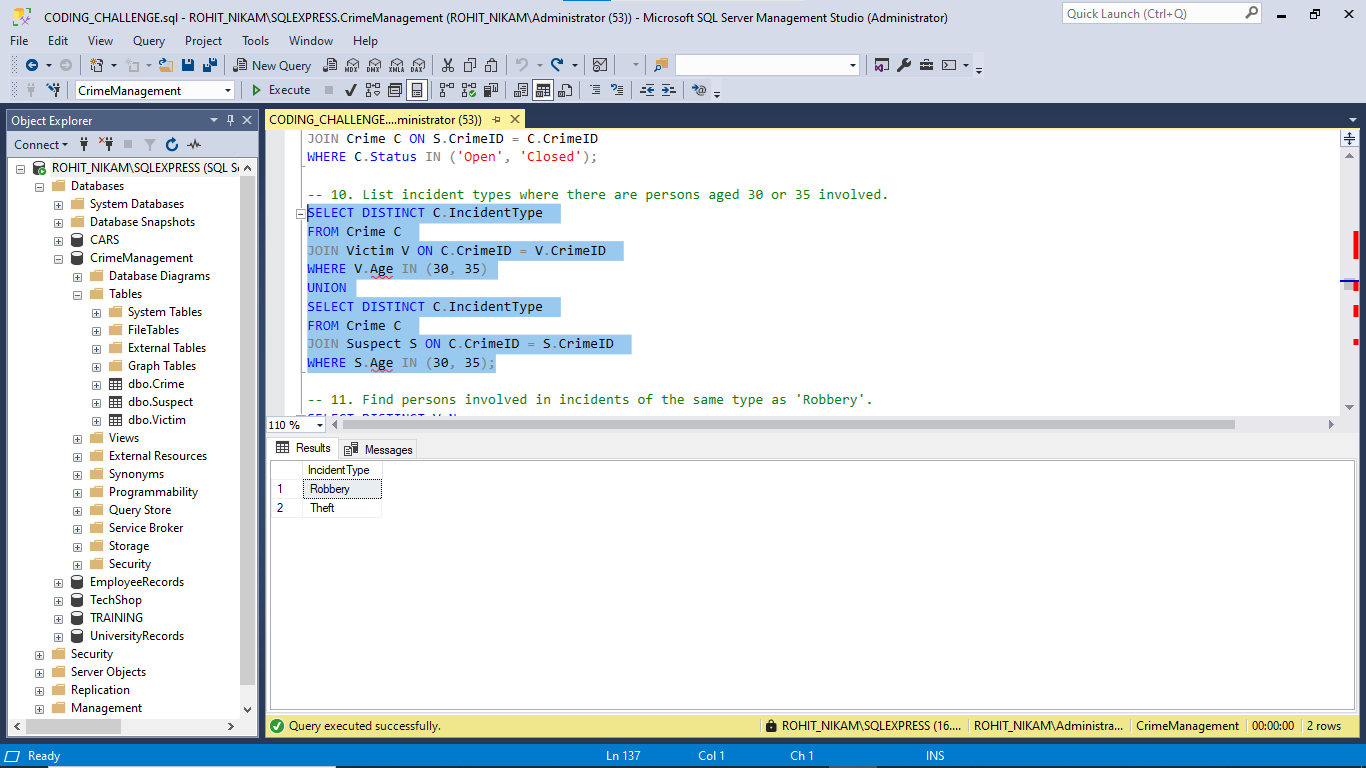
UNION

SELECT DISTINCT C.IncidentType

FROM Crime C

JOIN Suspect S ON C.CrimeID = S.CrimeID

WHERE S.Age IN (30, 35);



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

SELECT DISTINCT V.Name

FROM Victim V

JOIN Crime C ON V.CrimeID = C.CrimeID

WHERE C.IncidentType = 'Robbery'

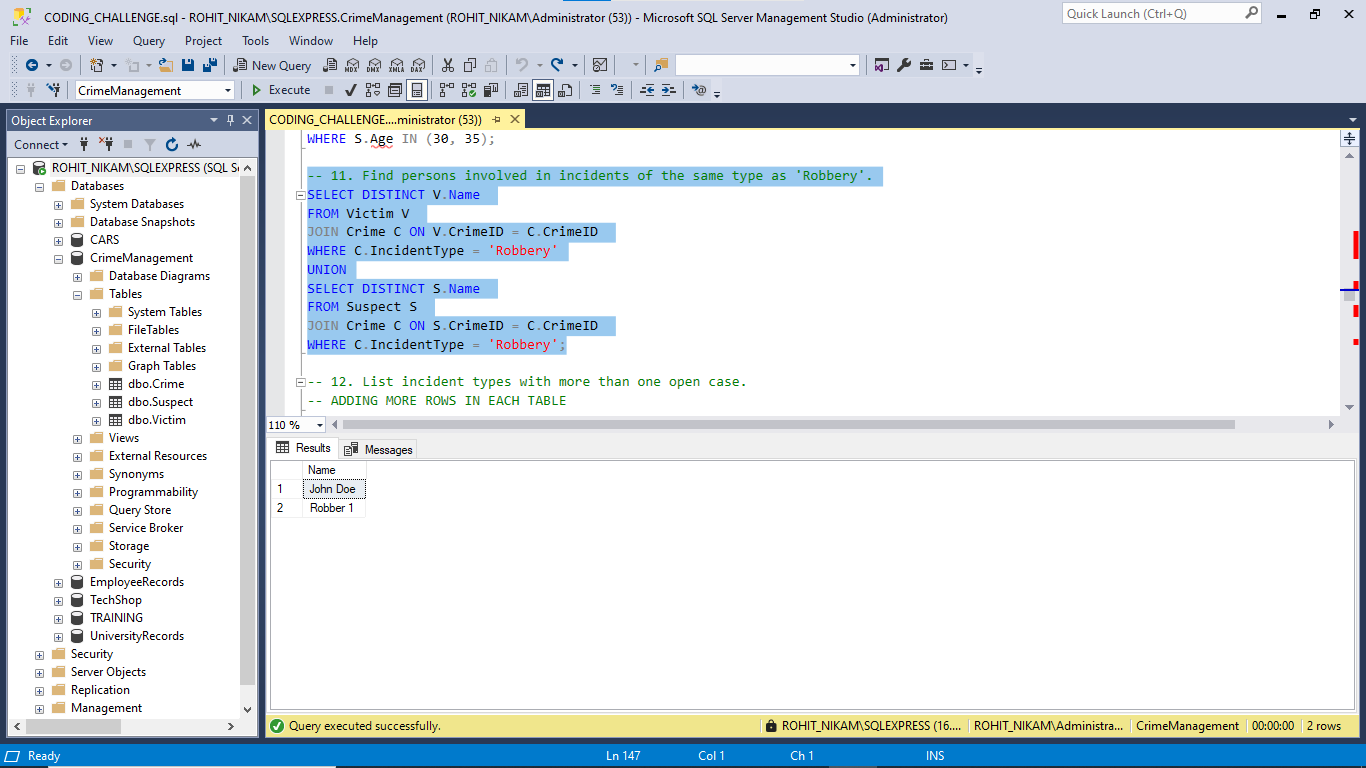
UNION

SELECT DISTINCT S.Name

FROM Suspect S

JOIN Crime C ON S.CrimeID = C.CrimeID

WHERE C.IncidentType = 'Robbery';



-- ADDING MORE ROWS IN EACH TABLE

INSERT INTO Crime (CrimeID, IncidentType, IncidentDate, Location, Description, Status)

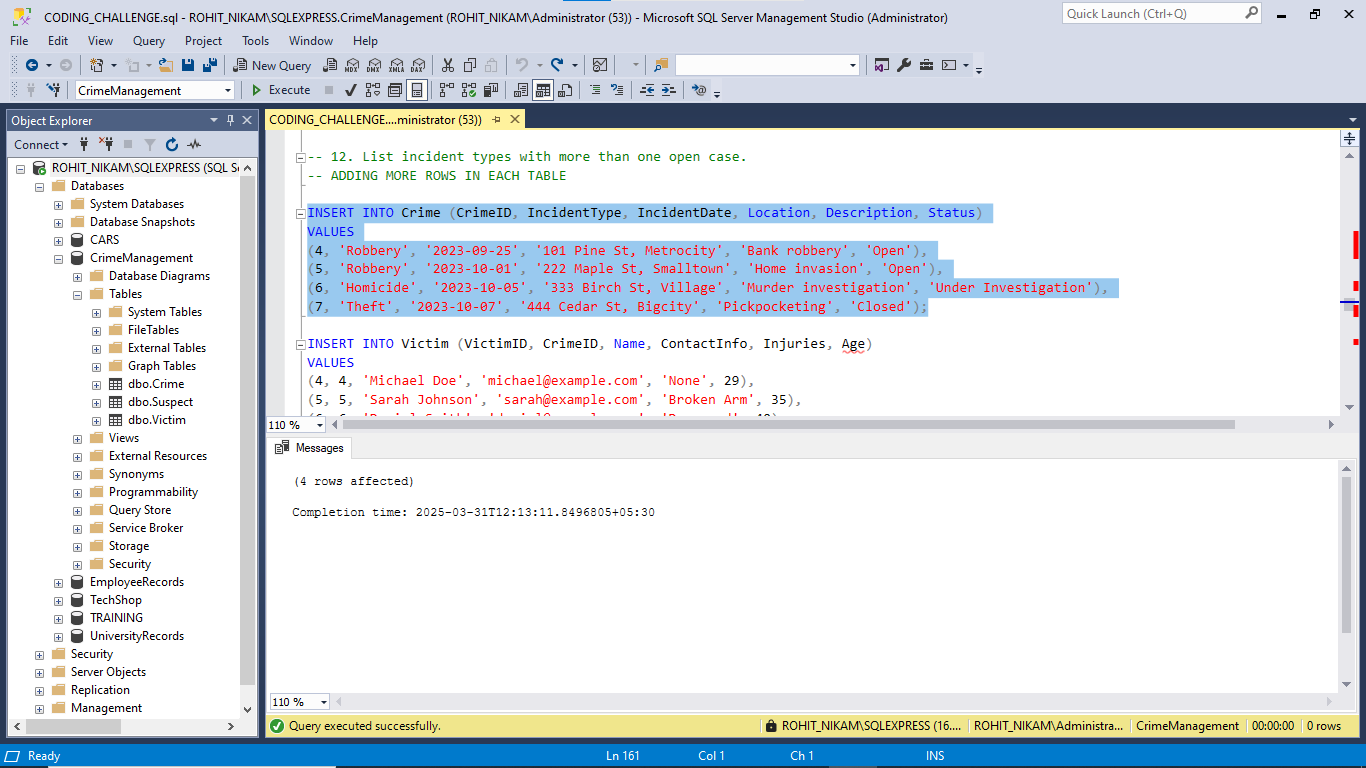
VALUES

(4, 'Robbery', '2023-09-25', '101 Pine St, Metrocity', 'Bank robbery', 'Open'),

(5, 'Robbery', '2023-10-01', '222 Maple St, Smalltown', 'Home invasion', 'Open'),

(6, 'Homicide', '2023-10-05', '333 Birch St, Village', 'Murder investigation', 'Under Investigation'),

(7, 'Theft', '2023-10-07', '444 Cedar St, Bigcity', 'Pickpocketing', 'Closed');



INSERT INTO Victim (VictimID, CrimeID, Name, ContactInfo, Injuries, Age)

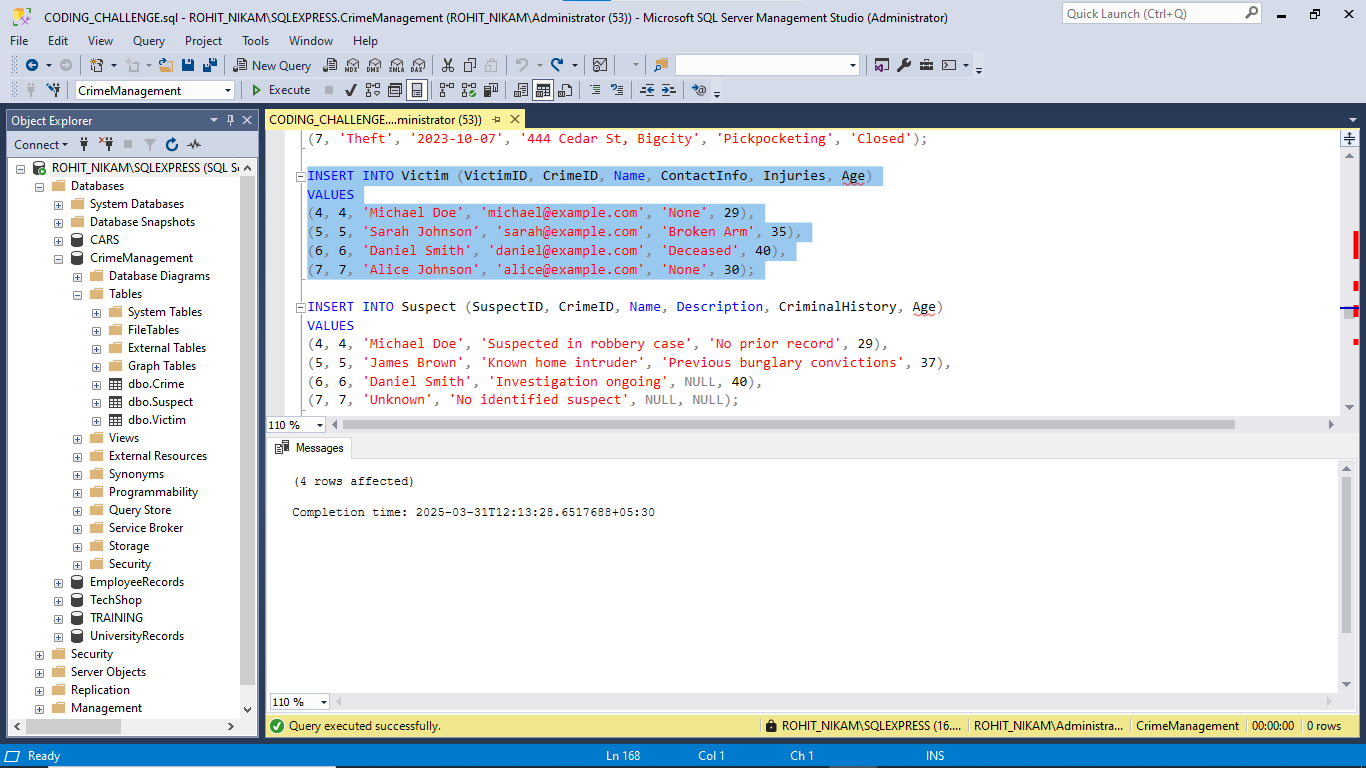
VALUES

(4, 4, 'Michael Doe', 'michael@example.com', 'None', 29),

(5, 5, 'Sarah Johnson', 'sarah@example.com', 'Broken Arm', 35),

(6, 6, 'Daniel Smith', 'daniel@example.com', 'Deceased', 40),

(7, 7, 'Alice Johnson', 'alice@example.com', 'None', 30);



INSERT INTO Suspect (SuspectID, CrimeID, Name, Description, CriminalHistory, Age)

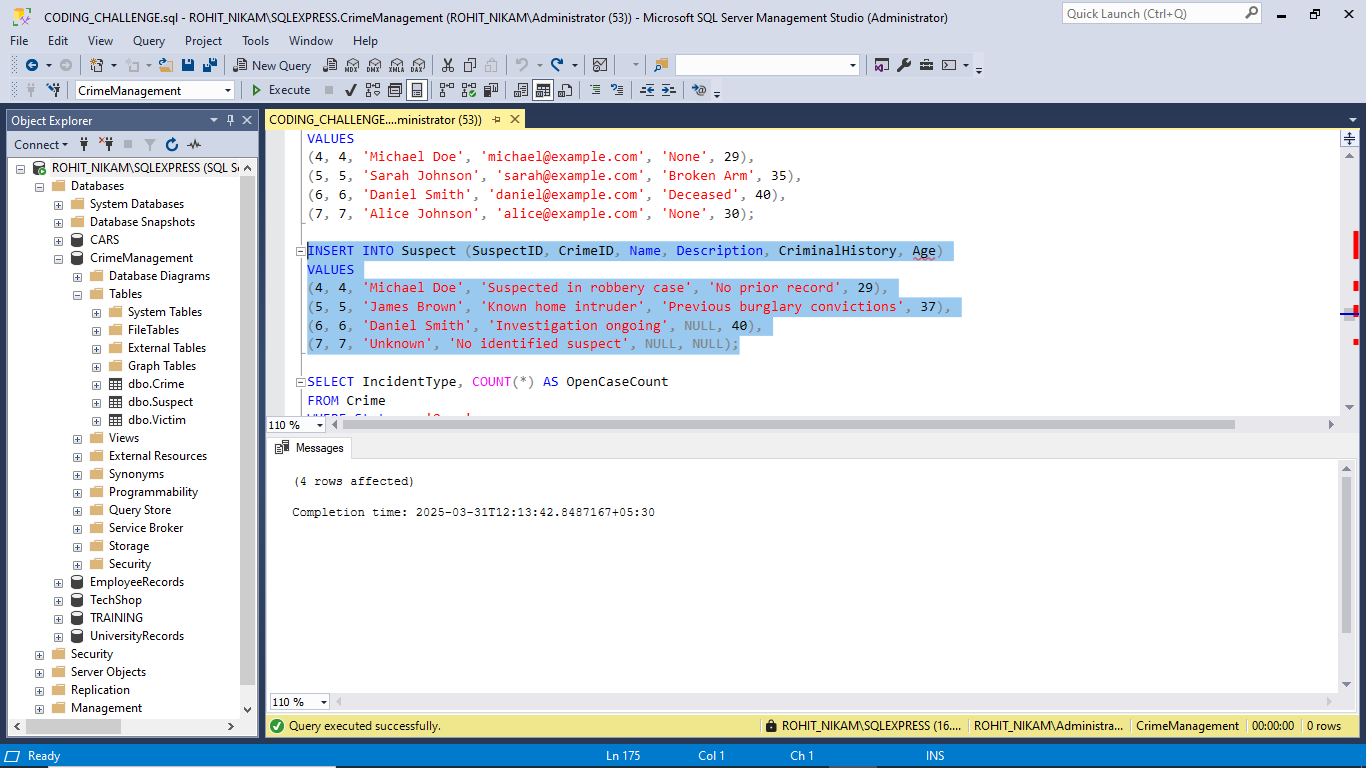
VALUES

(4, 4, 'Michael Doe', 'Suspected in robbery case', 'No prior record', 29),

(5, 5, 'James Brown', 'Known home intruder', 'Previous burglary convictions', 37),

(6, 6, 'Daniel Smith', 'Investigation ongoing', NULL, 40),

(7, 7, 'Unknown', 'No identified suspect', NULL, NULL);



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

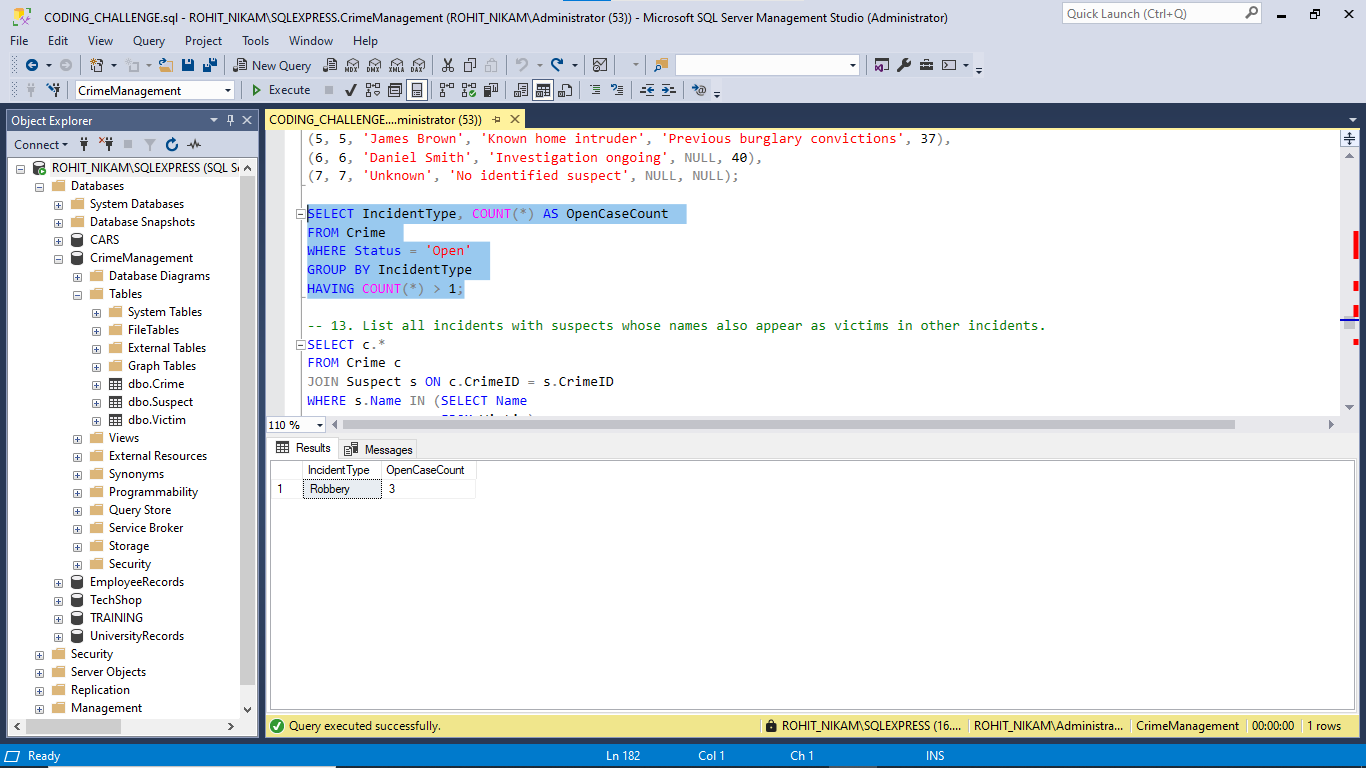
SELECT IncidentType, COUNT(\*) AS OpenCaseCount

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.

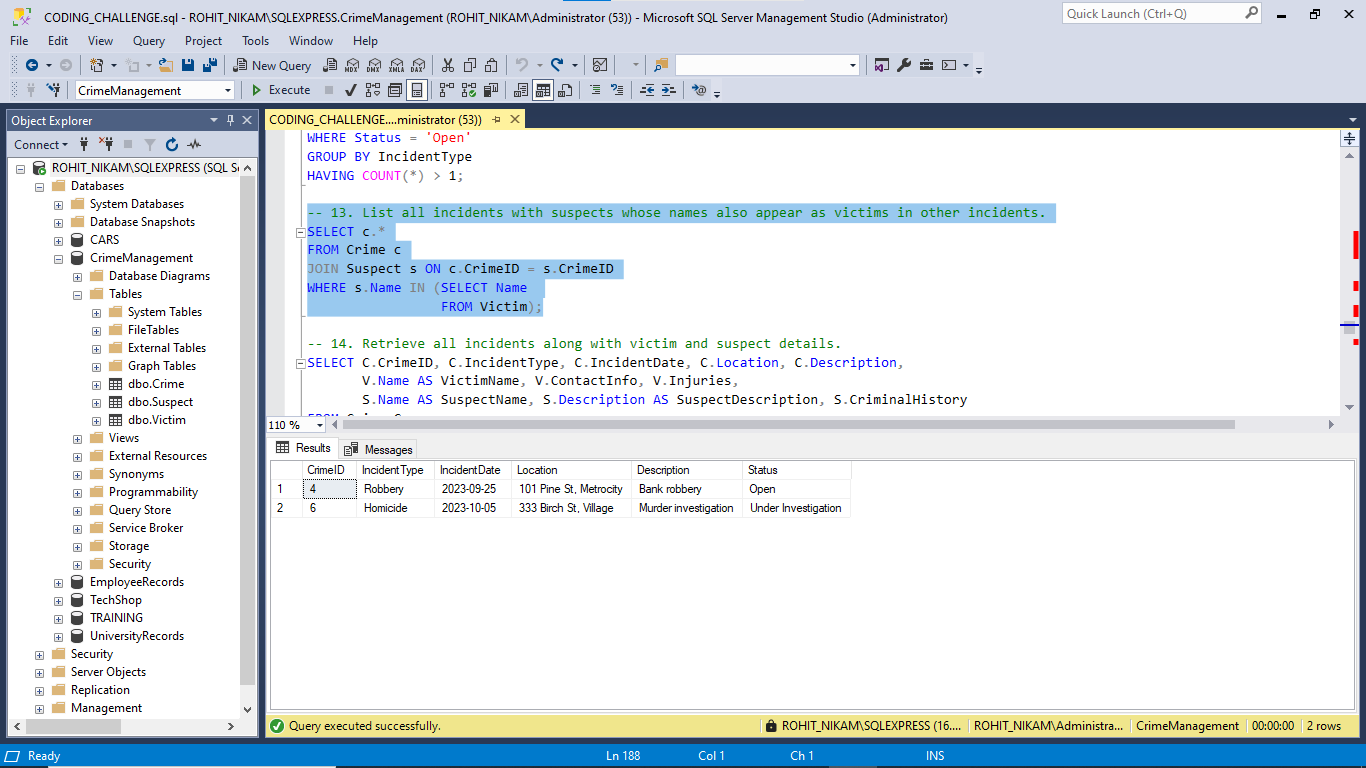
SELECT c.\*

FROM Crime c

JOIN Suspect s ON c.CrimeID = s.CrimeID

WHERE s.Name IN (SELECT Name

FROM Victim);



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

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

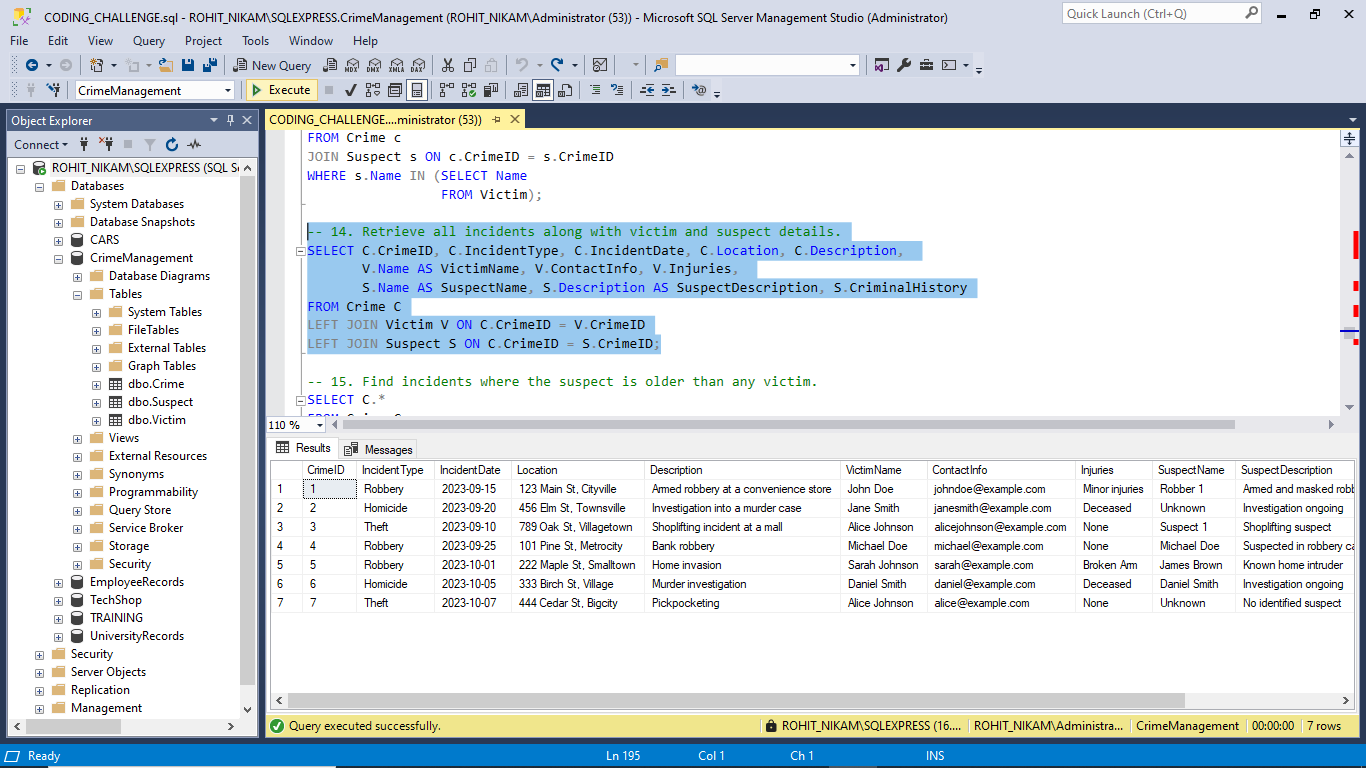
V.Name AS VictimName, V.ContactInfo, V.Injuries,

S.Name AS SuspectName, S.Description AS SuspectDescription, S.CriminalHistory

FROM Crime C

LEFT JOIN Victim V ON C.CrimeID = V.CrimeID

LEFT JOIN Suspect S ON C.CrimeID = S.CrimeID;



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

SELECT C.\*

FROM Crime C

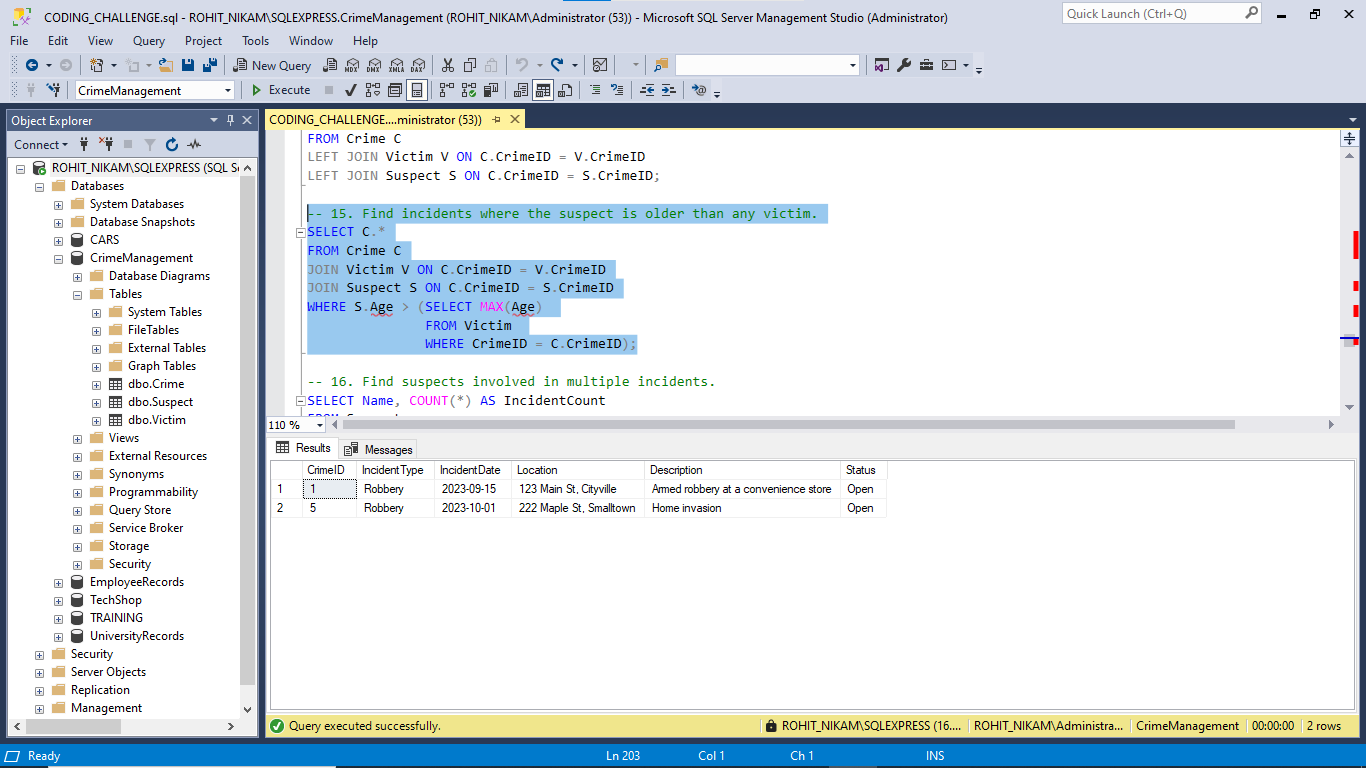
JOIN Victim V ON C.CrimeID = V.CrimeID

JOIN Suspect S ON C.CrimeID = S.CrimeID

WHERE S.Age > (SELECT MAX(Age)

FROM Victim

WHERE CrimeID = C.CrimeID);



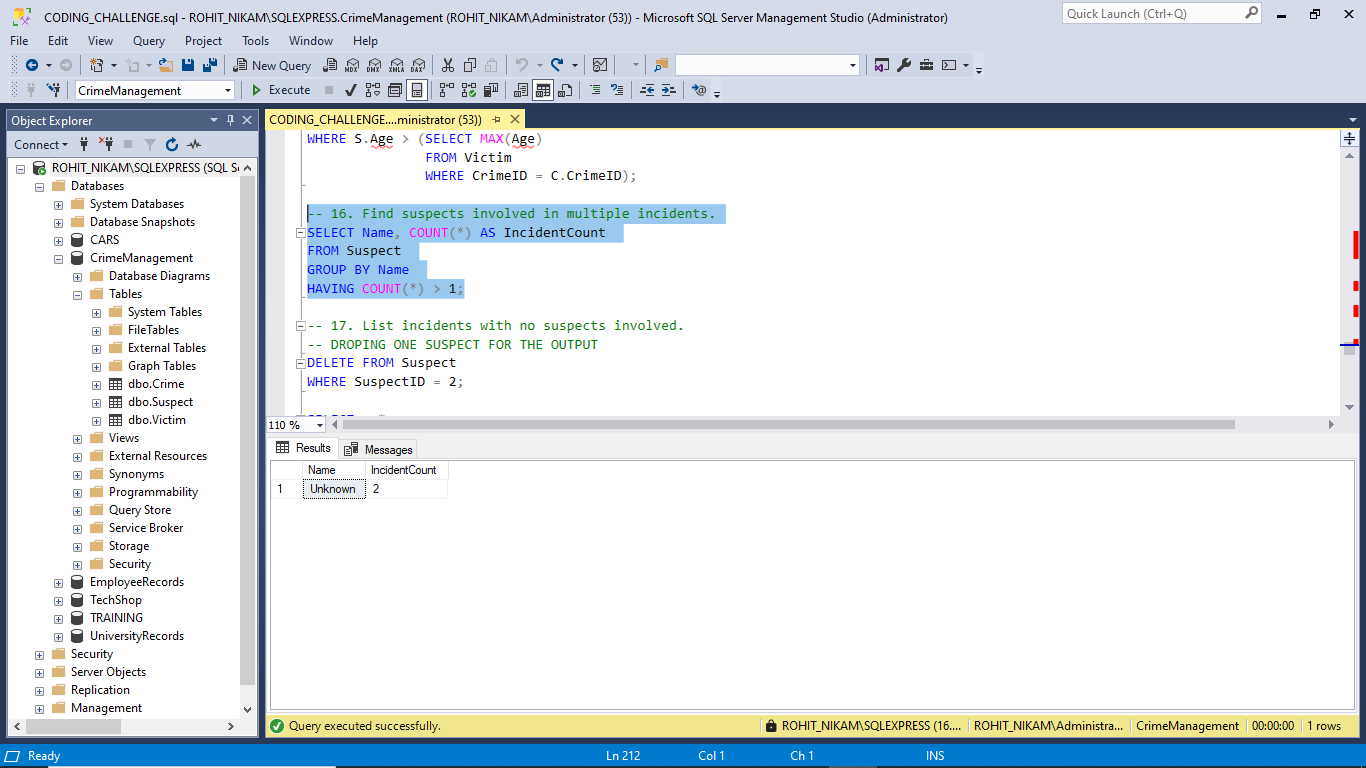
-- 16. Find suspects involved in multiple incidents.

SELECT Name, COUNT(\*) AS IncidentCount

FROM Suspect

GROUP BY Name

HAVING COUNT(\*) > 1;



-- DROPING ONE SUSPECT FOR THE OUTPUT

DELETE FROM Suspect

WHERE SuspectID = 2;



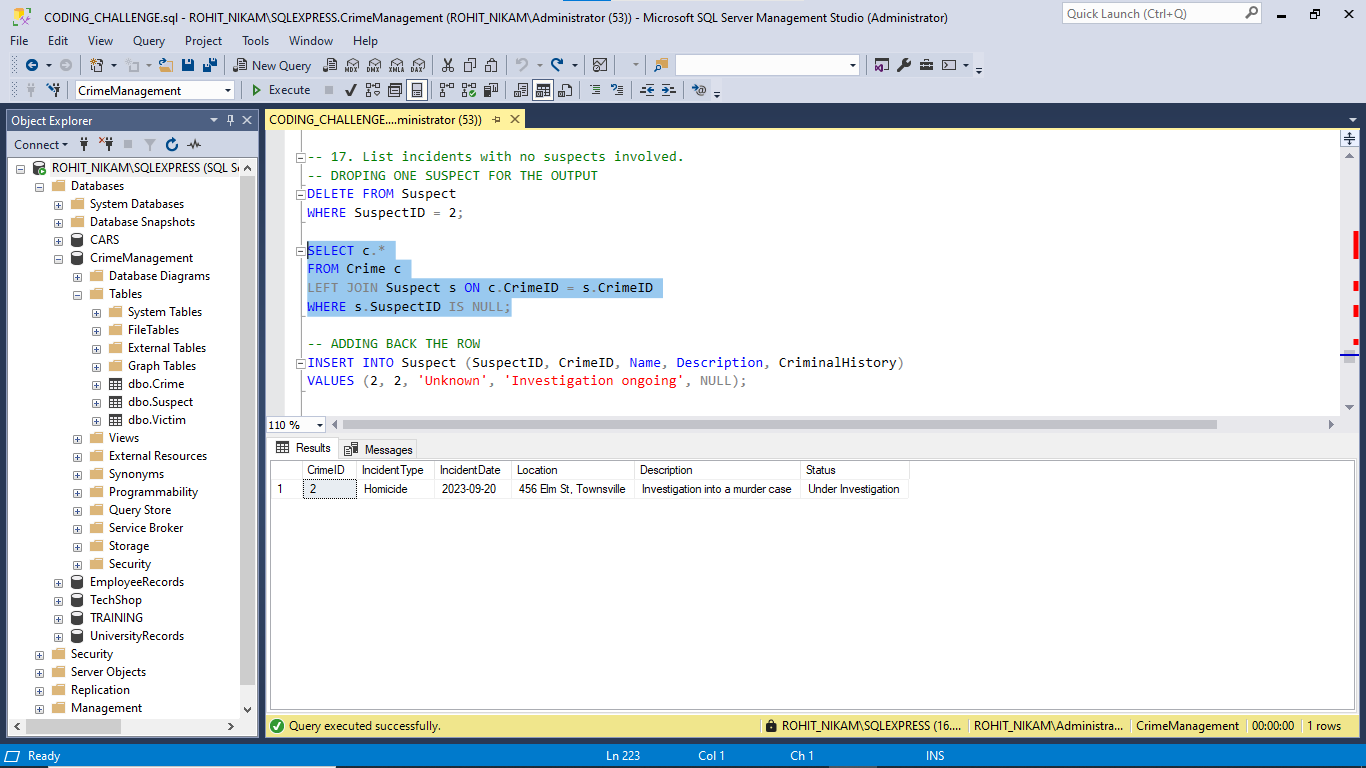
-- 17. List incidents with no suspects involved.

SELECT c.\*

FROM Crime c

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

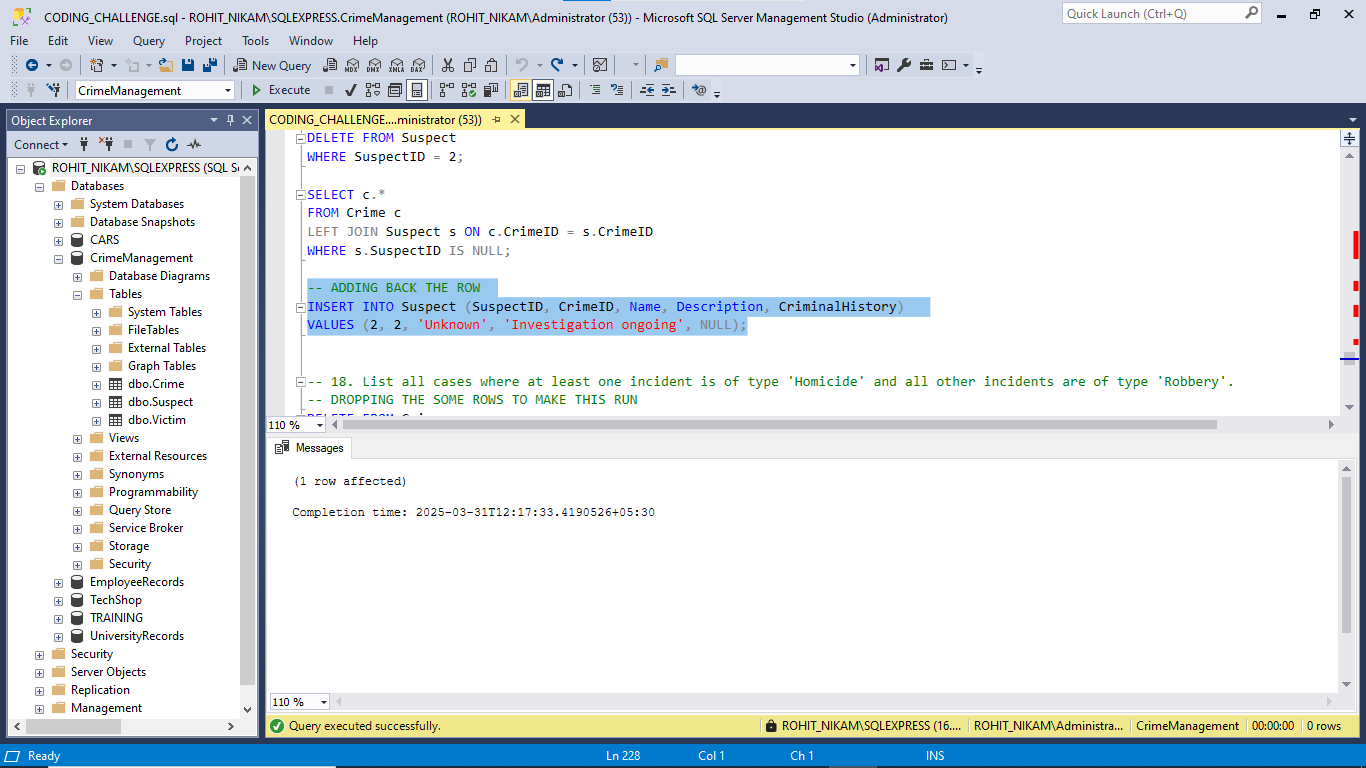
WHERE s.SuspectID IS NULL;



-- ADDING BACK THE ROW

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

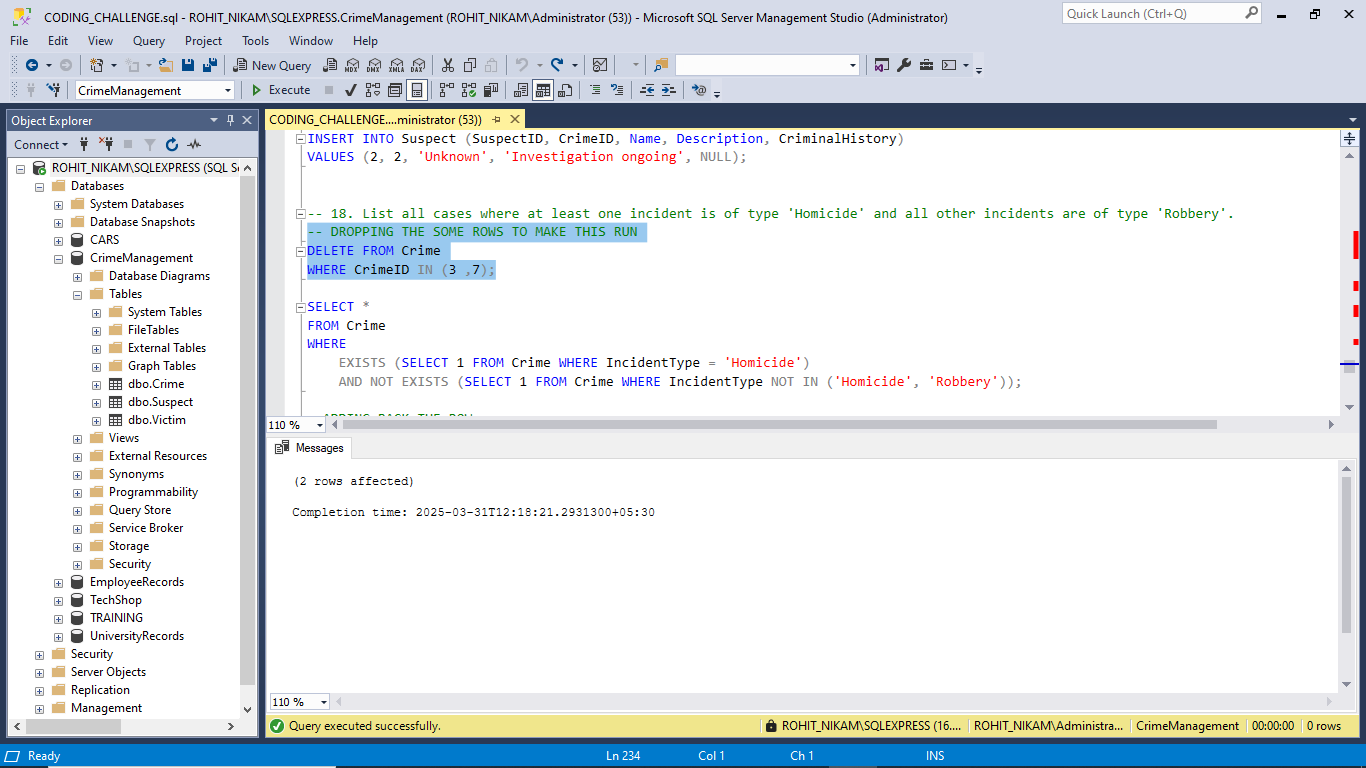
VALUES (2, 2, 'Unknown', 'Investigation ongoing', NULL);



-- DROPPING THE SOME ROWS TO MAKE THIS RUN

DELETE FROM Crime

WHERE CrimeID IN (3 ,7);



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

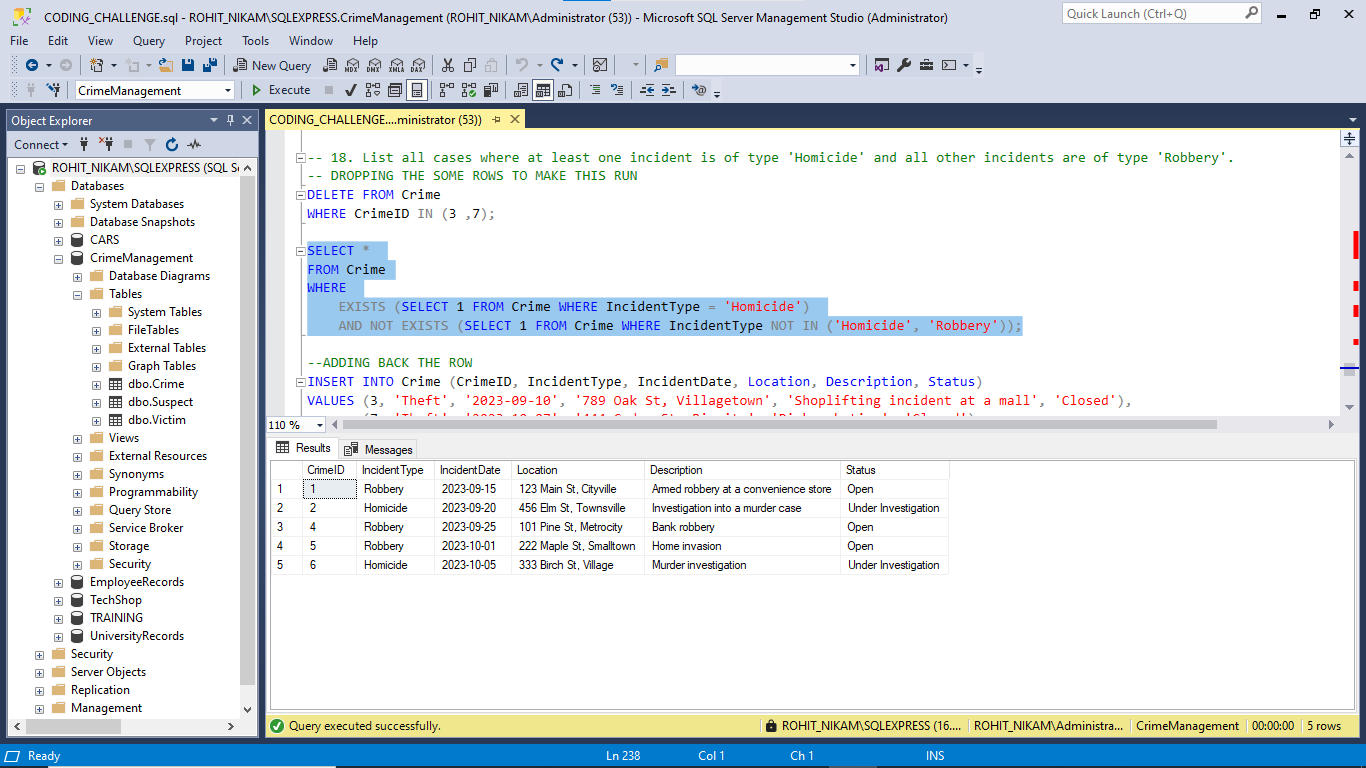
SELECT \*

FROM Crime

WHERE

EXISTS (SELECT 1 FROM Crime WHERE IncidentType = 'Homicide')

AND NOT EXISTS (SELECT 1 FROM Crime WHERE IncidentType NOT IN ('Homicide', 'Robbery'));

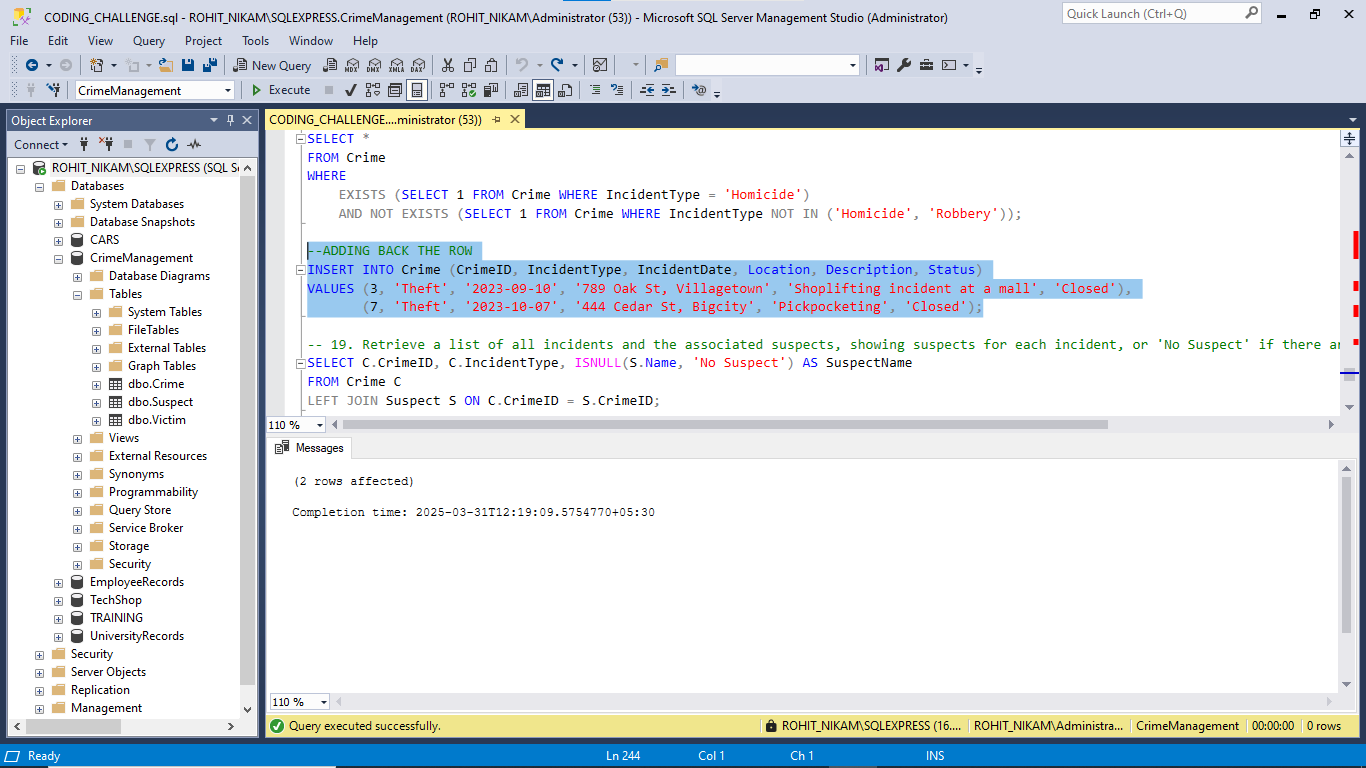


--ADDING BACK THE ROW

INSERT INTO Crime (CrimeID, IncidentType, IncidentDate, Location, Description, Status)

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

(7, 'Theft', '2023-10-07', '444 Cedar St, Bigcity', 'Pickpocketing', 'Closed');

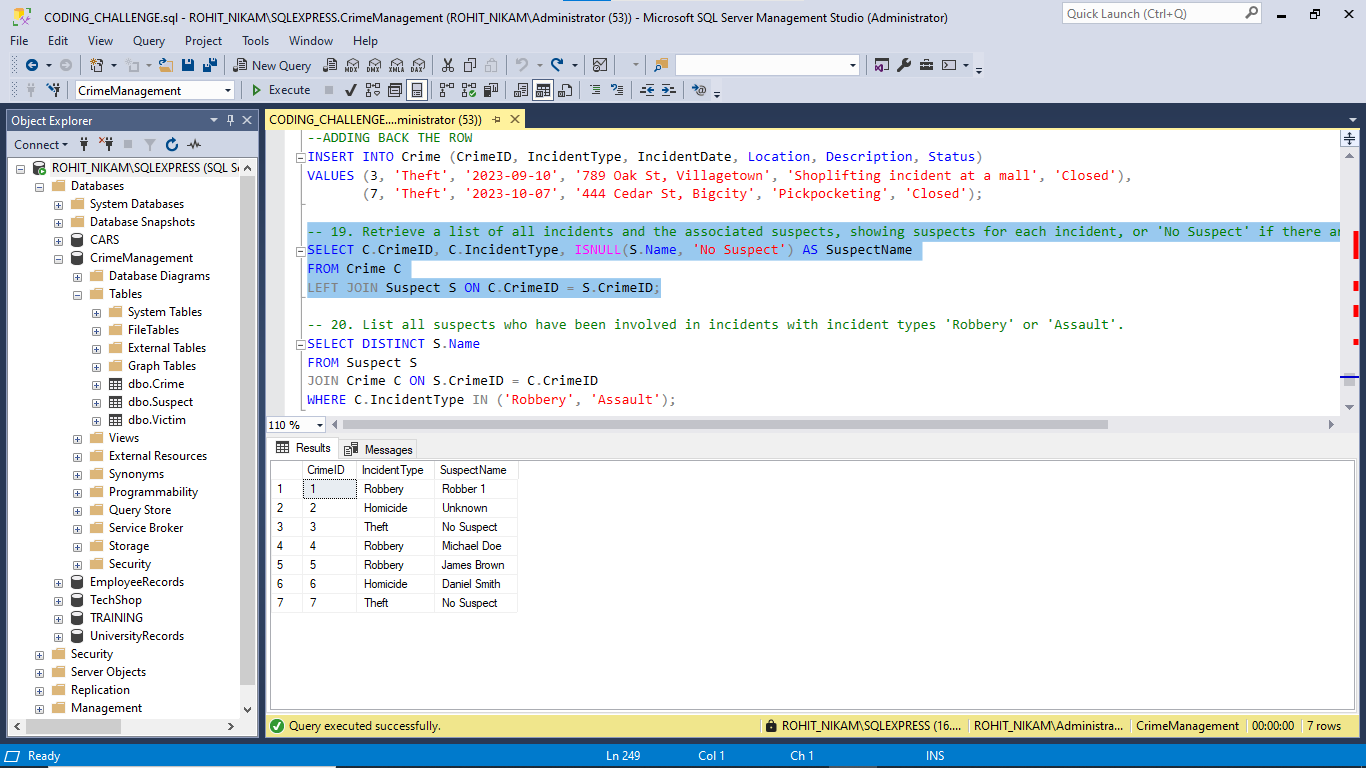


-- 19. Retrieve a list of all incidents and the associated suspects, showing suspects for each incident, or 'No Suspect' if there are none.

SELECT C.CrimeID, C.IncidentType, ISNULL(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.Name

FROM Suspect S

JOIN Crime C ON S.CrimeID = C.CrimeID

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

