

# **US CRIME DATA EXPLORATION AND ANALYSIS**

## **Problem Statement :**

**An organized quantitative and qualitative investigation is done to find trends in crime and disorder. Information on these patterns helps law enforcement agencies deploy resources more effectively. Crime analysis plays an important role in devising solutions to crime problems and formulating crime prevention strategies.**

## **Objectives :**

**It is required to delve deeper into data on different types of crimes and figure out the types of crimes which are more frequent and how they are trending over time.**

## **Dataset Description:**

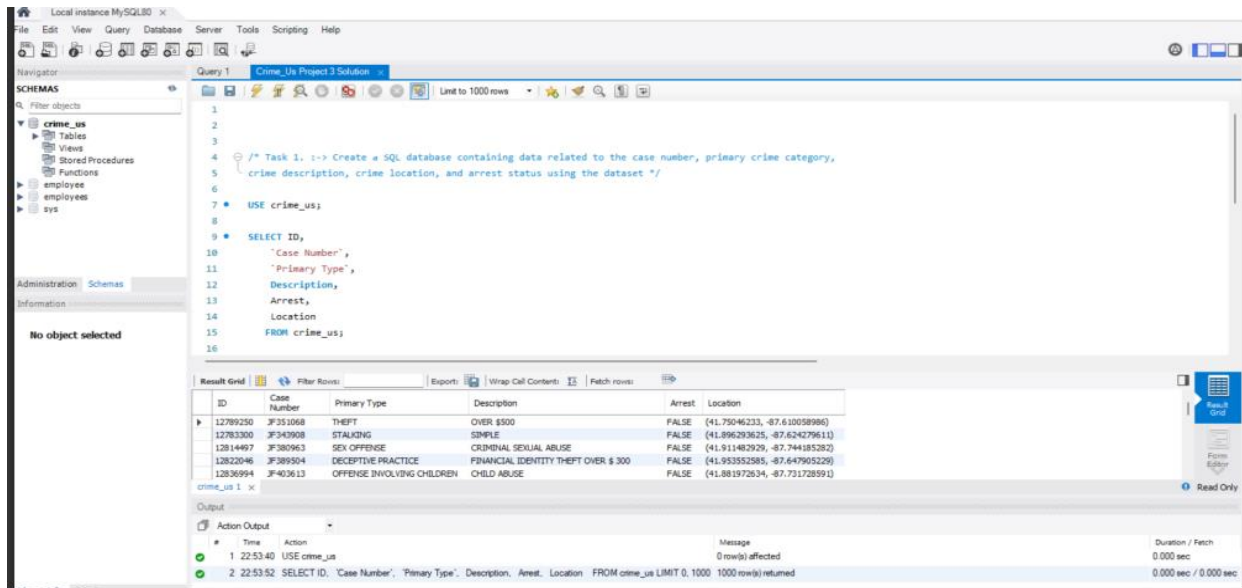
**The file Crime\_us.csv contains the details of crimes, that have occurred in a state of the US in the year 2022.**

## **Variable Description:**

Variable	Description
ID	Shows case id
Case Number	Shows case number
Date	Explicit the date and time of occurrence of the crime
Block	Explains the address where the crime had taken place
IUCR	Is a four-digit code that law enforcement agencies use to classify criminal incidents when taking individual reports
Primary Type	Classify the type of crime
Description	Demonstrates the crime event
Location Description	Explains the type of location where the crime has taken place
Arrest	Shows whether an arrest had been made or not
Domestic	Shows whether the crime was of domestic nature
Beat	Is the territory that a police officer is assigned to patrol
District	Shows the district of the victim
Ward	Shows the ward of the victim
Community Area	Shows the community of the victim
FBI Code	Is the investigation code used to find the criminals
X Coordinate	Shows various information about the location
Y Coordinate	Shows various information about the location
Year	Shows the year
Update On	Signifies the last updating date of the data
Latitude	Shows the latitude of the place
Longitude	Shows the longitude of the place
Location	Shows the location

# Tasks to Perform

1. Create a SQL database containing data related to the case number, primary crime category, crime description, crime location, and arrest status using the dataset.



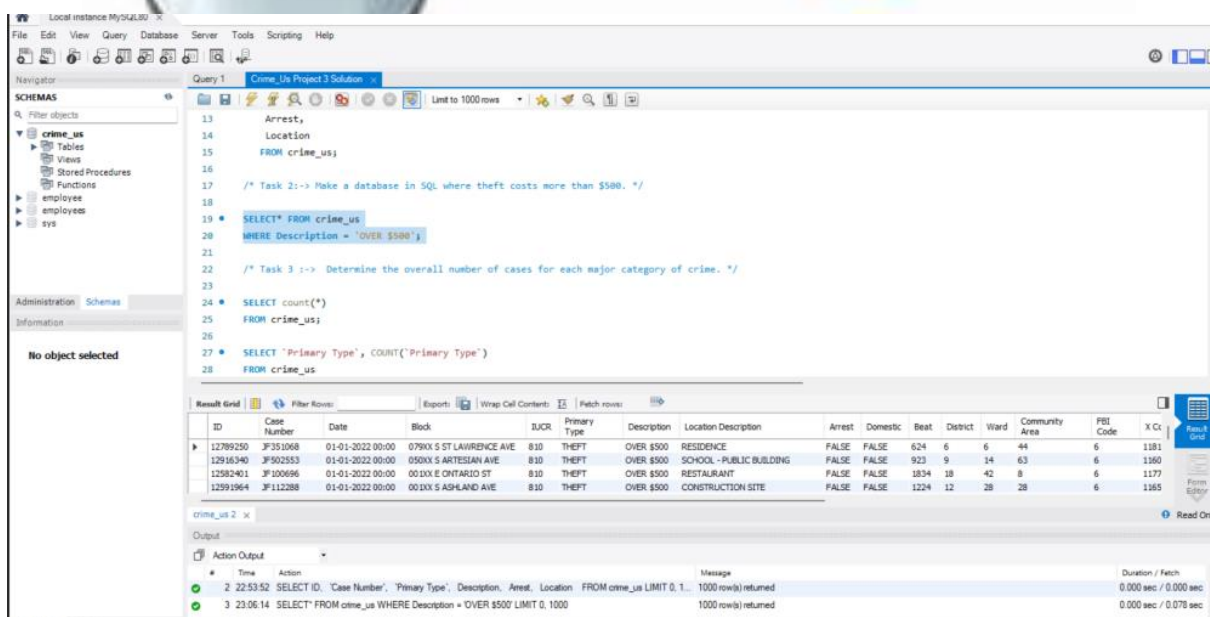
The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' list with 'crime\_us' selected. The main query editor contains the following SQL code:

```
1
2
3
4 /* Task 1: -> Create a SQL database containing data related to the case number, primary crime category,
5 crime description, crime location, and arrest status using the dataset */
6
7 * USE crime_us;
8
9 * SELECT ID,
10 'Case Number',
11 'Primary Type',
12 Description,
13 Arrest,
14 Location
15 FROM crime_us;
16
```

The 'Result Grid' at the bottom shows the output of the query, displaying columns: ID, Case Number, Primary Type, Description, Arrest, and Location. The data includes records for theft, stalking, sexual offense, deceptive practice, and offense involving children.

ID	Case Number	Primary Type	Description	Arrest	Location
12789250	JF351068	THEFT	OVER \$500	FALSE	(41.75046233, -87.61005898)
12783300	JF343908	STALKING	SIMPLE	FALSE	(41.896293625, -87.624279611)
12814487	JF380963	SEX OFFENSE	CORNUAL SEXUAL ABUSE	FALSE	(41.911403029, -87.744185202)
12822046	JF389504	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER \$ 300	FALSE	(41.93352585, -87.647905229)
12836994	JF403613	OFFENSE INVOLVING CHILDREN	CHILD ABUSE	FALSE	(41.881972634, -87.731728591)

2. Make a database in SQL where theft costs more than \$500.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' list with 'crime\_us' selected. The main query editor contains the following SQL code:

```
13
14 Arrest,
15 Location
16 FROM crime_us;
17
18 /* Task 2: -> Make a database in SQL where theft costs more than $500. */
19
20 * SELECT* FROM crime_us
21 WHERE Description = 'OVER $500';
22
23 /* Task 3: -> Determine the overall number of cases for each major category of crime. */
24
25 * SELECT count(*)
26 FROM crime_us;
27
28 * SELECT 'Primary Type', COUNT('Primary Type')
29 FROM crime_us

```

The 'Result Grid' at the bottom shows the output of the query, displaying columns: ID, Case Number, Date, Block, IUCR, Primary Type, Description, Location Description, Arrest, Domestic, Beat, District, Ward, Community Area, FBI Code, and X Co. The data includes records for theft, sexual offense, and deceptive practice.

ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District	Ward	Community Area	FBI Code	X Co
12789250	JF351068	01-01-2022 00:00	079XX S ST LAWRENCE AVE	810	THEFT	OVER \$500	RESIDENCE	FALSE	FALSE	624	6	6	44	6	1181
12916340	JF502553	01-01-2022 00:00	050XX S ARTESIAN AVE	810	THEFT	OVER \$500	SCHOOL - PUBLIC BUILDING	FALSE	FALSE	923	9	14	63	6	1160
12982401	JF100696	01-01-2022 00:00	001XX E ORTAKID ST	810	THEFT	OVER \$500	RESTAURANT	FALSE	FALSE	1834	18	42	8	6	1177
12991964	JF112388	01-01-2022 00:00	001XX S ASHLAND AVE	810	THEFT	OVER \$500	CONSTRUCTION SITE	FALSE	FALSE	1224	12	28	28	6	1165

### 3. Determine the overall number of cases for each major category of crime.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'crime\_us' database schema. The central pane contains the following SQL query:

```
19 SELECT* FROM crime_us
20 WHERE Description = 'OVER $500';
21
22 /* Task 3 :-> Determine the overall number of cases for each major category of crime. */
23
24 SELECT count(*)
25 FROM crime_us;
26
27 SELECT 'Primary Type', COUNT('Primary Type')
28 FROM crime_us
29 group by 'Primary Type'
30 order by count('Primary Type');
31
32 /* Task 4:-> Apply 1NF normalization to the dataset provided */
33
```

The 'Result Grid' at the bottom shows the output of the query, displaying the count of cases for each primary type:

Primary Type	COUNT('Primary Type')
OTHER OFFENSE	5381
ASSAULT	5779
DECEPTIVE PRACTICE	5815
CRIMINAL DAMAGE	8956
BATTERY	14815
THEFT	17986

The 'Output' pane shows the execution log with the following messages:

```
4 23:06:57 SELECT count(*) FROM crime_us LIMIT 0, 1000 1 row(s) returned 0.063 sec / 0.000 sec
5 23:07:26 SELECT 'Primary Type', COUNT('Primary Type') FROM crime_us group by 'Primary Type' order by count('Primary Type') 30 row(s) returned 0.359 sec / 0.000 sec
```

### 4. Apply 1NF normalization to the dataset provided.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'crime\_us' database schema. The central pane contains the following SQL query:

```
24 SELECT count(*)
25 FROM crime_us;
26
27 SELECT 'Primary Type', COUNT('Primary Type')
28 FROM crime_us
29 group by 'Primary Type'
30 order by count('Primary Type');
31
32 /* Task 4:-> Apply 1NF normalization to the dataset provided */
33
34 select cast(new_datetime as date) date_, cast(new_datetime as time) time_
35 from (select Date,
36 coalesce(str_to_date(Date, 'm-d-YY YY'), str_to_date(Date, 'm/YY YY')) as new_datetime
37 from crime_us) d;
```

The 'Result Grid' at the bottom shows the output of the query, displaying the date and time for each row:

date_	time_
2022-01-01	00:00:00
2022-01-01	00:00:00
2022-01-01	00:00:00
2022-01-01	00:00:00
2022-01-01	00:00:00
2022-01-01	00:00:00
2022-01-01	00:00:00
2022-01-01	00:00:00

The 'Output' pane shows the execution log with the following messages:

```
5 23:07:26 SELECT 'Primary Type', COUNT('Primary Type') FROM crime_us group by 'Primary Type' order by count('Primary Type') 30 row(s) returned 0.359 sec / 0.000 sec
6 23:08:39 select cast(new_datetime as date) date_, cast(new_datetime as time) time_ from (select Date, Coalesce(str_to_date(Date, 'm-d-YY YY'), str_to_date(Date, 'm/YY YY')) as new_datetime from crime_us) d 1000 row(s) returned 0.000 sec / 0.000 sec
```

**THANK YOU**

