CSCI 5408
Summer 2018

Assignment 5

Using Data Lake Analytics to capture and analyze real time traffic data set and summarize findings

Submitted by:-
Nikhil Dhirmalani: B00775542
Khushboo Siwal: B00781497

TABLE OF CONTENTS

1. Task Description
2. Tool Selection
3. Data Loading
4. Dashboard
5. Output
6. Code Submission
7. References

LIST OF FIGURES

Figure 1 Data Loading Query 5
Figure 2 Data Loaded Output 6
Figure 3 Query 1 7
Figure 4 Output for the query 8
Figure 5 Query 2 9
Figure 6 Output from Query 2 9
Figure 7 Query 3 10
Figure 8 Output from the query 11
Figure 9 Query 4 12
Figure 10 Output from Query 4 12
Figure 11 Query 5 14
Figure 12 Output from Query 5 14
Figure 13 Bar graph plotted for query 1, reflecting that WILLAM R ALLEN RD has highest average for vehicles traffic for all the years 15
Figure 14 Pie chart plotted for query 2, YONGE ST had maximum percentage of traffic for the past 5 years 16
Figure 15 Line graph plotted for query 3 which shows that year 2016 has maximum traffic 17
Figure 16 Surface chart plotted for the query 4 which shows Wednesday has maximum average of total pedestrian and vehicle traffic for all years 18
Figure 17 3D Scatter plot plotted for query 5 which reflects BAY ST having maximum traffic on Thursday 19

1. TASK DESCRIPTION

The tasks for this assignment is to learn data analytics using MS Azure Data Lake and learn how to work on data analytics dashboard. The activities involved us to load Traffic Signal Vehicle and Pedestrian Volumes data to MS Azure Data Lake and performing some queries for the given scenarios. Then the queried data needs to be loaded into visual analytical tool for plotting graphs for the specific cases.

2. TOOL SELECTION

Plotly library was chosen for data visualization as it is compatible with python and easy to learn. The library is very intuitive to use. The wide ranges of graphs in Plotly makes it easy to create visual representations of datasets, and the wide range of visualisation types and exploration options gives access to meaningful insights. We have used python programming language in order to get the output csv files from Data Lake into Pandas data frame and use Plotly to plot data from dataframe.

Azure Data Lake has capabilities like storing data of any size, shape or speed. It can do all types of processing and analytics across different platforms and languages. Data Lake analytics include U-SQL, query language that extends the familiar, simple, declarative nature of SQL with the expressive power of C#. It helps develop faster, debug and optimize smartly using familiar tools.

3. DATA LOADING AND QUERIES

We have used the given query to extract data from uploaded data.csv file into format suitable for the USQL. The results are stored in DataForUsql.csv file in output folder in azure Data Lake.

```
@data =
  EXTRACT
        Tcs int,
        Main string,
        Side_1_Route string,
        Activation_Date DateTime,
        Latitude double,
        Longitude double,
        Count_Date DateTime,
        eight_Peak_Hr_Vehicle_Volume int,
        eight_Peak_Hr_Pedestrian_Volume int
  FROM "/data/data.csv"
  USING Extractors.Csv(encoding: Encoding.UTF8, skipFirstNRows: 1);
```

```
OUTPUT @data
TO "/output/DataForUsql.csv"
USING Outputters.Csv(outputHeader: true);
```

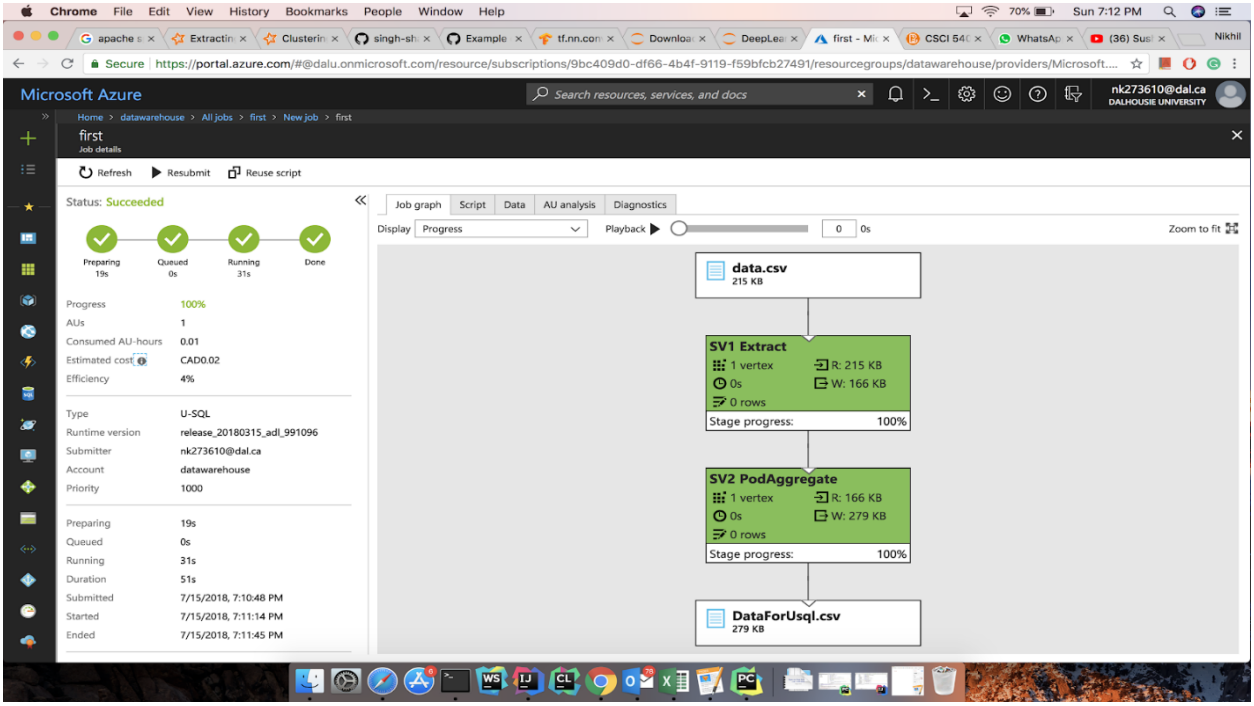
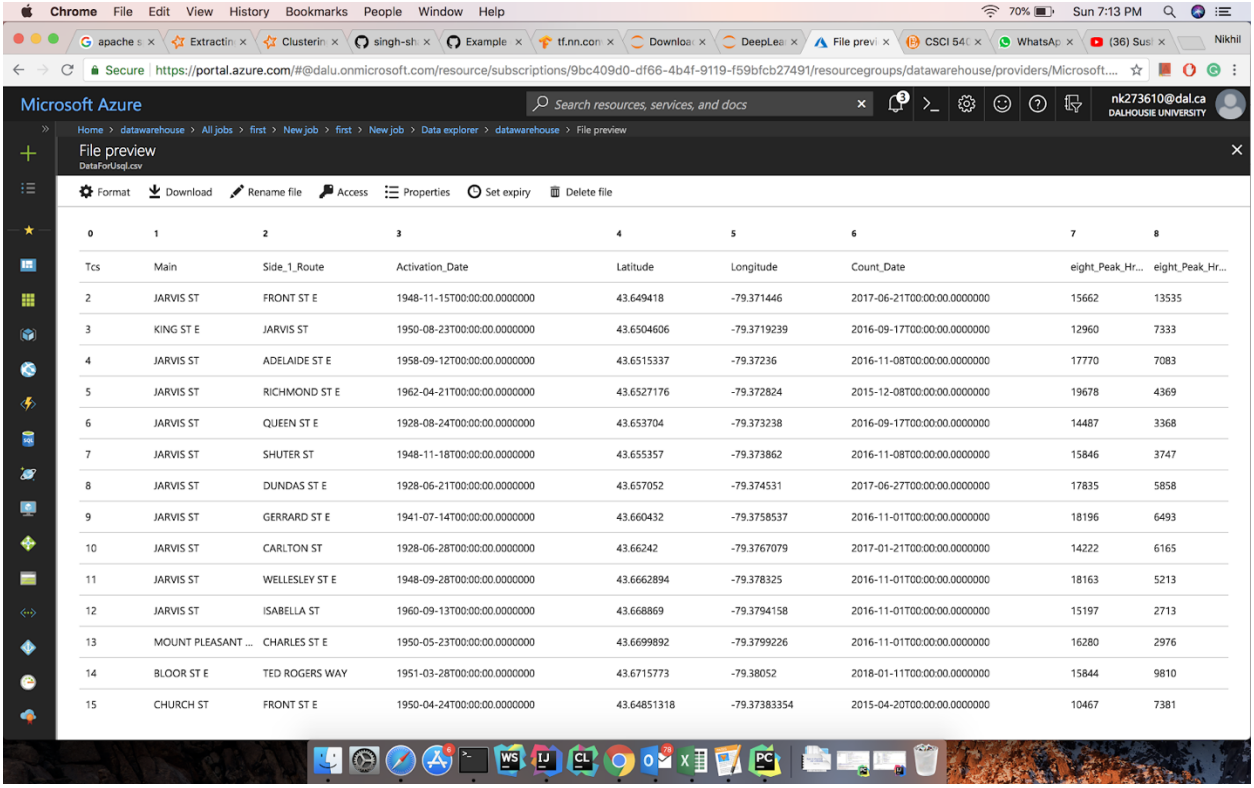


Figure 1 Data Loading Query



0	1	2	3	4	5	6	7	8
Tcs	Main	Side_1_Route	Activation_Date	Latitude	Longitude	Count_Date	eight_Peak_Hr...	eight_Peak_Hr...
2	JARVIS ST	FRONT ST E	1948-11-15T00:00:00.0000000	43.649418	-79.371446	2017-06-21T00:00:00.0000000	15662	13535
3	KING ST E	JARVIS ST	1950-08-23T00:00:00.0000000	43.6504606	-79.3719239	2016-09-17T00:00:00.0000000	12960	7333
4	JARVIS ST	ADELAIDE ST E	1958-09-12T00:00:00.0000000	43.6515337	-79.37236	2016-11-08T00:00:00.0000000	17770	7083
5	JARVIS ST	RICHMOND ST E	1962-04-21T00:00:00.0000000	43.6527176	-79.372824	2015-12-08T00:00:00.0000000	19678	4369
6	JARVIS ST	QUEEN ST E	1928-08-24T00:00:00.0000000	43.653704	-79.373238	2016-09-17T00:00:00.0000000	14487	3368
7	JARVIS ST	SHUTER ST	1948-11-18T00:00:00.0000000	43.655357	-79.373862	2016-11-08T00:00:00.0000000	15846	3747
8	JARVIS ST	DUNDAS ST E	1928-06-21T00:00:00.0000000	43.657052	-79.374531	2017-06-27T00:00:00.0000000	17835	5858
9	JARVIS ST	GERRARD ST E	1941-07-14T00:00:00.0000000	43.660432	-79.3758537	2016-11-01T00:00:00.0000000	18196	6493
10	JARVIS ST	CARLTON ST	1928-06-28T00:00:00.0000000	43.66242	-79.3767079	2017-01-21T00:00:00.0000000	14222	6165
11	JARVIS ST	WELLESLEY ST E	1948-09-28T00:00:00.0000000	43.6662894	-79.378325	2016-11-01T00:00:00.0000000	18163	5213
12	JARVIS ST	ISABELLA ST	1960-09-13T00:00:00.0000000	43.668869	-79.3794158	2016-11-01T00:00:00.0000000	15197	2713
13	MOUNT PLEASANT ...	CHARLES ST E	1950-05-23T00:00:00.0000000	43.6699892	-79.3799226	2016-11-01T00:00:00.0000000	16280	2976
14	BLOOR ST E	TED ROGERS WAY	1951-03-28T00:00:00.0000000	43.6715773	-79.38052	2018-01-11T00:00:00.0000000	15844	9810
15	CHURCH ST	FRONT ST E	1950-04-24T00:00:00.0000000	43.64851318	-79.37383354	2015-04-20T00:00:00.0000000	10467	7381

Figure 2 Data Loaded Output

Query 1: Aggregate results based on "Main Street Name"and calculate average volume of vehicles for all available years provided in the dataset. You need to calculate one average value for each individual "Main Street Name".

```
@data =
  EXTRACT
        Tcs int,
        Main string,
        Side_1_Route string,
        Activation_Date DateTime,
        Latitude double,
        Longitude double,
        Count_Date DateTime,
        eight_Peak_Hr_Vehicle_Volume int,
        eight_Peak_Hr_Pedestrian_Volume int
  FROM "/data/data.csv"
  USING Extractors.Csv(encoding: Encoding.UTF8, skipFirstNRows: 1);

@rs1=
SELECT Main, AVG(eight_Peak_Hr_Vehicle_Volume) AS Average_Vehicles
FROM @data  GROUP BY Main;

OUTPUT @rs1
TO "/output/dataquery1.csv"
```

```
USING Outputters.Csv(outputHeader: true);
```

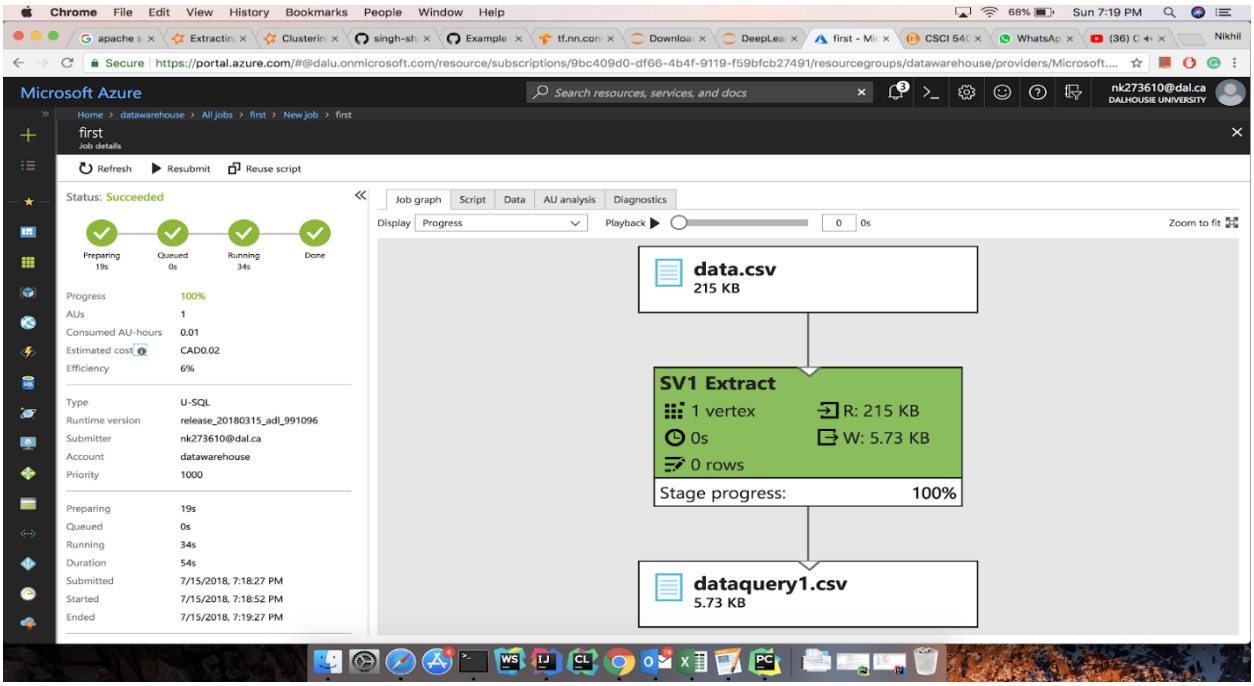


Figure 3 Query 1

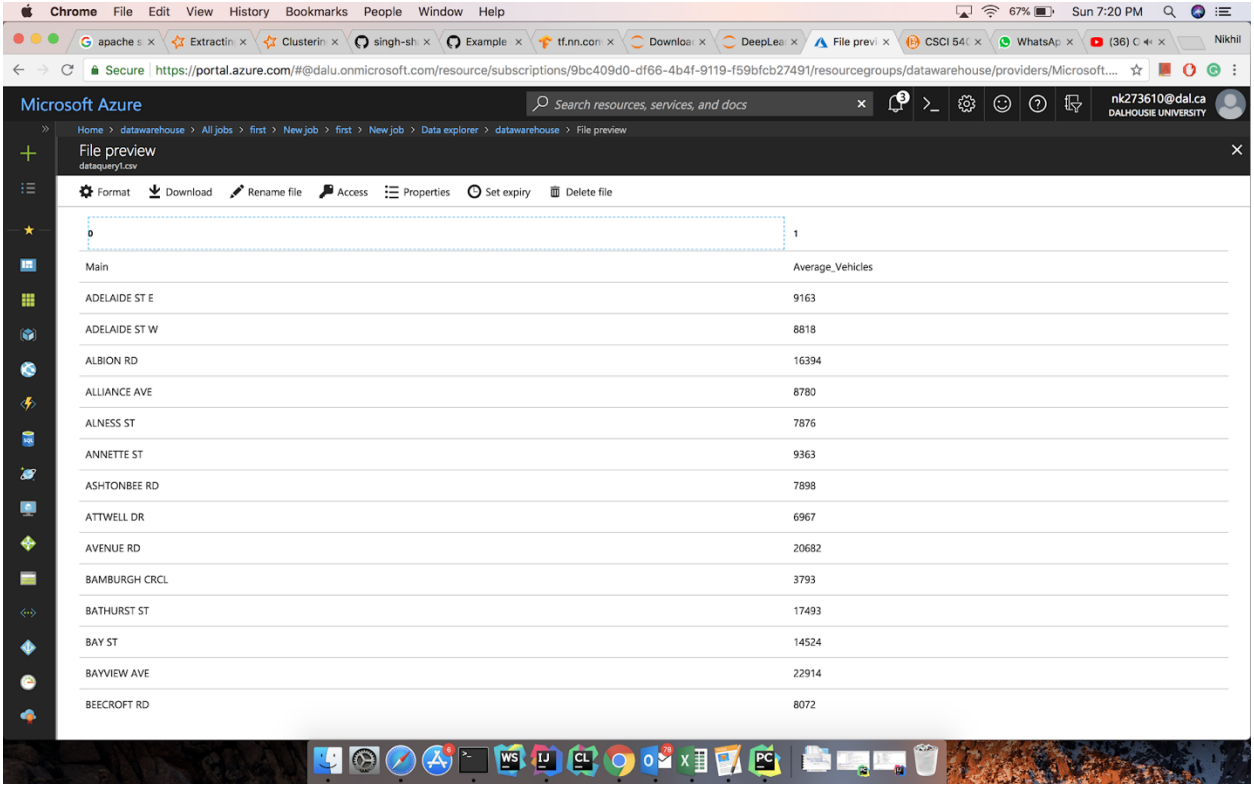


Figure 4 Output for the query

Query 2: Based on past 5 years of data, identify which 10 traffic locations are busiest during peak hours (consider both vehicle traffic and pedestrian traffic).

```
@data =
  EXTRACT
        Tcs int,
        Main string,
        Side_1_Route string,
        Activation_Date DateTime,
        Latitude double,
        Longitude double,
        Count_Date DateTime,
        eight_Peak_Hr_Vehicle_Volume int,
        eight_Peak_Hr_Pedestrian_Volume int
  FROM "/data/data.csv"
  USING Extractors.Csv(encoding: Encoding.UTF8, skipFirstNRows: 1);

@rs1=
SELECT Main, SUM(eight_Peak_Hr_Vehicle_Volume + eight_Peak_Hr_Pedestrian_Volume) AS Total_Traffic
FROM @data WHERE Count_Date.Year >= 2013 && Count_Date.Year <= 2018 GROUP BY Main ORDER BY
Total_Traffic DESC FETCH 10 ROWS;

OUTPUT @rs1
TO "/output/dataquery2.csv"
```

```
USING Outputters.Csv(outputHeader: true);
```

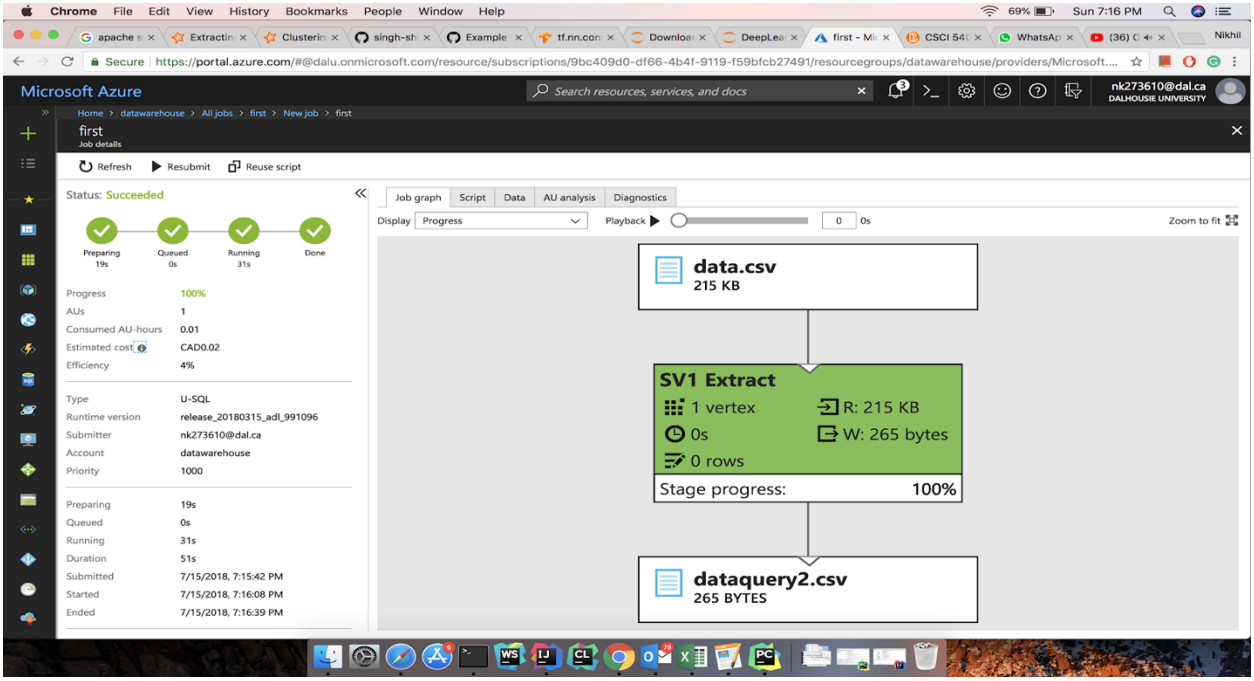


Figure 5 Query 2

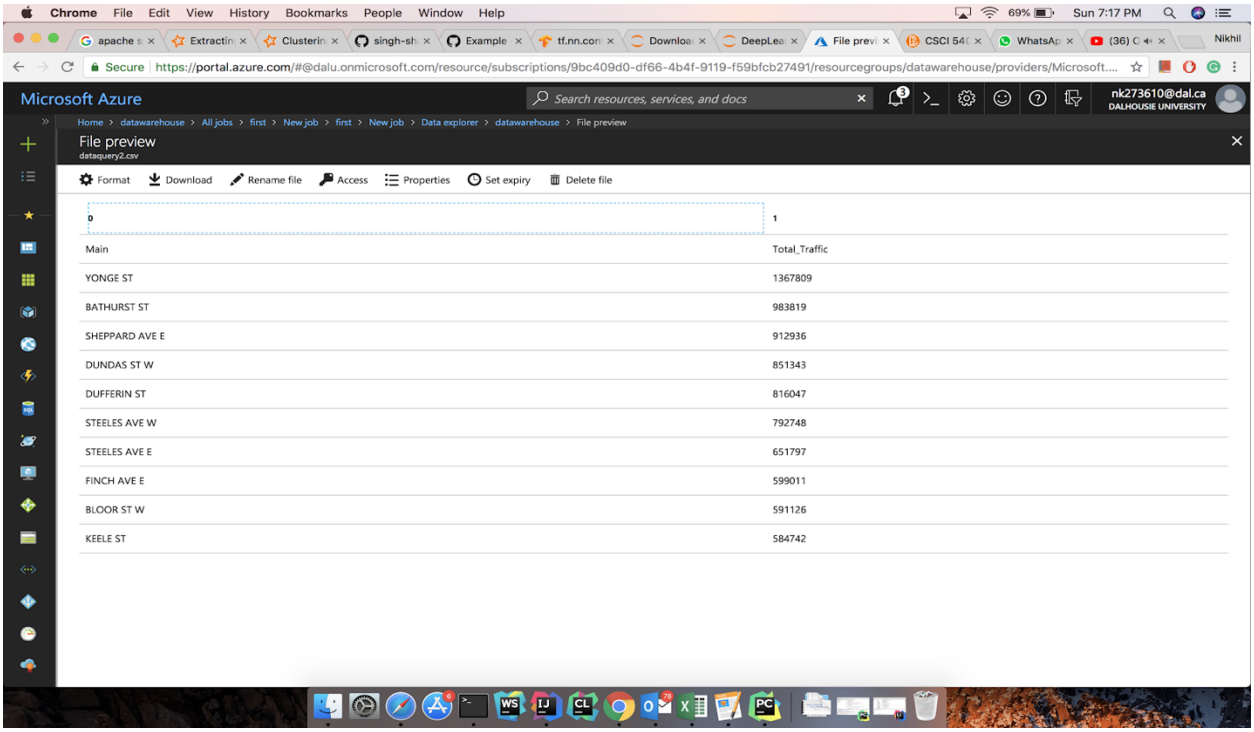


Figure 6 Output from Query 2

Query 3: Aggregate results based on individual year (2017,2016,2015, etc.)and calculate sum of vehicles and pedestrians traffic count for all available 2locations. You need to do sum on all available locations and group them based on individual years.

```
@data =
   EXTRACT
         Tcs int,
         Main string,
         Side_1_Route string,
         Activation_Date DateTime,
         Latitude double,
         Longitude double,
         Count_Date DateTime,
         eight_Peak_Hr_Vehicle_Volume int,
         eight_Peak_Hr_Pedestrian_Volume int
   FROM "/data/data.csv"
   USING Extractors.Csv(encoding: Encoding.UTF8, skipFirstNRows: 1);

@rs1=
SELECT Count_Date.Year, SUM(eight_Peak_Hr_Vehicle_Volume+eight_Peak_Hr_Pedestrian_Volume) AS Total_Traffic
FROM @data  GROUP BY Count_Date.Year;

OUTPUT @rs1
TO "/output/dataquery3.csv"
USING Outputters.Csv(outputHeader: true);
```

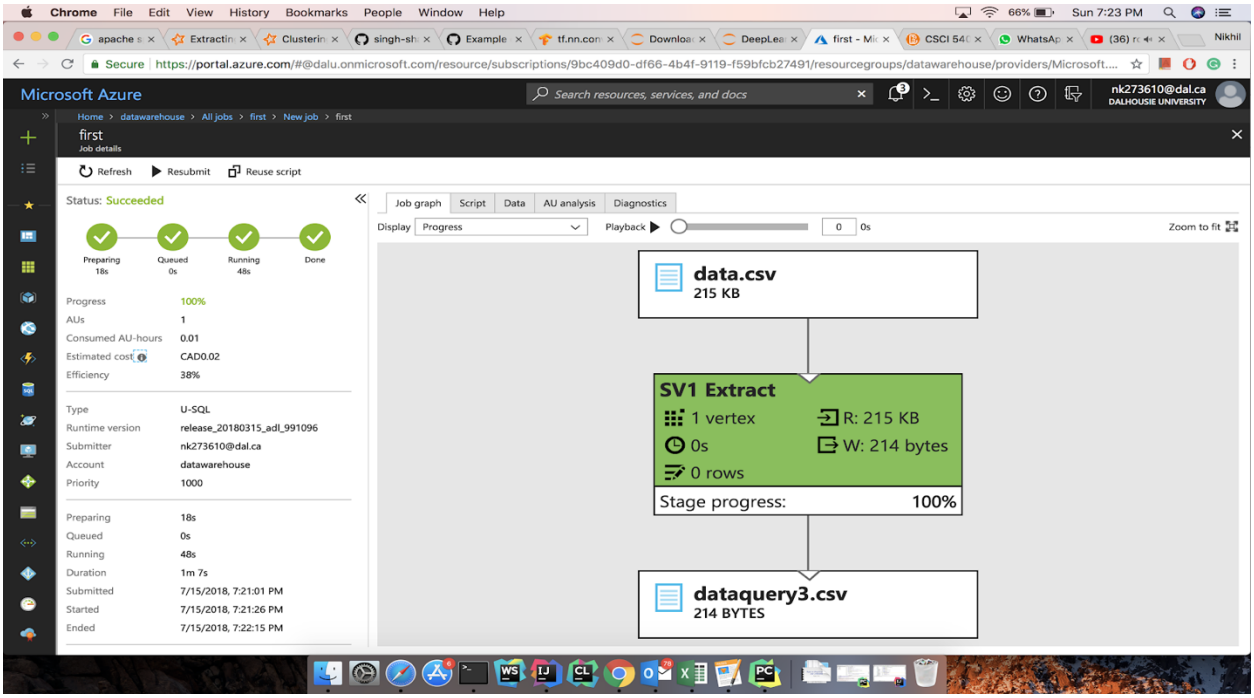
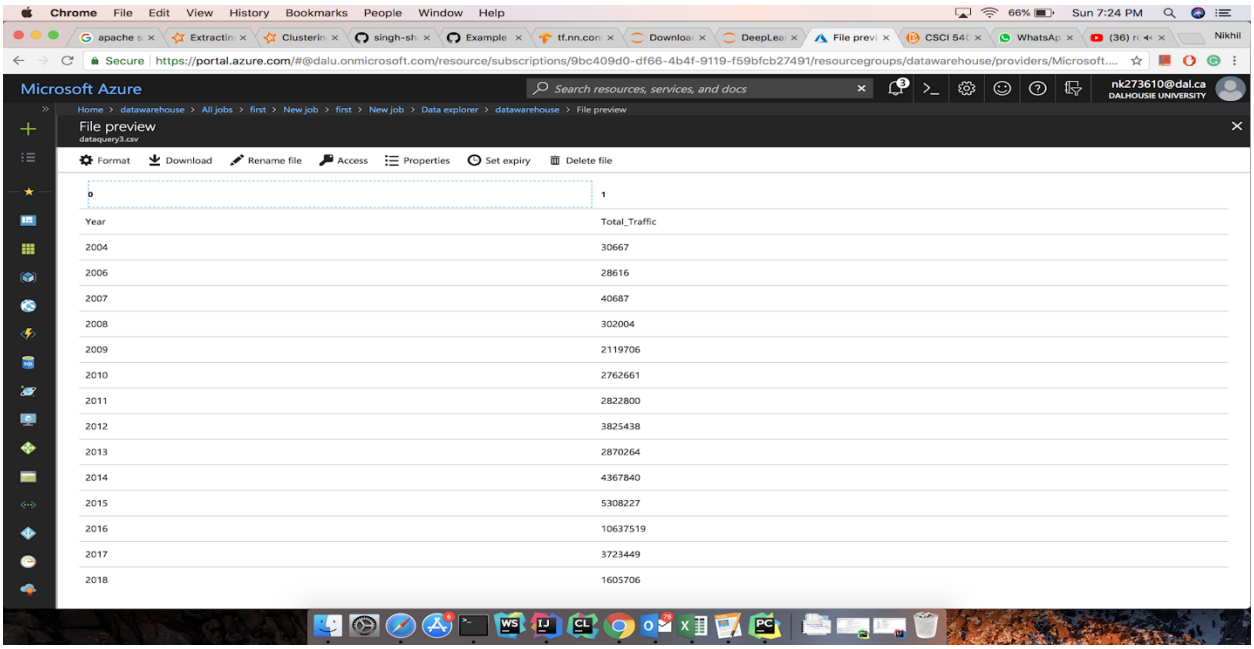


Figure 7 Query 3



0	1
Year	Total_Traffic
2004	30667
2006	28616
2007	40687
2008	302004
2009	2119706
2010	2762661
2011	2822800
2012	3825438
2013	2870264
2014	4367840
2015	5308227
2016	10637519
2017	3723449
2018	1605706

Figure 8 Output from the query

Query 4: Considering all historic years of data and all available locations, identify which day of the week(out of 7 days in a week) has been the busiest with vehicle and pedestrian traffic. Export sum of final counts for all 7 days of week for plotting.

```
@data =
  EXTRACT
        Tcs int,
        Main string,
        Side_1_Route string,
        Activation_Date DateTime,
        Latitude double,
        Longitude double,
        Count_Date DateTime,
        eight_Peak_Hr_Vehicle_Volume int,
        eight_Peak_Hr_Pedestrian_Volume int
  FROM "/data/data.csv"
  USING Extractors.Csv(encoding: Encoding.UTF8, skipFirstNRows: 1);

@rs1=
SELECT Count_Date.DayOfWeek.ToString("f") AS
Day,(eight_Peak_Hr_Vehicle_Volume+eight_Peak_Hr_Pedestrian_Volume) AS Total_Traffic
FROM @data;

@rs2=
SELECT Day, SUM(Total_Traffic) AS Total FROM @rs1 GROUP BY Day ORDER BY Total DESC OFFSET 0 ROWS;

OUTPUT @rs2
```

```
TO "/output/dataquery4.csv"
USING Outputters.Csv(outputHeader: true);
```

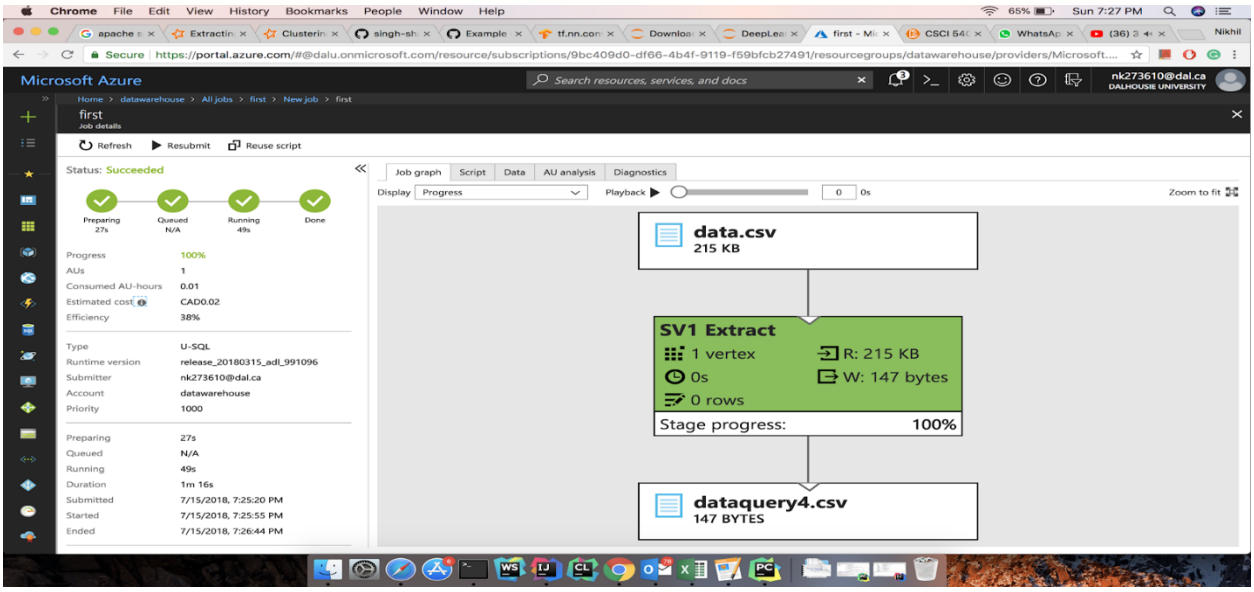


Figure 9 Query 4

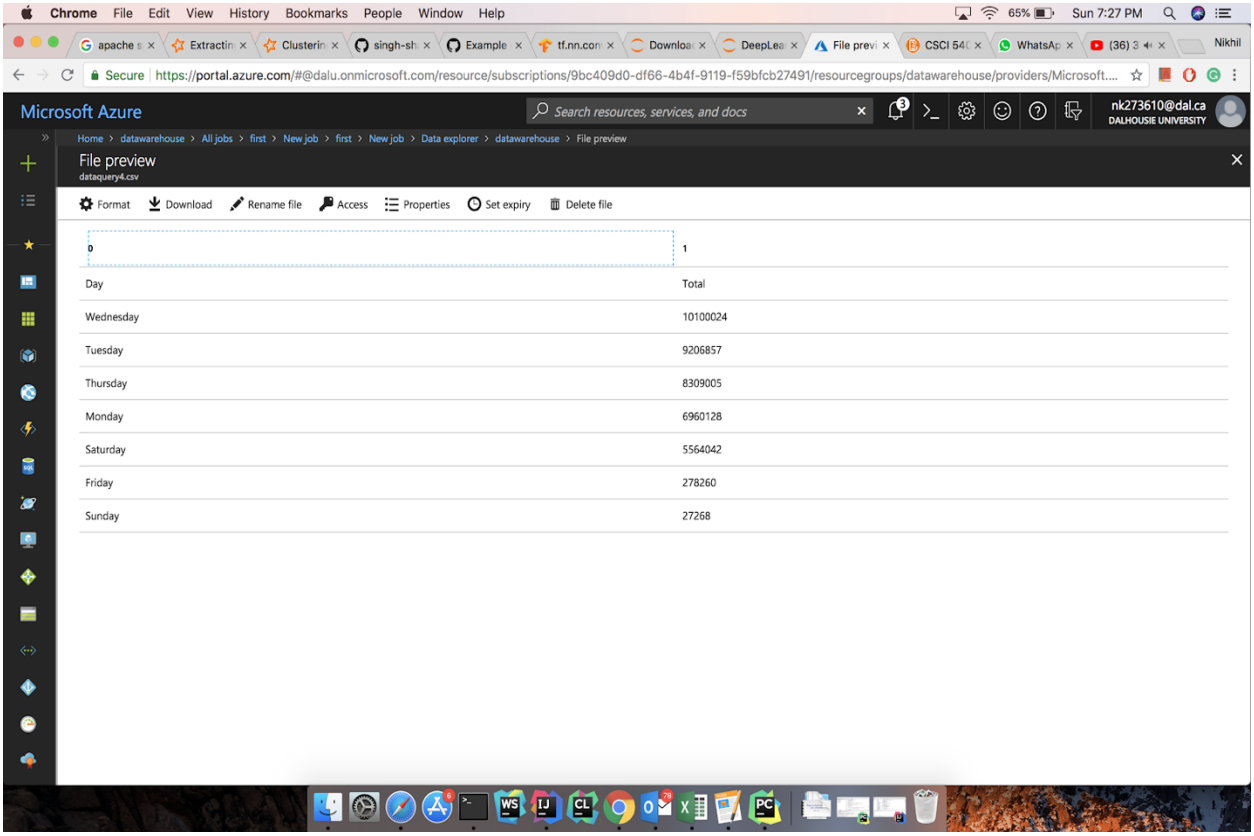


Figure 10 Output from Query 4

Query 5: Aggregate results based on "Main Street Name", identify which day of the week(out of 7 days in a week) has been the busiest withvehicle and pedestrian traffic for each individual location. Include all historic data in observation. [HINT: Group By Main Street Name, Day of Week and calculate SUM(Vehicle Traffic + Pedestrian Traffic)].

```
@data =
  EXTRACT
      Tcs int,
      Main string,
      Side_1_Route string,
      Activation_Date DateTime,
      Latitude double,
      Longitude double,
      Count_Date DateTime,
      eight_Peak_Hr_Vehicle_Volume int,
      eight_Peak_Hr_Pedestrian_Volume int
  FROM "/data/data.csv"
  USING Extractors.Csv(encoding: Encoding.UTF8, skipFirstNRows: 1);

@rs1=
SELECT Main, Count_Date.DayOfWeek.ToString("f") AS Day,
(int?)(eight_Peak_Hr_Vehicle_Volume+eight_Peak_Hr_Pedestrian_Volume) AS Total_Traffic
FROM @data;

@rs2=
SELECT Main ,MAX(Total_Traffic) AS Total FROM @rs1 GROUP BY Main;

@rs3=
SELECT s.Main, s.Total, p.Day FROM @rs2 AS s INNER JOIN @rs1 AS p ON s.Total==p.Total_Traffic AND
s.Main==p.Main;

OUTPUT @rs3
TO "/output/dataquery5.csv"
USING Outputters.Csv(outputHeader: true);
```

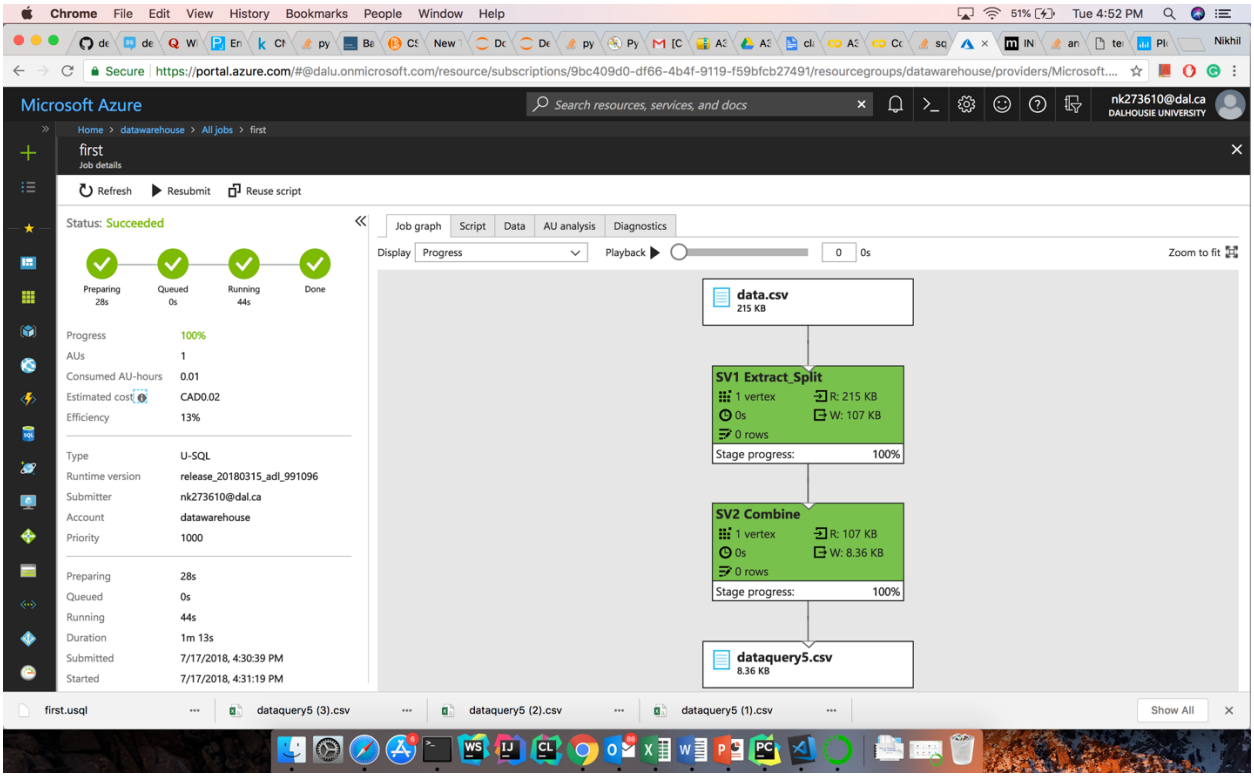
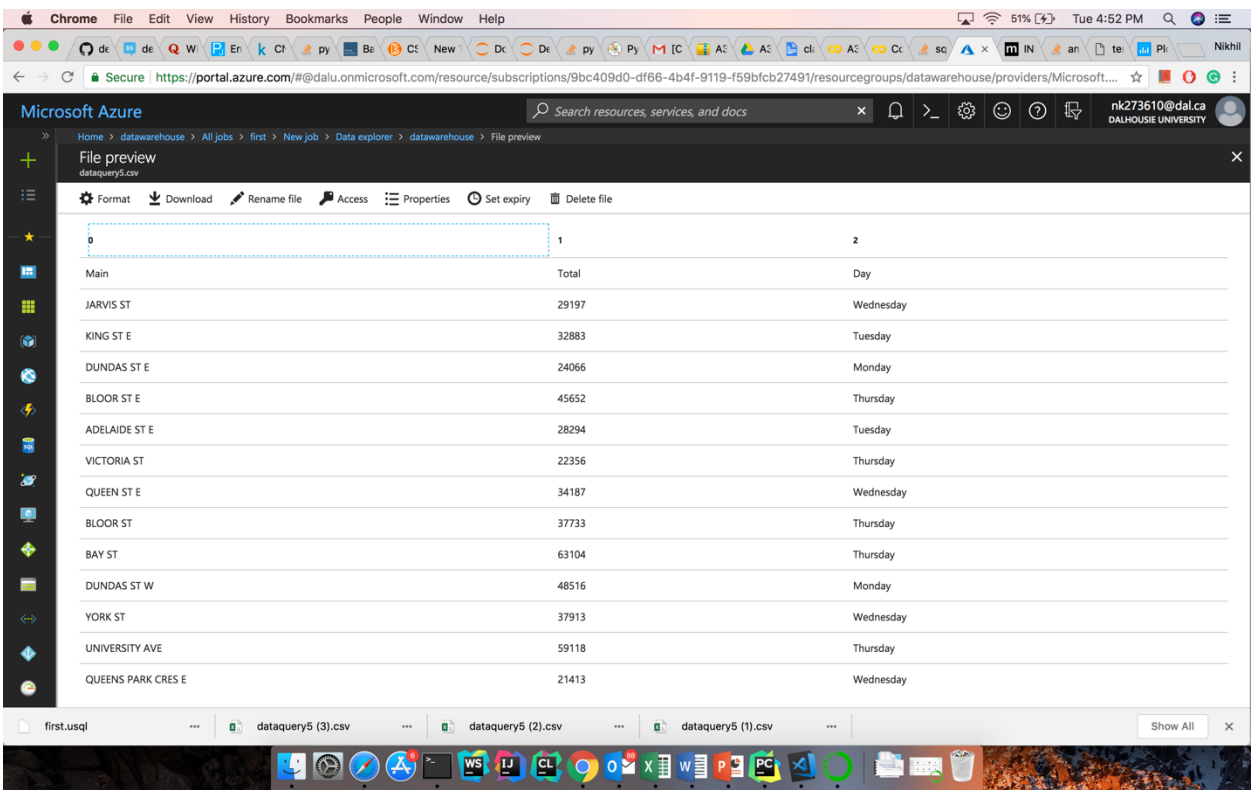


Figure 11 Query 5



0	1	2
Main	Total	Day
JARVIS ST	29197	Wednesday
KING ST E	32883	Tuesday
DUNDAS ST E	24066	Monday
BLOOR ST E	45652	Thursday
ADELAIDE ST E	28294	Tuesday
VICTORIA ST	22356	Thursday
QUEEN ST E	34187	Wednesday
BLOOR ST	37733	Thursday
BAY ST	63104	Thursday
DUNDAS ST W	48516	Monday
YORK ST	37913	Wednesday
UNIVERSITY AVE	59118	Thursday
QUEENS PARK CRES E	21413	Wednesday

Figure 12 Output from Query 5

4. DASHBOARD

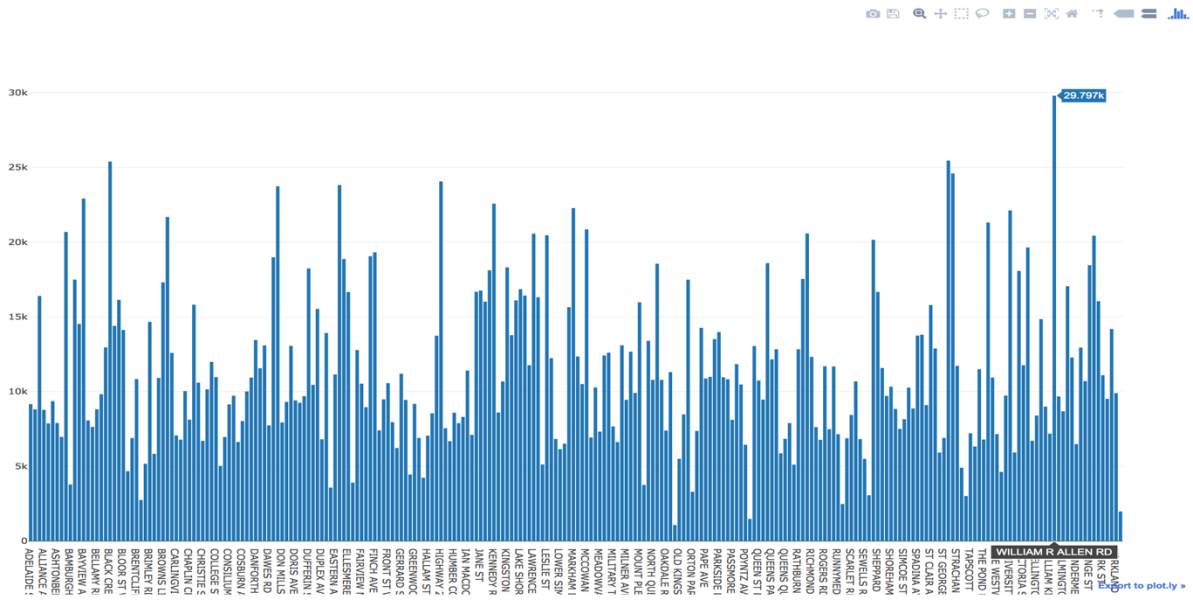


Figure 13 Bar graph plotted for query 1, reflecting that WILLAM R ALLEN RD has highest average for vehicles traffic for all the years

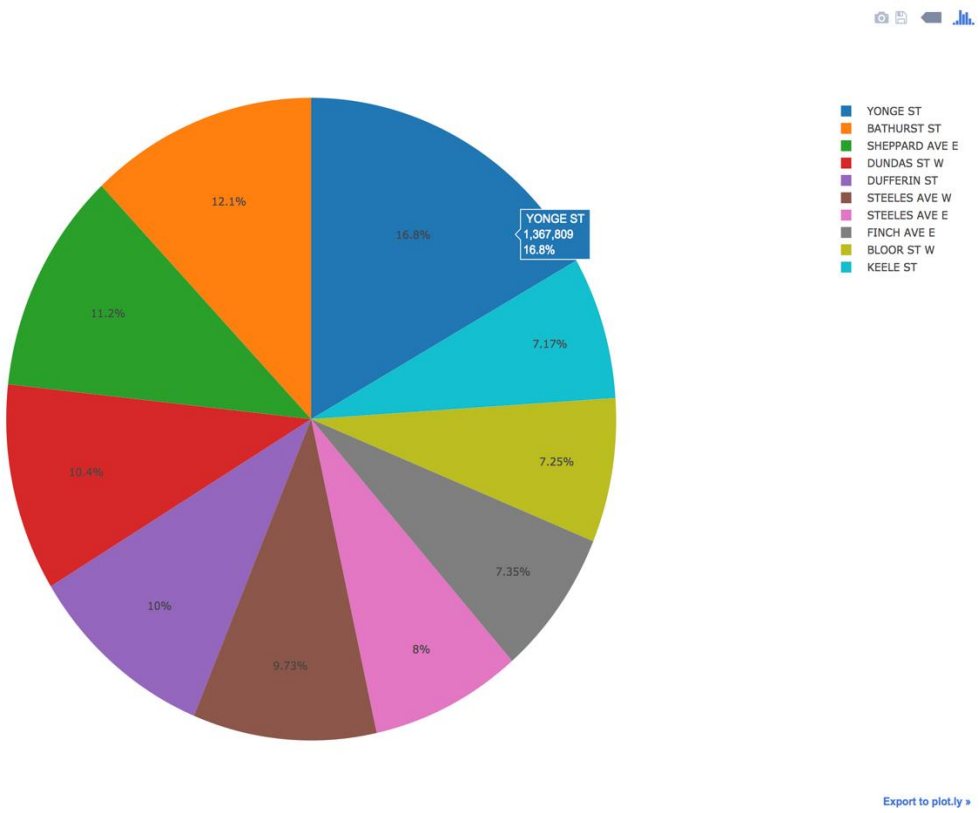


Figure 14 Pie chart plotted for query 2, YONGE ST had maximum percentage of traffic for the past 5 years

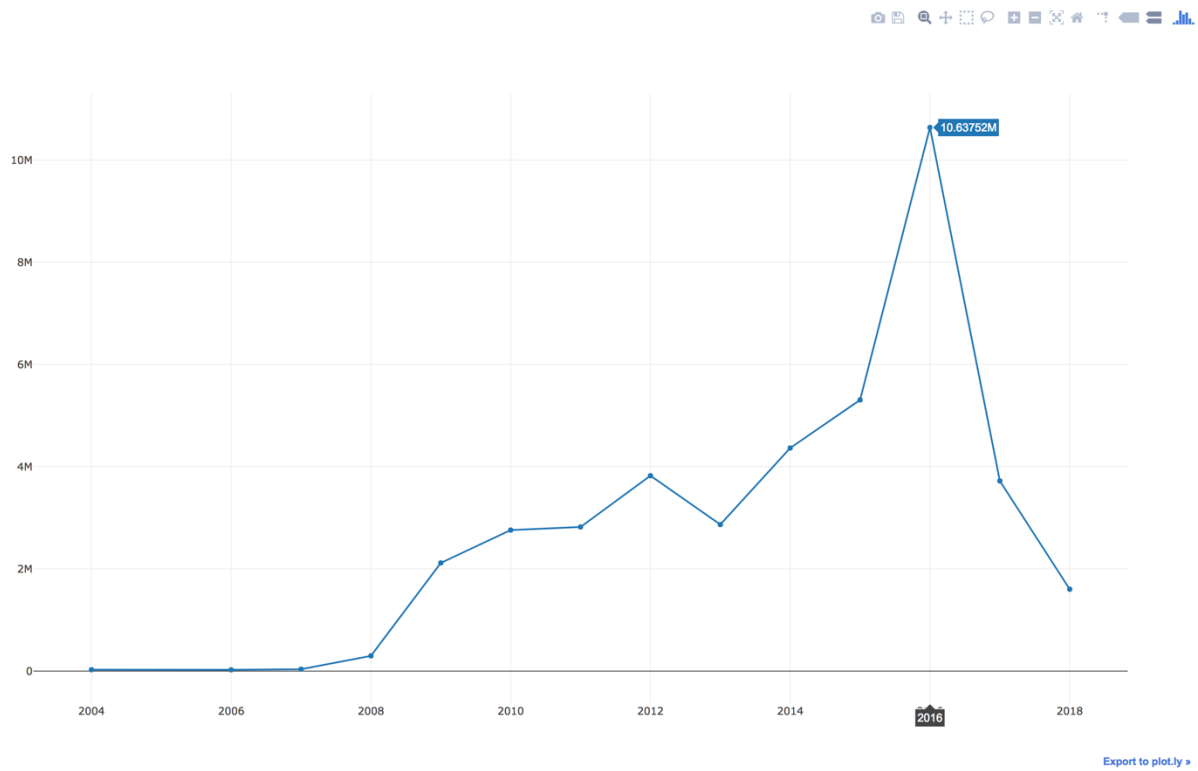


Figure 15 Line graph plotted for query 3 which shows that year 2016 has maximum traffic

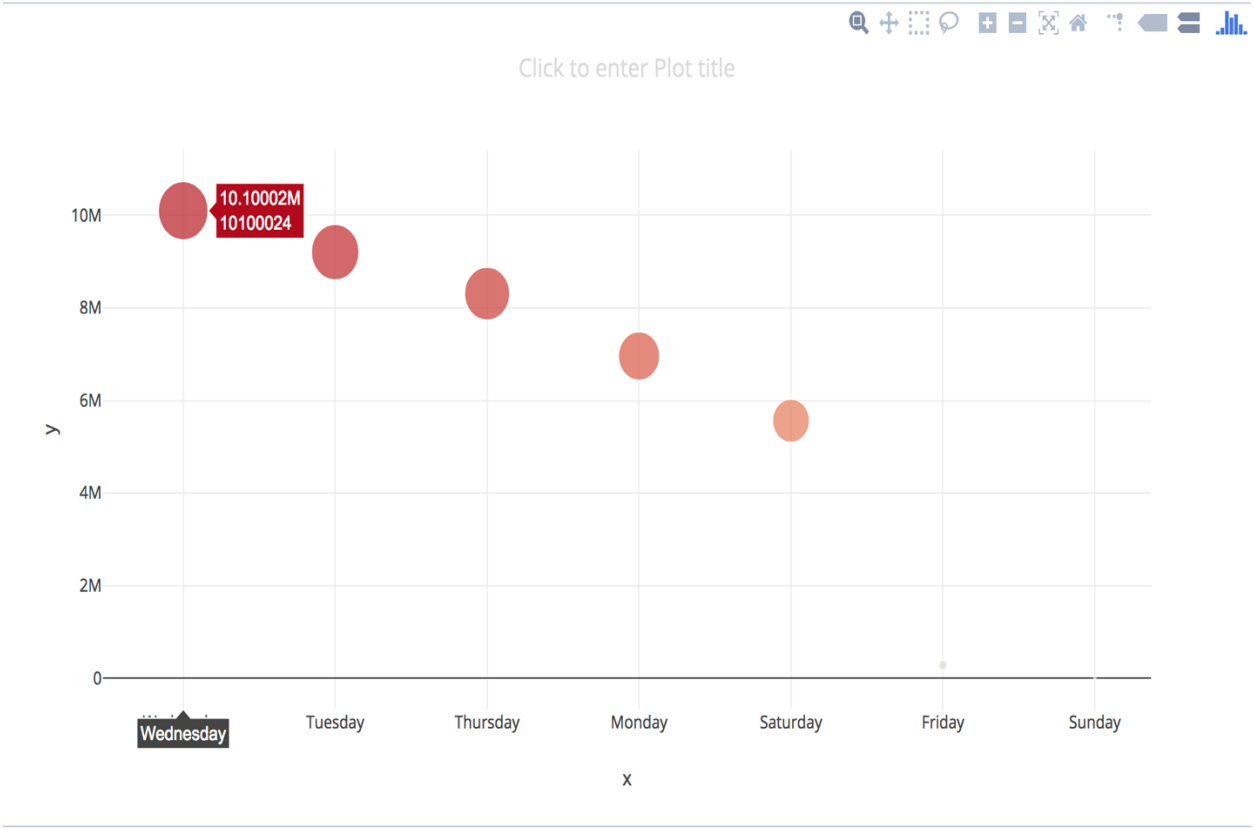


Figure 16 Surface chart plotted for the query 4 which shows Wednesday has maximum average of total pedestrian and vehicle traffic for all years

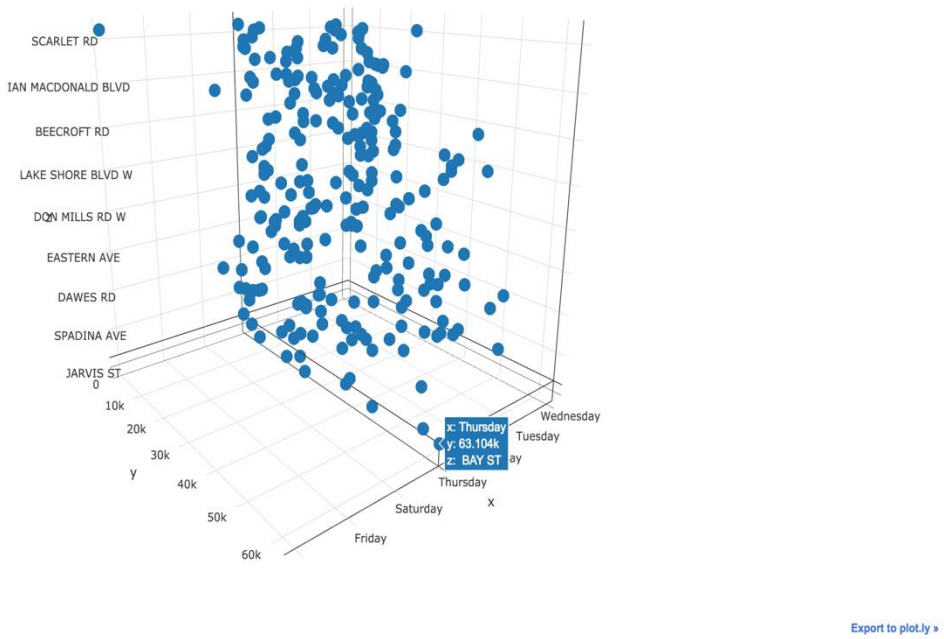


Figure 17 3D Scatter plot plotted for query 5 which reflects BAY ST having maximum traffic on Thursday

5. OUTPUT

- From the Pie chart plotted for query 2 we can see that for past 5 years (2013-2018) YONGE ST has maximum percentage of traffic (pedestrian and vehicle) i.e. 16.8% in number 1,367,809 which is maximum then we have BATHURST ST at 12.1% and so on.
- From the line graph plotted for query 3 we can see that year 2016 has maximum traffic (pedestrian and vehicle) i.e. around 10.6 Million. We can also see that the graph of total traffic vs Year increases till 2016 and with significant change then decreases after 2016 till 2018.
- From the Bar graph plotted for query 1 we can clearly see that WILLAM R ALLEN RD has highest average for vehicles traffic for all the years. We can also see that OLD FINCHE AVE has only 1081 average for vehicle traffic for all years which is also lowest. We can figure out from the graph the average volume of vehicle traffic for all years for any given street.
- From the Surface chart plotted for the query 4 we can clearly see that Wednesday has maximum average of total pedestrian and vehicle traffic for all years. The

average value of total traffic for all years for Wednesday is 10.2 Million. Then we have Tuesday, Thursday, Monday, Saturday, Friday and Sunday in decreasing order.

- From the 3D Scatter plot plotted for query 5 we have Days (Mon-Sun) on x-axis, Total traffic on y-axis and Street on z-axis, we can clearly see that BAY ST has maximum traffic on Thursday i.e. 63.10K, UNIVERSITY AVE has also maximum traffic on Thursday i.e. 59.1K, ZOO RD has maximum traffic on SUNDAY i.e. 1998. We can clearly see that from the graph for SUNDAY we have only ZOO RD for the maximum traffic. We have plotted for every street that has maximum traffic on a day for all years.

6. CODE SUBMISSION

We performed all the steps given in tutorial in order to connect our Azure Data lake through python. We have downloaded the required packages mentioned in tutorial.

CONNECT PYTHON CLIENT TO DATA LAKE

- There are a couple of steps prior to making a Python code.
- First, we need to setup the access and permissions on Azure Portal (can be done through PowerShell as well).

REQUIRED PACKAGES

- Install required packages:

```
pip install azure-mgmt-resource
pip install azure-mgmt-datalake-store
pip install azure-datalake-store
```

We successfully created the client id, access key and connected to the azure Data Lake. We have written a python script which reads our response files for all the queries from output folder and converts it to pandas Dataframe and we have used Plotly for plotting the data. In order to run the script please run our python file code.py in Python Code folder and you will be able to see all the visualizations that we have created.

Below is the link for the GitHub repository and all the outputs and files are submitted here.

https://github.com/NK273610/DataWarehouseAssignment3/tree/master/DataWarehousingAssignment5

7. REFERENCES

OVERVIEW OF AZURE DATA LAKE ANALYTICS

In-text: ("Overview of Azure Data Lake Analytics", 2018)

Your Bibliography: Overview of Azure Data Lake Analytics. (2018). Retrieved from https://docs.microsoft.com/en-us/azure/data-lake-analytics/data-lake-analytics-overview

DATA LAKE | MICROSOFT AZURE

In-text: ("Data Lake | Microsoft Azure", 2018)

Your Bibliography: Data Lake | Microsoft Azure. (2018). Retrieved from https://azure.microsoft.com/en-ca/solutions/data-lake/