Lab notebook Week 9

Submitted by: Vishrut Sharma (OdinID: vishrut)

Table of Contents

09.1g: BigQuery, BigLake........ 1

3. Create dataset........ 1
4. Query data........ 2
9. Query data........ 3

09.2g: Jupyter Notebooks........ 3

3. BigQuery query........ 3
6. Run queries........ 4
8. Mobility........ 4
9. Airport traffic........ 4
10. Mortality........ 5
11. Run example queries........ 5
12. Write queries........ 7

09.3g: Dataproc........ 8

6. Run computation........ 8
8. Run computation again........ 8

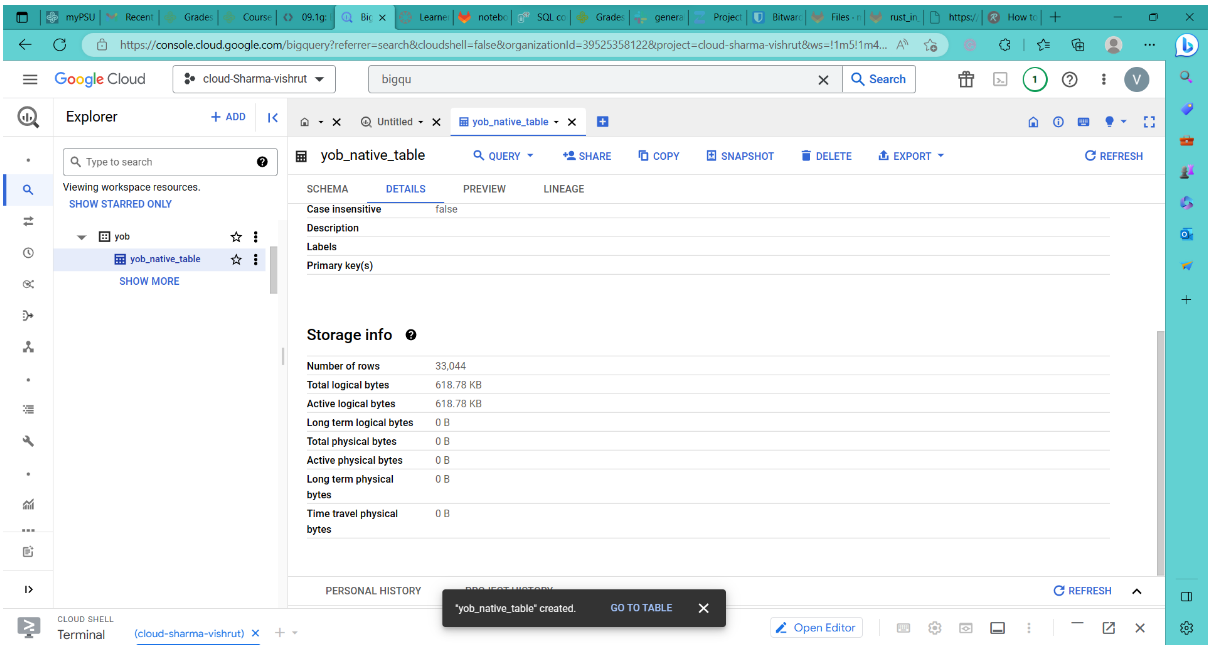
09.4g: Dataflow........ 8

3. Beam code........ 8
4. Run pipeline locally........ 9
5. Dataflow Lab #2 (Word count)........ 10
6. Run code locally........ 11
9. Run code using Dataflow runner........ 11
12. View raw data from PubSub........ 12
14. Run Dataflow job from template........ 13
15. Query data in BigQuery........ 13
16. Data visualization........ 15

09.1g: BigQuery, BigLake

3. Create dataset

- **Take a screenshot of the table's details that includes the number of rows in the table.**



yob_native_table

Case insensitive	false
Description	
Labels	
Primary key(s)	

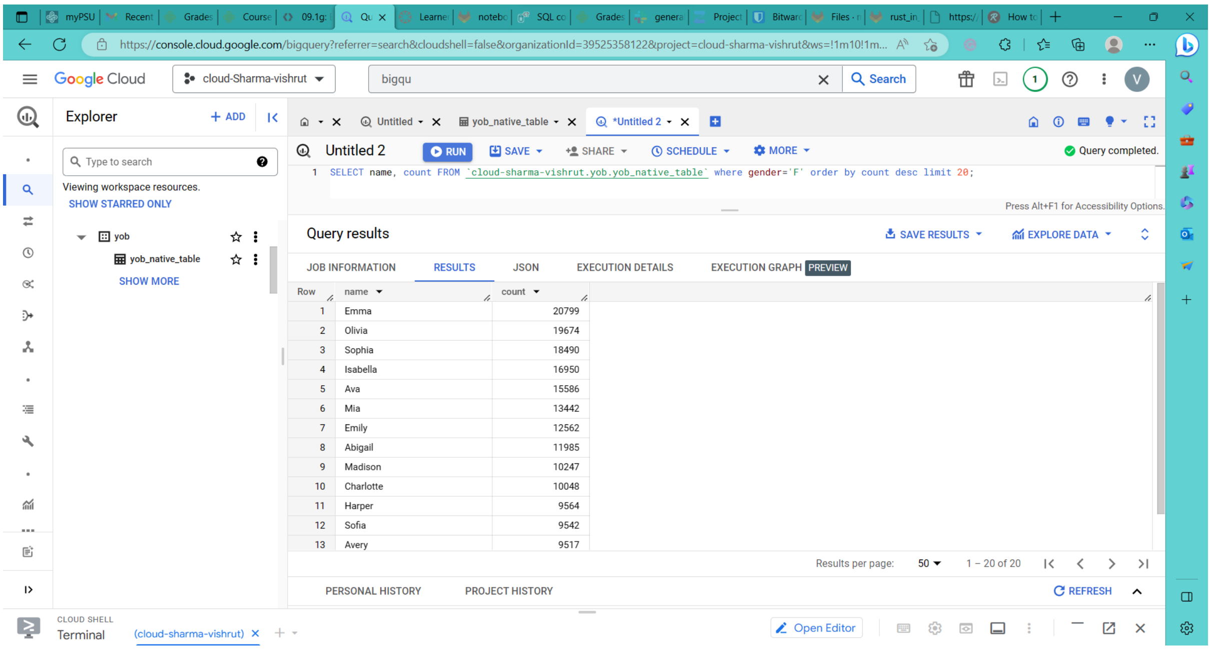
Storage info

Number of rows	33,044
Total logical bytes	618.78 KB
Active logical bytes	618.78 KB
Long term logical bytes	0 B
Total physical bytes	0 B
Active physical bytes	0 B
Long term physical bytes	0 B
Time travel physical bytes	0 B

"yob_native_table" created.

4. Query data

- **Screenshot the query results and include it in your lab notebook**



Untitled 2

```
SELECT name, count FROM `cloud-sharma-vishrut.yob.yob_native_table` where gender='F' order by count desc limit 20;
```

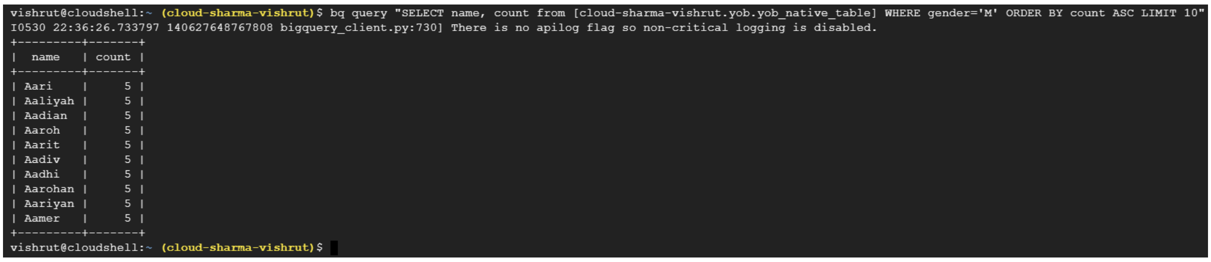
Query completed.

Query results

Row	name	count
1	Emma	20799
2	Olivia	19674
3	Sophia	18490
4	Isabella	16950
5	Ava	15586
6	Mia	13442
7	Emily	12562
8	Abigail	11985
9	Madison	10247
10	Charlotte	10048
11	Harper	9564
12	Sofia	9542
13	Avery	9517

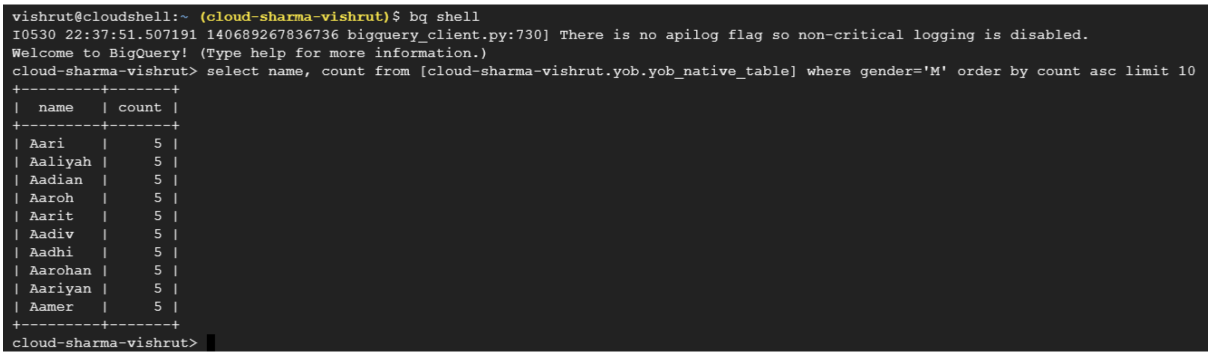
Results per page: 50 1 – 20 of 20

- **Screenshot your results and include it in your lab notebook**



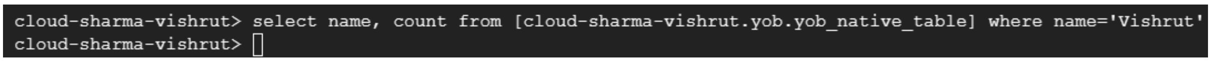
```
vishrut@cloudshell:~ (cloud-sharma-vishrut)$ bq query "SELECT name, count from [cloud-sharma-vishrut.yob.yob_native_table] WHERE gender='M' ORDER BY count ASC LIMIT 10"
I0530 22:36:26.733797 140627648767808 bigquery_client.py:730] There is no apilog flag so non-critical logging is disabled.
+---------+-------+
|  name   | count |
+---------+-------+
| Aari    |     5 |
| Aaliyah |     5 |
| Aadian  |     5 |
| Aaroh   |     5 |
| Aarit   |     5 |
| Aadiv   |     5 |
| Aadhi   |     5 |
| Aarohan |     5 |
| Aariyan |     5 |
| Aamer   |     5 |
+---------+-------+
vishrut@cloudshell:~ (cloud-sharma-vishrut)$
```

- **Screenshot your results and include it in your lab notebook**



```
vishrut@cloudshell:~ (cloud-sharma-vishrut)$ bq shell
I0530 22:37:51.507191 140689267836736 bigquery_client.py:730] There is no apilog flag so non-critical logging is disabled.
Welcome to BigQuery! (Type help for more information.)
cloud-sharma-vishrut> select name, count from [cloud-sharma-vishrut.yob.yob_native_table] where gender='M' order by count asc limit 10
+---------+-------+
|  name   | count |
+---------+-------+
| Aari    |     5 |
| Aaliyah |     5 |
| Aadian  |     5 |
| Aaroh   |     5 |
| Aarit   |     5 |
| Aadiv   |     5 |
| Aadhi   |     5 |
| Aarohan |     5 |
| Aariyan |     5 |
| Aamer   |     5 |
+---------+-------+
cloud-sharma-vishrut>
```

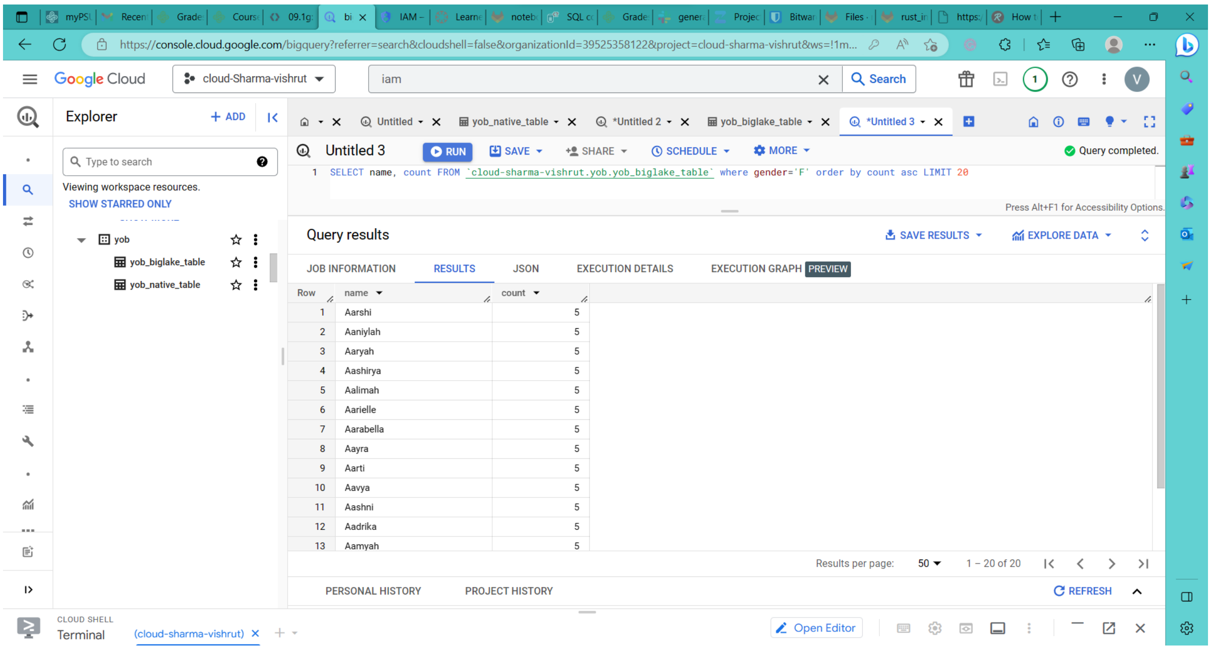
- **Screenshot your results and include it in your lab notebook**



```
cloud-sharma-vishrut> select name, count from [cloud-sharma-vishrut.yob.yob_native_table] where name='Vishrut'
cloud-sharma-vishrut>
```

9. Query data

- **Screenshot the query results and include it in your lab notebook**



https://console.cloud.google.com/bigquery?referrer=search&cloudshell=false&organizationId=39525358122&project=cloud-sharma-vishrut&ws=!1m...

```
SELECT name, count FROM `cloud-sharma-vishrut.yob.yob_biglake_table` where gender='F' order by count asc LIMIT 20
```

Row	name	count
1	Aarshi	5
2	Aaniylah	5
3	Aaryah	5
4	Aashirya	5
5	Aalimah	5
6	Aarielle	5
7	Aarabella	5
8	Aayra	5
9	Aarti	5
10	Aavya	5
11	Aashni	5
12	Aadrika	5
13	Aamyah	5

09.2g: Jupyter Notebooks

3. BigQuery query

- **How much less data does this query process compared to the size of the table?**

Answer: 18.89 GB less data

- **How many twins were born during this time range?**

Answer: 375362

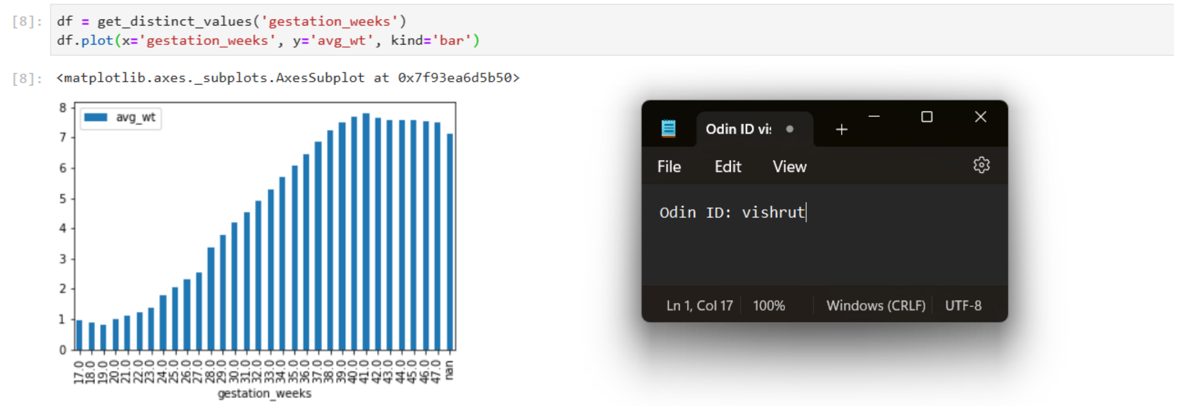
- **How much lighter on average are they compared to single babies?**

Answer: 2.17116045789 lighter

6. Run queries

- **Show the plots generated for the two most important features for your lab notebook**

Answer:

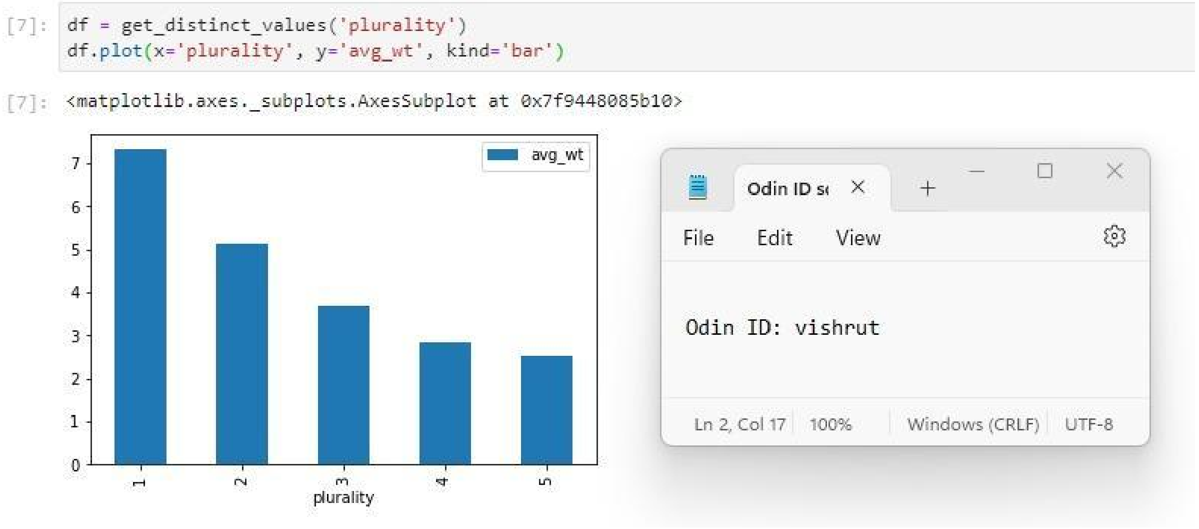


```
[8]: df = get_distinct_values('gestation_weeks')
     df.plot(x='gestation_weeks', y='avg_wt', kind='bar')

[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7f93ea6d5b50>
```

Odin ID: vishrut

Ln 1, Col 17 | 100% | Windows (CRLF) | UTF-8



```
[7]: df = get_distinct_values('plurality')
     df.plot(x='plurality', y='avg_wt', kind='bar')

[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9448085b10>
```

Odin ID: vishrut

Ln 2, Col 17 | 100% | Windows (CRLF) | UTF-8

8. Mobility

- **What day saw the largest spike in trips to grocery and pharmacy stores?**

Answer: 2020-03-13

- **On the day the stay-at-home order took effect (3/23/2020), what was the total impact on workplace trips?**

Answer: -49

9. Airport traffic

- **Which three airports were impacted the most in April 2020 (the month when lockdowns became widespread)?**

Answer: Newark Liberty International, Daniel K. Inouye International, Chicago OHare International

- **Run the query again using the month of August 2020. Which three airports were impacted the most?**

Answer: Newark Liberty International, Charlotte Douglas International, Dallas/Fort Worth International

10. Mortality

- **What table and columns identify the place name, the starting date, and the number of excess deaths from COVID-19?**

Answer: Table name: excess_deaths

Columns: placename, start_date, excess_deaths

- **What table and columns identify the date, county, and deaths from COVID-19?**

Answer: Table name: us_counties

Columns: date, county, deaths

- **What table and columns identify the date, state, and confirmed cases of COVID-19?**

Answer: Table name: us_states

Columns: date, state_name, confirmed_cases

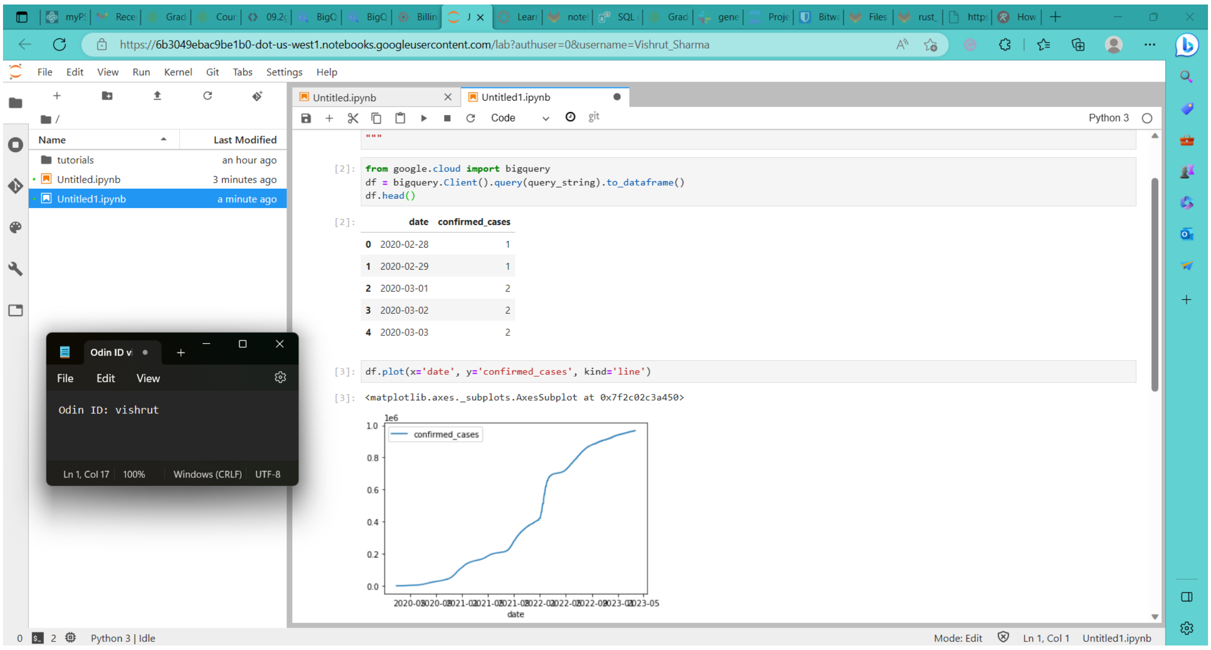
- **What table and columns identify a county code and the percentage of its residents that report they always wear masks?**

Answer: Table name: mask_use_by_county

Columns: county_fips_code, always

11. Run example queries

- **Show a screenshot of the plot and the code used to generate it for your lab notebook**



```
"""
```

```
[2]: from google.cloud import bigquery
     df = bigquery.Client().query(query_string).to_dataframe()
     df.head()
```

[2]:

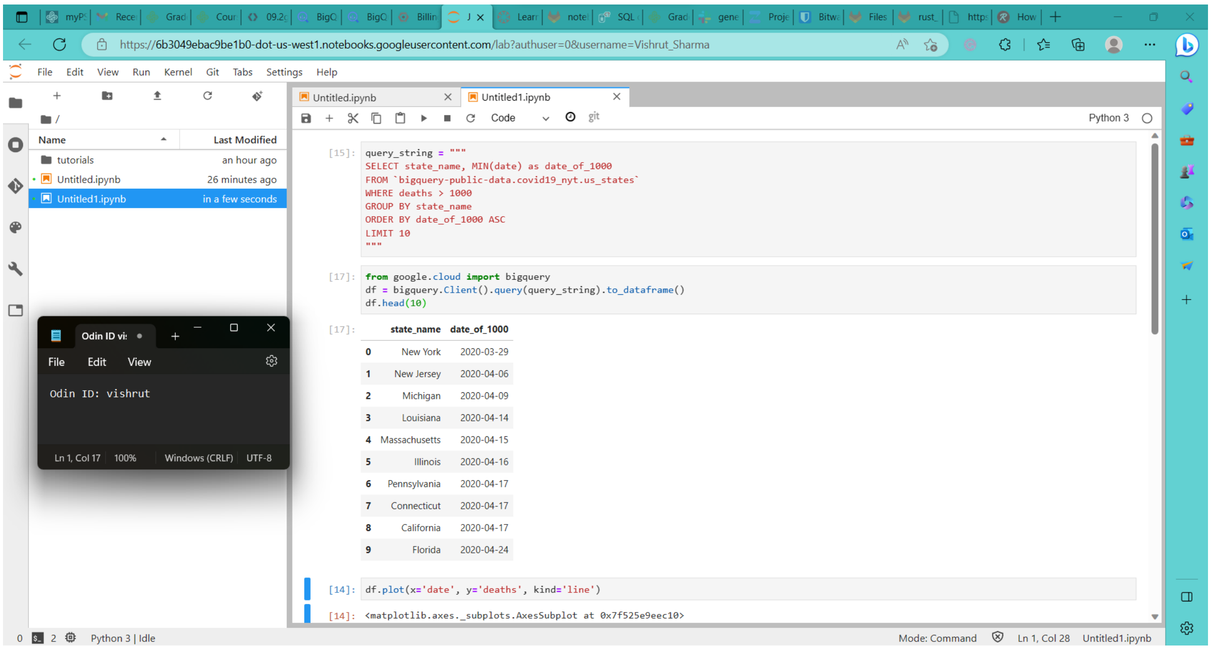
	date	confirmed_cases
0	2020-02-28	1
1	2020-02-29	1
2	2020-03-01	2
3	2020-03-02	2
4	2020-03-03	2

```
[3]: df.plot(x='date', y='confirmed_cases', kind='line')
```

```
[3]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2c02c3a450>
```

Odin ID: vishrut

- **From within your Jupyter notebook, run the query and write code that shows the first 10 states that reached 1000 deaths from COVID-19. Take a screenshot for your lab notebook.**



```
[15]: query_string = """
      SELECT state_name, MIN(date) as date_of_1000
      FROM `bigquery-public-data.covid19_nyt.us_states`
      WHERE deaths > 1000
      GROUP BY state_name
      ORDER BY date_of_1000 ASC
      LIMIT 10
      """
```

```
[17]: from google.cloud import bigquery
      df = bigquery.Client().query(query_string).to_dataframe()
      df.head(10)
```

[17]:

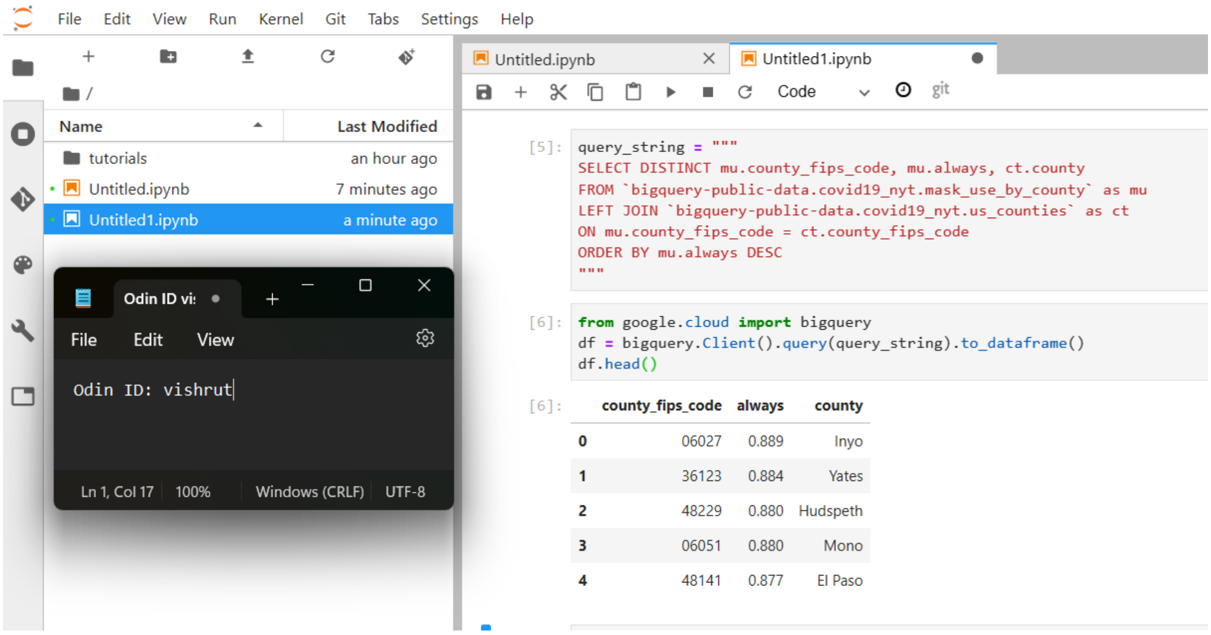
	state_name	date_of_1000
0	New York	2020-03-29
1	New Jersey	2020-04-06
2	Michigan	2020-04-09
3	Louisiana	2020-04-14
4	Massachusetts	2020-04-15
5	Illinois	2020-04-16
6	Pennsylvania	2020-04-17
7	Connecticut	2020-04-17
8	California	2020-04-17
9	Florida	2020-04-24

```
[14]: df.plot(x='date', y='deaths', kind='line')
```

```
[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7f525e9eec10>
```

Odin ID: vishrut

- **Take a screenshot for your lab notebook of the Top 5 counties and the states they are located in.**



Name	Last Modified
tutorials	an hour ago
Untitled.ipynb	7 minutes ago
Untitled1.ipynb	a minute ago

```
Odin ID: vishrut
```

```
[5]: query_string = """
     SELECT DISTINCT mu.county_fips_code, mu.always, ct.county
     FROM `bigquery-public-data.covid19_nyt.mask_use_by_county` as mu
     LEFT JOIN `bigquery-public-data.covid19_nyt.us_counties` as ct
     ON mu.county_fips_code = ct.county_fips_code
     ORDER BY mu.always DESC
     """
```

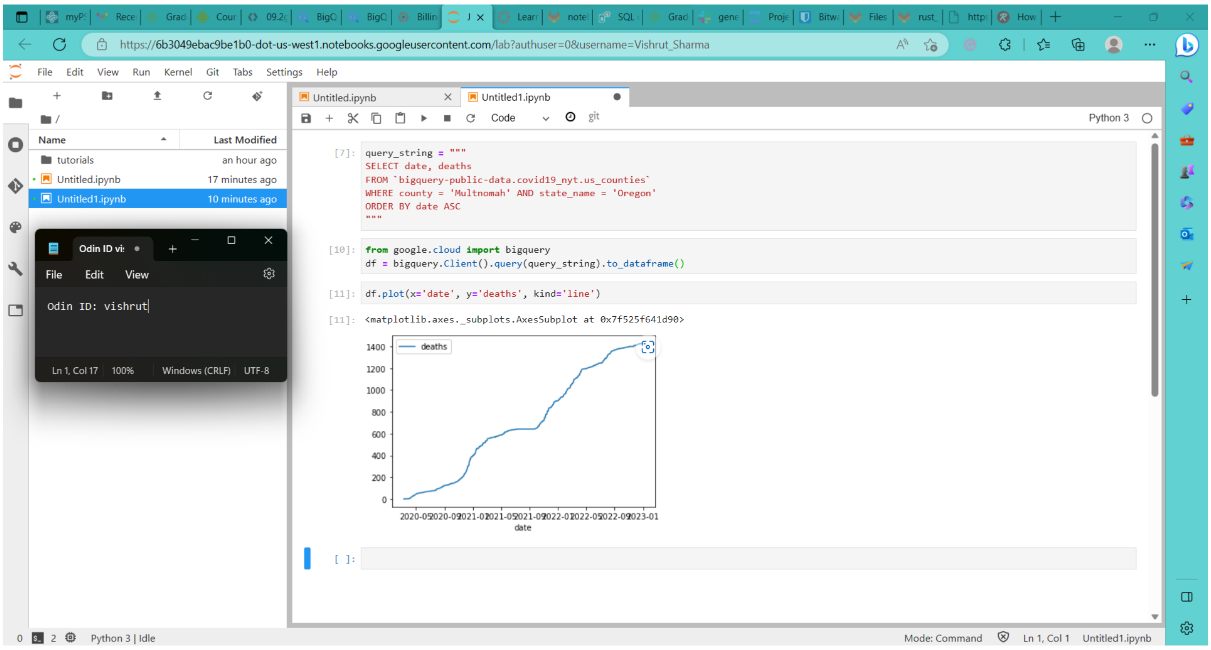
```
[6]: from google.cloud import bigquery
     df = bigquery.Client().query(query_string).to_dataframe()
     df.head()
```

[6]:

	county_fips_code	always	county
0	06027	0.889	Inyo
1	36123	0.884	Yates
2	48229	0.880	Hudspeth
3	06051	0.880	Mono
4	48141	0.877	El Paso

12. Write queries

- **Plot the results and take a screenshot for your lab notebook.**



Name	Last Modified
tutorials	an hour ago
Untitled.ipynb	17 minutes ago
Untitled1.ipynb	10 minutes ago

```
Odin ID: vishrut
```

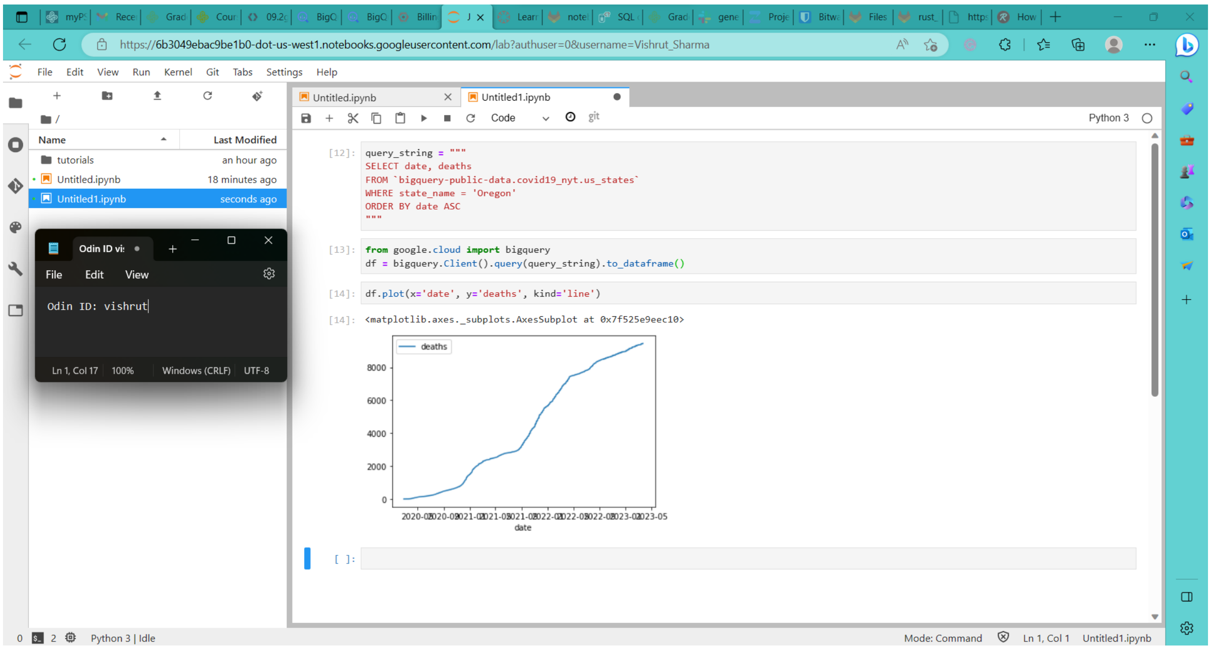
```
[7]: query_string = """
     SELECT date, deaths
     FROM `bigquery-public-data.covid19_nyt.us_counties`
     WHERE county = 'Multnomah' AND state_name = 'Oregon'
     ORDER BY date ASC
     """
```

```
[10]: from google.cloud import bigquery
      df = bigquery.Client().query(query_string).to_dataframe()
```

```
[11]: df.plot(x='date', y='deaths', kind='line')
```

```
[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f525f641d90>
```

- **Plot the results and take a screenshot for your lab notebook.**



```
[12]: query_string = """
SELECT date, deaths
FROM `bigquery-public-data.covid19_nyt.us_states`
WHERE state_name = 'Oregon'
ORDER BY date ASC
"""
```

```
[13]: from google.cloud import bigquery
df = bigquery.Client().query(query_string).to_dataframe()
```

```
[14]: df.plot(x='date', y='deaths', kind='line')
```

```
[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7f525e9eec10>
```

```
Odin ID: vishrut
```

<u>09.3g: Dataproc</u>

<u>6. Run computation</u>

- **How long did the job take to execute?**

<u>Answer</u>: Approximately 30 seconds

- **Examine output.txt and show the estimate of π calculated.**

<u>Answer</u>: 3.141495591414956

<u>8. Run computation again</u>

- **How long did the job take to execute? How much faster did it take?**

<u>Answer</u>: Approximately 19 seconds. It was 11 seconds faster.

- **Examine output2.txt and show the estimate of π calculated.**

<u>Answer</u>: 3.1415917714145079

<u>09.4g: Dataflow</u>

<u>3. Beam code</u>

- **Where is the input taken from by default?**

<u>Answer</u>: input is taken from '../javahelp/src/main/java/com/google/cloud/training/dataanalyst/javahelp/'

- **Where does the output go by default?**

Answer: '/tmp/output'

- **Examine both the getPackages() function and the splitPackageName() function. What operation does the 'PackageUse()' transform implement?**

Answer: The PackageUse**()** transform takes a line of code and a keyword and returns a tuple with a count of 1 for each package used.

- **Look up Beam's CombinePerKey. What operation does the TotalUse operation implement?**

Answer: The TotalUse operation implements the Reduce operation using the CombinePerKey transform. It combines the counts for each packet and summarizes them based on the packet key.

- **Which operations correspond to a "Map"?**

Answer: GetImports, PackageUse

- **Which operation corresponds to a "Shuffle-Reduce"?**

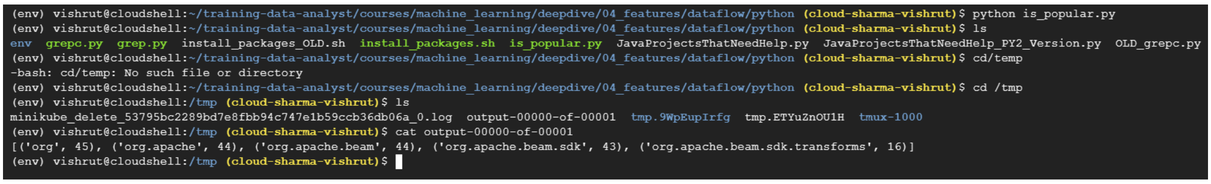
Answer: TotalUse

- **Which operation corresponds to a "Reduce"?**

Answer: Top_5

4. Run pipeline locally

- **Take a screenshot of its contents**



```
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$ python is_popular.py
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$ ls
env  grepc.py  grep.py  install_packages_OLD.sh  install_packages.sh  is_popular.py  JavaProjectsThatNeedHelp.py  JavaProjectsThatNeedHelp_PY2_Version.py  OLD_grepc.py
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$ cd/temp
-bash: cd/temp: No such file or directory
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$ cd /tmp
(env) vishrut@cloudshell:/tmp (cloud-sharma-vishrut)$ ls
minikube_delete_53795bc2289bd7e8fbb94c747e1b59ccb36db06a_0.log  output-00000-of-00001  tmp.9WpEupIrfg  tmp.ETYuZnOU1H  tmux-1000
(env) vishrut@cloudshell:/tmp (cloud-sharma-vishrut)$ cat output-00000-of-00001
[('org', 45), ('org.apache', 44), ('org.apache.beam', 44), ('org.apache.beam.sdk', 43), ('org.apache.beam.sdk.transforms', 16)]
(env) vishrut@cloudshell:/tmp (cloud-sharma-vishrut)$ 
```

- **Explain what the data in this output file corresponds to based on your understanding of the program.**

Answer: The output file will contain the top 5 Java packages that are most frequently used in the input Java code files. Each line in the output file will consist of a package name followed by its corresponding count, separated by a tab or space.

The program scans through the input Java code files and identifies lines starting with the keyword 'import'. It extracts the package names from those lines and keeps track of the occurrences of each package. The counts are combined using the 'CombinePerKey' transform.

The 'Top_5' transform selects the 5 packages with the highest counts, based on their occurrence values. Finally, the selected packages and their counts are written to the output file.

Consequently, the output file provides valuable insights into the commonly used Java packages in the analyzed codebase. This information helps developers identify frequently employed dependencies and potentially optimize their code or manage dependencies more effectively.

5. Dataflow Lab #2 (Word count)

- **What are the names of the stages in the pipeline?**

Answer: Read, Split, PairWithOne, GroupAndSum, Format, Write,

- **Describe what each stage does.**

Answer:

1. **Read**:
    - Utilizes the ReadFromText function to read the specified text file mentioned in the input argument.
    - Generates a PCollection named "lines" that contains the individual lines from the input text file.
2. **Split**:
    - Applies the WordExtractingDoFn function using ParDo.
    - Divides each line of text into separate words using regular expression matching.
    - Produces a PCollection consisting of the individual words.
3. **PairWithOne**:
    - Maps each word to a key-value pair where the word becomes the key and the value is set as 1.
    - Assigns an initial count of 1 to each word.
4. **GroupAndSum**:
    - Groups the key-value pairs by their keys (words).
    - Combines the values (counts) associated with each key (word) using the CombinePerKey transform along with the sum function.
    - Generates a PCollection of key-value pairs where each key represents a unique word, and the value is the sum of counts for that word.
5. **Format**:
    - Maps each key-value pair to a formatted string using the format_result function.
    - Formats the word and its count using the pattern "%s: %d", where %s represents the word and %d represents the count.

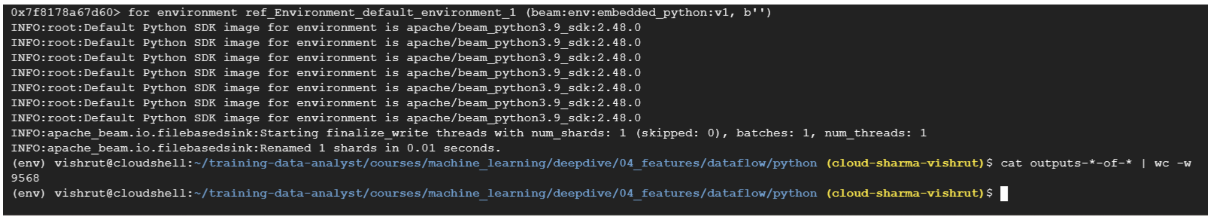
- Produces a PCollection of formatted strings.

6. **Write**:

   - Uses the WriteToText transform to write the output PCollection to the output file specified in the --output argument.
   - Each stage in the pipeline carries out a specific transformation or action on the data, such as reading from a text file, splitting lines into words, counting word occurrences, formatting results, and writing the output to a file.

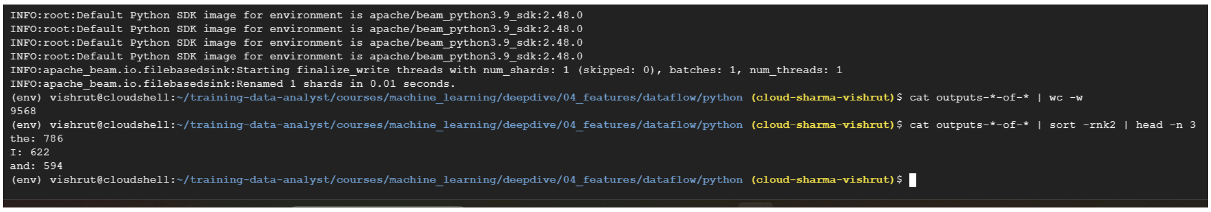
6. Run code locally

- **Use wc with an appropriate flag to determine the number of unique words in King Lear.**



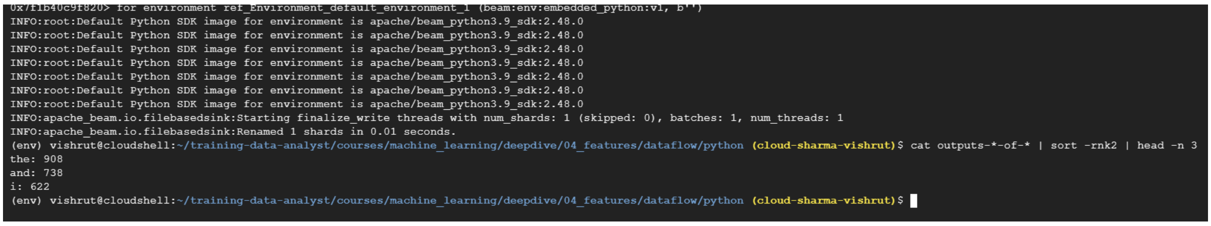
```
0x7f8178a67d60> for environment ref_Environment_default_environment_1 (beam:env:embedded_python:v1, b'')
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:apache_beam.io.filebasedsink:Starting finalize_write threads with num_shards: 1 (skipped: 0), batches: 1, num_threads: 1
INFO:apache_beam.io.filebasedsink:Renamed 1 shards in 0.01 seconds.
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$ cat outputs-*-of-* | wc -w
9568
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$
```

- **Use sort with appropriate flags to perform a *numeric* sort on the *key field* containing the count for each word in *descending* order. Pipe the output into head to show the top 3 words in King Lear and the number of times they appear**



```
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:apache_beam.io.filebasedsink:Starting finalize_write threads with num_shards: 1 (skipped: 0), batches: 1, num_threads: 1
INFO:apache_beam.io.filebasedsink:Renamed 1 shards in 0.01 seconds.
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$ cat outputs-*-of-* | wc -w
9568
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$ cat outputs-*-of-* | sort -rnk2 | head -n 3
the: 786
I: 622
and: 594
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$
```

- **Use the previous method to show the top 3 words in King Lear, case-insensitive, and the number of times they appear.**

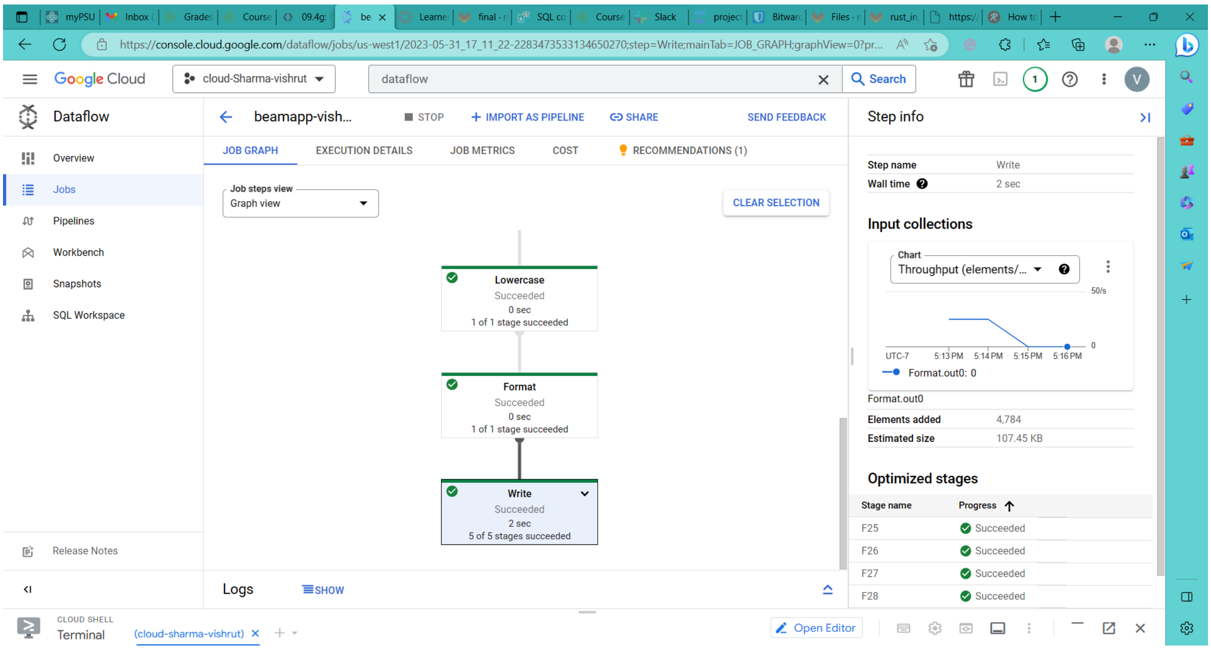


```
0x7f1b40c9f820> for environment ref_Environment_default_environment_1 (beam:env:embedded_python:v1, b'')
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.48.0
INFO:apache_beam.io.filebasedsink:Starting finalize_write threads with num_shards: 1 (skipped: 0), batches: 1, num_threads: 1
INFO:apache_beam.io.filebasedsink:Renamed 1 shards in 0.01 seconds.
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$ cat outputs-*-of-* | sort -rnk2 | head -n 3
the: 908
and: 738
i: 622
(env) vishrut@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-sharma-vishrut)$
```

9. Run code using Dataflow runner

- **The part of the job graph that has taken the longest time to complete.**

Answer: Write took 2 seconds



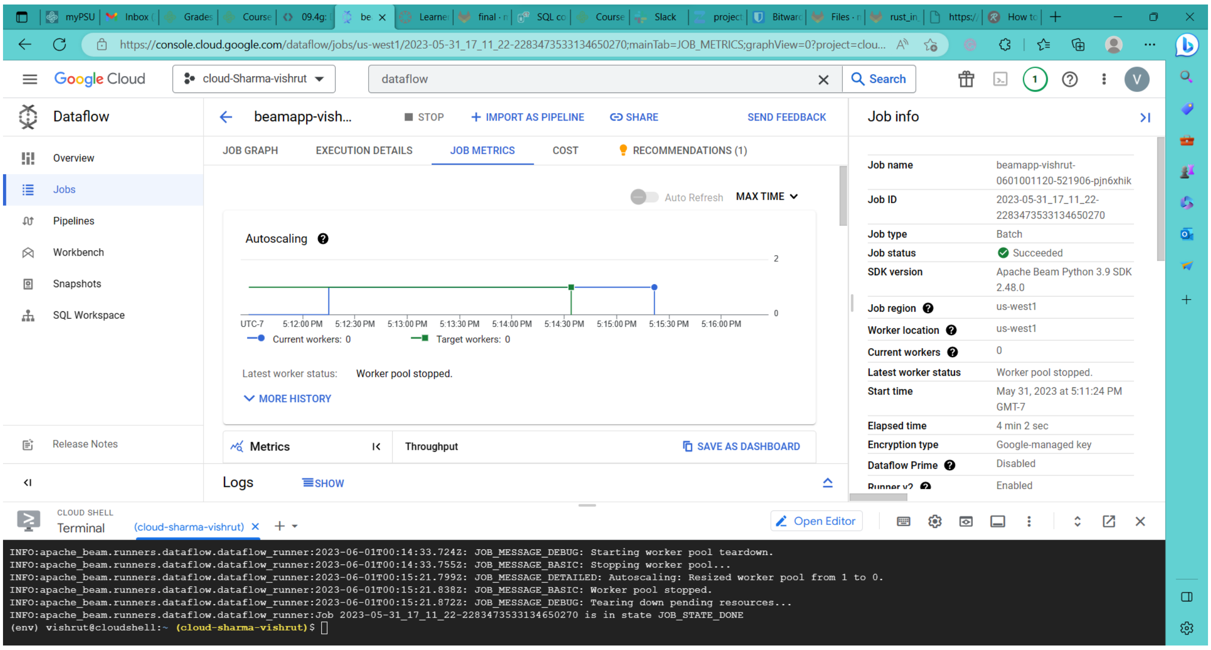
https://console.cloud.google.com/dataflow/jobs/us-west1/2023-05-31_17_11_22-2283473533134650270;step=Write;mainTab=JOB_GRAPH;graphView=0?pr...

Step name: Write
Wall time: 2 sec

Format.out0
Elements added: 4,784
Estimated size: 107.45 KB

Stage name	Progress
F25	Succeeded
F26	Succeeded
F27	Succeeded
F28	Succeeded

- **The autoscaling graph showing when the worker was created and stopped.**



https://console.cloud.google.com/dataflow/jobs/us-west1/2023-05-31_17_11_22-2283473533134650270;mainTab=JOB_METRICS;graphView=0?project=clou...

Latest worker status: Worker pool stopped.

Job name: beamapp-vishrut-0601001120-521906-pjn6xhik
Job ID: 2023-05-31_17_11_22-2283473533134650270
Job type: Batch
Job status: Succeeded
SDK version: Apache Beam Python 3.9 SDK 2.48.0
Job region: us-west1
Worker location: us-west1
Current workers: 0
Latest worker status: Worker pool stopped.
Start time: May 31, 2023 at 5:11:24 PM GMT-7
Elapsed time: 4 min 2 sec
Encryption type: Google-managed key
Dataflow Prime: Disabled
Runner v2: Enabled

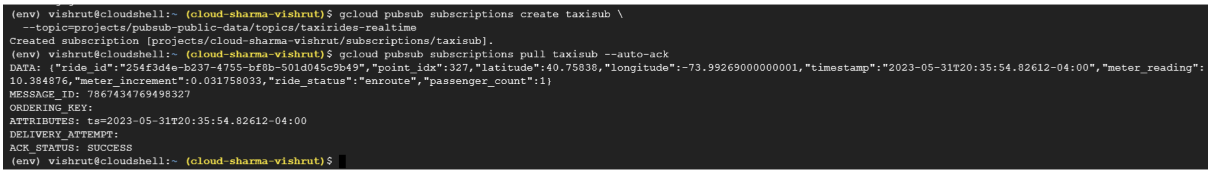
```
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-06-01T00:14:33.724Z: JOB_MESSAGE_DEBUG: Starting worker pool teardown.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-06-01T00:14:33.755Z: JOB_MESSAGE_BASIC: Stopping worker pool...
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-06-01T00:15:21.799Z: JOB_MESSAGE_DETAILED: Autoscaling: Resized worker pool from 1 to 0.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-06-01T00:15:21.838Z: JOB_MESSAGE_BASIC: Worker pool stopped.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-06-01T00:15:21.872Z: JOB_MESSAGE_DEBUG: Tearing down pending resources...
INFO:apache_beam.runners.dataflow.dataflow_runner:Job 2023-05-31_17_11_22-2283473533134650270 is in state JOB_STATE_DONE
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$
```

- **Examine the output directory in Cloud Storage. How many files has the final write stage in the pipeline created?**

Answer: 6 files

12. View raw data from PubSub

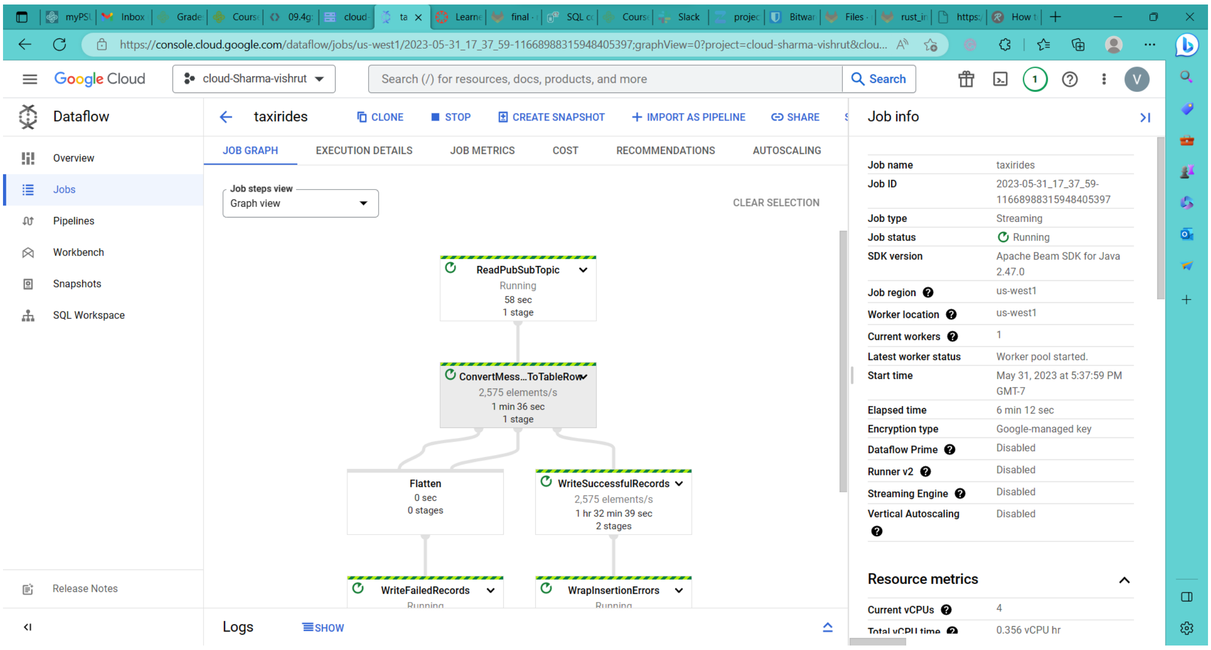
- **Take a screenshot listing the different fields of this object.**



```
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$ gcloud pubsub subscriptions create taxisub \
  --topic=projects/pubsub-public-data/topics/taxirides-realtime
Created subscription [projects/cloud-sharma-vishrut/subscriptions/taxisub].
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$ gcloud pubsub subscriptions pull taxisub --auto-ack
DATA: {"ride_id":"254f3d4e-b237-4755-bf8b-501d045c9b49","point_idx":327,"latitude":40.75838,"longitude":-73.99269000000001,"timestamp":"2023-05-31T20:35:54.82612-04:00","meter_reading":
10.384876,"meter_increment":0.031758033,"ride_status":"enroute","passenger_count":1}
MESSAGE_ID: 7867434769498327
ORDERING_KEY:
ATTRIBUTES: ts=2023-05-31T20:35:54.82612-04:00
DELIVERY_ATTEMPT:
ACK_STATUS: SUCCESS
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$
```

14. Run Dataflow job from template

- **Take a screenshot of the pipeline that includes its stages and the number of elements per second being handled by individual stages.**



https://console.cloud.google.com/dataflow/jobs/us-west1/2023-05-31_17_37_59-11668988315948405397;graphView=0?project=cloud-sharma-vishrut&clou...

ReadPubSubTopic: Running, 58 sec, 1 stage

ConvertMess...ToTableRow: 2,575 elements/s, 1 min 36 sec, 1 stage

Flatten: 0 sec, 0 stages

WriteSuccessfulRecords: 2,575 elements/s, 1 hr 32 min 39 sec, 2 stages

WriteFailedRecords: Running

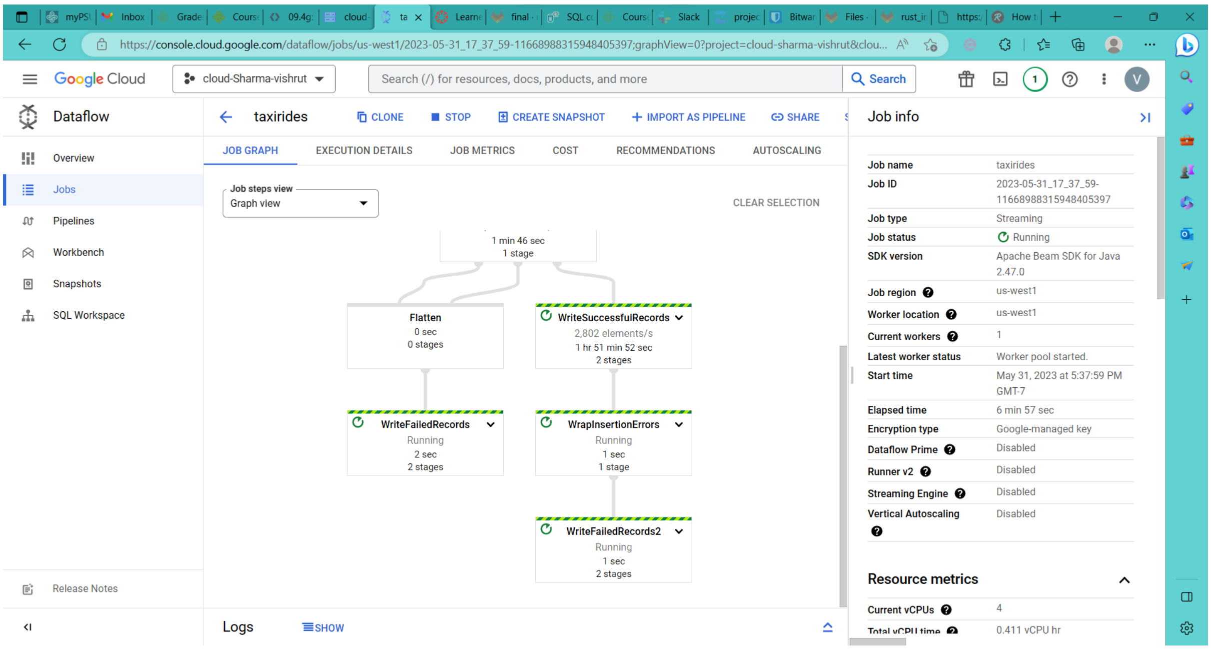
WrapInsertionErrors: Running

Job info

Job name	taxirides
Job ID	2023-05-31_17_37_59-11668988315948405397
Job type	Streaming
Job status	Running
SDK version	Apache Beam SDK for Java 2.47.0
Job region	us-west1
Worker location	us-west1
Current workers	1
Latest worker status	Worker pool started.
Start time	May 31, 2023 at 5:37:59 PM GMT-7
Elapsed time	6 min 12 sec
Encryption type	Google-managed key
Dataflow Prime	Disabled
Runner v2	Disabled
Streaming Engine	Disabled
Vertical Autoscaling	Disabled

Resource metrics

Current vCPUs	4
Total vCPU time	0.356 vCPU hr



https://console.cloud.google.com/dataflow/jobs/us-west1/2023-05-31_17_37_59-11668988315948405397;graphView=0?project=cloud-sharma-vishrut&clou...

1 min 46 sec, 1 stage

Flatten: 0 sec, 0 stages

WriteSuccessfulRecords: 2,802 elements/s, 1 hr 51 min 52 sec, 2 stages

WriteFailedRecords: Running, 2 sec, 2 stages

WrapInsertionErrors: Running, 1 sec, 1 stage

WriteFailedRecords2: Running, 1 sec, 2 stages

Job info

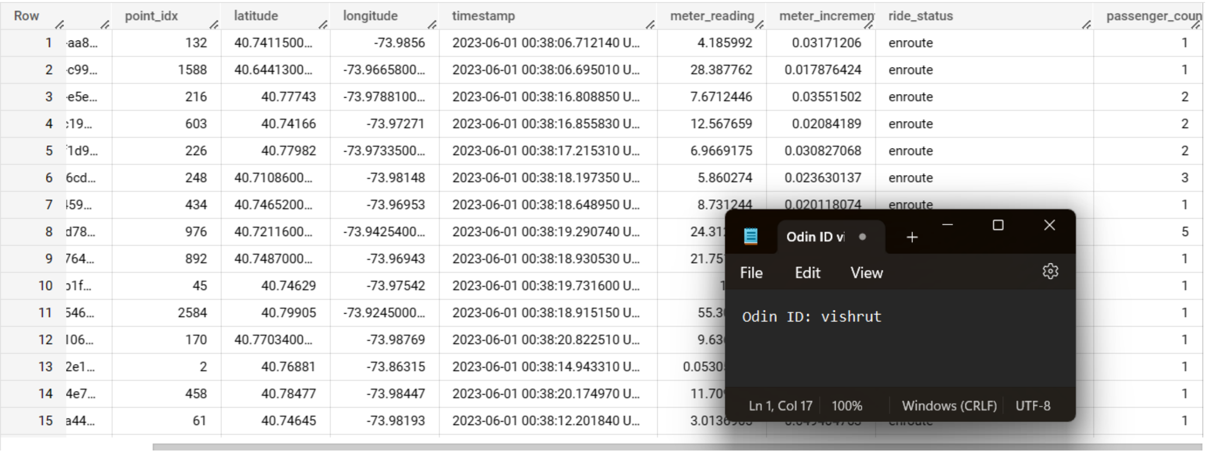
Job name	taxirides
Job ID	2023-05-31_17_37_59-11668988315948405397
Job type	Streaming
Job status	Running
SDK version	Apache Beam SDK for Java 2.47.0
Job region	us-west1
Worker location	us-west1
Current workers	1
Latest worker status	Worker pool started.
Start time	May 31, 2023 at 5:37:59 PM GMT-7
Elapsed time	6 min 57 sec
Encryption type	Google-managed key
Dataflow Prime	Disabled
Runner v2	Disabled
Streaming Engine	Disabled
Vertical Autoscaling	Disabled

Resource metrics

Current vCPUs	4
Total vCPU time	0.411 vCPU hr

15. Query data in BigQuery

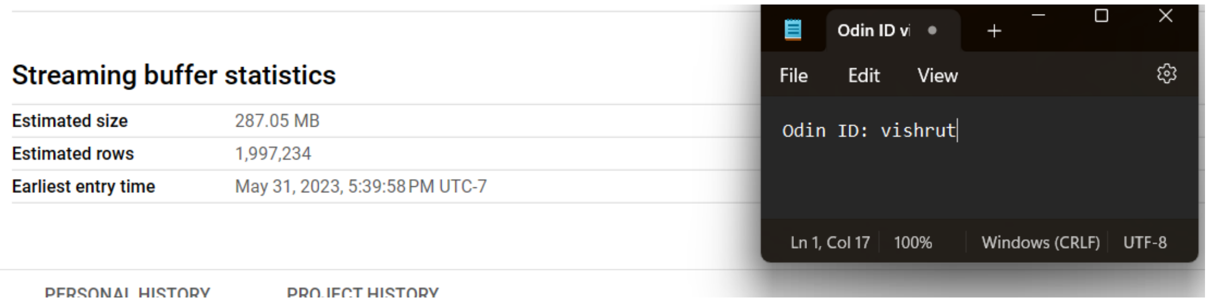
- **Take a screenshot showing the number of passengers and the amount paid for the first ride**



Row		point_idx	latitude	longitude	timestamp	meter_reading	meter_incremen	ride_status	passenger_coun
1	aa8…	132	40.7411500…	-73.9856	2023-06-01 00:38:06.712140 U…	4.185992	0.03171206	enroute	1
2	c99…	1588	40.6441300…	-73.9665800…	2023-06-01 00:38:06.695010 U…	28.387762	0.017876424	enroute	1
3	e5e…	216	40.77743	-73.9788100…	2023-06-01 00:38:16.808850 U…	7.6712446	0.03551502	enroute	2
4	:19…	603	40.74166	-73.97271	2023-06-01 00:38:16.855830 U…	12.567659	0.02084189	enroute	2
5	1d9…	226	40.77982	-73.9733500…	2023-06-01 00:38:17.215310 U…	6.9669175	0.030827068	enroute	2
6	6cd…	248	40.7108600…	-73.98148	2023-06-01 00:38:18.197350 U…	5.860274	0.023630137	enroute	3
7	159…	434	40.7465200…	-73.96953	2023-06-01 00:38:18.648950 U…	8.731244	0.020118074	enroute	1
8	d78…	976	40.7211600…	-73.9425400…	2023-06-01 00:38:19.290740 U…	24.31			5
9	764…	892	40.7487000…	-73.96943	2023-06-01 00:38:18.930530 U…	21.75			1
10	o1f…	45	40.74629	-73.97542	2023-06-01 00:38:19.731600 U…	1			1
11	546…	2584	40.79905	-73.9245000…	2023-06-01 00:38:18.915150 U…	55.3			1
12	106…	170	40.7703400…	-73.98769	2023-06-01 00:38:20.822510 U…	9.63			1
13	2e1…	2	40.76881	-73.86315	2023-06-01 00:38:14.943310 U…	0.0530			1
14	4e7…	458	40.78477	-73.98447	2023-06-01 00:38:20.174970 U…	11.70			1
15	a44…	61	40.74645	-73.98193	2023-06-01 00:38:12.201840 U…	3.0136			1

```
Odin ID: vishrut
```

- **Take a screenshot showing the estimated number of rows in the table.**

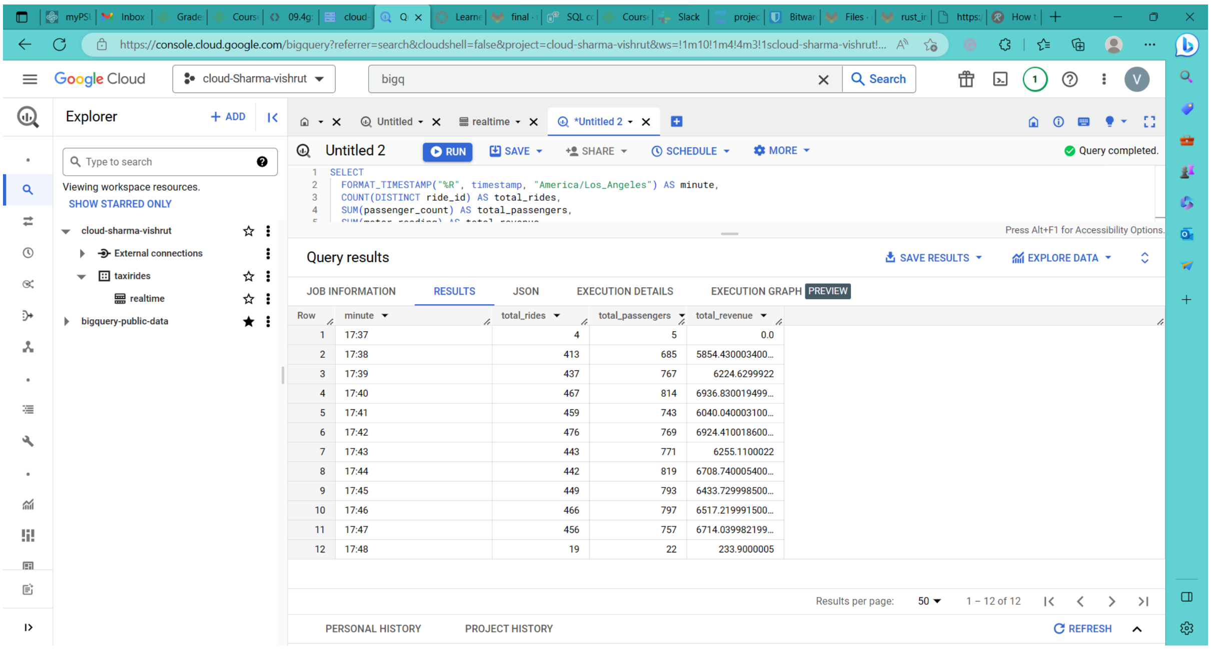


Streaming buffer statistics

Estimated size	287.05 MB
Estimated rows	1,997,234
Earliest entry time	May 31, 2023, 5:39:58 PM UTC-7

```
Odin ID: vishrut
```

- **Take a screenshot showing the per-minute number of rides, passengers, and revenue for the data collected**



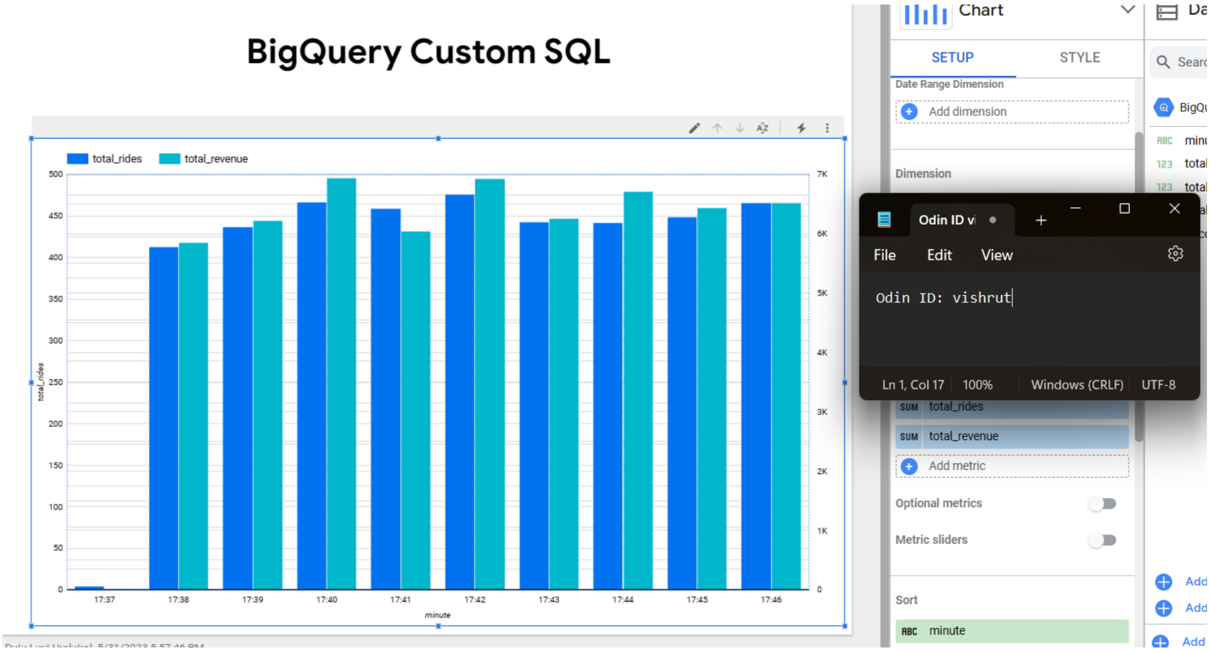
```
SELECT
  FORMAT_TIMESTAMP("%R", timestamp, "America/Los_Angeles") AS minute,
  COUNT(DISTINCT ride_id) AS total_rides,
  SUM(passenger_count) AS total_passengers,
  SUM(meter_reading) AS total_revenue
```

Query results

Row	minute	total_rides	total_passengers	total_revenue
1	17:37	4	5	0.0
2	17:38	413	685	5854.430003400…
3	17:39	437	767	6224.6299922
4	17:40	467	814	6936.830019499…
5	17:41	459	743	6040.040003100…
6	17:42	476	769	6924.410018600…
7	17:43	443	771	6255.1100022
8	17:44	442	819	6708.740005400…
9	17:45	449	793	6433.729998500…
10	17:46	466	797	6517.219991500…
11	17:47	456	757	6714.039982199…
12	17:48	19	22	233.9000005

16. Data visualization

- **Take a screenshot showing the plot for your data for your lab notebook**



Data Last Updated: 5/31/2023 5:57:46 PM

Date Range Dimension

Add dimension

Dimension

SUM total_rides

SUM total_revenue

Add metric

Optional metrics

Metric sliders

Sort

ABC minute

```
Odin ID: vishrut
```

Ln 1, Col 17 | 100% | Windows (CRLF) | UTF-8