

ADVANCED DATABASE SYSTEMS

CSE -6331-002

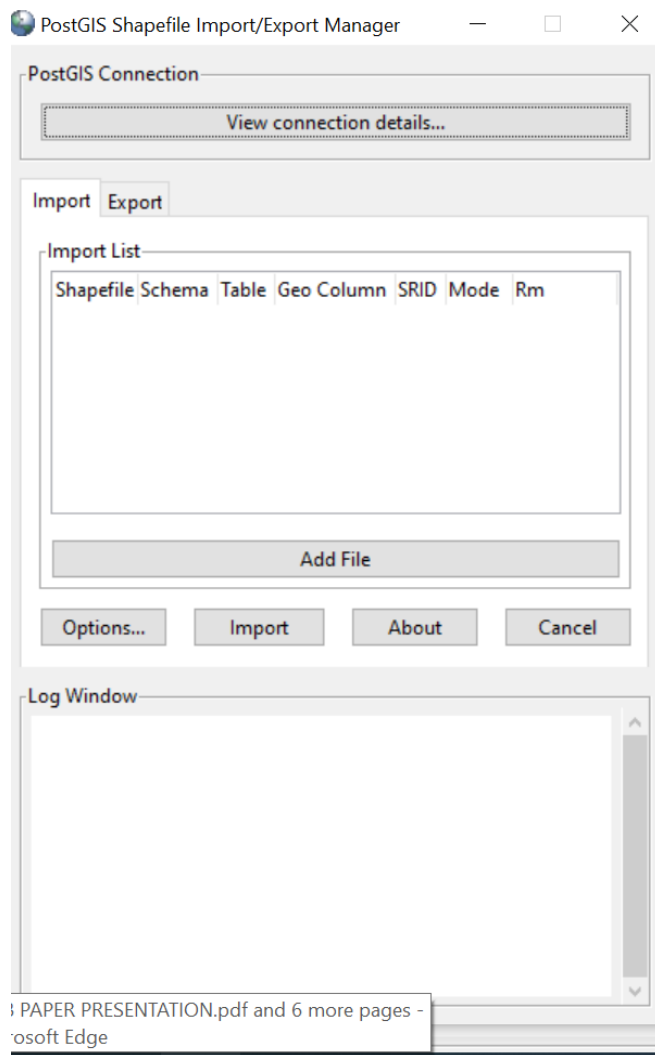
PROJECT – MILESTONE 1

TEAM 2:

- Jeyamaruthi Jayakumar – 1001757737
- Mujahed Khaled Sheikh – 1001764918

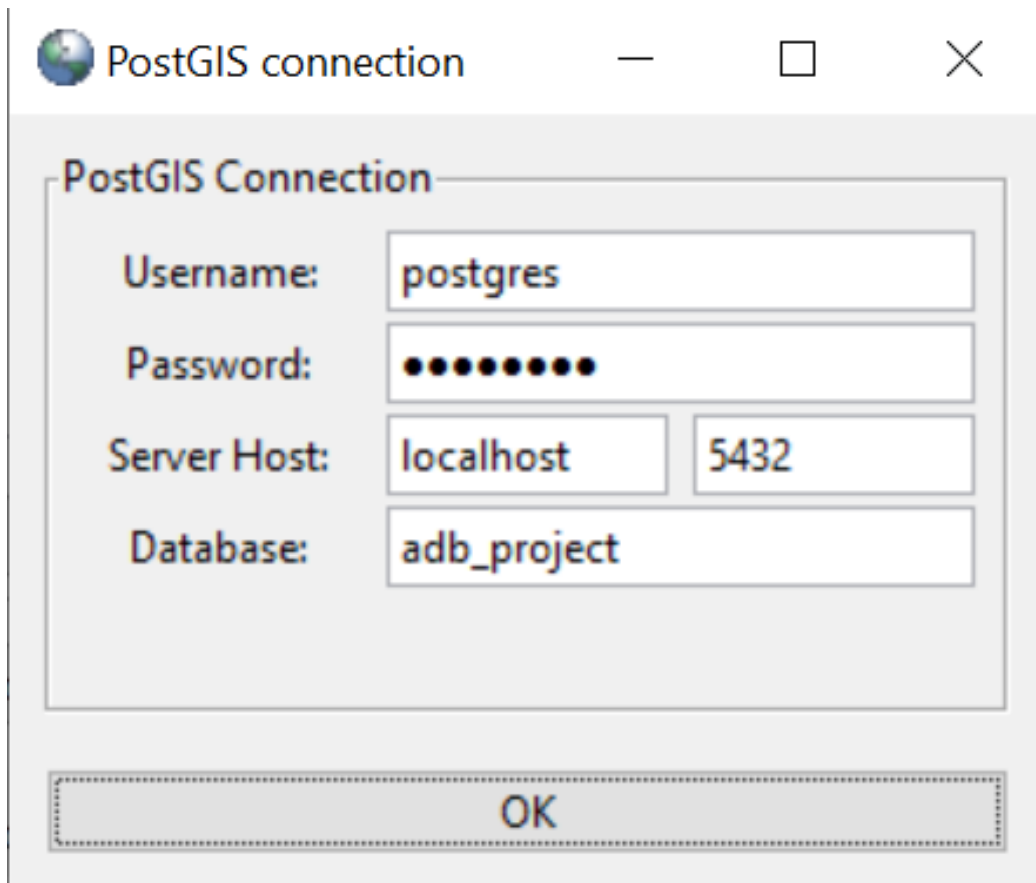
LOADING DATA INTO THE DATABASE

- 1) Using "postGIS PostGIS Bundle 3 for PostgreSQL x64 12 Shapefile and DBF Loader Exporter" which helps in loading the data to the database in POSTGIS
- 2) We get the below window where we first click the "View Connection Details" in order to connect to the database and then we load the files



Step 2:

- 1) Now we Connect to the database using Username, password with server host as local host and port as 5432 and database we want to connect to, our database is adb_project.
- 2) This helps us connecting to the database and upload the required files into the database of postgis.



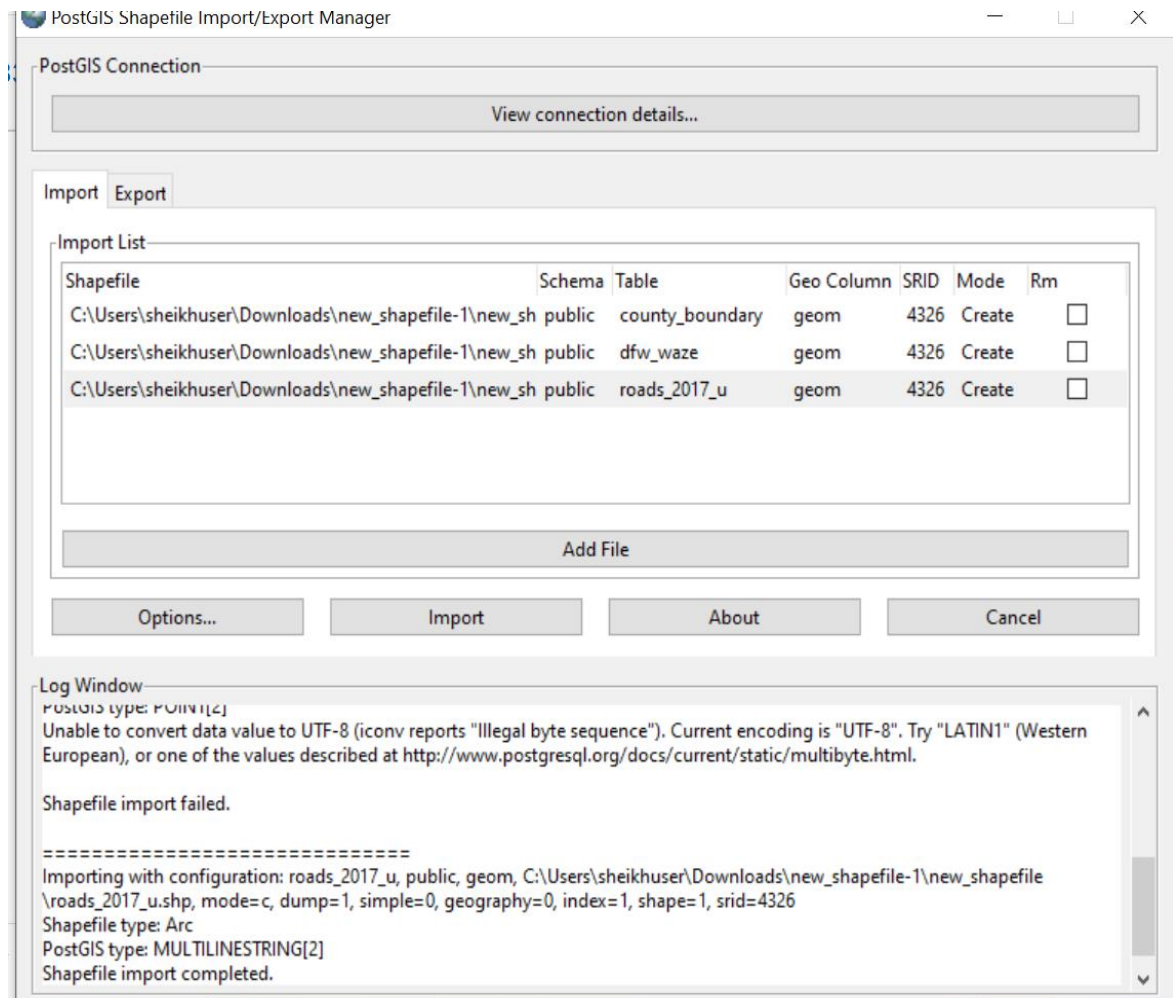
A screenshot of a 'PostGIS connection' dialog box. The window has a title bar with a globe icon, the text 'PostGIS connection', and standard minimize, maximize, and close buttons. The main area is titled 'PostGIS Connection' and contains four labeled input fields: 'Username:' with the text 'postgres', 'Password:' with ten black dots, 'Server Host:' with two sub-fields containing 'localhost' and '5432', and 'Database:' with the text 'adb_project'. At the bottom, there is a large rectangular button labeled 'OK'.

Username:	postgres	
Password:	●●●●●●●●●●	
Server Host:	localhost	5432
Database:	adb_project	

OK

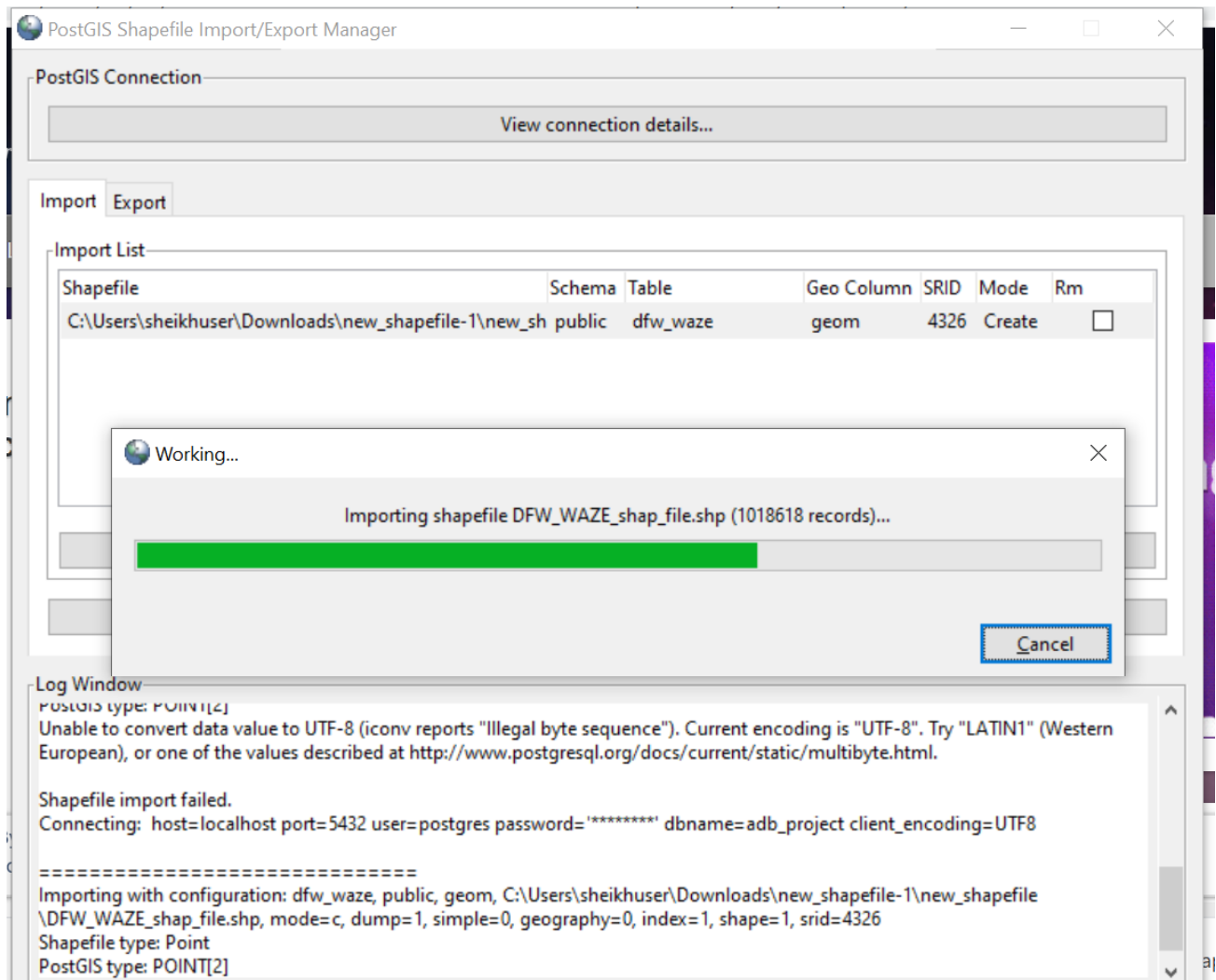
Step 3:

- 1) Once connected we need to add files we want to upload to the databases, in order to do that we click the “Add file” in the above mentioned image, these are the files the file we upload to the database.



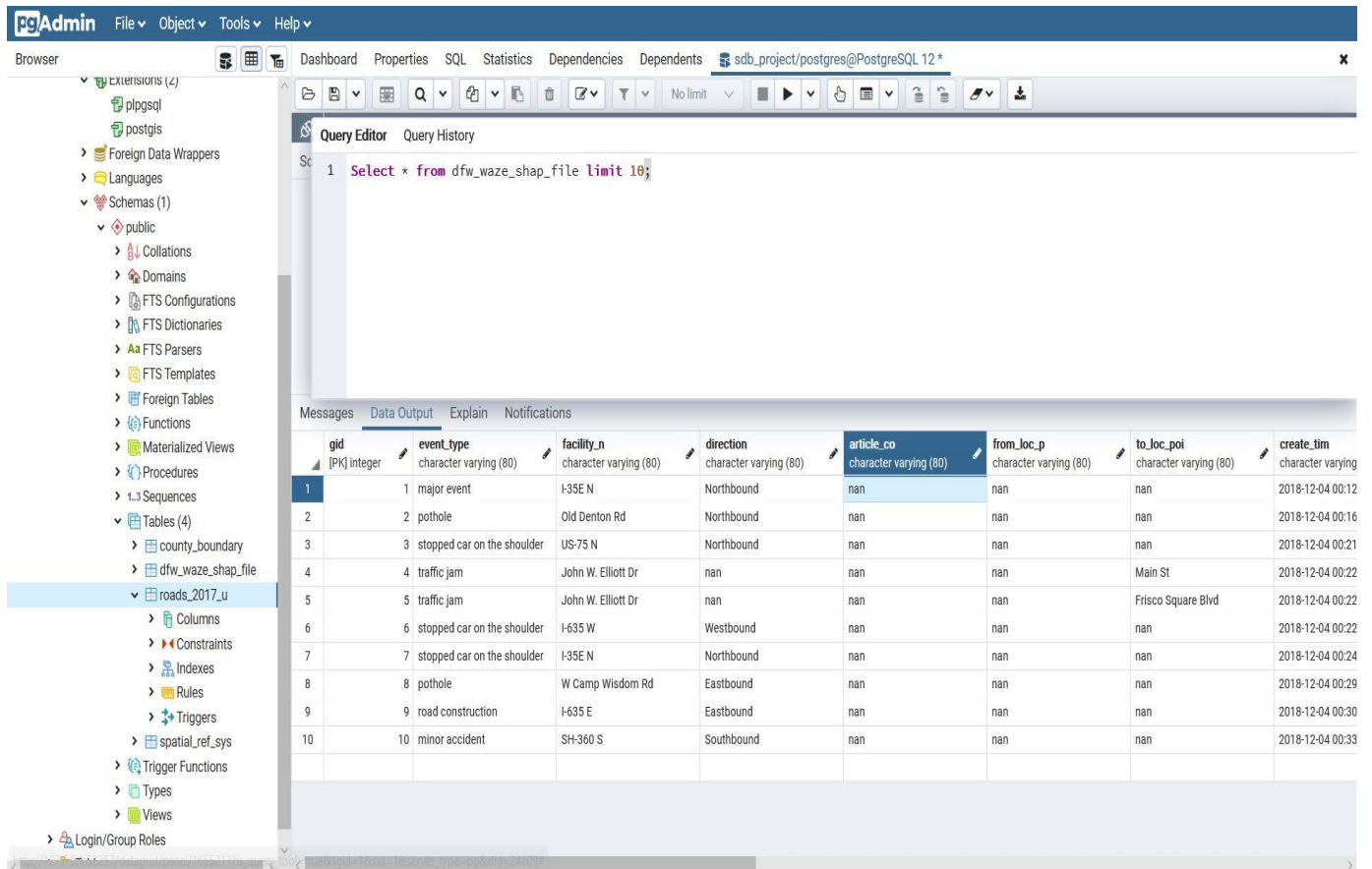
Step 4:

1) Here we uploading the data into the database



ROWS IN THE TABLES OF DATABASE:

TABLE dfw_waze:



The screenshot shows the pgAdmin interface with a query executed in the Query Editor. The query is `Select * from dfw_waze_shap_file limit 10;`. The result is displayed in the Data Output tab, showing 10 rows of data from the `dfw_waze_shap_file` table. The table has columns: `gid` (integer), `event_type` (character varying), `facility_n` (character varying), `direction` (character varying), `article_co` (character varying), `from_loc.p` (character varying), `to_loc.poi` (character varying), and `create_tim` (character varying).

	gid	event_type	facility_n	direction	article_co	from_loc.p	to_loc.poi	create_tim
1	1	major event	I-35E N	Northbound	nan	nan	nan	2018-12-04 00:12
2	2	pothole	Old Denton Rd	Northbound	nan	nan	nan	2018-12-04 00:16
3	3	stopped car on the shoulder	US-75 N	Northbound	nan	nan	nan	2018-12-04 00:21
4	4	traffic jam	John W. Elliott Dr	nan	nan	nan	Main St	2018-12-04 00:22
5	5	traffic jam	John W. Elliott Dr	nan	nan	nan	Frisco Square Blvd	2018-12-04 00:22
6	6	stopped car on the shoulder	I-635 W	Westbound	nan	nan	nan	2018-12-04 00:22
7	7	stopped car on the shoulder	I-35E N	Northbound	nan	nan	nan	2018-12-04 00:24
8	8	pothole	W Camp Wisdom Rd	Eastbound	nan	nan	nan	2018-12-04 00:29
9	9	road construction	I-635 E	Eastbound	nan	nan	nan	2018-12-04 00:30
10	10	minor accident	SH-360 S	Southbound	nan	nan	nan	2018-12-04 00:33

TABLE county_boundary:

pgAdmin 4 File Object Tools Help

Browser Dashboard Properties SQL Statistics Dependencies Dependents sdb_project/postgres@PostgreSQL 12 *

Extensions (2)
plpgsql
postgis
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (4)
county_boundary
dfw_waze_shap_file
roads_2017_u
Columns
Constraints
Indexes
Rules
Triggers
spatial_ref_sys
Trigger Functions
Types
Views
Login/Group Roles

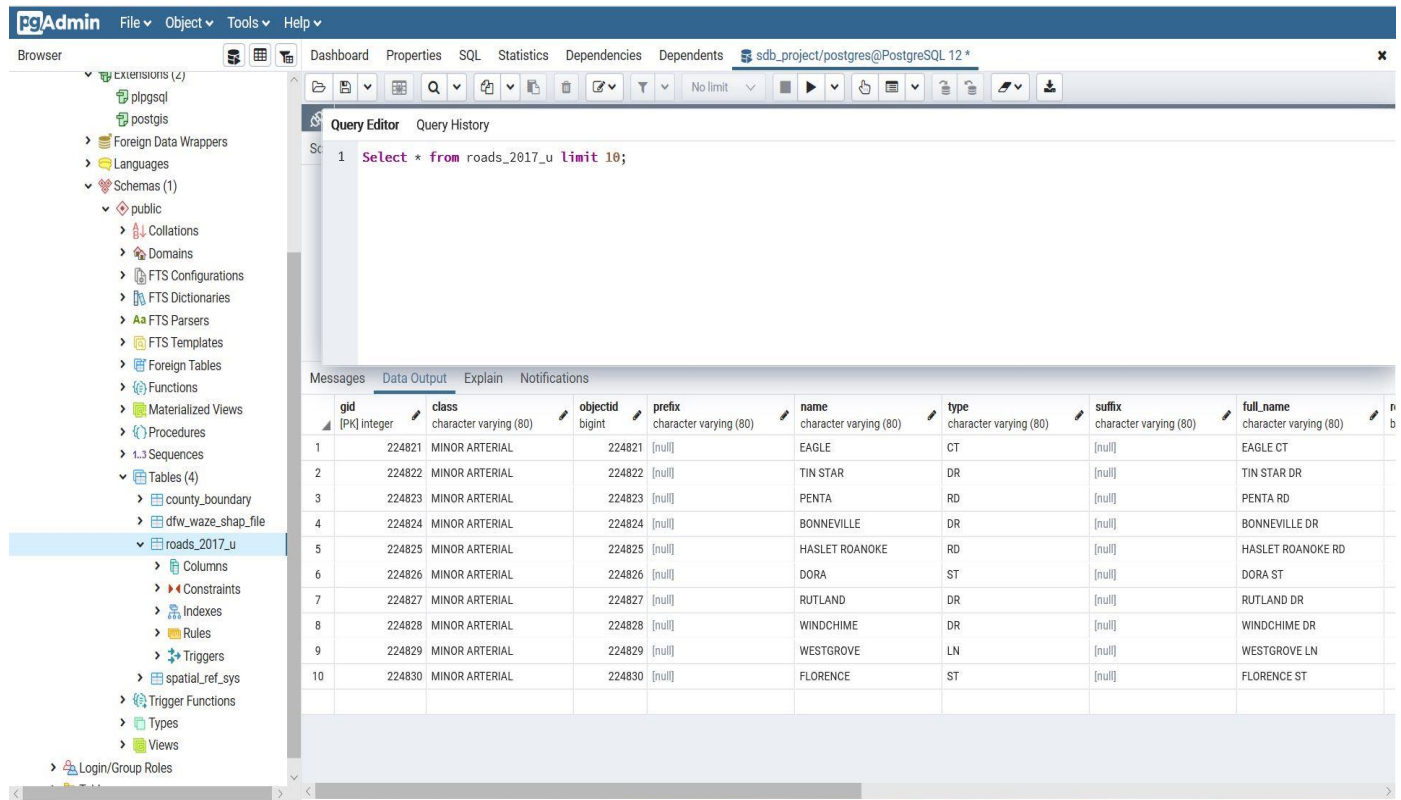
Query Editor Query History

```
1 Select * from county_boundary limit 10;
```

Messages Data Output Explain Notifications

	gid	objectid	cmprtl_cnt	cnty_nm	dps_cnty	fips_st_cn	txdot_cnty	txdot_dist	_gid	shape_are	shape_len	geom
	[PK] integer	bigint	bigint	character varying (80)	bigint	character varying (80)	bigint	bigint	bigint	numeric	numeric	geometry
1	1	103	111	Hood	111	48221	112	2	156	8.2890601158142	324043400993105	0106000020E610X
2	2	104	126	Johnson	126	48251	127	2	157	6.8593797683715	745777122006984	0106000020E610X
3	3	105	70	Ellis	70	48139	71	18	158	7.5898399353027	344544736982789	0106000020E610X
4	4	108	129	Kaufman	129	48257	130	18	161	9.4531297683715	351448850985616	0106000020E610X
5	5	110	234	Van Zandt	234	48467	234	10	163	6.3710899353027	126792976981960	0106000020E610X
6	6	121	182	Palo Pinto	182	48363	182	2	174	4.2304701805114	340522014000453	0106000020E610X
7	7	122	184	Parker	184	48367	184	2	175	4.5976600646972	182904147997033	0106000020E610X
8	8	158	18	Bosque	18	48035	18	9	138	6.3828101158142	392265425017104	0106000020E610X
9	9	161	109	Hill	109	48217	110	9	141	2.6757798194885	374097201973200	0106000020E610X
10	10	162	213	Somervell	213	48425	213	2	142	1.5625000000000	103201183993951	0106000020E610X

TABLE roads_2017_u:



The screenshot shows the pgAdmin 4 interface with a query executed in the Query Editor. The query is `Select * from roads_2017_u limit 10;`. The result is displayed in the Data Output tab as a table with 10 rows and 10 columns. The columns are: gid (PK) integer, class character varying (80), objectid bigint, prefix character varying (80), name character varying (80), type character varying (80), suffix character varying (80), and full_name character varying (80). The data represents road segments with various attributes like gid, class (MINOR ARTERIAL), objectid, prefix, name, type, suffix, and full_name.

	gid (PK) integer	class character varying (80)	objectid bigint	prefix character varying (80)	name character varying (80)	type character varying (80)	suffix character varying (80)	full_name character varying (80)	
1	224821	MINOR ARTERIAL	224821	[null]	EAGLE	CT	[null]	EAGLE CT	
2	224822	MINOR ARTERIAL	224822	[null]	TIN STAR	DR	[null]	TIN STAR DR	
3	224823	MINOR ARTERIAL	224823	[null]	PENTA	RD	[null]	PENTA RD	
4	224824	MINOR ARTERIAL	224824	[null]	BONNEVILLE	DR	[null]	BONNEVILLE DR	
5	224825	MINOR ARTERIAL	224825	[null]	HASLET ROANOKE	RD	[null]	HASLET ROANOKE RD	
6	224826	MINOR ARTERIAL	224826	[null]	DORA	ST	[null]	DORA ST	
7	224827	MINOR ARTERIAL	224827	[null]	RUTLAND	DR	[null]	RUTLAND DR	
8	224828	MINOR ARTERIAL	224828	[null]	WINDCHIME	DR	[null]	WINDCHIME DR	
9	224829	MINOR ARTERIAL	224829	[null]	WESTGROVE	LN	[null]	WESTGROVE LN	
10	224830	MINOR ARTERIAL	224830	[null]	FLORENCE	ST	[null]	FLORENCE ST	

Functions Used:

- st_length
- st_within
- st_centroid
- st_dimension
- st_intersects
- st_area
- st_touches
- st_x
- st_y

Roads_2017_u

This table represents the information regarding the roads.

Attribute Name	Description
GID	It is a primary key, which uniquely describes different tuples.
Class	It specifies different types of class, e.g.: class maybe a high-capacity urban road like “Minor/Major Arterial” & normal “Service Road”
ObjectID	Object ID which represents unique transaction, similar to primary key
Prefix	It holds the prefix of the Road Name
Name	It holds the main name of the Road
Type	It holds the type of the road, e.g.: Street, boulevard etc
Suffix	It holds the suffix of the Road Name

Road_Class	A numerical differentiation to describe a class of a road, e.g.: for Minor Arterial it is 101.
City_I	It holds the data of City Name where the road presents.
County_I	It holds the data of County Name where the road presents.
State	State of the Road
Postal_I	Postal code (if it exists) of the road
Zip_I	Zip code (if it exists) of the road
One_way	Represents the symbol of the One Way
Speed	It represents the max speed limit in a road.
Geom	The Geometric Shape of the Table

County Boundary

This table represents the information regarding the counties.

Attribute Name	Description
Gid	It is a primary key, which uniquely describes different tuples.
Objectid	Object ID which represents unique transaction, similar to primary key.
Cmptrl_cnt	The data holds Texas Comptroller value.
Cnty_nm	The field holds Name of the County.
Dps_cnty_n	The data holds County value of the Department of Public Safety, which is the same value as the Comptroller Value.
Fips_st_cn	It is a foreign key map to the table "Roads_2017_u".
Txdot_cnty	The data holds the Texas Department of Transport county

	value, which is the same as Comptroller Value.
Txdot_dist	The data holds the distance from the county.
..gid	It is a foreign key maps
Shape_are	Attribute represents Shape Area
Shape_len	Attribute represents Shape Length
Geom	The Geometric shape of the Table

Dfw_waze_shap_file

This table represents the information regarding the events occurred in DFW.

Attribute Name	Description
Gid	It is a primary key, which uniquely describes different tuples.
Event_type	It holds the data of event happened during a specific time interval
Facility_n	It holds the data of the road where the event has occurred
Direction	Optional Attribute which specifies the direction of the event
Article_co	If a value if from_loc_p exists, then Article_co will always have value, it is because: article_co state clause regarding the event
From_loc_p	It holds the data of locations where start of the event occurred
To_loc_poi	End of the event occurred

Create_tim	The time at which the event is created
Last_updat	The time at which the event is updated aka transaction time
Close_time	The time at which the transaction is closed
Event_Desc	A brief description of the event
City	The city of place where the event occurred
County	The county where the event occurred
State	The state where the event occurred
Update_num	Which holds the amount of time the data has been updated
Pk_uid	It is a foreign key, from the table 'roads_2017_u'
Day	It is the day when the event has occurred
Geom	The Geometric shape of the Table

SQL QUERIES:

Spatial Queries:

1- (Number) What is the total area of Tarrant county?

Answer:

```
select st_area(C.geom) as "Area" from county_boundary C where  
C.cnty_nm='Tarrant'
```

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Info', 'Table', and 'Preview'. Below these is a toolbar with a 'Query (adb_project)' button and a close icon. A 'Saved query' dropdown menu is set to 'Name', with buttons for 'Save', 'Delete', 'Load File', and 'Save As File'. The main text area contains the SQL query:

```
1 select st_area(C.geom) as "Area"  
2 from county_boundary C where C.cnty_nm='Tarrant'  
3
```

 Below the query editor is a status bar showing 'Execute', '1 rows, 0.254 seconds', 'Create a view', and 'Clear'. A 'Query History' button is also present. The results table is displayed below the status bar, with a single column 'Area' and one row containing the value '1 0.22529427028...'. The table has a light gray header and a white body.

	Area
1	0.22529427028...

2- (List of counties -Text) Which counties are located around Dallas county (share a boundary with Dallas county)?

Answer:

```
SELECT C1.cnty_nm AS "Dallas Neighbours" FROM county_boundary C1, county_boundary C2 WHERE st_touches(C1.geom,C2.geom)='1' AND C2.cnty_nm ='Dallas'
```

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Info', 'Table', and 'Preview'. Below the tabs, there is a 'Query (adb_project)' tab. The main area contains a SQL query. Below the query, there are buttons for 'Execute', '5 rows, 0.518 seconds', 'Create a view', and 'Clear'. To the right of these buttons is a 'Query History' button. Below the query editor, there is a table titled 'Dallas Neighbours' with 5 rows of data.

```
1
2 SELECT C1.cnty_nm AS "Dallas Neighbours" FROM county_boundary C1, county_boundary C2
3 WHERE st_touches(C1.geom,C2.geom)='1'
4 AND C2.cnty_nm ='Dallas'
5
6
```

Dallas Neighbours	
1	Ellis
2	Kaufman
3	Rockwall
4	Collin
5	Tarrant

3) What is the total length of MATLOCK RD that is located in Tarrant county?

Answer:

```
select sum(st_length(geom)) from roads_2017_u
where full_name='MATLOCK RD'
and county_l = 'TARRANT'
group by full_name
```

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Info', 'Table', and 'Preview'. Below these is a toolbar with a 'SQL' icon, a 'Saved query' dropdown, a 'Name' input field, and buttons for 'Save', 'Delete', 'Load File', and 'Save As File'. The main area contains a SQL query with line numbers 1 through 6. Below the query is a horizontal scrollbar. At the bottom, there is a status bar with 'Execute', '1 rows, 0.973 seconds', 'Create a view', and 'Clear' buttons. To the right of the status bar is a 'Query History' button. Below the status bar is a table with one column 'sum' and one row containing the value '0.32336596576...'. The table has a light gray header and a white body.

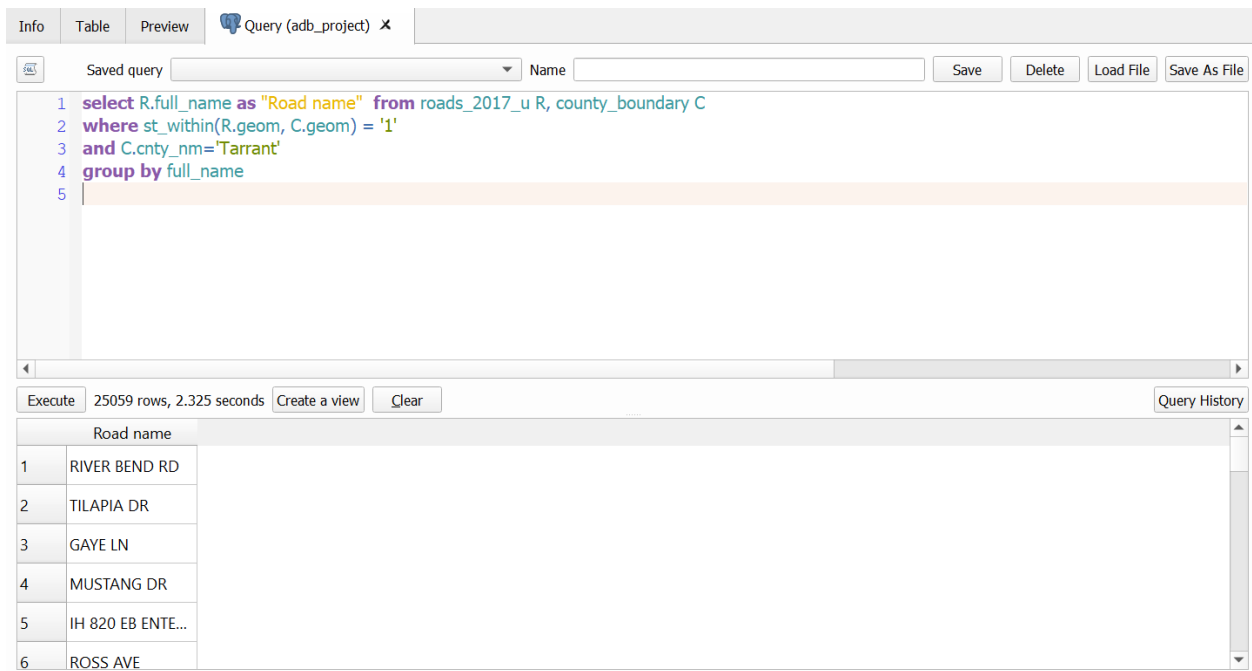
```
1 select sum(st_length(geom)) from roads_2017_u
2 where full_name='MATLOCK RD'
3 and county_l = 'TARRANT'
4 group by full_name
5
6
```

sum
0.32336596576...

4) Return the list of all roads that are totally inside Tarrant county boundary. Note: some roads has more than one record, you need to do group by to return each road name only once

Answer:

```
select R.full_name as "Road name" from roads_2017_u R, county_boundary C
where st_within(R.geom, C.geom) = '1'
and C.cnty_nm='Tarrant'
group by full_name
```



Info Table Preview Query (adb_project) X

Save Saved query Name Save Delete Load File Save As File

```
1 select R.full_name as "Road name" from roads_2017_u R, county_boundary C
2 where st_within(R.geom, C.geom) = '1'
3 and C.cnty_nm='Tarrant'
4 group by full_name
5
```

Execute 25059 rows, 2.325 seconds Create a view Clear Query History

	Road name
1	RIVER BEND RD
2	TILAPIA DR
3	GAYE LN
4	MUSTANG DR
5	IH 820 EB ENTE...
6	ROSS AVE

5) (XY coordinate) Return the coordinates of the center point of Dallas county as text.

Answer:

```
select st_x(st_centroid(geom)), st_y(st_centroid(geom)) from county_boundary where cnty_nm = 'Dallas'
```

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Info', 'Table', and 'Preview'. Below these is a toolbar with buttons for 'Save', 'Delete', 'Load File', and 'Save As File'. The main area contains a SQL query:

```
1 select st_x(st_centroid(geom)), st_y(st_centroid(geom)) from county_boundary
2 where cnty_nm = 'Dallas'
3
```

 Below the query editor, there is a status bar showing 'Execute 1 rows, 0.468 seconds' and buttons for 'Create a view' and 'Clear'. To the right of the status bar is a 'Query History' button. Below the status bar is a table with two columns: 'st_x' and 'st_y'. The table contains one row of data:


	st_x	st_y
1	-96.777826141...	32.7665431978...

6)(Number) Returns the dimension of the geometric object of Hood county?

```
select st_dimension(geom) from county_boundary where cnty_nm = 'Hood'
```

InfoTablePreview

Query (adb_project) X



Saved query

Name

Save

Delete

Load File

Save As File

1

select st_dimension(geom) from county_boundary where cnty_nm = 'Hood'

2

Execute

1 rows, 0.392 seconds

Create a view

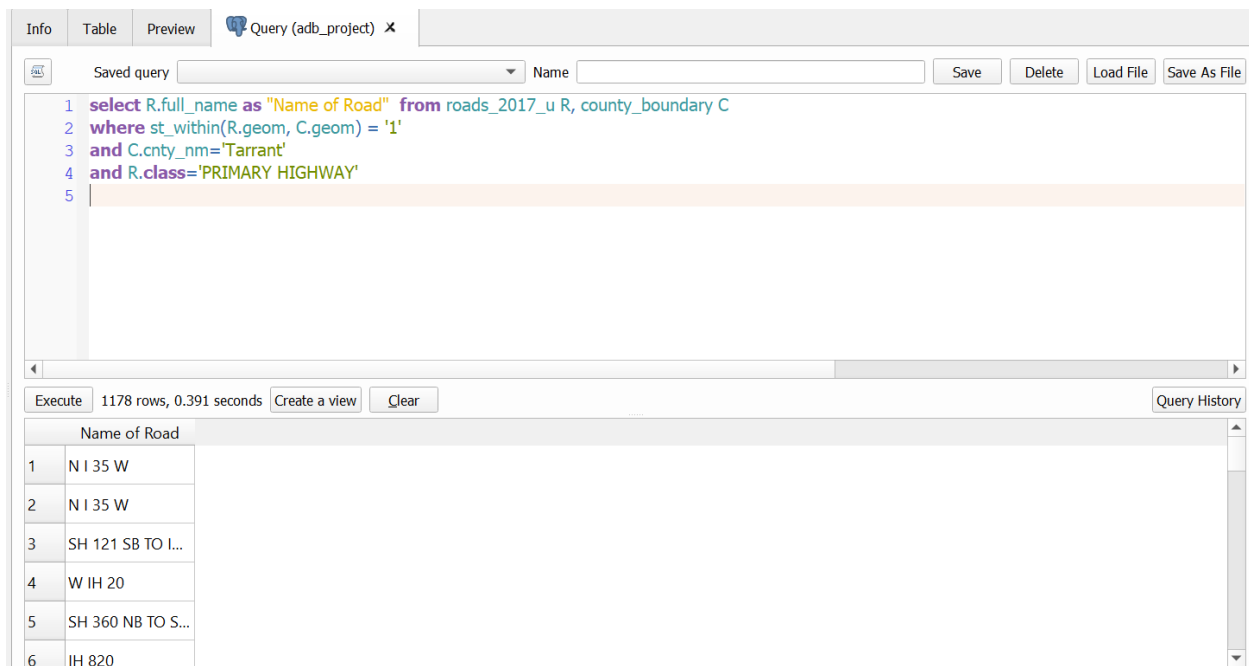
Clear

Query History

	st_dimension
1	2

7)(List of roads) Return the names of the roads with class 'PRIMARY HIGHWAY' in Tarrant county.

```
select R.full_name as "Name of Road" from roads_2017_u R, county_boundary C
where st_within(R.geom, C.geom) and C.cnty_nm='Tarrant' and R.class='PRIMARY HIGHWAY'
```



Query (adb_project) X

Saved query: [dropdown] Name: [input] Save Delete Load File Save As File

```
1 select R.full_name as "Name of Road" from roads_2017_u R, county_boundary C
2 where st_within(R.geom, C.geom) = '1'
3 and C.cnty_nm='Tarrant'
4 and R.class='PRIMARY HIGHWAY'
5
```

Execute 1178 rows, 0.391 seconds Create a view Clear Query History

	Name of Road
1	N I 35 W
2	N I 35 W
3	SH 121 SB TO I...
4	W IH 20
5	SH 360 NB TO S...
6	IH 820

8) (List) Return all the roads that intersect with 'S COOPER ST' in Arlington

```
select R1.name from roads_2017_u R1, roads_2017_u R2
where st_intersects(R1.geom,R2.geom) = '1'
and R1.full_name = 'S COOPER ST'
and R2.city_l = 'ARLINGTON'
```

InfoTablePreviewQuery (adb_project) X

Copy

Saved query

Name

Save

Delete

Load File

Save As File

```
1 select R1.name from roads_2017_u R1, roads_2017_u R2
2 where st_intersects(R1.geom,R2.geom) = '1'
3 and R1.full_name = 'S COOPER ST'
4 and R2.city_l = 'ARLINGTON'
5 |
```

Execute1625 rows, 1.146 secondsCreate a viewClear

Query History

	name
1	COOPER
2	COOPER
3	COOPER
4	COOPER
5	COOPER
6	COOPER

9)(County name and size) Return the name of the largest county in size?

```
select c.cnty_nm, st_area(c.geom) from county_boundary c where st_area(c.geom)
= (select max(st_area(c2.geom)) from county_boundary c2)
```

InfoTablePreviewQuery (adb_project) X

Saved query

Name

SaveDeleteLoad FileSave As File

1

2

3

4

5

```
select c.cnty_nm as "Largest county name", st_area(c.geom) as "Size" from county_boundary c where st_area(c.geom)
= (select max(st_area(c2.geom)) from county_boundary c2)
```

Execute1 rows, 0.476 secondsCreate a viewClear

Query History

	Largest county name	Size
1	Erath	0.26951936937...

Spatio-Temporal:

1) (streets name, GPS points) Return the street name and location (as GPS points) of event with event type "accident" that happened in Tarrant county on 12/09/2018 between 6:00 and 19:00?

```
select Waze.facility_n, st_x(Waze.geom) as "X coord of GPS", st_y(Waze.geom) as "Y coord of GPS" from dfw_waze_shap_file Waze where create_tim > '2018-12-09 06:00:00' and create_tim < '2018-12-09 19:00:00' and event_type = 'accident' and county = 'Tarrant'
```

```
1 select Waze.facility_n, st_x(Waze.geom) as "X coord of GPS", st_y(Waze.geom) as "Y coord of GPS"
2 from dfw_waze_shap_file Waze
3 where create_tim > '2018-12-09 06:00:00'
4 and create_tim < '2018-12-09 19:00:00'
5 and event_type = 'accident' and county = 'Tarrant'
6
7
```

Execute	42 rows, 5.082 seconds	Create a view	Clear
	facility_n	X coord of GPS	Y coord of GPS
1	SH-114 W	-97.057579	32.925104
2	E Arkansas Ln	-97.061668000...	32.706088
3	SH-121 S	-97.048314	32.930224
4	SH-114 E	-97.058968999...	32.923724
5	Crowley Rd	-97.348531000...	32.656588
6	Exit 30: SH-360 ...	-97.057362	32.7598280000...

2) (Number) Retrieve the number of traffic jams in Collin county on 12/27/2018 between 7:00:00 and 15:00:00

```
Select count(*) from dfw_waze_shap_file
where event_type = 'traffic jam' and
create_tim between '2018-12-27 07:00:00' and '2018-12-27 12:00:00'
and (st_dimension(geom), county) = (select st_dimension(geom), county from
dfw_waze_shap_file where county = 'Collin' limit 1)
group by event_type
```

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Info', 'Table', 'Preview', and a 'Query (adb_project)' tab. Below the tabs, there is a 'Saved query' dropdown menu, a 'Name' input field, and buttons for 'Save', 'Delete', 'Load File', and 'Save As File'. The main area contains a SQL query with line numbers 1 through 6. Below the query, there is a 'Query History' button. At the bottom, there is a 'Query History' button. The results section shows a table with one row and one column, 'count', with the value '1 258'.

```
1 Select count(*) from dfw_waze_shap_file
2 where event_type = 'traffic jam' and
3 create_tim between '2018-12-27 07:00:00' and '2018-12-27 12:00:00'
4 and (st_dimension(geom), county) = (select st_dimension(geom), county from dfw_waze_shap_file where county = 'Collin' limit 1)
5 group by event_type
6
```

Execute 1 rows, 5.556 seconds Create a view Clear Query History

count
1 258

Non spatial :

1- (list) List the unique event types from WAZE data set.

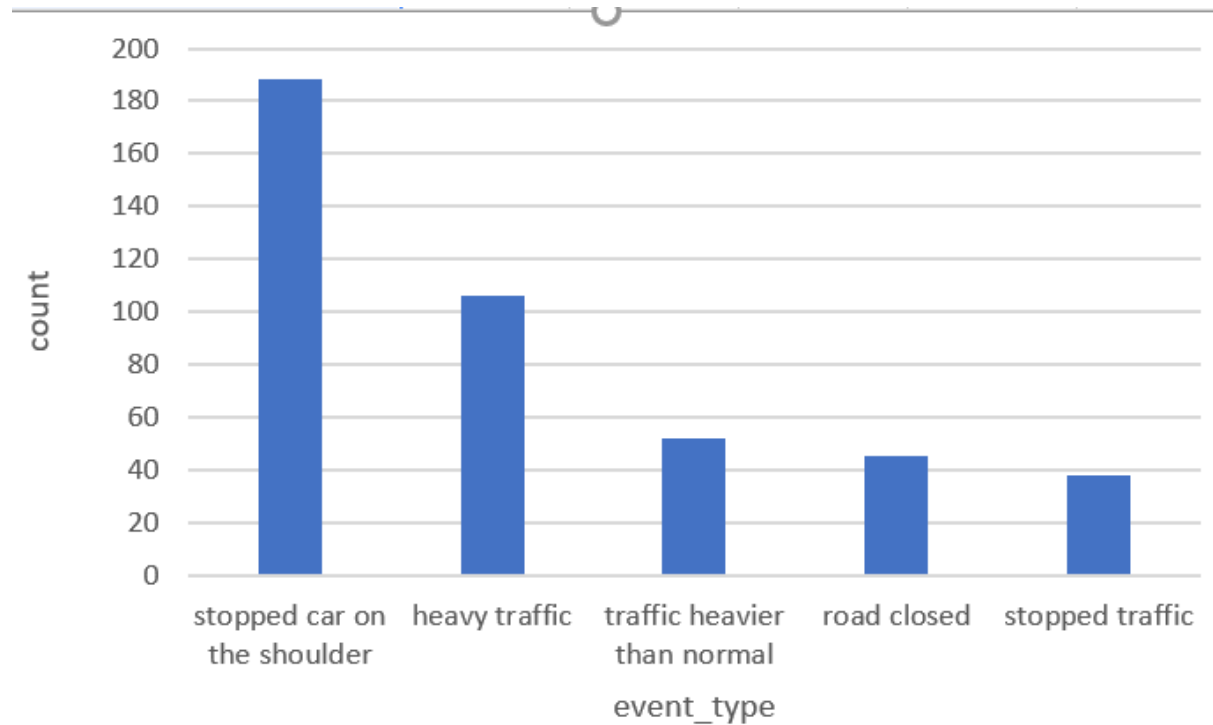
```
select distinct event_type from dfw_waze_shap_file
```

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Info', 'Table', and 'Preview'. Below these is a toolbar with a 'Query (adb_project)' button and a close icon. A 'Saved query' dropdown menu is set to 'Name', with buttons for 'Save', 'Delete', 'Load File', and 'Save As File'. The query editor area contains the SQL statement: `1 select distinct event_type from dfw_waze_shap_file`. Below the editor, there is a status bar showing 'Execute', '32 rows, 4.479 seconds', 'Create a view', and 'Clear'. To the right of the status bar is a 'Query History' button. The results table is displayed below the status bar, with a single column 'event_type'. The table contains six rows of data:

	event_type
1	malfunctioning ...
2	weather hazard
3	heavy traffic
4	stopped car on ...
5	large traffic jam
6	animal on the s...

2) (bar chart- only the result from the 2nd row to the 6th row) Return the WAZE event type and total of each type that happened in Arlington on 12/08/2018 order from the largest.

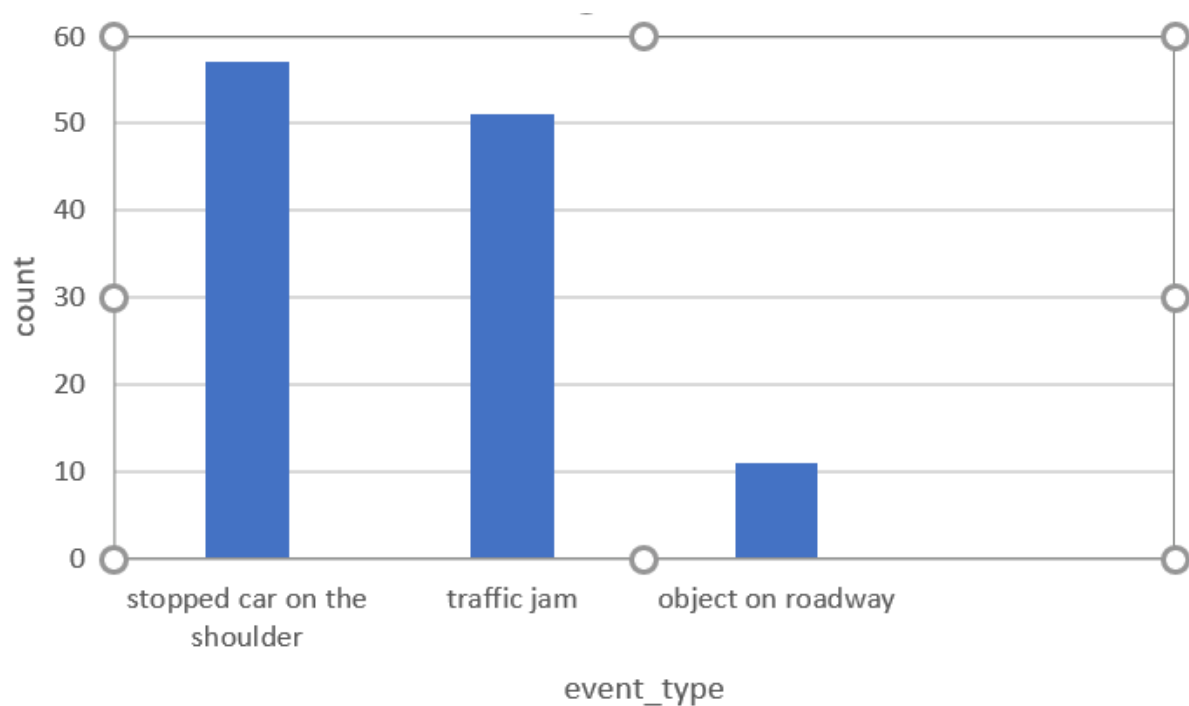
```
select D.event_type, count(*) from dfw_waze_shap_file D where D.city='Arlington'  
and D.create_tim > '2018-12-08 00:00:00' and D.create_tim < '2018-12-08 23:59:59'  
group by D.event_type order by count(*) desc
```



3) (bar chart – only the result from the 1st row to the 3rd row) Return the WAZE event type and total of each type that happened in Dallas on 12/25/2018 between 10:00:00 and 12:00:00 order from the largest

Answer:

```
select D.event_type, count(*) from dfw_waze_shap_file D
where D.city = 'Dallas'
and D.create_time between '2018-12-25 10:00:00' and '2018-12-25 12:00:00' group by
D.event_type order by count(*) desc LIMIT 3
```



4- (Table with event type and total) Retrieve total of each WAZE events that happened on the third Sunday in Dallas city.

```
select event_type as " Event Type", count(*) as "count" from dfw_waze_shap_file
where create_tim >= '2018-12-15' and create_tim <= '2018-12-21'
and day = 'Sunday'
and city = 'Dallas'
group by event_type;
```

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Info', 'Table', and 'Preview', followed by a 'Query (Project) X' tab. Below the tabs, there is a 'Saved query' dropdown menu and a 'Name' input field. To the right of these are buttons for 'Save', 'Delete', 'Load File', and 'Save As File'. The main area contains a SQL query with line numbers 1 through 6. Below the query, there is a status bar showing 'Execute', '0 rows, 0.425 seconds', 'Create a view', and 'Clear'. On the far right of the status bar is a 'Query History' button. Below the status bar, there is a table header with two columns: 'Event Type' and 'count'. The table body is currently empty.

```
1 select event_type as " Event Type", count(*) as "count" from dfw_waze_shap_file
2 where create_tim >= '2018-12-15' and create_tim <= '2018-12-21'
3 and day = 'Sunday'
4 and city = 'Dallas'
5 group by event_type;
6
```

Execute 0 rows, 0.425 seconds Create a view Clear Query History

Event Type	count
------------	-------