SEP 720 – Cloud Computing : Assignment 2

Qwiklabs- 1

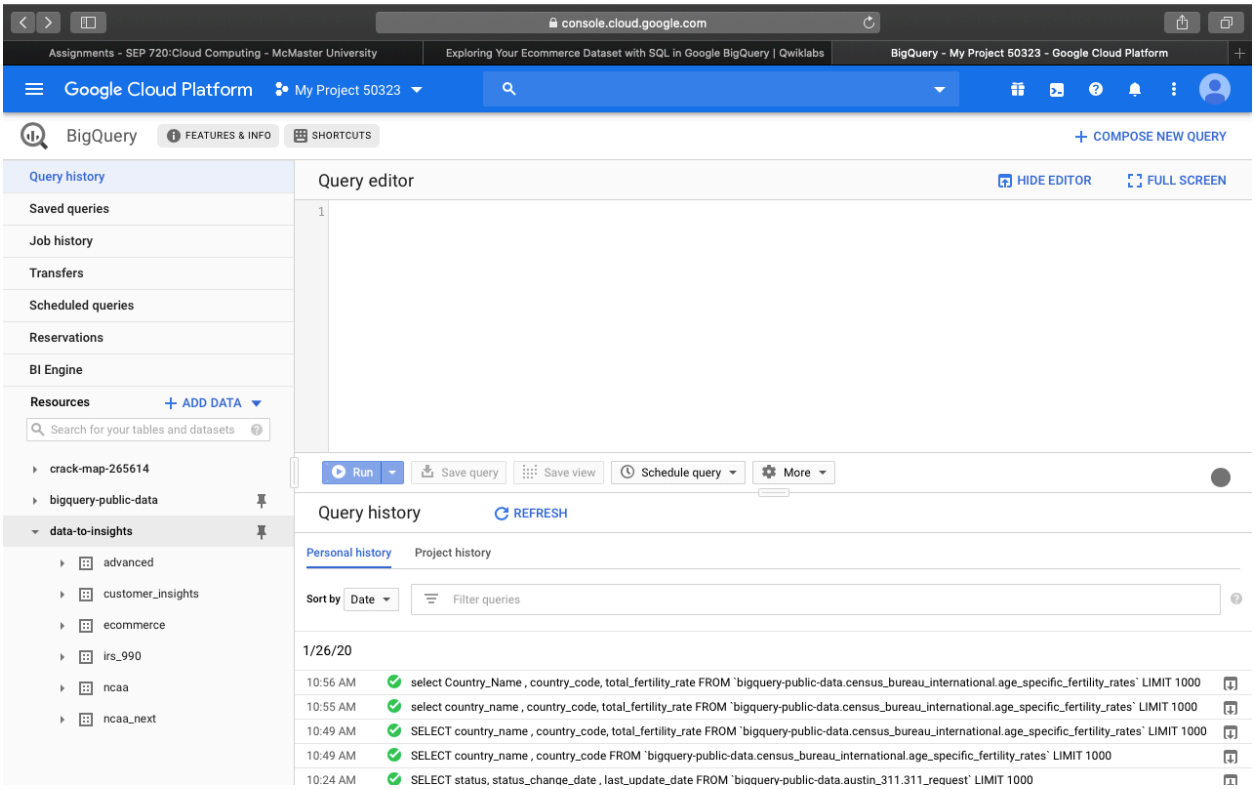
Submitted by,

Greeshma Gopal(gopalg)

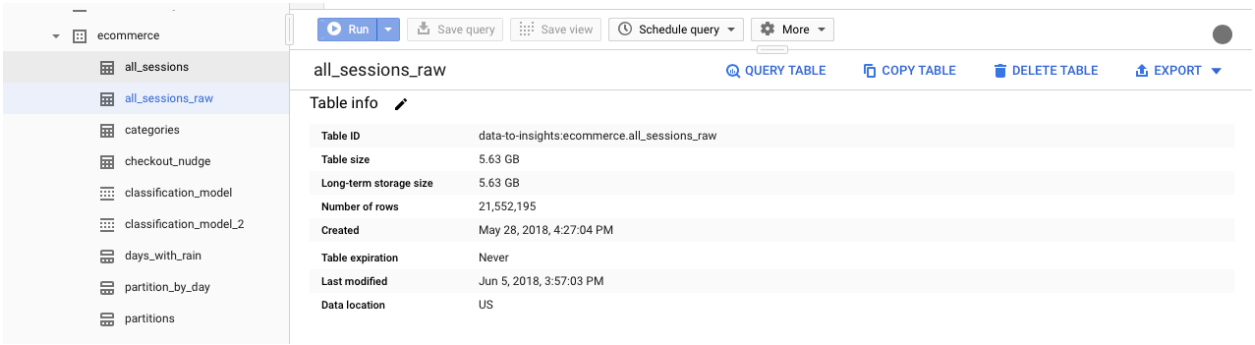
ID- 400245291

Lab 1: Exploring Your Ecommerce Dataset with SQL in Google BigQuery

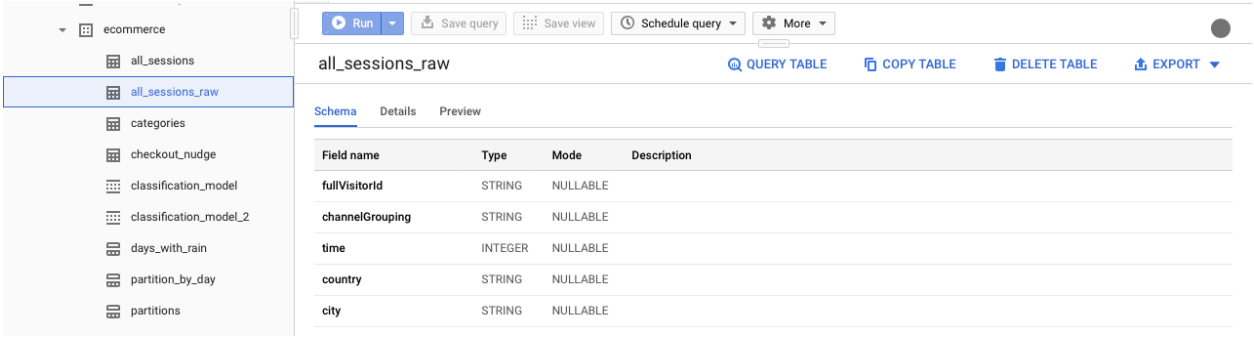
- The primary objective of this lab is to explore a dataset which is already existing using BigQuery. As a first step, the data was loaded, and it is related e-commerce.



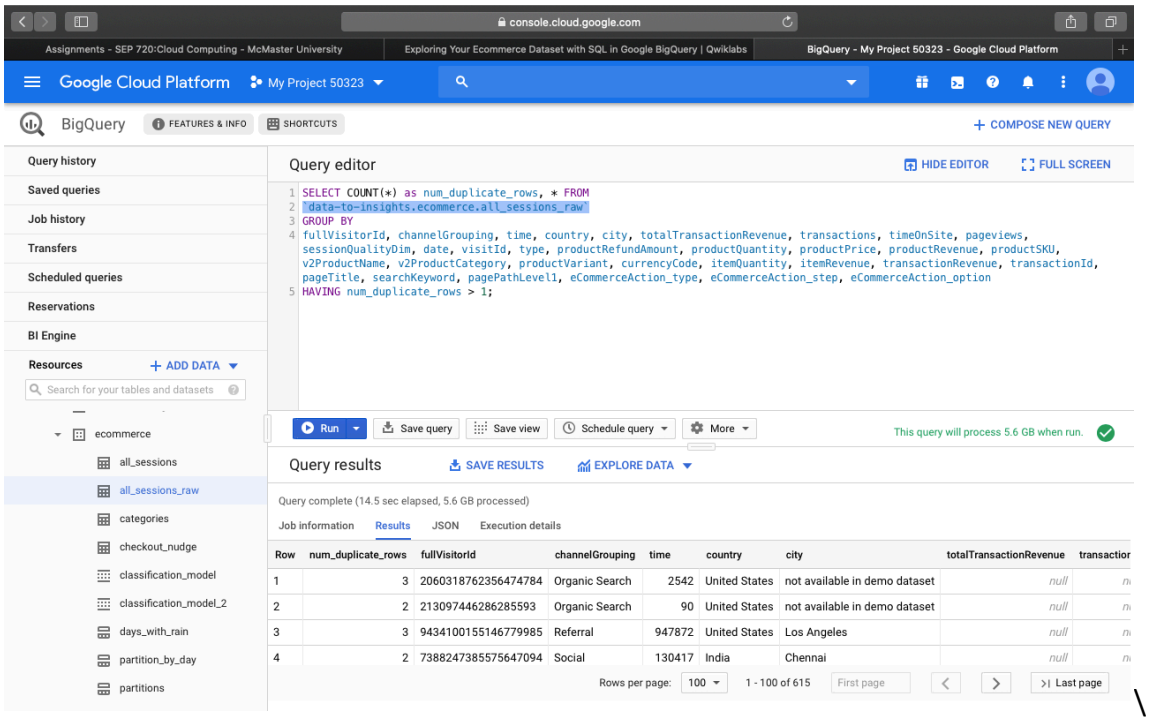
- all_sessions_raw is the dataset which was explored.



- Viewing the schema of the dataset



- Verifying the duplicate rows using the query.



Query editor

HIDE EDITOR FULL SCREEN

```
visitId, # a visitor can have multiple visits
date, # session date stored as string YYYYMMDD
time, # time of the individual site hit  (can be 0 to many per visitor session)
v2ProductName, # not unique since a product can have variants like Color
productSKU, # unique for each product
type, # a visitor can visit Pages and/or can trigger Events (even at the same time)
eCommerceAction_type, # maps to 'add to cart', 'completed checkout'
eCommerceAction_step,
eCommerceAction_option,
  transactionRevenue, # revenue of the order
  transactionId, # unique identifier for revenue bearing transaction
COUNT(*) as row_count
FROM
`data-to-insights.ecommerce.all_sessions`
GROUP BY 1,2,3 ,4, 5, 6, 7, 8, 9, 10,11,12
HAVING row_count > 1 # find duplicates
```

Run | Save query | Save view | Schedule query | More

This query will process 2.3 GB when run.

Query results

Query complete (8.7 sec elapsed, 2.3 GB processed)

Job information | Results | JSON | Execution details

⚠ This query returned no results.

- Fetching the unique visitors.

Query editor

```
#standardSQL
SELECT
  COUNT(*) AS product_views,
  COUNT(DISTINCT fullVisitorId) AS unique_visitors
FROM `data-to-insights.ecommerce.all_sessions`;
```

Run | Save query | Save view | Schedule query | More

Query results SAVE RESULTS EXPLORE DATA

Query complete (3.6 sec elapsed, 428 MB processed)

Job information | Results | JSON | Execution details

Row	product_views	unique_visitors
1	21493109	389934

- Fetching the unique visitors grouping them based on the purpose of visit of site

Query editor

```
#standardSQL
SELECT
  COUNT(DISTINCT fullVisitorId) AS unique_visitors,
  channelGrouping
FROM `data-to-insights.ecommerce.all_sessions`
GROUP BY channelGrouping
ORDER BY channelGrouping DESC;
```

Run | Save query | Save view | Schedule query

Query results — SAVE RESULTS — EXPLORE DATA

Query complete (4.1 sec elapsed, 684.6 MB processed)

Job information | Results | JSON | Execution details

Row	unique_visitors	channelGrouping
1	38101	Social
2	57308	Referral
3	11865	Paid Search
4	211993	Organic Search
5	3067	Display
6	75688	Direct
7	5966	Affiliates
8	62	(Other)

- Querying the product names of the website

Query editor

```
#standardSQL
SELECT
  (v2ProductName) AS ProductName
FROM `data-to-insights.ecommerce.all_sessions`
GROUP BY ProductName
ORDER BY ProductName
```

Run | Save query | Save view | Schedule query | More

Query results — SAVE RESULTS — EXPLORE DATA

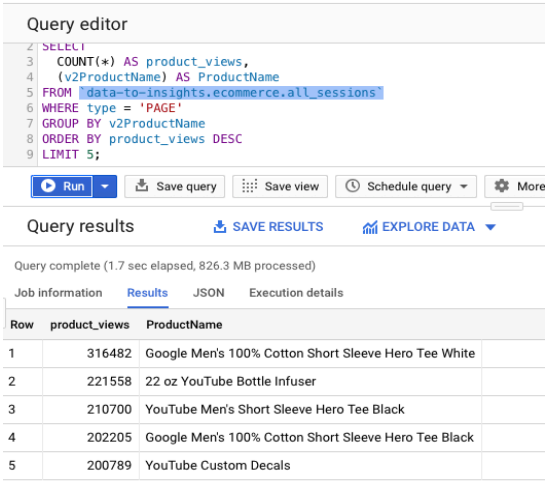
Query complete (0.8 sec elapsed, 702.6 MB processed)

Job information | Results | JSON | Execution details

Row	ProductName
1	1 oz Hand Sanitizer
2	14oz Ceramic Google Mug
3	15 oz Ceramic Mug
4	15" Android Squishable - Online
5	16 oz. Hot and Cold Tumbler
6	16 oz. Hot/Cold Tumbler
7	20 oz Stainless Steel Insulated Tumbler
8	22 oz Android Bottle

Rows per page: 100 — 1 - 100 of 633

- Fetching product views for each product



Query editor

```
SELECT
  COUNT(*) AS product_views,
  (v2ProductName) AS ProductName
FROM `data-to-insights.ecommerce.all_sessions`
WHERE type = 'PAGE'
GROUP BY v2ProductName
ORDER BY product_views DESC
LIMIT 5;
```

Run | Save query | Save view | Schedule query | More

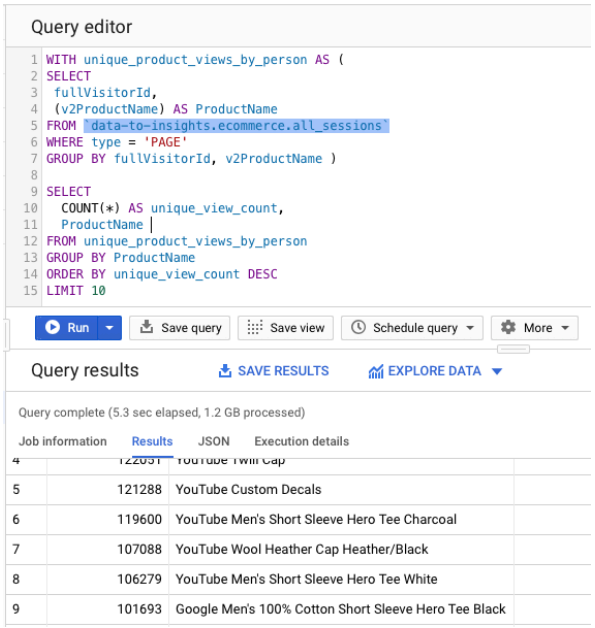
Query results | SAVE RESULTS | EXPLORE DATA

Query complete (1.7 sec elapsed, 826.3 MB processed)

Job information | Results | JSON | Execution details

Row	product_views	ProductName
1	316482	Google Men's 100% Cotton Short Sleeve Hero Tee White
2	221558	22 oz YouTube Bottle Infuser
3	210700	YouTube Men's Short Sleeve Hero Tee Black
4	202205	Google Men's 100% Cotton Short Sleeve Hero Tee Black
5	200789	YouTube Custom Decals

- Fetching unique views of every product



Query editor

```
WITH unique_product_views_by_person AS (
SELECT
 fullVisitorId,
 (v2ProductName) AS ProductName
FROM `data-to-insights.ecommerce.all_sessions`
WHERE type = 'PAGE'
GROUP BY fullVisitorId, v2ProductName )

SELECT
  COUNT(*) AS unique_view_count,
  ProductName
FROM unique_product_views_by_person
GROUP BY ProductName
ORDER BY unique_view_count DESC
LIMIT 10
```

Run | Save query | Save view | Schedule query | More

Query results | SAVE RESULTS | EXPLORE DATA

Query complete (5.3 sec elapsed, 1.2 GB processed)

Job information | Results | JSON | Execution details

4	122051	YouTube Twill Cap
5	121288	YouTube Custom Decals
6	119600	YouTube Men's Short Sleeve Hero Tee Charcoal
7	107088	YouTube Wool Heather Cap Heather/Black
8	106279	YouTube Men's Short Sleeve Hero Tee White
9	101693	Google Men's 100% Cotton Short Sleeve Hero Tee Black

- Querying the number of orders for every product and the quantity of the same ordered.

Query editor

```
#standardSQL
SELECT
  COUNT(*) AS product_views,
  COUNT(productQuantity) AS orders,
  SUM(productQuantity) AS quantity_product_ordered,
  v2ProductName
FROM `data-to-insights.ecommerce.all_sessions`
WHERE type = 'PAGE'
GROUP BY v2ProductName
ORDER BY product_views DESC
LIMIT 5;
```

Run | Save query | Save view | Schedule query | More

Query results | SAVE RESULTS | EXPLORE DATA

Query complete (0.9 sec elapsed, 829.1 MB processed)

Job information | Results | JSON | Execution details

1	316482	3158	6352	Google Men's 100% Cotton Short Sleeve Hero Tee White
2	221558	508	4769	22 oz YouTube Bottle Infuser
3	210700	949	1114	YouTube Men's Short Sleeve Hero Tee Black
4	202205	2713	8072	Google Men's 100% Cotton Short Sleeve Hero Tee Black
5	200789	1703	11336	YouTube Custom Decals

```
SELECT
  COUNT(*) AS product_views,
  COUNT(productQuantity) AS orders,
  SUM(productQuantity) AS quantity_product_ordered,
  SUM(productQuantity) / COUNT(productQuantity) AS avg_per_order,
  (v2ProductName) AS ProductName
FROM `data-to-insights.ecommerce.all_sessions`
WHERE type = 'PAGE'
GROUP BY v2ProductName
ORDER BY product_views DESC
LIMIT 5;
```

Run | Save query | Save view | Schedule query | More

Query results | SAVE RESULTS | EXPLORE DATA

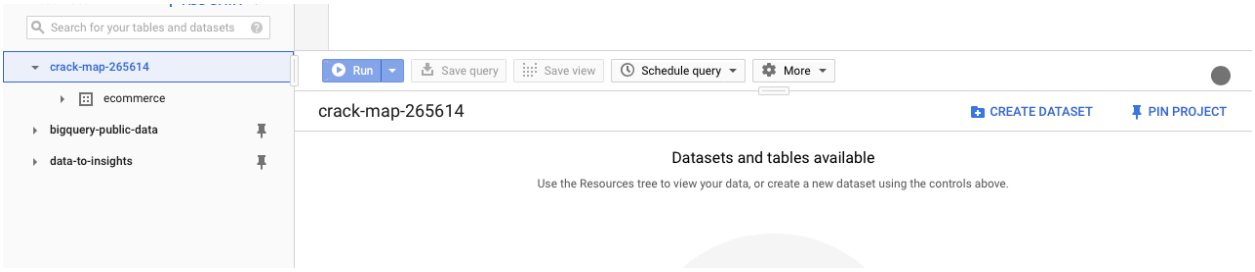
Query complete (0.9 sec elapsed, 829.1 MB processed)

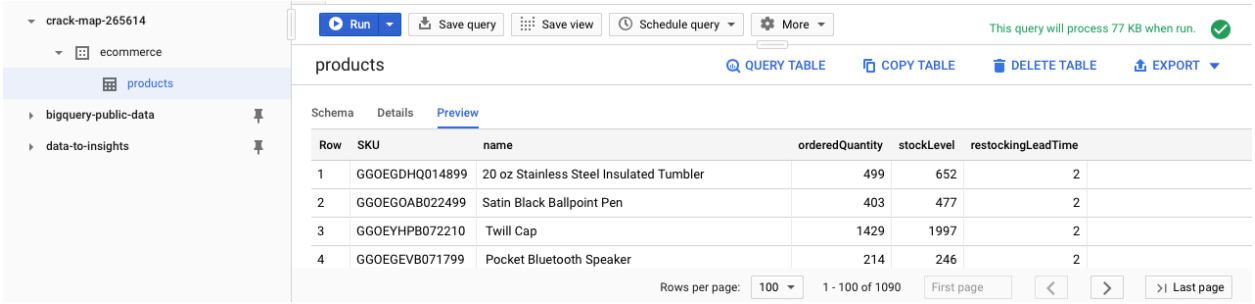
Job information | Results | JSON | Execution details

Row	product_views	orders	quantity_product_ordered	v2ProductName
1	316482	3158	6352	Google Men's 100% Cotton Short Sleeve Hero Tee White
2	221558	508	4769	22 oz YouTube Bottle Infuser
3	210700	949	1114	YouTube Men's Short Sleeve Hero Tee Black
4	202205	2713	8072	Google Men's 100% Cotton Short Sleeve Hero Tee Black
5	200789	1703	11336	YouTube Custom Decals

Lab 2: Ingesting New Datasets into BigQuery

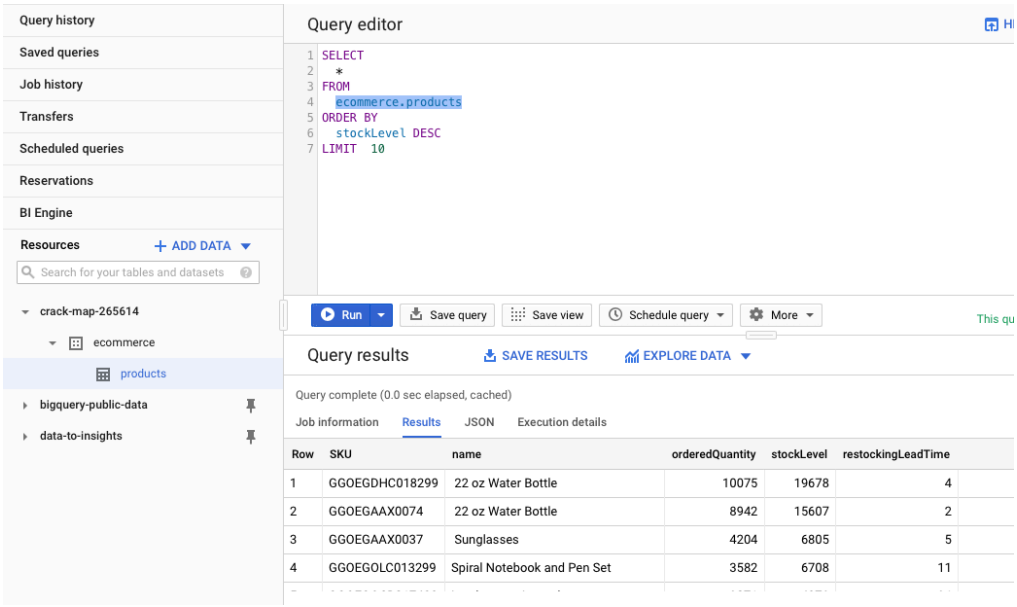
- Created a new table with the name ‘product’. I had created it under the dataset ecommerce.





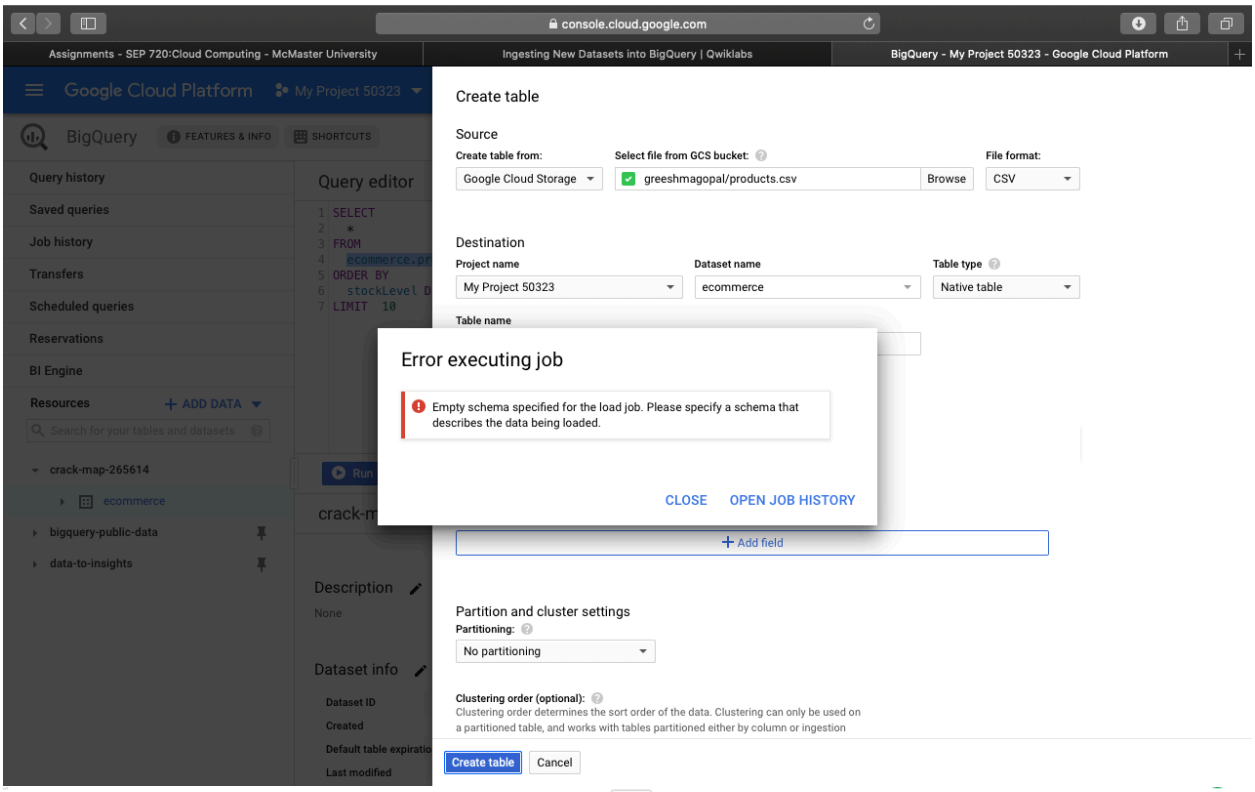
Loading data from the excel

- The data for the table has been loaded with the excel which was downloaded from qwiklabs which is product.csv.

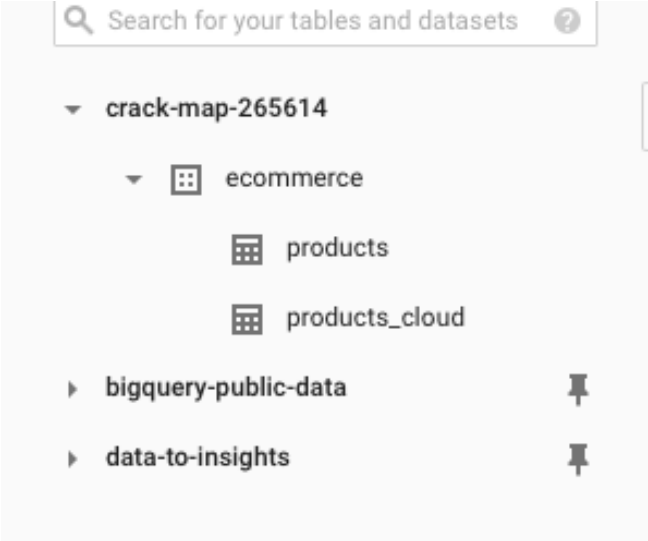


Loading the data from the cloud

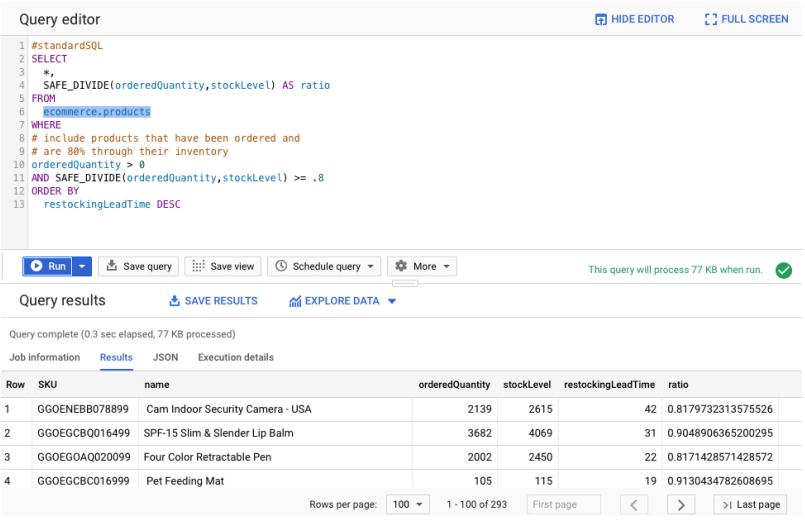
- Instead of loading the data using excel, it was loaded from cloud. Since the schema did not match it threw the error.



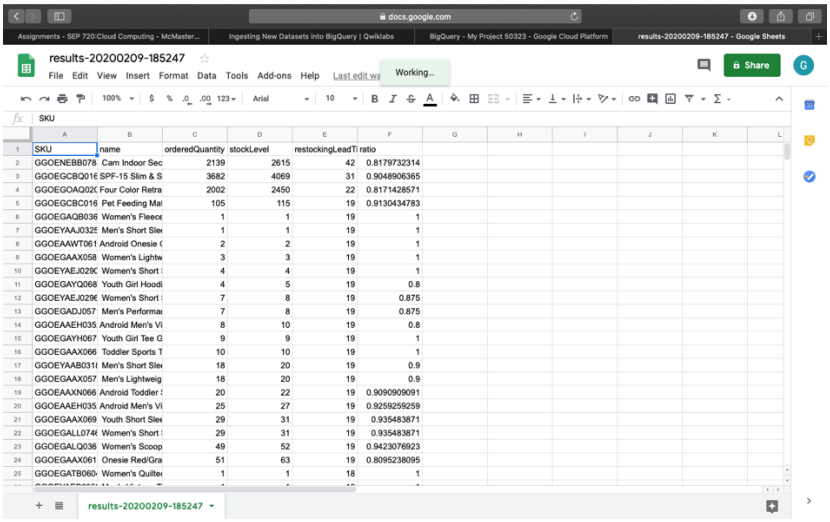
- The data was uploaded to the bucket which was created earlier, and this was loaded to the table ‘products_cloud’



- Querying the data which was loaded from the cloud.

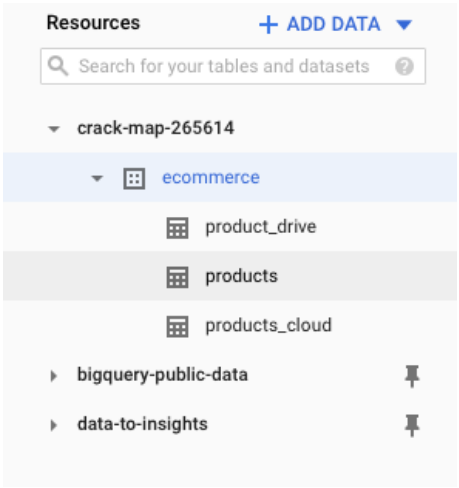


- The results of the data is saved in google sheets. A new column ‘comments’ was added.

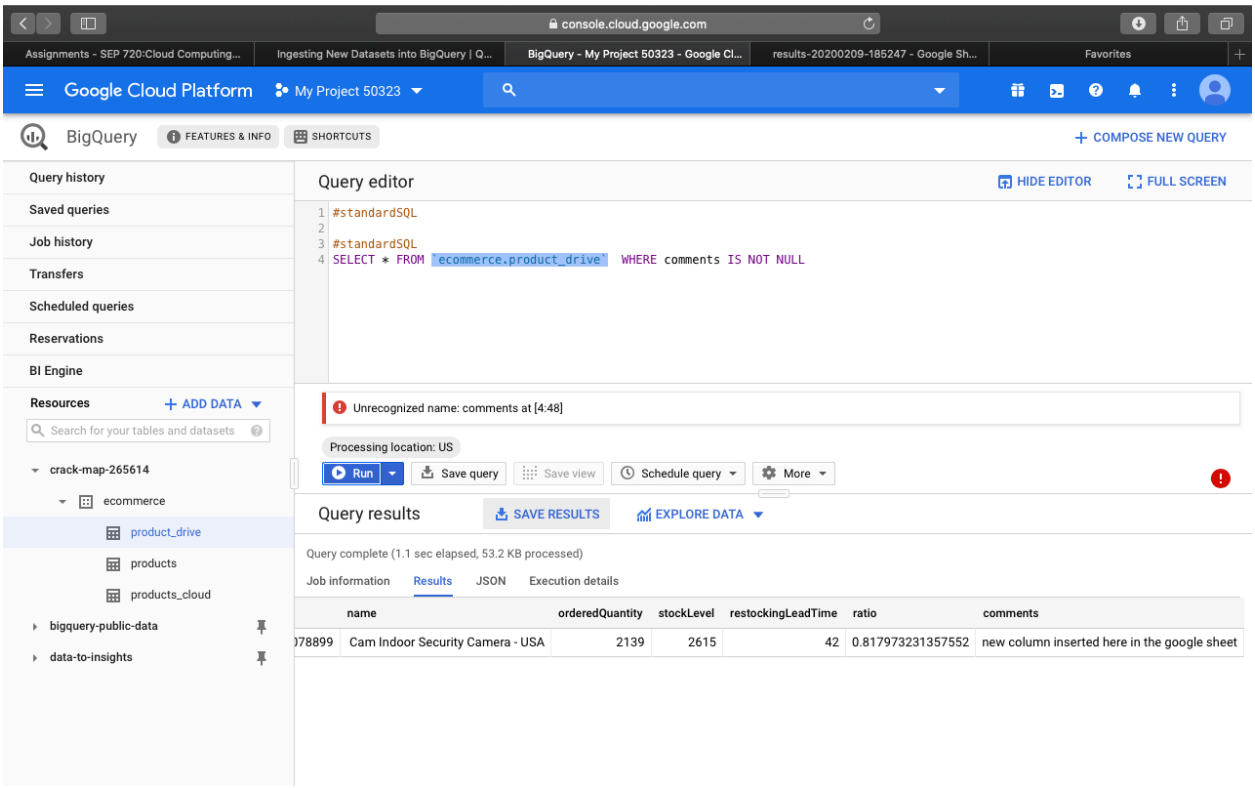


Loading the data from google drive

- A new table ‘products_drive’ was created wherein the google sheet was loaded for the data.

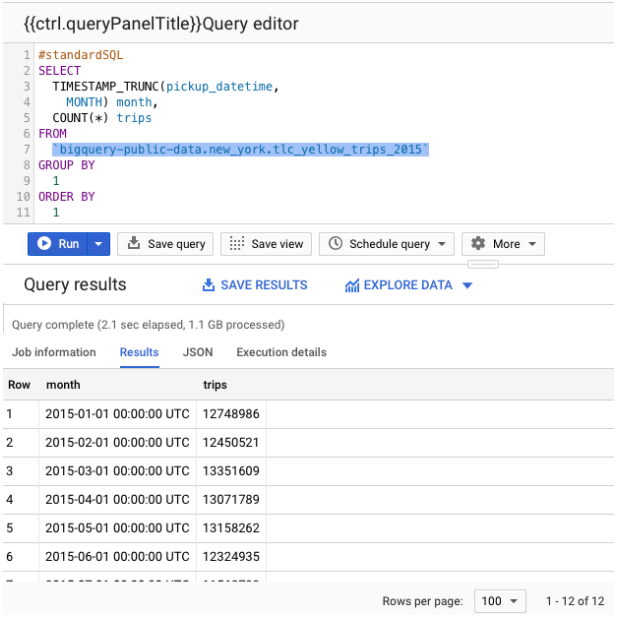


- Fetching the data with comments as not null(which was added newly in the google sheet).

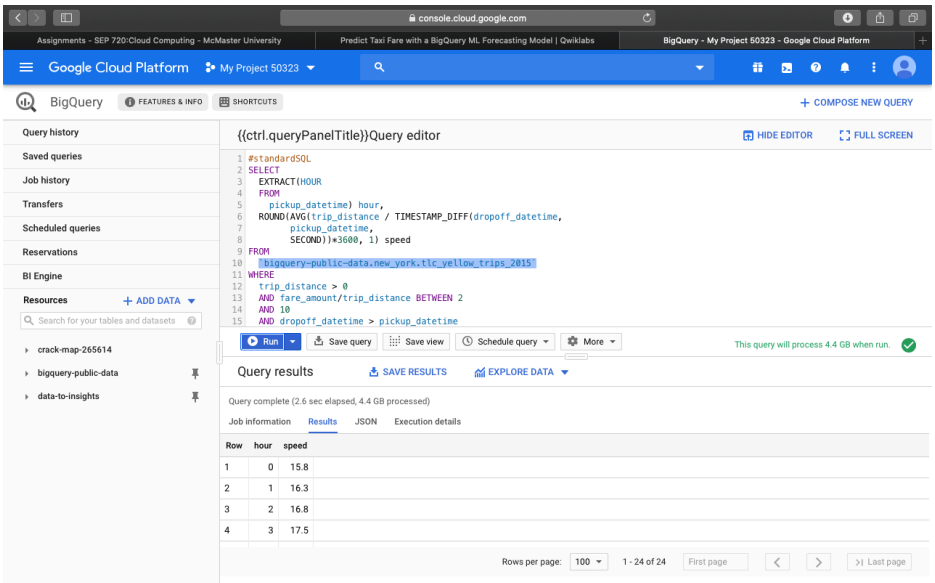


Lab 3: Predict Taxi Fare with a BigQuery ML Forecasting Model

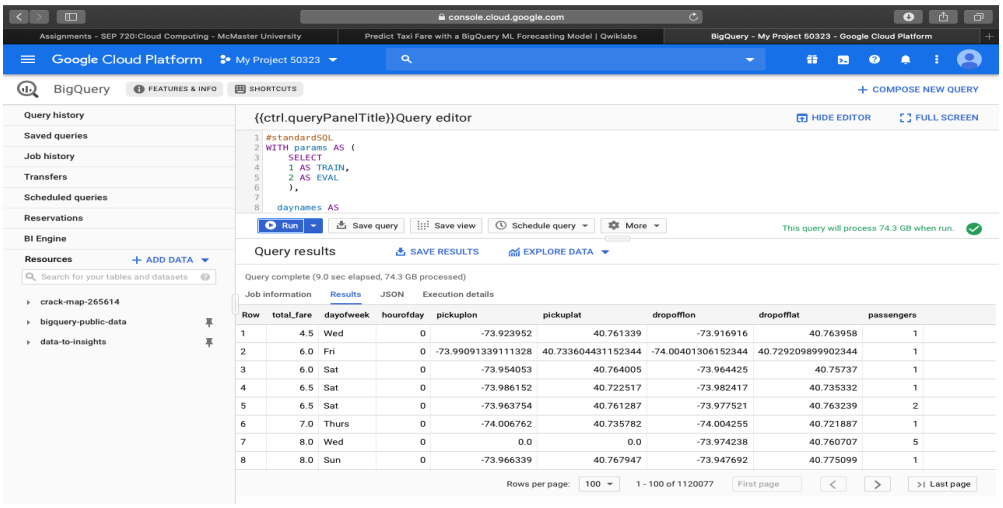
- The dataset chosen was about the taxi rides in NYC. The data was loaded and queried for finding the total number of trips every month.



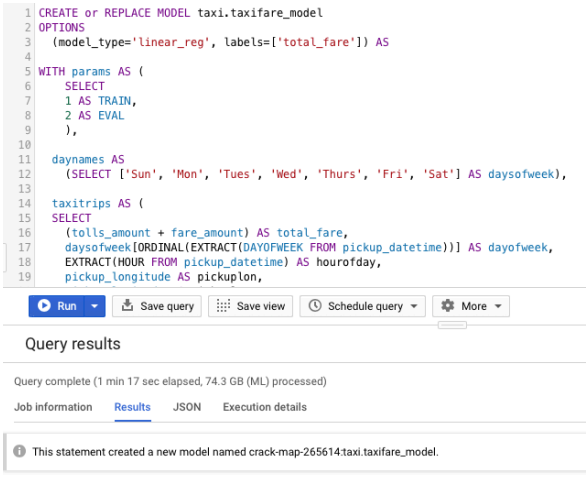
- The data was explored further, to find the speed of the taxis hourly.



- Selecting the major features to create the training dataset. The WHERE statement filters out the data which don't want to be trained on.



- Choosing the model for the dataset. Since the price varies linear with the distance and time, linear regression can be chosen as the model. The query result will be 'model created'.



- Root mean square is chosen as the loss function to validate the predicted data. The rmse is higher in this case.

```
#standardSQL
SELECT
  SQRT(mean_squared_error) AS rmse
FROM
  ML.EVALUATE(MODEL taxi.taxifare_model,
  (

  WITH params AS (
    SELECT
    1 AS TRAIN,
    2 AS EVAL
    ),

  daynames AS
    (SELECT ['Sun', 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat'] AS daysofweek),

  taxitrips AS (
  SELECT
    (tolls_amount + fare_amount) AS total_fare,
```

Run | Save query | Save view | Schedule query | More

Query results | SAVE RESULTS | EXPLORE DATA

Query complete (4.8 sec elapsed, 74.3 GB processed)

Job information | Results | JSON | Execution details

Row	rmse
1	9.476637019770271

- Since there are outliners in the dataset (fares are negative and some are more than $50,000), we need to filter out this data for better prediction.

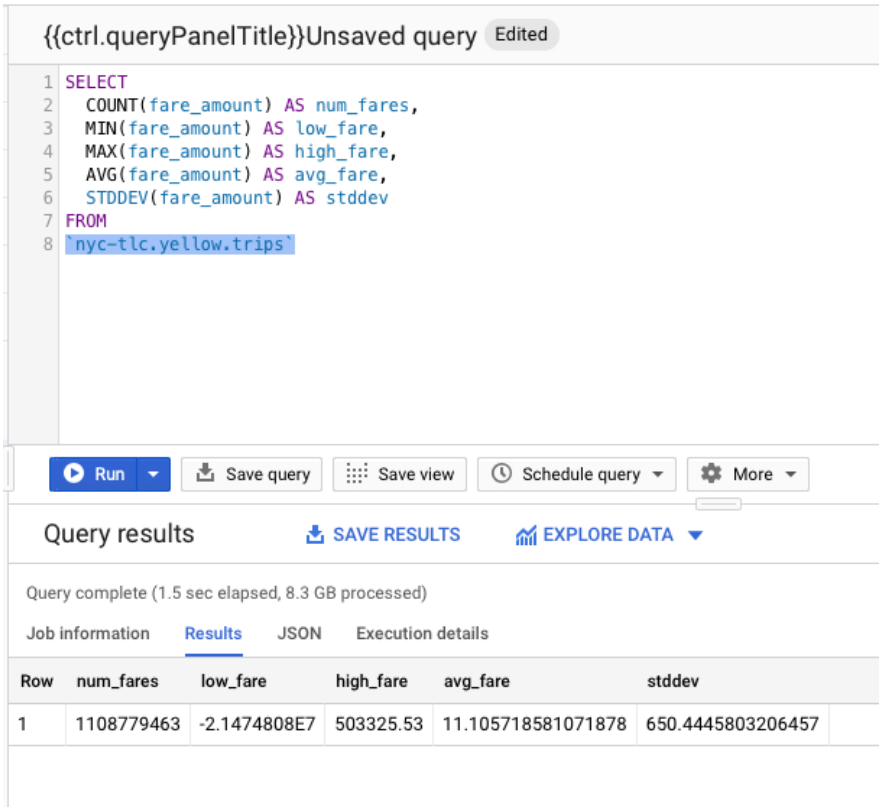
Query results | SAVE RESULTS | EXPLORE DATA

Query complete (12.0 sec elapsed, 74.3 GB processed)

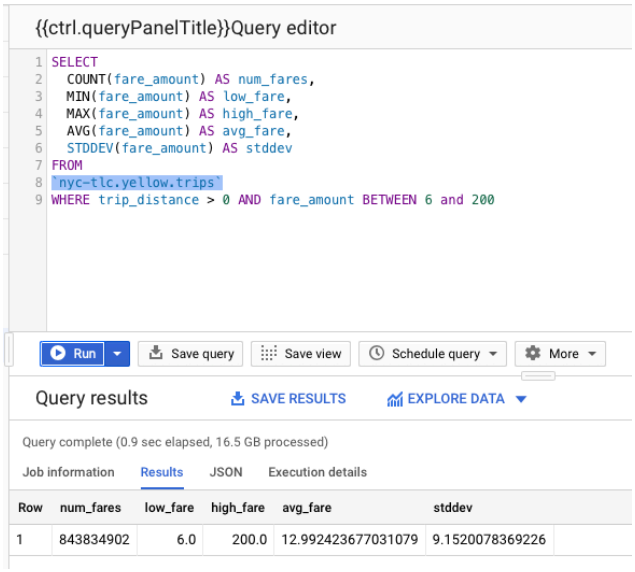
Job information | Results | JSON | Execution details

Row	predicted_total_fare	total_fare	dayofweek	hourofday	pickuplon	pickuplat	dropofflon	dropofflat	pass
1	11.502443386235797	3.5	Wed	0	-74.003658	40.74031	-73.997352	40.737686	
2	11.95925140979752	4.0	Wed	0	-73.954907	40.76972	-73.956013	40.775472	
3	11.222868131199554	4.0	Sat	0	-73.938094	40.796674	-73.944337	40.788055	
4	11.222907321676589	4.5	Sat	0	-73.985415	40.731798	-73.987391	40.72688	
5	11.396090850265768	5.0	Tues	0	-73.98422	40.749068	-73.97328	40.743545	
6	11.648081503904997	5.5	Thurs	0	-73.997277	40.721625	-73.992042	40.734562	
7	11.502480822029849	6.0	Wed	0	-74.00328063964844	40.72758483886719	-74.00617980957031	40.73969650268555	
8	11.396114108211755	6.5	Tues	0	-73.985543	40.768008	-74.005034	40.74127	
9	11.648057063763122	6.5	Thurs	0	-73.862743	40.768847	-73.876562	40.764175	
10	11.738753301431082	6.5	Tues	0	-73.99188	40.7351	-73.999481	40.743964	
11	12.219062247997142	7.5	Thurs	0	-73.9981	40.761095	-73.996765	40.757902	

Rows per page: 100 | 1 - 100 of 1111106 | First page | < | > | >| Last page

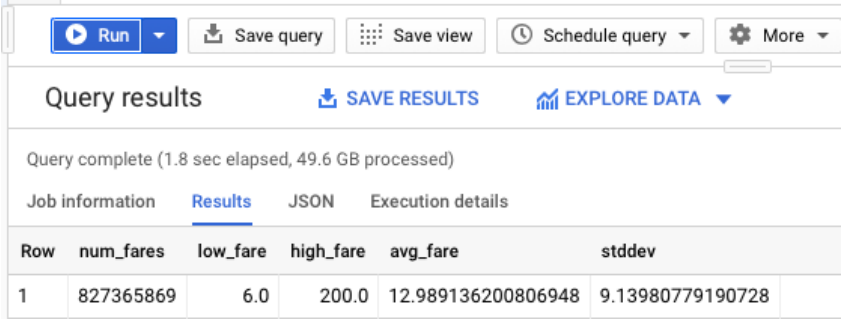


- The data has been filtered to choose the ones with fares between 6 to 200.

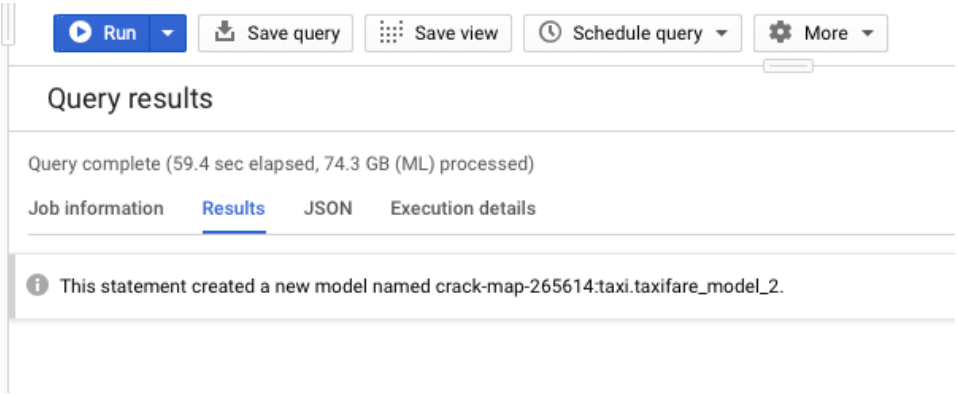


- Also we will pick the data which is specific to New York city.

```
8  `nyc-tlc.yellow.trips`
9  WHERE trip_distance > 0 AND fare_amount BETWEEN 6 and 200
10     AND pickup_longitude > -75
11     AND pickup_longitude < -73
12     AND dropoff_longitude > -75
13     AND dropoff_longitude < -73
14     AND pickup_latitude > 40
15     AND pickup_latitude < 42
16     AND dropoff_latitude > 40
```



- The different model(same linear regression) was created taxifare_model_2 for retraining.



- The root mean square was calculated again and was lesser than the first one after when the outliners were filtered out.