

Assignment 3- Gathering, Scraping, Munging and Cleaning Data

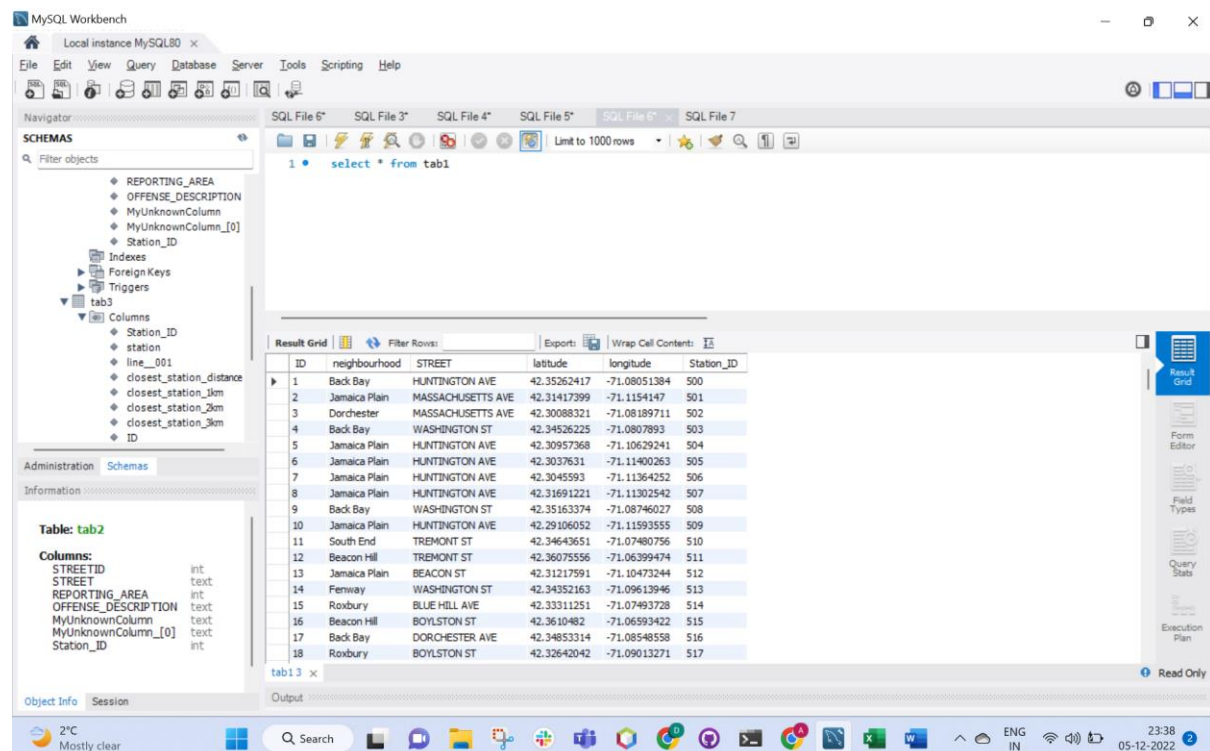
Topic Name-NEU Transportation System

Google Colab Link-

<https://colab.research.google.com/drive/1TUc2razXpRgsD4SscANjOvPMCMASSEVK?usp=sharing>

Sample Table-

Table 1 - Location



The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists a database named 'tab3' with columns: Station_ID, station, line_001, closest_station_distance, closest_station_1km, closest_station_3km, and ID. The 'Query' pane in the center shows the SQL query 'select * from tab1'. The 'Result Grid' on the right displays 18 rows of data with columns: ID, neighbourhood, STREET, latitude, longitude, and Station_ID. The bottom status bar shows the system time as 23:38 on 05-12-2022.

ID	neighbourhood	STREET	latitude	longitude	Station_ID
1	Back Bay	HUNTINGTON AVE	42.35262417	-71.08051384	500
2	Jamaica Plain	MASSACHUSETTS AVE	42.31417399	-71.1154147	501
3	Dorchester	MASSACHUSETTS AVE	42.30088321	-71.08189711	502
4	Back Bay	WASHINGTON ST	42.34526225	-71.0807893	503
5	Jamaica Plain	HUNTINGTON AVE	42.30957368	-71.10629241	504
6	Jamaica Plain	HUNTINGTON AVE	42.3037631	-71.11400263	505
7	Jamaica Plain	HUNTINGTON AVE	42.3045593	-71.11364252	506
8	Jamaica Plain	HUNTINGTON AVE	42.31691221	-71.11302542	507
9	Back Bay	WASHINGTON ST	42.35163374	-71.08746027	508
10	Jamaica Plain	HUNTINGTON AVE	42.29106052	-71.11593555	509
11	South End	TREMONT ST	42.34643651	-71.07480756	510
12	Beacon Hill	TREMONT ST	42.36075556	-71.06399474	511
13	Jamaica Plain	BEACON ST	42.31217591	-71.10473244	512
14	Fenway	WASHINGTON ST	42.34352163	-71.09613946	513
15	Roxbury	BLUE HILL AVE	42.33311251	-71.07493728	514
16	Beacon Hill	BOYLSTON ST	42.3610482	-71.06593422	515
17	Back Bay	DORCHESTER AVE	42.34853314	-71.08548558	516
18	Roxbury	BOYLSTON ST	42.32642042	-71.09013271	517

Table 2- Crime

MySQL Workbench interface showing the 'Crime' table (tab2) with 16 rows of data. The table structure is as follows:

STREETID	STREET	REPORTING_AREA	OFFENSE_DESCRIPTION	Station_ID
1	HUNTINGTON AVE	146	LARCENY SHOPLIFTING	500
2	MASSACHUSETTS AVE	185	VERBAL DISPUTE	501
3	MASSACHUSETTS AVE	185	LARCENY ALL OTHERS	502
4	WASHINGTON ST	450	HARASSMENT	503
5	HUNTINGTON AVE	600	ASSAULT - AGGRAVATED	504
6	HUNTINGTON AVE	600	ASSAULT SIMPLE - BATTERY	505
7	HUNTINGTON AVE	600	DISORDERLY CONDUCT	506
8	HUNTINGTON AVE	600	TRESPASSING	507
9	WASHINGTON ST	562	LARCENY THEFT FROM MV - NON-ACCESSORY	508
10	HUNTINGTON AVE	594	VERBAL DISPUTE	509
11	TREMONT ST	111	ASSAULT - AGGRAVATED - BATTERY	510
12	TREMONT ST	122	M/V - LEAVING SCENE - PROPERTY DAMAGE	511
13	BEACON ST	512	INVESTIGATE PERSON	512
14	WASHINGTON ST	752	ASSAULT - SIMPLE	513
15	BLUE HILL AVE	425	INVESTIGATE PROPERTY	514
16	BOYLSTON ST	117	TRESPASSING	515

Table 3- MBTA

MySQL Workbench interface showing the 'MBTA' table (tab2) with 11 rows of data. The table structure is as follows:

Station_ID	station	line_001	closest_station_distance	closest_station_1km	closest_station_2km	closest_station_3km	ID
500	Copley Station	green	0.388674513	Copley Station	Copley Station	Copley Station	1
501	Green Street Station	orange	0.756237262	Green Street Station	Green Street Station	Green Street Station	2
502	Shawmut Station	red	1.576817878	Shawmut Station	Shawmut Station	Shawmut Station	3
503	Prudential Station	green	0.091817796	Prudential Station	Prudential Station	Prudential Station	4
504	Green Street Station	orange	0.151704646	Green Street Station	Green Street Station	Green Street Station	5
505	Forest Hills Station	orange	0.378969426	Forest Hills Station	Forest Hills Station	Forest Hills Station	6
506	Forest Hills Station	orange	0.471992524	Forest Hills Station	Forest Hills Station	Forest Hills Station	7
507	Stony Brook Station	orange	0.68090611	Stony Brook Station	Stony Brook Station	Stony Brook Station	8
508	Hynes Convention Center	green	0.412716007	Hynes Convention Center	Hynes Convention Center	Hynes Convention Center	9
509	Forest Hills Station	orange	1.065981674	Forest Hills Station	Forest Hills Station	Forest Hills Station	10
510	Bark Rav Station	orange	0.14770921	Bark Rav Station	Bark Rav Station	Bark Rav Station	11

USE CASES

1. Created a store procedure for the crime reports more than 900

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure. The 'maindatabase' is expanded, showing 'tab1' and 'tab2'. 'tab1' has columns: ID, neighbourhood, STREET, latitude, longitude, and Station_ID. 'tab2' has columns: STREETID, STREET, REPORTING_AREA, OFFENSE_DESCRIPTION, and MyUnknownColumn. The 'Administration' tab is selected, and the 'Schemas' sub-tab is active. The main query editor shows a SQL script:

```
1 delimiter &&
2 • create procedure top_STREETW()
3 begin
4 select STREET, REPORTING_AREA
5 from neut.maindatabase where REPORTING_AREA >900;
6 End &&
7 delimiter ;
8 • call top_STREETW();
```

Below the query editor, the 'Result Grid' shows the output of the query:

STREET	REPORTING_AREA
RIVER ST	951
WASHINGTON ST	933
HEATH ST	906
FIDELIS WAY	938

2. Which are the stations near to particular neighbourhood ?

The screenshot shows the same database management tool interface. The 'SCHEMAS' pane on the left is the same. The main query editor shows a SQL query:

```
1 • select distinct tab1.neighbourhood ,tab3.closest_station_1km from tab1 left join tab3 on tab1.station_id = tab3.station_id;
```

Below the query editor, the 'Result Grid' shows the output of the query:

neighbourhood	closest_station_1km
Back Bay	Copley Station
Jamaica Plain	Green Street Station
Dorchester	Prudential Station
Back Bay	Forest Hills Station
Jamaica Plain	Stony Brook Station
Back Bay	Hynes Convention Center
Jamaica Plain	Back Bay Station
South End	Bowdoin Station
Beacon Hill	Kenmore Station
Fenway	Roxbury Crossing Station
Roxbury	Haymarket Station
North End	Bridgeway Station

- To find the nearest station , distance , neighbourhood from the street name
MASSACHUSETTS AVE

The screenshot shows a SQL IDE interface with a Navigator on the left and a SQL editor on the right. The Navigator displays a schema named 'maindatabase' with two tables: 'tab1' and 'tab2'. 'tab1' has columns: ID, neighbourhood, STREET, latitude, longitude, and Station_ID. 'tab2' has columns: STREETID, STREET, REPORTING_AREA, OFFENSE_DESCRIPTION, and MyUnknownColumn. The SQL editor contains the following query:

```
1 SELECT neighbourhood ,STREET , min(closest_station_distance), closest_station_1km
2 from neut.maindatabase where STREET = "MASSACHUSETTS AVE";
```

The result grid shows the following data:

neighbourhood	STREET	min(closest_station_distance)	closest_station_1km
Jamaica Plain	MASSACHUSETTS AVE	0.106319184	Green Street Station

- To count the number of street in each neighbourhood

The screenshot shows a SQL IDE interface with a Navigator on the left and a SQL editor on the right. The Navigator displays a schema named 'maindatabase' with two tables: 'tab1' and 'tab2'. 'tab1' has columns: ID, neighbourhood, STREET, latitude, longitude, and Station_ID. 'tab2' has columns: STREETID, STREET, REPORTING_AREA, OFFENSE_DESCRIPTION, and MyUnknownColumn. The SQL editor contains the following query:

```
1 select distinct neighbourhood ,count(STREET) from neut.maindatabase group by neighbourhood order by STREET DESC;
```

The result grid shows the following data:

neighbourhood	count(STREET)
Fenway	16
Mission Hill	6
Downtown	19
South Boston	12
South End	23
Beacon Hill	24
East Boston	6
West Roxbury	2
Jamaica Plain	30
Dorchester	20
Brighton	12
Allston	9
Mattapan	2
Back Bay	21

- To find the street, neighbourhood name where reporting crime counts are more than 800

The screenshot shows a SQL IDE interface with a Navigator on the left and a SQL editor on the right. The Navigator displays a schema named 'maindatabase' with two tables: 'tab1' and 'tab2'. 'tab1' has columns: ID, neighbourhood, STREET, latitude, longitude, and Station_ID. 'tab2' has columns: STREETID, STREET, REPORTING_AREA, OFFENSE_DESCRIPTION, and MyUnknownColumn. The SQL editor contains the following query:

```
1 SELECT DISTINCT neighbourhood ,STREET, REPORTING_AREA FROM neut.maindatabase WHERE REPORTING_AREA > 800
2
3 GROUP BY STREET,REPORTING_AREA ORDER BY STREET
```

The result grid shows the following data:

neighbourhood	STREET	REPORTING_AREA
South End	BRIGHTON AVE	802
South End	DORCHESTER AVE	817
Back Bay	FIDELIS WAY	938
Allston	HEATH ST	906
Jamaica Plain	RIVER ST	951
Brighton	WASHINGTON ST	933

6. What is the count of reporting crime of a particular neighbourhood ?

The screenshot shows the SQL Fire interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including 'maindatabase', 'tab1', and 'tab2'. The 'tab2' table is selected, showing columns: STREETID, STREET, REPORTING_AREA, OFFENSE_DESCRIPTION, and MyUnknownColumn. The main query editor contains the following SQL statement:

```
1 • select tab1.neighbourhood, tab2.REPORTING_AREA from tab1 inner join tab2 on tab1.STREET = tab2.STREET group by neighbourhood;
```

The 'Result Grid' shows the output of the query, with columns 'neighbourhood' and 'REPORTING_AREA'. The data is grouped by neighbourhood, showing the count of reporting areas for each.

neighbourhood	REPORTING_AREA
Back Bay	283
Jamaica Plain	172
Dorchester	172
South End	102
Beacon Hill	102
Fenway	282
Roxbury	444
North End	646
Mission Hill	282
Charlestown	404
Roslindale	646
Downtown	282
South Boston	282
Chinatown	283

7. To display street name where Reporting crime count more than the avg reporting count

The screenshot shows the SQL Fire interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including 'maindatabase', 'tab1', and 'tab2'. The 'tab2' table is selected, showing columns: STREETID, STREET, REPORTING_AREA, OFFENSE_DESCRIPTION, and MyUnknownColumn. The main query editor contains the following SQL statement:

```
1 • select STREET, REPORTING_AREA
2 from neut.maindatabase where REPORTING_AREA > (select avg( REPORTING_AREA) from neut.maindatabase)
```

The 'Result Grid' shows the output of the query, with columns 'STREET' and 'REPORTING_AREA'. The data lists streets where the reporting count is greater than the average reporting count.

STREET	REPORTING_AREA
WASHINGTON ST	450
HUNTINGTON AVE	600
HUNTINGTON AVE	600
HUNTINGTON AVE	600
HUNTINGTON AVE	600
WASHINGTON ST	562
HUNTINGTON AVE	594
BEACON ST	512
WASHINGTON ST	752
BLUE HILL AVE	425
WASHINGTON ST	778
TALBOT AVE	441
WASHINGTON ST	777
BOYLSTON ST	627
WASHINGTON ST	416

8. To select the station and neighbourhood name which has more than one train connection

The screenshot shows the SQL Fire interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including 'maindatabase', 'tab1', and 'tab2'. The 'tab2' table is selected, showing columns: STREETID, STREET, and REPORTING_AREA. The main query editor contains the following SQL statement:

```
1 • select neighbourhood,station,line__001 from maindatabase where station="Park Street Station"
```

The 'Result Grid' shows the output of the query, with columns 'neighbourhood', 'station', and 'line__001'. The data lists neighbourhoods and stations that have more than one train connection.

neighbourhood	station	line__001
Beacon Hill	Park Street Station	red
Downtown	Park Street Station	orange
Beacon Hill	Park Street Station	green
Downtown	Park Street Station	red

9. To select the street name and station who has committed the offence of Verbal Dispute

The screenshot shows the SQL Studio interface. The left pane displays the database schema with 'tab1' and 'tab2' tables. The main editor shows the following SQL query:

```
1 select tab2.STREET,tab3.station,tab2.OFFENSE_DESCRIPTION
2 from tab2 inner join tab3 on tab2.Station_ID=tab3.Station_ID
3 where tab2.OFFENSE_DESCRIPTION= "VERBAL DISPUTE"
```

The 'Result Grid' at the bottom displays the following data:

STREET	station	OFFENSE_DESCRIPTION
MASSACHUSETTS AVE	Green Street Station	VERBAL DISPUTE
HUNTINGTON AVE	Forest Hills Station	VERBAL DISPUTE
MASSACHUSETTS AVE	Charles/MGH Station	VERBAL DISPUTE
HARVARD ST	Stony Brook Station	VERBAL DISPUTE
RIVER ST	Green Street Station	VERBAL DISPUTE
BLUE HILL AVE	Haymarket Station	VERBAL DISPUTE
WASHINGTON ST	Community College Station	VERBAL DISPUTE
WARREN ST	Cooley Station	VERBAL DISPUTE

10. To select the neighbourhood, street and reported crime area which has the most minimum reported crime

The screenshot shows the SQL Studio interface. The left pane displays the database schema. The main editor shows the following SQL query:

```
1 select tab1.neighbourhood, tab1.STREET,min(tab2.REPORTING_AREA) as Minimum_Reported_Crime
2 from tab1 left join tab2 on tab1.STREET=tab2.STREET
3
```

The 'Result Grid' at the bottom displays the following data:

neighbourhood	STREET	Minimum_Reported_Crime
Back Bay	HUNTINGTON AVE	23

11. To find the neighbourhood name and street name who has reported crime less than 100

The screenshot shows the SQL Studio interface. The left pane displays the database schema. The main editor shows the following SQL query:

```
1 select DISTINCT neighbourhood,STREET from tab1 where exists (select REPORTING_AREA,STREET
2 from tab2 where tab1.STREET = tab2.STREET AND REPORTING_AREA < 100)
3
```

The 'Result Grid' at the bottom displays the following data:

neighbourhood	STREET
Dorchester	CAMBRIDGE ST
Dorchester	BOWDOIN ST
Mission Hill	CAMBRIDGE ST
Fenway	FANEUIL HALL MARKETPLACE
Hyde Park	CAMBRIDGE ST
Beacon Hill	BENNINGTON ST
Hyde Park	BENNINGTON ST
South Boston	BOWDOIN ST

12. To find the Neighbourhood name and street name and who has reported the offense of property lost

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is as follows:

```

1 SELECT distinct tab1.neighbourhood, tab1.STREET,tab2.OFFENSE_DESCRIPTION FROM
2 tab1 inner join tab2 on tab1.STREET=tab2.STREET
3 WHERE OFFENSE_DESCRIPTION ="PROPERTY - LOST"
4

```

The result grid displays the following data:

neighbourhood	STREET	OFFENSE_DESCRIPTION
Back Bay	HUNTINGTON AVE	PROPERTY - LOST
Jamaica Plain	MASSACHUSETTS AVE	PROPERTY - LOST
Dorchester	MASSACHUSETTS AVE	PROPERTY - LOST
Back Bay	WASHINGTON ST	PROPERTY - LOST
Jamaica Plain	HUNTINGTON AVE	PROPERTY - LOST
South End	TREMONT ST	PROPERTY - LOST
Beacon Hill	TREMONT ST	PROPERTY - LOST
Fenway	WASHINGTON ST	PROPERTY - LOST

13. Check for the closest station from 1,2,3km using case expression

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is as follows:

```

1 SELECT distinct Station , closest_station_1km,closest_station_2km,closest_station_3km
2 FROM neut.maindatabase
3 ORDER BY
4 (CASE
5 WHEN closest_station_1km IS NULL THEN closest_station_2km
6 WHEN closest_station_2km IS NULL THEN closest_station_3km
7 ELSE closest_station_1km
8 END);

```

The result grid displays the following data:

Station	closest_station_1km	closest_station_2km	closest_station_3km
Forest Hills Station		Boston College Station	Boston College Station
Boston College Station		Shawmut Station	Shawmut Station
Shawmut Station		South Street Station	South Street Station
South Street Station		Harvard Avenue Station	Harvard Avenue Station
Harvard Avenue Station		Broadway Station	Broadway Station
Broadway Station		Forest Hills Station	Forest Hills Station
Forest Hills Station		Andrew Station	Andrew Station
Andrew Station		Packards Corner Station	Packards Corner Station
Packards Corner Station		Andrew Station	Andrew Station
Andrew Station		Symphony Station	Symphony Station
Symphony Station			

14. To find the recent 5 offense, reported crime of the streets

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is as follows:

```

1 SELECT CONCAT(STREET,REPORTING_AREA,OFFENSE_DESCRIPTION) AS
2 Result FROM neut.maindatabase limit 5

```

The result grid displays the following data:

Result
HUNTINGTON AVE146LARCENY SHOPLIFTING
MASSACHUSETTS AVE185VERBAL DISPUTE
MASSACHUSETTS AVE185LARCENY ALL OTHERS
WASHINGTON ST450HARASSMENT
HUNTINGTON AVE600ASSAULT - AGGRAVATED

15. To find the station, the train line of the street using "USING Clause"

The screenshot shows a SQL IDE interface. On the left, the 'SCHEMAS' pane displays a tree view of database objects, including 'tab3' with columns: Station_ID, station, line__001, closest_station_distance, closest_station_1km, closest_station_2km, and closest_station_3km. The main editor shows the following SQL query:

```
1 • SELECT Distinct e.STREET,d.station, d.line__001
2 FROM tab1 e JOIN tab3 d
3 USING(Station_ID);
```

The 'Result Grid' displays the following data:

STREET	station	line__001
HUNTINGTON AVE	Copley Station	green
MASSACHUSETTS AVE	Green Street Station	orange
MASSACHUSETTS AVE	Shawmut Station	red
WASHINGTON ST	Prudential Station	green
HUNTINGTON AVE	Green Street Station	orange
HUNTINGTON AVE	Forest Hills Station	orange
HUNTINGTON AVE	Stony Brook Station	orange
WASHINGTON ST	Hynes Convention Center	green
TREMONT ST	Back Bay Station	orange
TREMONT ST	Bowdoin Station	blue
BEACON ST	Green Street Station	orange
WASHINGTON ST	Kenmore Station	green
BLUE HILL AVE	Symphony Station	green
BOYLSTON ST	Bowdoin Station	blue
DORCHESTER AVE	Hynes Convention Center	green

The 'Information' pane at the bottom left shows details for 'Table: tab2', including columns: STREETID (int), STREET (text), and REPORTING_AREA (int).

16. To find the closest station, distance from a particular neighbourhood and street.

The screenshot shows a SQL IDE interface. On the left, the 'SCHEMAS' pane displays a tree view of database objects, including 'tab3' with columns: Station_ID, station, line__001, closest_station_distance, closest_station_1km, closest_station_2km, closest_station_3km, and ID. The main editor shows the following SQL query:

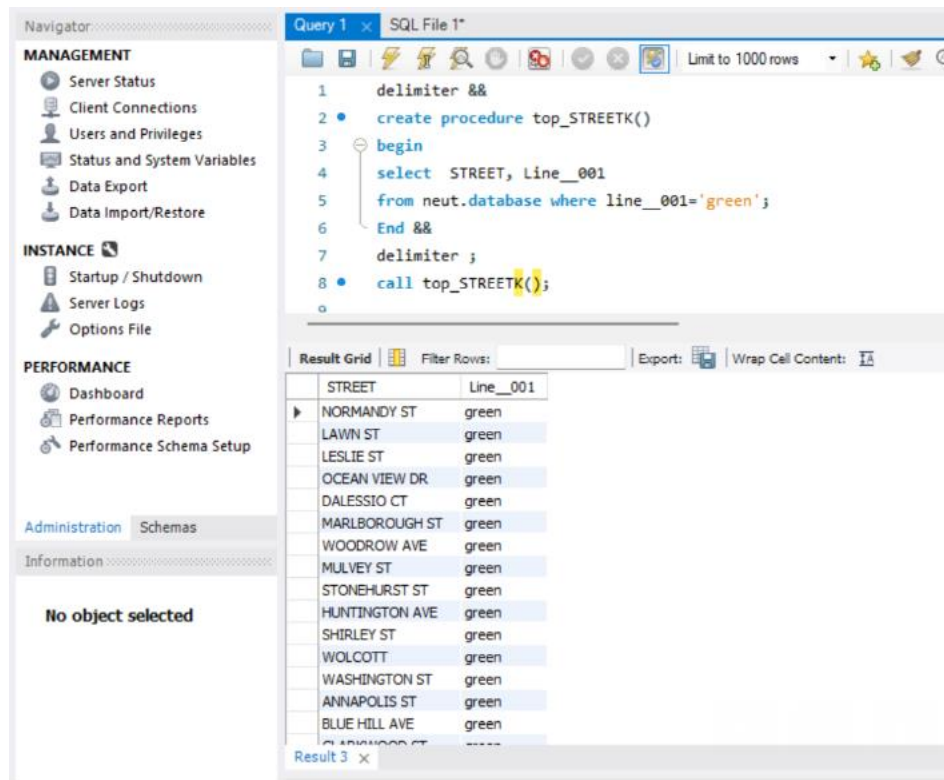
```
1 • select DISTINCT tab1.neighbourhood,tab1.STREET, tab3.closest_station_distance,tab3.closest_station_1km
2 from tab1 inner join tab3 on tab1.ID=tab3.ID
```

The 'Result Grid' displays the following data:

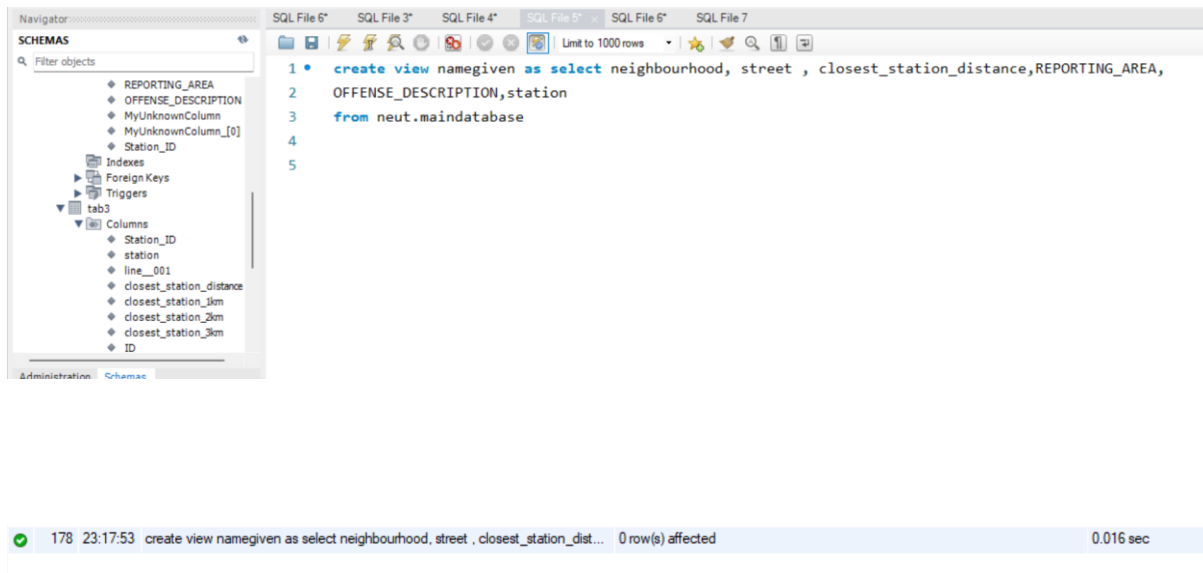
neighbourhood	STREET	closest_station_distance	closest_station_1km
Back Bay	HUNTINGTON AVE	0.388674513	Copley Station
Jamaica Plain	MASSACHUSETTS AVE	0.756237262	Green Street Station
Dorchester	MASSACHUSETTS AVE	1.576817878	
Back Bay	WASHINGTON ST	0.091817796	Prudential Station
Jamaica Plain	HUNTINGTON AVE	0.151704646	Green Street Station
Jamaica Plain	HUNTINGTON AVE	0.378969426	Forest Hills Station
Jamaica Plain	HUNTINGTON AVE	0.471992524	Forest Hills Station
Jamaica Plain	HUNTINGTON AVE	0.68090611	Stony Brook Station
Back Bay	WASHINGTON ST	0.412716007	Hynes Convention Center
Jamaica Plain	HUNTINGTON AVE	1.065981674	
South End	TREMONT ST	0.14220921	Back Bay Station
Beacon Hill	TREMONT ST	0.171207855	Bowdoin Station
Jamaica Plain	BEACON ST	0.323239118	Green Street Station
Fenway	WASHINGTON ST	0.605215481	Kenmore Station
Dorchester	BLUE HILL AVE	1.260077002	

The 'Information' pane at the bottom left shows details for 'Table: tab2', including columns: STREETID (int), STREET (text), REPORTING_AREA (int), OFFENSE_DESCRIPTION (text), MyUnknownColumn (text), and MyUnknownColumn_0 (text).

17.To create a stored procedure to show the street which are connected green line



18.We have created a view and given the name (namegiven)



Now we have called the view which is to find the neighbourhood , street , closest station , reporting area offense description and station

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'REPORTING_AREA', 'OFFENSE_DESCRIPTION', and 'Station_ID'. The 'Information' pane shows details for 'Table: tab2', including columns 'STREETID', 'STREET', 'REPORTING_AREA', and 'OFFENSE_DESCRIPTION'. The main pane displays a query result grid for the query 'select * from namegiven'. The grid contains 17 rows of data, with columns: neighbourhood, street, closest_station_distance, REPORTING_AREA, OFFENSE_DESCRIPTION, and station. The bottom status bar shows an error message: 'Error Code: 1305. PROCEDURE neut.top_STREETE does not exist'.

neighbourhood	street	closest_station_distance	REPORTING_AREA	OFFENSE_DESCRIPTION	station
Back Bay	HUNTINGTON AVE	0.388674513	146	LARCENY SHOPLIFTING	Copley Station
Jamaica Plain	MASSACHUSETTS AVE	0.756237262	185	VERBAL DISPUTE	Green Street Station
Dorchester	MASSACHUSETTS AVE	1.576817878	185	LARCENY ALL OTHERS	Shawmut Station
Back Bay	WASHINGTON ST	0.091817796	450	HARASSMENT	Prudential Station
Jamaica Plain	HUNTINGTON AVE	0.151704646	600	ASSAULT - AGGRAVATED	Green Street Station
Jamaica Plain	HUNTINGTON AVE	0.378969426	600	ASSAULT SIMPLE - BATTERY	Forest Hills Station
Jamaica Plain	HUNTINGTON AVE	0.471992524	600	DISORDERLY CONDUCT	Forest Hills Station
Jamaica Plain	HUNTINGTON AVE	0.68090611	600	TRESPASSING	Stony Brook Station
Back Bay	WASHINGTON ST	0.412716007	562	LARCENY THEFT FROM MV - NON-ACCESSORY	Hynes Convention Center
Jamaica Plain	HUNTINGTON AVE	1.065981674	594	VERBAL DISPUTE	Forest Hills Station
South End	TREMONT ST	0.14220921	111	ASSAULT - AGGRAVATED - BATTERY	Back Bay Station
Beacon Hill	TREMONT ST	0.171207855	122	M/V - LEAVING SCENE - PROPERTY DAMAGE	Bowdoin Station

19. To find street name, neighbourhood and offense description on the basis on reporting crime

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'maindatabase', 'tab1', and 'tab2'. The 'Information' pane shows details for 'Table: tab2', including columns 'STREETID', 'STREET', 'REPORTING_AREA', and 'OFFENSE_DESCRIPTION'. The main pane displays a query result grid for the query 'SELECT DISTINCT tab1.neighbourhood, tab1.STREET, tab2.REPORTING_AREA, tab2.OFFENSE_DESCRIPTION FROM tab1 inner join tab2 on tab1.STREET=tab2.STREET WHERE REPORTING_AREA > 700 GROUP BY STREET,REPORTING_AREA ORDER BY STREET'. The grid contains 17 rows of data, with columns: neighbourhood, STREET, REPORTING_AREA, and OFFENSE_DESCRIPTION. The bottom status bar shows the result count: 'Result 56'.

neighbourhood	STREET	REPORTING_AREA	OFFENSE_DESCRIPTION
Jamaica Plain	BEACON ST	749	INVESTIGATE PROPERTY
Allston	BRIGHTON AVE	796	LARCENY ALL OTHERS
Allston	BRIGHTON AVE	800	LARCENY THEFT FROM BUILDING
Allston	BRIGHTON AVE	802	VANDALISM
Dorchester	CAMBRIDGE ST	795	VAL - VIOLATION OF AUTO LAW - OTHER
Downtown	COMMONWEALTH AVE	786	INVESTIGATE PERSON
Downtown	COMMONWEALTH AVE	788	TOWED MOTOR VEHICLE
Downtown	COMMONWEALTH AVE	793	TOWED MOTOR VEHICLE
Downtown	COMMONWEALTH AVE	794	LANDLORD - TENANT SERVICE
Downtown	COMMONWEALTH AVE	796	LARCENY ALL OTHERS
Back Bay	DORCHESTER AVE	817	TOWED MOTOR VEHICLE
West End	FANEUIL ST	759	LARCENY THEFT FROM MV - NON-ACCES...
Back Bay	FIDELIS WAY	938	SICK/INJURED/MEDICAL - PERSON
Allston	HEATH ST	906	VANDALISM

20.To find the station which is farrest from the street along with the distance using “PARTITION BY” clause

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is as follows:

```
1 • SELECT STREET, station,  
2     MAX(closest_station_distance) OVER (PARTITION BY ID) AS distance  
3 FROM   neut.maindatabase;  
4
```

The result grid displays the following data:

STREET	station	distance
HUNTINGTON AVE	Copley Station	0.388674513
MASSACHUSETTS AVE	Green Street Station	0.756237262
MASSACHUSETTS AVE	Shawmut Station	1.576817878
WASHINGTON ST	Prudential Station	0.091817796
HUNTINGTON AVE	Green Street Station	0.151704646
HUNTINGTON AVE	Forest Hills Station	0.378969426
HUNTINGTON AVE	Forest Hills Station	0.471992524
HUNTINGTON AVE	Stony Brook Station	0.68090611
WASHINGTON ST	Hynes Convention Center	0.412716007
HUNTINGTON AVE	Forest Hills Station	1.065981674
TREMONT ST	Back Bay Station	0.14220921
TREMONT ST	Bowdoin Station	0.171207855
BEACON ST	Green Street Station	0.323239118
WASHINGTON ST	Kenmore Station	0.605215481