PROJECT FOR SQL MODULE

CRIME ANALYSIS

Institute name: I. T. Vedant

Name: Ashwini P Sawant

Email: ashwinisawant999@gmail.com

Date of Submission:30/07/2024

Project Aims:

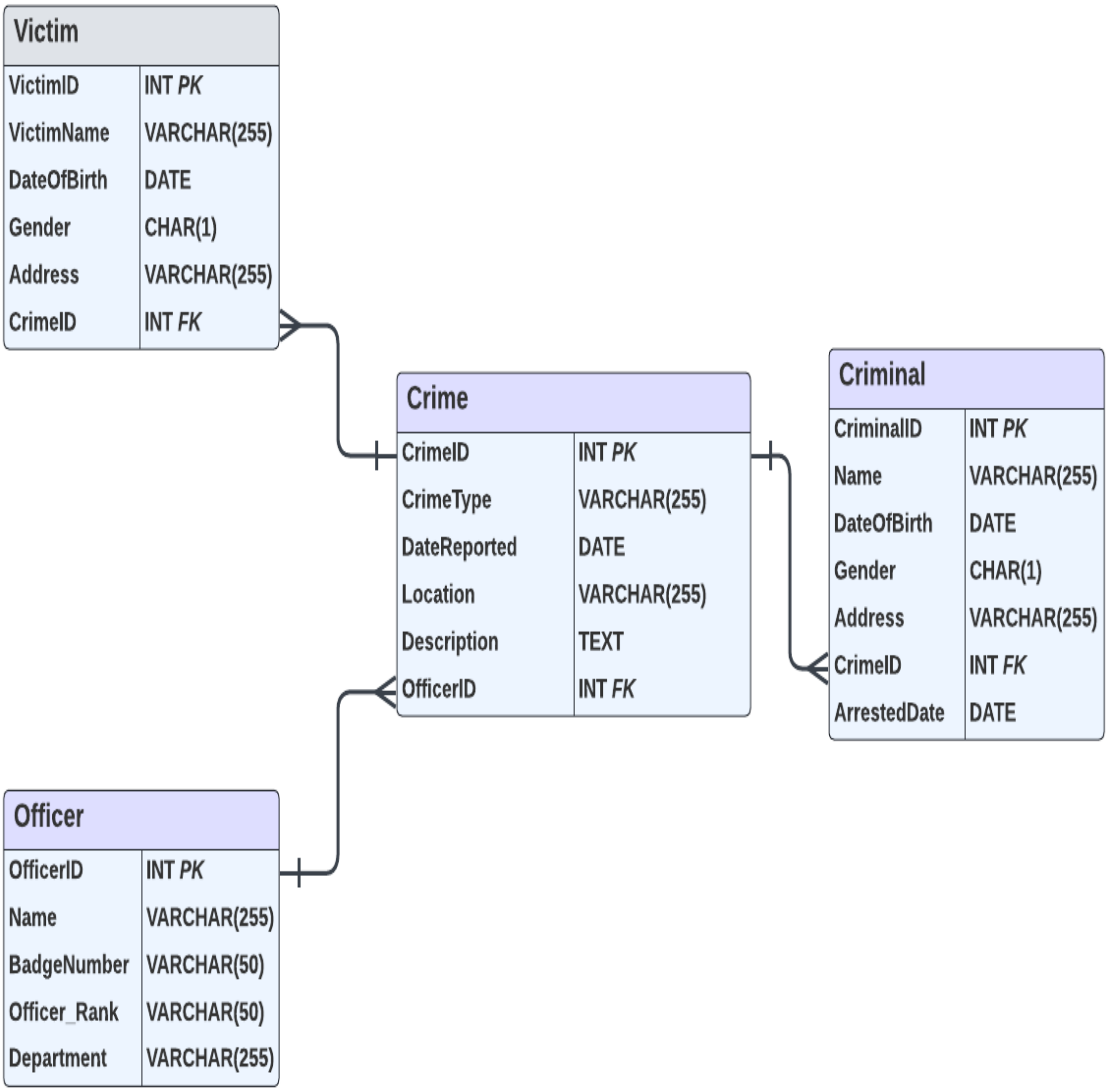
1. **Database Design and Implementation:**
   - **Aim:** To design and implement a relational database that accurately models the relationships between crimes, criminals, victims, and law enforcement officers.
   - **Outcome:** A normalized database schema with tables for crime, criminal, victim, and officer that ensures data integrity and efficient querying.
2. **Data Population and Management:**
   - **Aim:** To populate the database with realistic data and manage this data effectively.
   - **Outcome:** At least 100 rows of data in each table, ensuring diversity and realism in the data entries.
3. **Data Retrieval and Analysis:**
   - **Aim:** To develop and execute SQL queries that retrieve and analyze data to support law enforcement activities.
   - **Outcome:** A set of SQL queries that demonstrate the ability to extract meaningful insights from the database, such as identifying trends in criminal activity, tracking arrest records, and analyzing victim profiles.

4. **Query Optimization:**
    - **Aim:** To optimize SQL queries for performance and efficiency.
    - **Outcome:** Well-structured queries that minimize execution time and resource consumption, demonstrated through performance benchmarks.
5. **Learning and Skill Development:**
    - **Aim:** To develop and enhance skills in database management, SQL querying, and data analysis.
    - **Outcome:** A comprehensive understanding of relational databases, proficiency in SQL, and experience in data analysis and reporting.
    -

Project Objective:

This project aims to design a structured relational database for crime, criminal, victim, and officer data, populated with at least 100 rows per table. We aim to develop efficient SQL queries for data retrieval and analysis and generate clear reports and visualizations to aid decision-making. Security measures will be implemented to protect sensitive data, and performance optimization will ensure efficient operations. This database will support law enforcement activities and enhance our skills in database management, SQL querying, and data analysis.

ER diagram:



Victim	
VictimID	INT *PK*
VictimName	VARCHAR(255)
DateOfBirth	DATE
Gender	CHAR(1)
Address	VARCHAR(255)
CrimeID	INT *FK*

Crime	
CrimeID	INT *PK*
CrimeType	VARCHAR(255)
DateReported	DATE
Location	VARCHAR(255)
Description	TEXT
OfficerID	INT *FK*

Criminal	
CriminalID	INT *PK*
Name	VARCHAR(255)
DateOfBirth	DATE
Gender	CHAR(1)
Address	VARCHAR(255)
CrimeID	INT *FK*
ArrestedDate	DATE

Officer	
OfficerID	INT *PK*
Name	VARCHAR(255)
BadgeNumber	VARCHAR(50)
Officer_Rank	VARCHAR(50)
Department	VARCHAR(255)

Table description:

1. **OFFICER:**

Field	Type	Null	Key	Default	Extra
OfficerID	int	NO	PRI	NULL	auto_increment
OfficerName	varchar(255)	NO		NULL	
BadgeNumber	varchar(50)	NO	UNI	NULL	
Officer_Rank	varchar(50)	YES		NULL	
Department	varchar(255)	YES		NULL	

2. CRIME:

Field	Type	Null	Key	Default	Extra
CrimeID	int	NO	PRI	NULL	auto_increment
CrimeType	varchar(255)	NO		NULL	
DateReported	date	NO		NULL	
Location	varchar(255)	NO		NULL	
Description	text	YES		NULL	
OfficerID	int	YES	MUL	NULL	

3. CRIMINAL:

Field	Type	Null	Key	Default	Extra
CriminalID	int	NO	PRI	NULL	auto_increment
Name	varchar(255)	NO		NULL	
DateOfBirth	date	YES		NULL	
Gender	char(1)	YES		NULL	
Address	varchar(255)	YES		NULL	
CrimeID	int	YES	MUL	NULL	
ArrestedDate	date	YES		NULL	

4. VICTIM:

Field	Type	Null	Key	Default	Extra
VictimID	int	NO	PRI	NULL	auto_increment
VictimName	varchar(255)	NO		NULL	
DateOfBirth	date	YES		NULL	
Gender	char(1)	YES		NULL	
Address	varchar(255)	YES		NULL	
CrimeID	int	YES	MUL	NULL	

Commands:

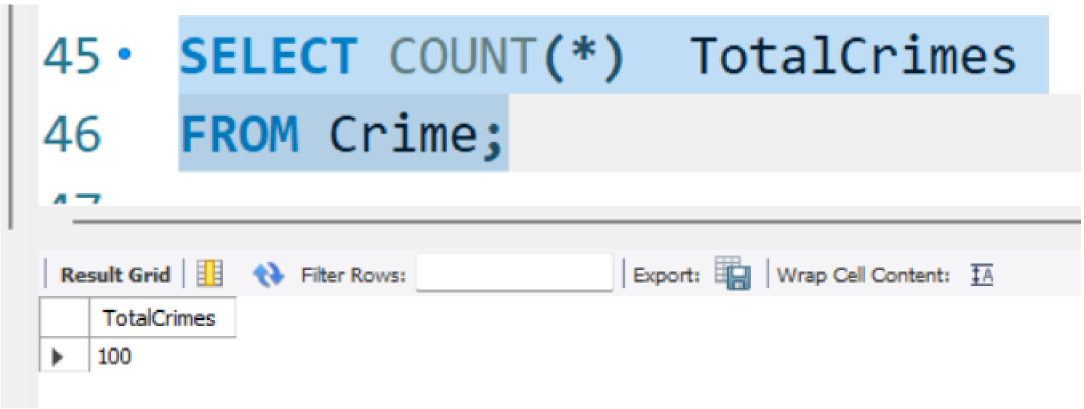
```
CREATE TABLE Officer (

   OfficerID INT AUTO_INCREMENT PRIMARY KEY ,

   Name VARCHAR(255) NOT NULL,

   BadgeNumber VARCHAR(50) NOT NULL UNIQUE,

   Officer_Rank VARCHAR(50),

   Department VARCHAR(255)

);
```

```
CREATE TABLE Crime (
    CrimeID INT AUTO_INCREMENT PRIMARY KEY ,
    CrimeType VARCHAR(255) NOT NULL,
    DateReported DATE NOT NULL,
    Location VARCHAR(255) NOT NULL,
    Description TEXT,
    OfficerID INT,
   Constraint Fk_Crime_Officer FOREIGN KEY (OfficerID) REFERENCES Officer(OfficerID)
);

CREATE TABLE Criminal (
    CriminalID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL,
    DateOfBirth DATE,
    Gender CHAR(1),
    Address VARCHAR(255),
    CrimeID INT,
    ArrestedDate DATE,
    FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)
);

CREATE TABLE Victim (
    VictimID INT PRIMARY KEY AUTO_INCREMENT,
    VictimName VARCHAR(255) NOT NULL,
    DateOfBirth DATE,
    Gender CHAR(1),
    Address VARCHAR(255),
    CrimeID INT,
    FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID)
);
```

SQL Queries:

1. Total number of crimes:

SELECT COUNT(*) TotalCrimes FROM Crime;

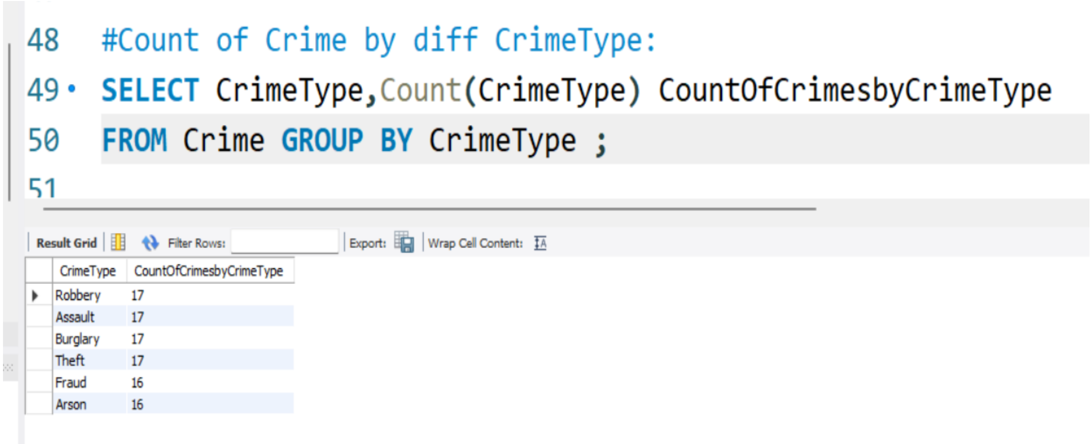


```
SELECT COUNT(*)   TotalCrimes
FROM Crime;
```

	TotalCrimes
▶	100

2. Crime by different CrimeType:

SELECT CrimeType,Count(CrimeType) CountOfCrimesbyCrimeType FROM Crime GROUP BY CrimeType

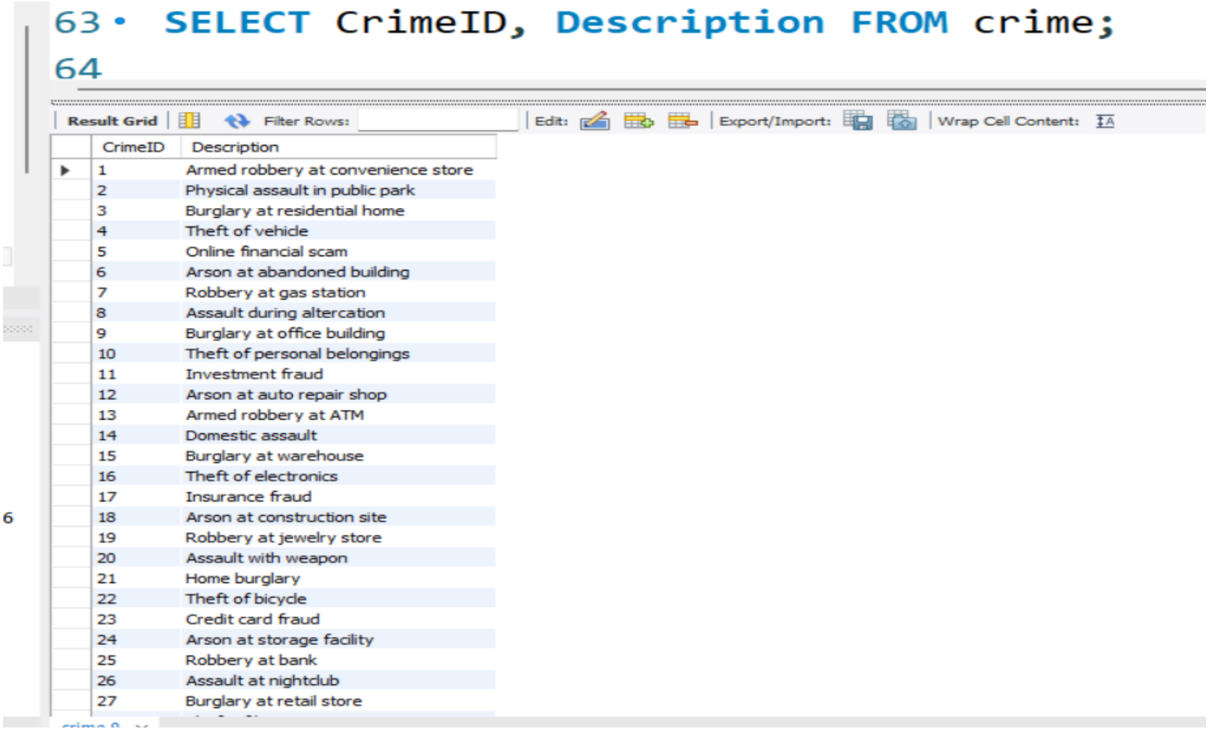


```
#Count of Crime by diff CrimeType:
SELECT CrimeType,Count(CrimeType) CountOfCrimesbyCrimeType
FROM Crime GROUP BY CrimeType ;
```

	CrimeType	CountOfCrimesbyCrimeType
▶	Robbery	17
	Assault	17
	Burglary	17
	Theft	17
	Fraud	16
	Arson	16

3. List all crimes:

SELECT CrimeID, Description FROM crime;

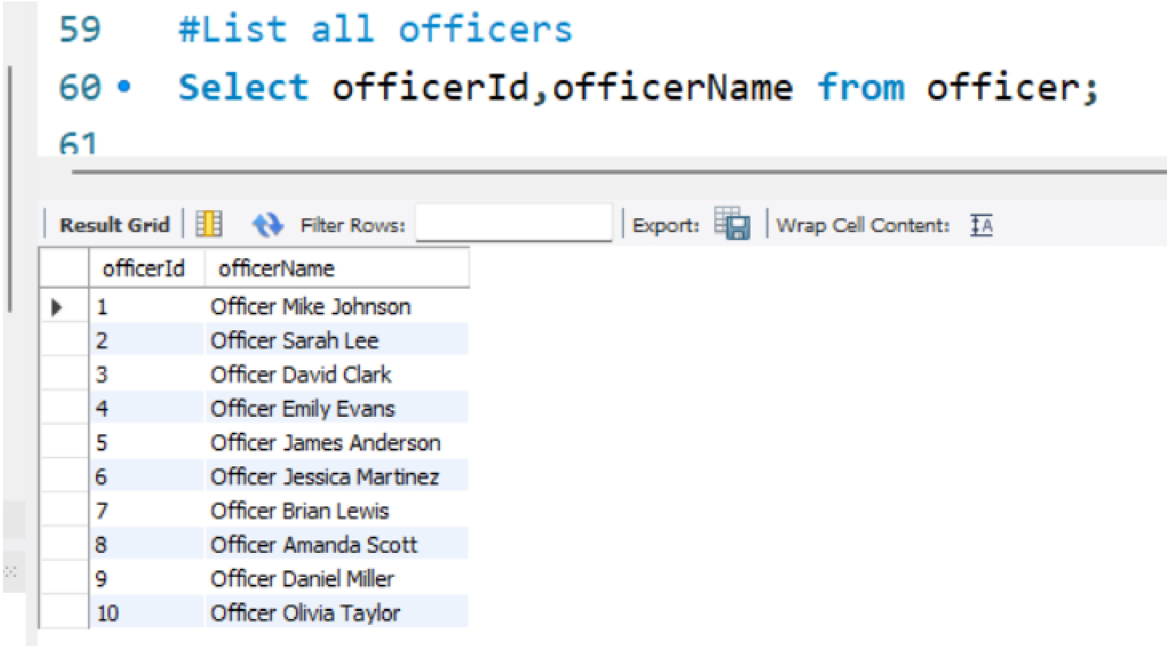


```
SELECT CrimeID, Description FROM crime;
```

CrimeID	Description
1	Armed robbery at convenience store
2	Physical assault in public park
3	Burglary at residential home
4	Theft of vehicle
5	Online financial scam
6	Arson at abandoned building
7	Robbery at gas station
8	Assault during altercation
9	Burglary at office building
10	Theft of personal belongings
11	Investment fraud
12	Arson at auto repair shop
13	Armed robbery at ATM
14	Domestic assault
15	Burglary at warehouse
16	Theft of electronics
17	Insurance fraud
18	Arson at construction site
19	Robbery at jewelry store
20	Assault with weapon
21	Home burglary
22	Theft of bicycle
23	Credit card fraud
24	Arson at storage facility
25	Robbery at bank
26	Assault at nightclub
27	Burglary at retail store

4. List all officers:

Select officerId, officerName from officer;

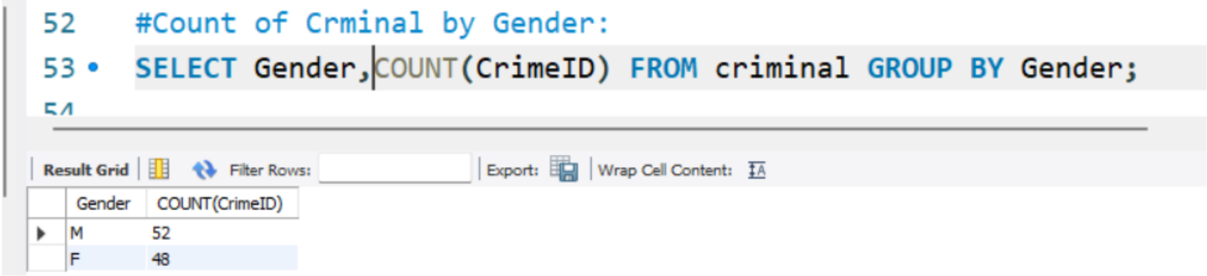


```
#List all officers
Select officerId,officerName from officer;
```

officerId	officerName
1	Officer Mike Johnson
2	Officer Sarah Lee
3	Officer David Clark
4	Officer Emily Evans
5	Officer James Anderson
6	Officer Jessica Martinez
7	Officer Brian Lewis
8	Officer Amanda Scott
9	Officer Daniel Miller
10	Officer Olivia Taylor

5. Total number of Criminal by Gender:

SELECT Gender,COUNT(CrimeID) FROM criminal GROUP BY Gender;



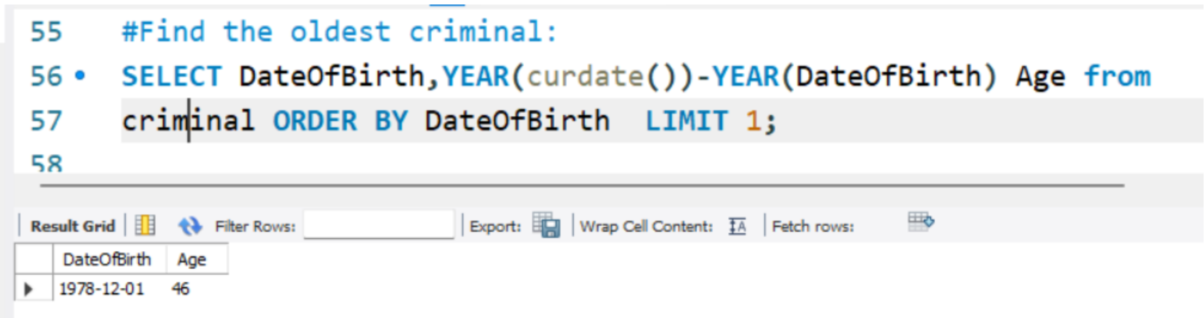
```
#Count of Crminal by Gender:
SELECT Gender,COUNT(CrimeID) FROM criminal GROUP BY Gender;
```

Gender	COUNT(CrimeID)
M	52
F	48

6. Find the oldest criminal:

SELECT DateOfBirth,YEAR(curdate())-YEAR(DateOfBirth) Age from

criminal ORDER BY DateOfBirth LIMIT 1;



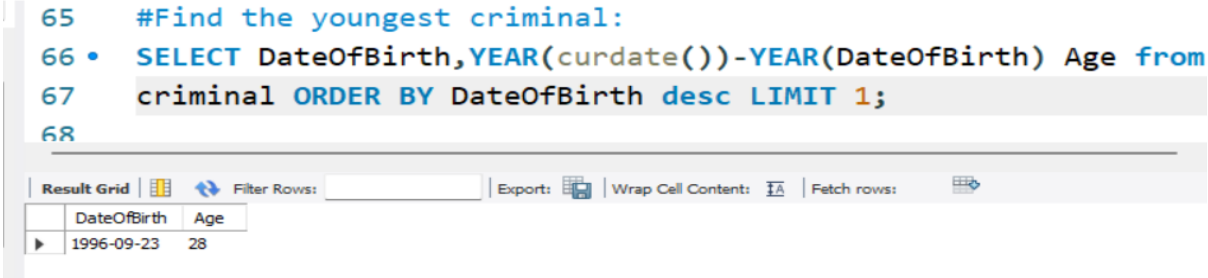
```
#Find the oldest criminal:
SELECT DateOfBirth,YEAR(curdate())-YEAR(DateOfBirth) Age from
criminal ORDER BY DateOfBirth  LIMIT 1;
```

DateOfBirth	Age
1978-12-01	46

7. Find the Youngest criminal:

SELECT DateOfBirth,YEAR(curdate())-YEAR(DateOfBirth) Age from

criminal ORDER BY DateOfBirth desc LIMIT 1;

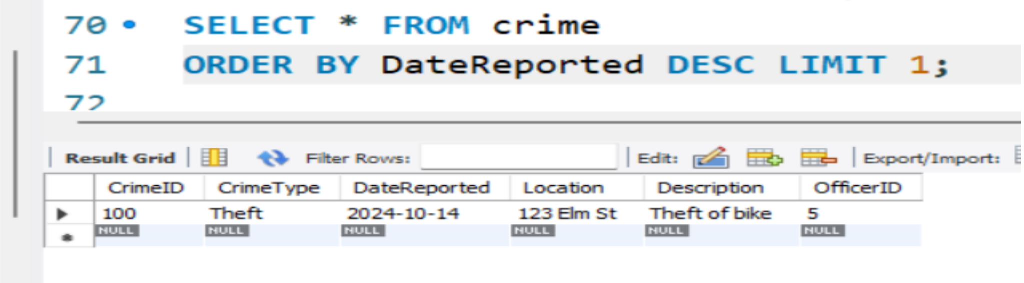


```
#Find the youngest criminal:
SELECT DateOfBirth,YEAR(curdate())-YEAR(DateOfBirth) Age from
criminal ORDER BY DateOfBirth desc LIMIT 1;
```

DateOfBirth	Age
1996-09-23	28

8. Find the most recent crime reported:

SELECT * FROM crime ORDER BY DateReported DESC LIMIT 1;

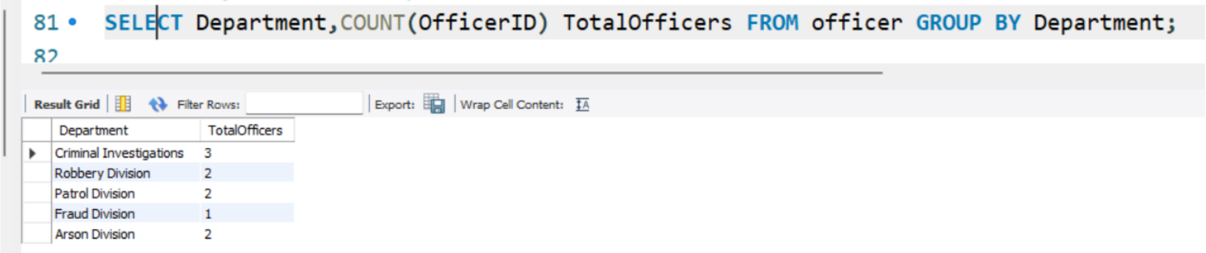


```
SELECT * FROM crime
ORDER BY DateReported DESC LIMIT 1;
```

CrimeID	CrimeType	DateReported	Location	Description	OfficerID
100	Theft	2024-10-14	123 Elm St	Theft of bike	5
NULL	NULL	NULL	NULL	NULL	NULL

9. Crimes by Officer's Department:

SELECT Department,COUNT(OfficerID) TotalOfficers FROM officer GROUP BY Department;

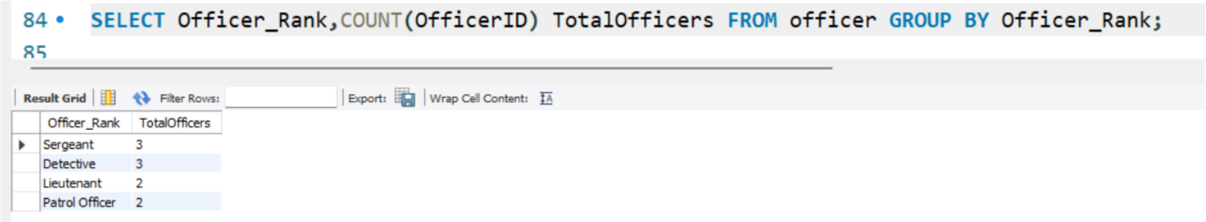


```
SELECT Department,COUNT(OfficerID) TotalOfficers FROM officer GROUP BY Department;
```

Department	TotalOfficers
Criminal Investigations	3
Robbery Division	2
Patrol Division	2
Fraud Division	1
Arson Division	2

10.Count the total number of Officers by Officer_Rank:

SELECT Officer_Rank,COUNT(OfficerID) TotalOfficers FROM officer GROUP BY Officer_Rank;

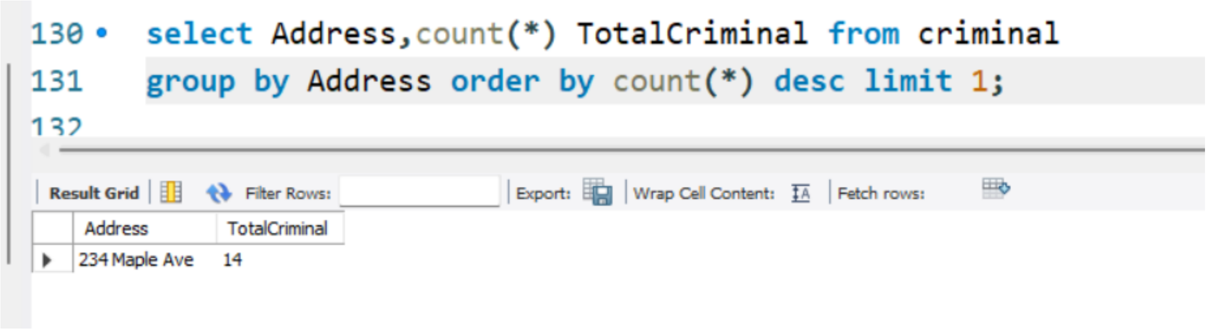


```
SELECT Officer_Rank,COUNT(OfficerID) TotalOfficers FROM officer GROUP BY Officer_Rank;
```

Officer_Rank	TotalOfficers
Sergeant	3
Detective	3
Lieutenant	2
Patrol Officer	2

11.Address where most Criminal reside:

select Address,count(*) TotalCriminal from criminal group by Address order by count(*) desc limit 1;

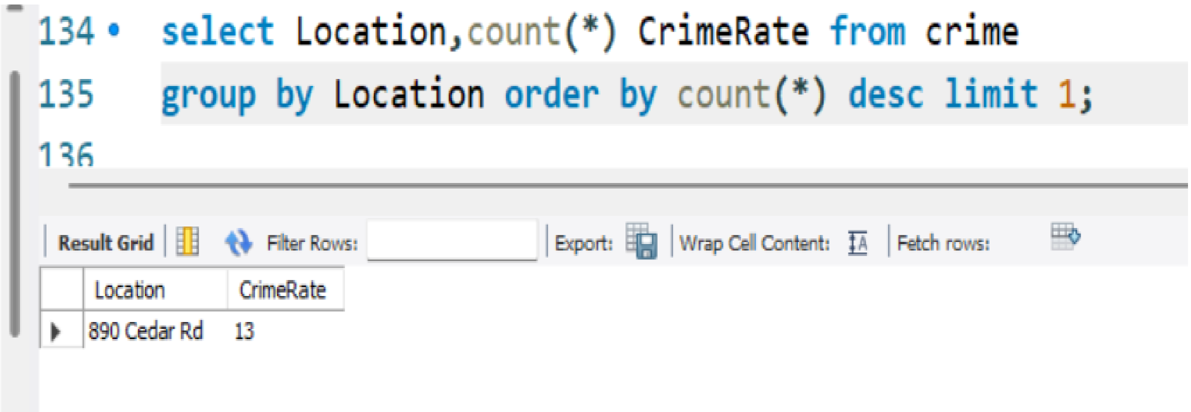


```
select Address,count(*) TotalCriminal from criminal
group by Address order by count(*) desc limit 1;
```

Address	TotalCriminal
234 Maple Ave	14

12.High-Crime Areas:

select Location,count(*) CrimeRate from crime group by Location order by count(*) desc limit 1;



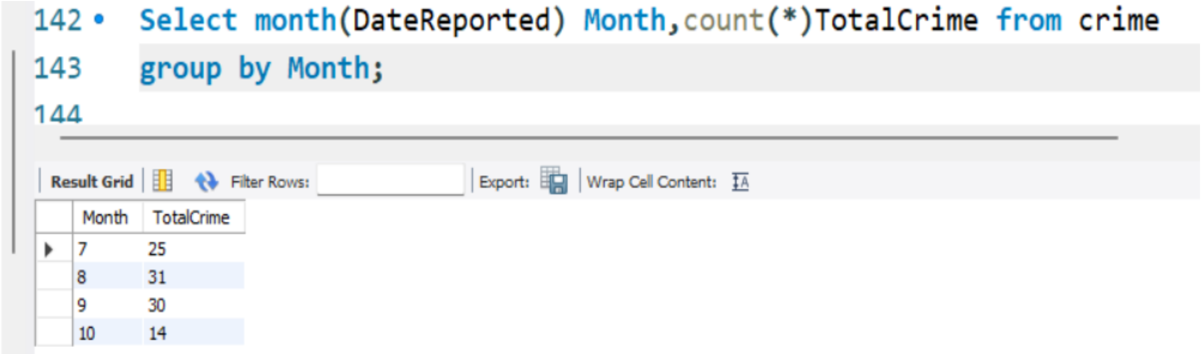
```
select Location,count(*) CrimeRate from crime
group by Location order by count(*) desc limit 1;
```

Location	CrimeRate
890 Cedar Rd	13

13.Crimes by Month:

Select month(DateReported) Month,count(*)TotalCrime from crime

group by Month;

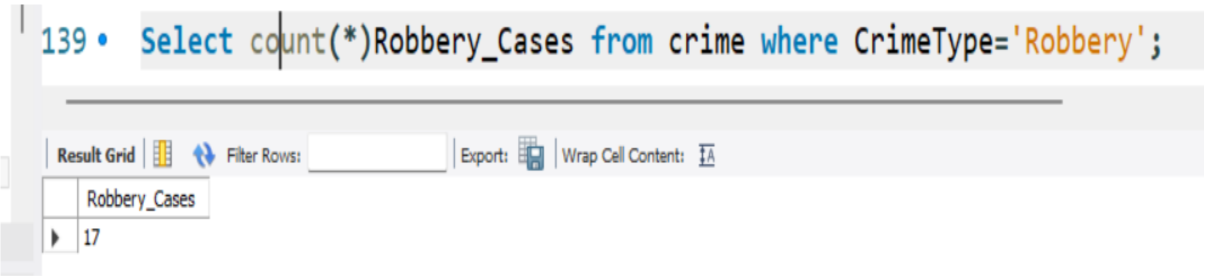


```
Select month(DateReported) Month,count(*)TotalCrime from crime
group by Month;
```

Month	TotalCrime
7	25
8	31
9	30
10	14

14.Total Robbery Cases:

Select count(*)Robbery_Cases from crime where CrimeType='Robbery';

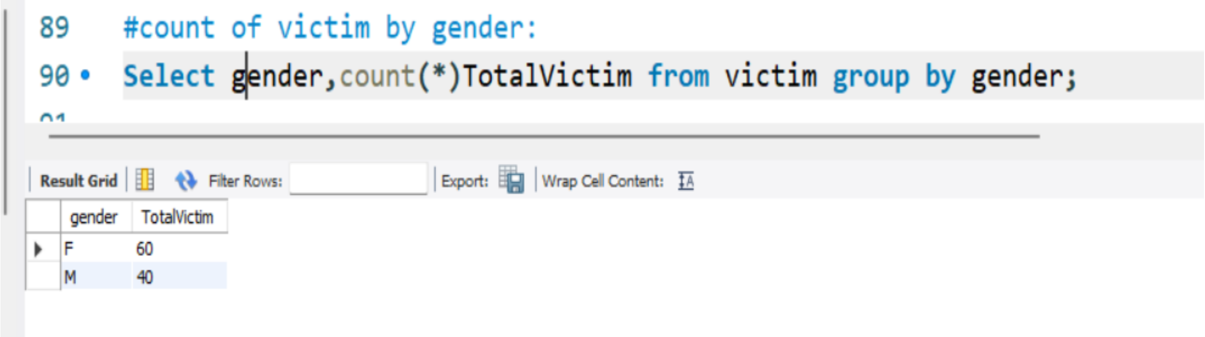


```
Select count(*)Robbery_Cases from crime where CrimeType='Robbery';
```

Robbery_Cases
17

15.Count of victim by gender:

Select gender,count(*)TotalVictim from victim group by gender;



```
#count of victim by gender:
Select gender,count(*)TotalVictim from victim group by gender;
```

gender	TotalVictim
F	60
M	40

Joins Queries:

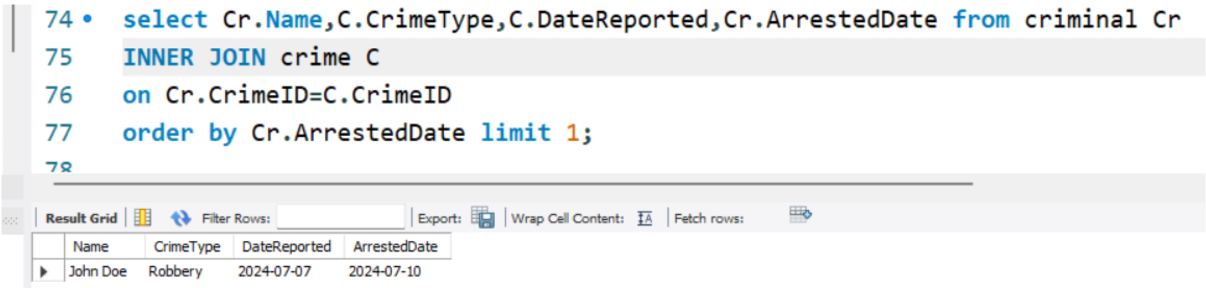
1. Find the most recent criminal and crime:

select Cr.Name,C.CrimeType,C.DateReported,Cr.ArrestedDate from criminal Cr

INNER JOIN crime C

on Cr.CrimeID=C.CrimeID

order by Cr.ArrestedDate limit 1;



```
select Cr.Name,C.CrimeType,C.DateReported,Cr.ArrestedDate from criminal Cr
INNER JOIN crime C
on Cr.CrimeID=C.CrimeID
order by Cr.ArrestedDate limit 1;
```

Name	CrimeType	DateReported	ArrestedDate
John Doe	Robbery	2024-07-07	2024-07-10

2. Find the name of officer who handled more cases:

SELECT   O.OfficerName, COUNT(C.crimeid) TotalCrimeHandled

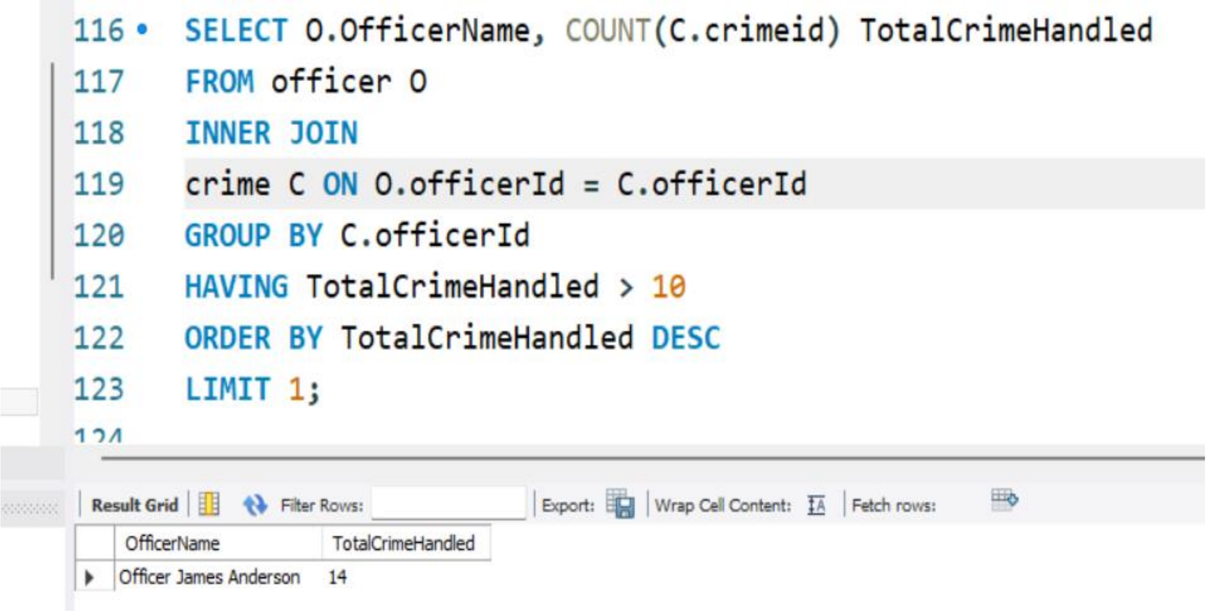
FROM officer O INNER JOIN

crime C ON O.officerId = C.officerId

GROUP BY C.officerId

HAVING TotalCrimeHandled > 10

ORDER BY TotalCrimeHandled DESC LIMIT 1;



```
SELECT O.OfficerName, COUNT(C.crimeid) TotalCrimeHandled
FROM officer O
INNER JOIN
crime C ON O.officerId = C.officerId
GROUP BY C.officerId
HAVING TotalCrimeHandled > 10
ORDER BY TotalCrimeHandled DESC
LIMIT 1;
```

OfficerName	TotalCrimeHandled
Officer James Anderson	14

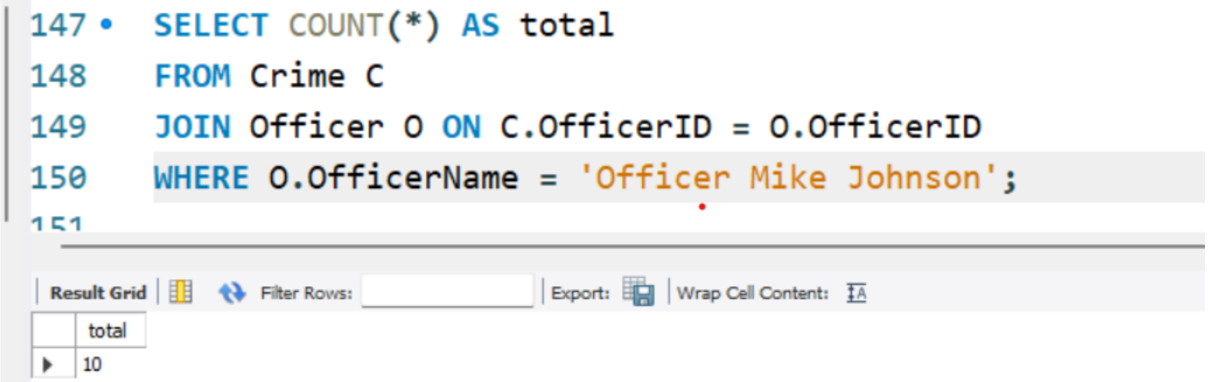
3. Crimes Involving a Specific Officer:

SELECT COUNT(*) AS total

FROM Crime C

JOIN Officer O ON C.OfficerID = O.OfficerID

WHERE O.OfficerName = 'Officer Mike Johnson';



```
SELECT COUNT(*) AS total
FROM Crime C
JOIN Officer O ON C.OfficerID = O.OfficerID
WHERE O.OfficerName = 'Officer Mike Johnson';
```

total
10

4. Crimes Involving a Specific Victim:

SELECT COUNT(*) AS total

FROM Crime C

JOIN Officer O ON C.OfficerID = O.OfficerID

WHERE O.OfficerName = 'Officer Mike Johnson';

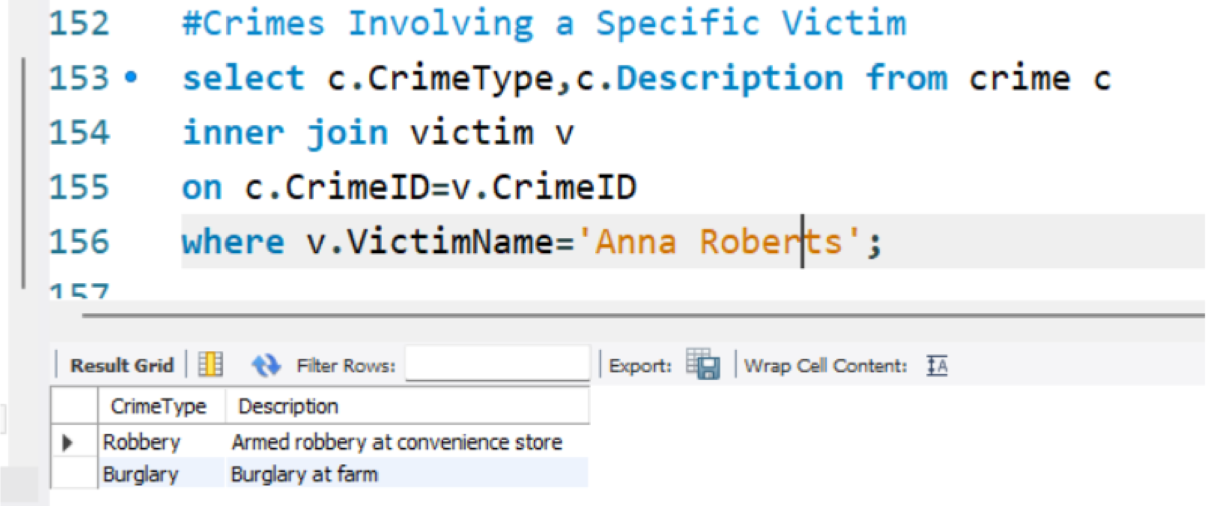
#Crimes Involving a Specific Victim

select c.CrimeType,c.Description from crime c

inner join victim v

on c.CrimeID=v.CrimeID

where v.VictimName='Anna Roberts';



```
#Crimes Involving a Specific Victim
select c.CrimeType,c.Description from crime c
inner join victim v
on c.CrimeID=v.CrimeID
where v.VictimName='Anna Roberts';
```

CrimeType	Description
Robbery	Armed robbery at convenience store
Burglary	Burglary at farm

5. Crimes by Victim's Age Group:

Select

CASE

WHEN year(curdate())-year(V.DateOfBirth) BETWEEN 0 AND 17 THEN '0-17'

WHEN year(curdate())-year(V.DateOfBirth) BETWEEN 18 AND 35 THEN '18-35'

WHEN year(curdate())-year(V.DateOfBirth) BETWEEN 36 AND 50 THEN '36-50'

ELSE '51+'

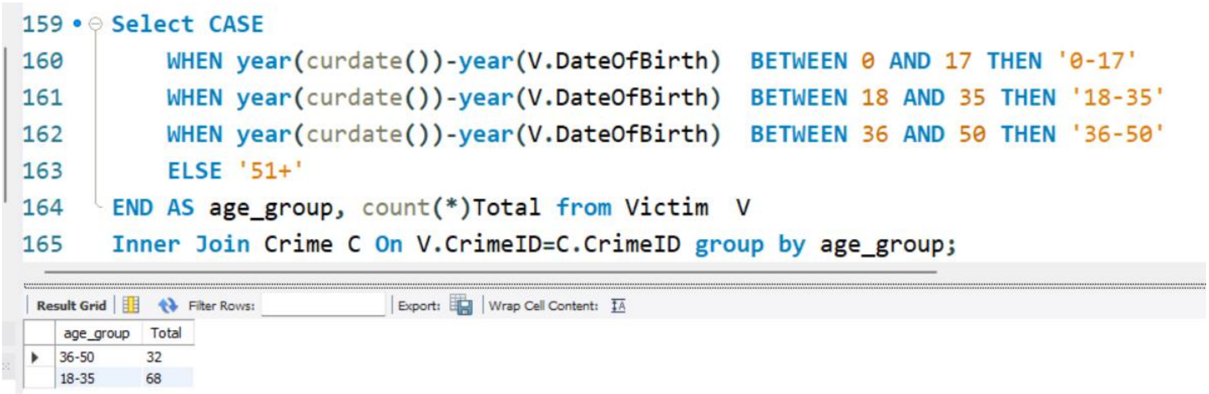
END AS age_group, count(*)Total

from Victim V

Inner Join Crime C

On V.CrimeID=C.CrimeID

group by age_group;



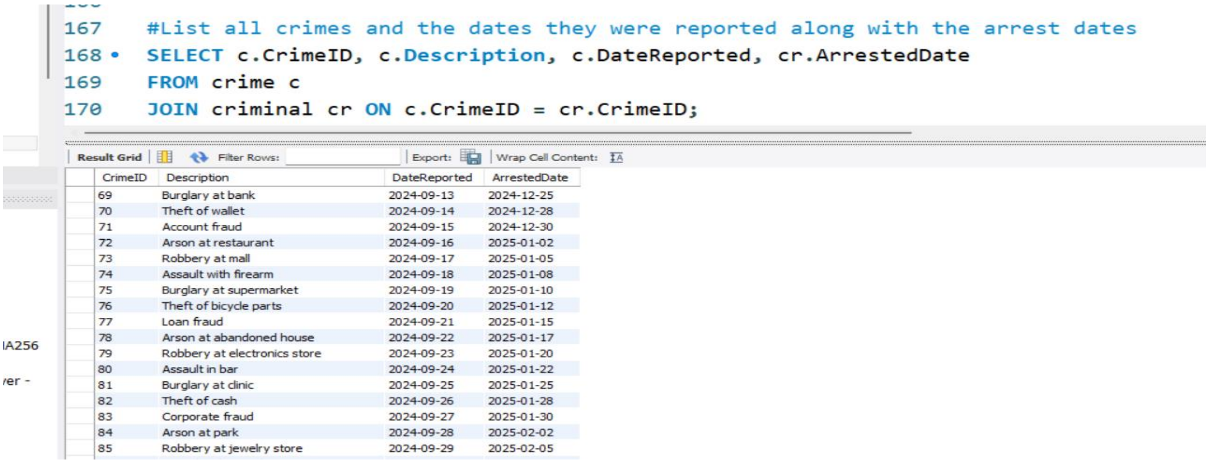
```
Select CASE
    WHEN year(curdate())-year(V.DateOfBirth)  BETWEEN 0 AND 17 THEN '0-17'
    WHEN year(curdate())-year(V.DateOfBirth)  BETWEEN 18 AND 35 THEN '18-35'
    WHEN year(curdate())-year(V.DateOfBirth)  BETWEEN 36 AND 50 THEN '36-50'
    ELSE '51+'
END AS age_group, count(*)Total from Victim  V
Inner Join Crime C On V.CrimeID=C.CrimeID group by age_group;
```

age_group	Total
36-50	32
18-35	68

6. List all crimes and the dates they were reported along with the arrest dates

SELECT c.CrimeID, c.Description, c.DateReported, cr.ArrestedDate

FROM crime c JOIN criminal cr ON c.CrimeID = cr.CrimeID;



```
#List all crimes and the dates they were reported along with the arrest dates
SELECT c.CrimeID, c.Description, c.DateReported, cr.ArrestedDate
FROM crime c
JOIN criminal cr ON c.CrimeID = cr.CrimeID;
```

CrimeID	Description	DateReported	ArrestedDate
69	Burglary at bank	2024-09-13	2024-12-25
70	Theft of wallet	2024-09-14	2024-12-28
71	Account fraud	2024-09-15	2024-12-30
72	Arson at restaurant	2024-09-16	2025-01-02
73	Robbery at mall	2024-09-17	2025-01-05
74	Assault with firearm	2024-09-18	2025-01-08
75	Burglary at supermarket	2024-09-19	2025-01-10
76	Theft of bicycle parts	2024-09-20	2025-01-12
77	Loan fraud	2024-09-21	2025-01-15
78	Arson at abandoned house	2024-09-22	2025-01-17
79	Robbery at electronics store	2024-09-23	2025-01-20
80	Assault in bar	2024-09-24	2025-01-22
81	Burglary at clinic	2024-09-25	2025-01-25
82	Theft of cash	2024-09-26	2025-01-28
83	Corporate fraud	2024-09-27	2025-01-30
84	Arson at park	2024-09-28	2025-02-02
85	Robbery at jewelry store	2024-09-29	2025-02-05

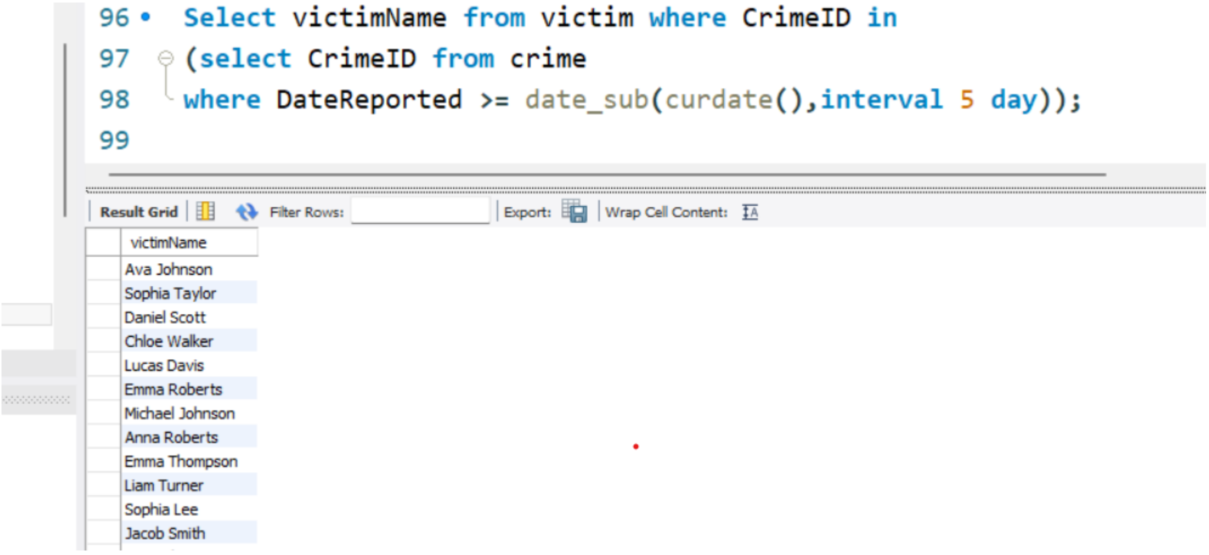
Subqueries Queries:

1. Find the names of victims who were involved in crimes that occurred in the past 5 days:

Select victimName from victim where CrimeID in

(select CrimeID from crime

where DateReported >= date_sub(curdate(),interval 5 day));

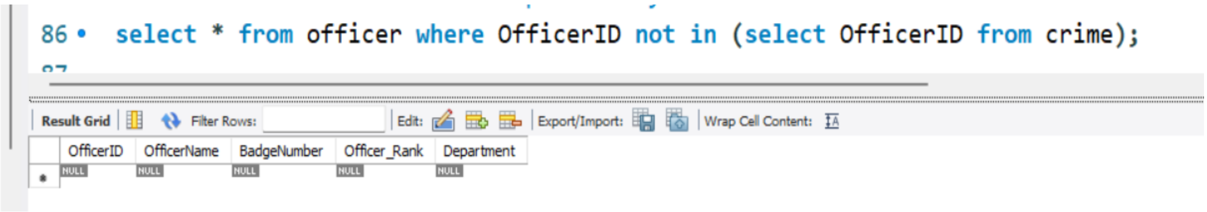


```
Select victimName from victim where CrimeID in
(select CrimeID from crime
where DateReported >= date_sub(curdate(),interval 5 day));
```

victimName
Ava Johnson
Sophia Taylor
Daniel Scott
Chloe Walker
Lucas Davis
Emma Roberts
Michael Johnson
Anna Roberts
Emma Thompson
Liam Turner
Sophia Lee
Jacob Smith

2. Find officers who have not reported any crimes:

select * from officer where OfficerID not in (select OfficerID from crime);



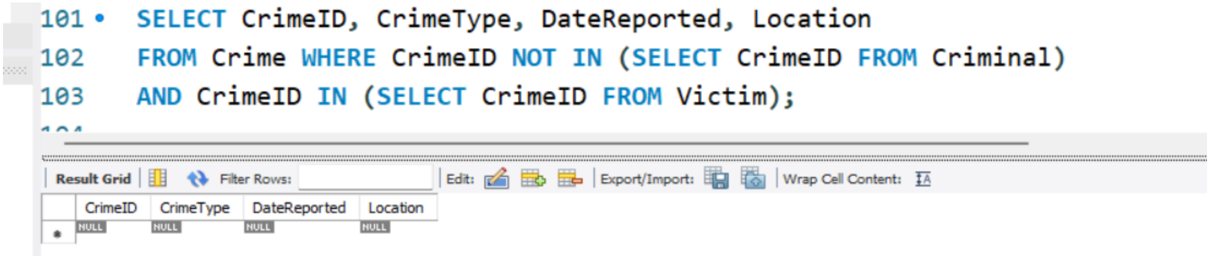
```
select * from officer where OfficerID not in (select OfficerID from crime);
```

OfficerID	OfficerName	BadgeNumber	Officer_Rank	Department
NULL	NULL	NULL	NULL	NULL

3. Get the details of crimes that have victims but no associated criminals:

SELECT CrimeID, CrimeType, DateReported, Location FROM Crime WHERE CrimeID

NOT IN (SELECT CrimeID FROM Criminal) AND CrimeID IN (SELECT CrimeID FROM Victim);



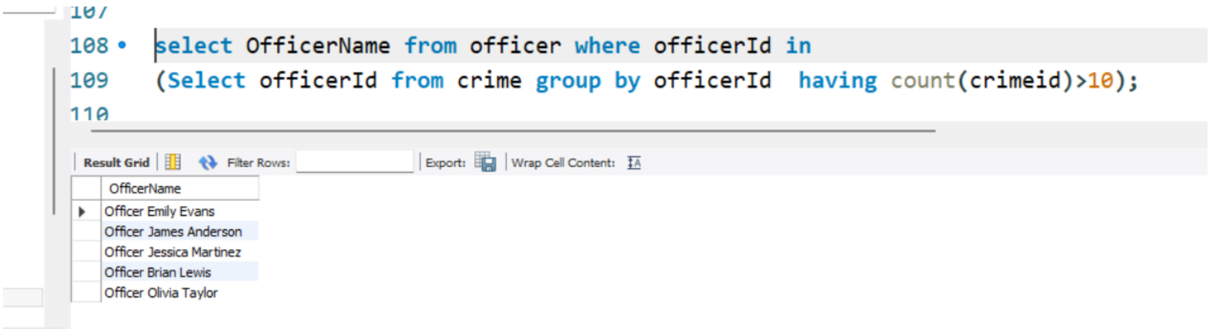
```
SELECT CrimeID, CrimeType, DateReported, Location
FROM Crime WHERE CrimeID NOT IN (SELECT CrimeID FROM Criminal)
AND CrimeID IN (SELECT CrimeID FROM Victim);
```

CrimeID	CrimeType	DateReported	Location
NULL	NULL	NULL	NULL

4. Find the names of officers who have handled more than 10 crimes:

select OfficerName from officer where officerId in

(Select officerId from crime group by officerId having count(crimeid)>10);



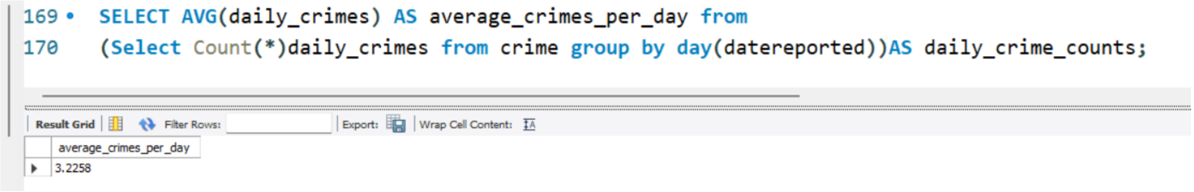
```
select OfficerName from officer where officerId in
(Select officerId from crime group by officerId  having count(crimeid)>10);
```

OfficerName
Officer Emily Evans
Officer James Anderson
Officer Jessica Martinez
Officer Brian Lewis
Officer Olivia Taylor

5. Average Number of Crimes Per Day:

SELECT AVG(daily_crimes) AS average_crimes_per_day from

(Select Count(*)daily_crimes from crime group by day(datereported))AS daily_crime_counts;



```
SELECT AVG(daily_crimes) AS average_crimes_per_day from
(Select Count(*)daily_crimes from crime group by day(datereported))AS daily_crime_counts;
```

average_crimes_per_day
3.2258

Conclusion:

In conclusion, this project successfully created a relational database to manage data on crimes, criminals, victims, and officers. The structured schema and realistic data entries have made it a valuable tool for analyzing crime patterns and supporting law enforcement activities. I developed efficient SQL queries for data retrieval and analysis. This project provided practical tools for law enforcement and enhanced our skills in database management, SQL querying, and data analysis.