

Tony Nguyen

Dr. Shawn Bowers

CPSC 324

26 February 2024

Homework 2

1. Step 1

a.

- Instead of “Compose a New Query,” it is now “Create SQL Query.”
- There is no “+” symbol to create a new query. It has been changed to “Query”.

b.

- Click “View actions” in the Explorer pane, then “Create dataset.”

Field	Value
Create table from	Google Cloud Storage
Select file from GCS bucket	sp1s/gsp072/baby-names/yob2014.txt
File format	CSV
Table	names_2014
Schema > Edit as text	Slide on, then add the following in the textbox: name:string,gender:string,count:integer

- Click “View actions” within the dataset, then “Create table.”

c.

- Query

```
SELECT
  name, count
FROM
  `babynames.names_2014`
WHERE
  gender = 'F'
ORDER BY count ASC LIMIT 10;
```

- Result

Row	name ▼	count ▼
1	Aamyah	5
2	Aalimah	5
3	Aaniylah	5
4	Aania	5
5	Aadrika	5
6	Aarion	5
7	Aarielle	5
8	Aamilah	5
9	Aaiza	5
10	Aabriella	5

2. Step 2

a.

- Examining

- `bq show bigquery-public-data:samples.shakespeare`

- Listing datasets

- `bq ls`
- `bq ls [Project ID]:`
 - List the datasets in that specific project
 - Ending with a colon

- Creating

- `bq mk [Dataset Name]`
- `bq` keyword is used for dealing with the dataset level (think like a folder)
- `ls`, normal command line args, deal with file level
- `unzip [filename]`
- Load a table

- `bq load babynames.names2010 yob2010.txt`
`name:string,gender:string,count:integer`

```
datasetID: babynames
tableID: names2010
source: yob2010.txt
schema: name:string,gender:string,count:integer
```

- Querying
 - `bq query --use_legacy_sql=fals \ [SQL_CODE]`
 - `bq query "[SQL_CODE]"`
- Removing
 - `bq rm -r babynames`

b.

- Query
 - `bq query "SELECT name, count FROM babynames.names2010 WHERE
gender = 'F' AND count >= 10000 ORDER BY count ASC LIMIT 5"`
- Result

name	count
Elizabeth	10276
Addison	10331
Mia	10646
Chloe	11761
Madison	13191

3. Step 3

a.

```
(cpssc322) tony@Tonys-MacBook-Pro CPSC324 % bq ls
(cpsc322) tony@Tonys-MacBook-Pro CPSC324 % gcloud projects list
PROJECT_ID          NAME                PROJECT_NUMBER
cnguyen4-cpsc324-hw1 cnguyen4-cpsc324-hw1 20928757838
cnguyen4-cpsc324-hw2-415105 cnguyen4-cpsc324-hw2 45274338188
(cpsc322) tony@Tonys-MacBook-Pro CPSC324 % gcloud config set project cnguyen4-cpsc324-hw2-415105

WARNING: Your active project does not match the quota project in your local Application Default Credentials file. This might result in unexpected quota issues.

To update your Application Default Credentials quota project, use the `gcloud auth application-default set-quota-project` command.
Updated property [core/project].
(cpsc322) tony@Tonys-MacBook-Pro CPSC324 % gcloud auth application-default set-quota-project
ERROR: (gcloud.auth.application-default.set-quota-project) argument QUOTA_PROJECT_ID: Must be specified.
Usage: gcloud auth application-default set-quota-project QUOTA_PROJECT_ID [optional flags]
       optional flags may be --help

For detailed information on this command and its flags, run:
  gcloud auth application-default set-quota-project --help
(cpsc322) tony@Tonys-MacBook-Pro CPSC324 % bq ls
(cpsc322) tony@Tonys-MacBook-Pro CPSC324 % bq mk babynames
Dataset 'cnguyen4-cpsc324-hw2-415105:babynames' successfully created.
(cpsc322) tony@Tonys-MacBook-Pro CPSC324 % curl -LO http://www.ssa.gov/OACT/babynames/names.zip
(cpsc322) tony@Tonys-MacBook-Pro CPSC324 % curl -LO http://www.ssa.gov/OACT/babynames/names.zip
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Dload  % Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0      0      0      0  --:--:-- --:--:-- --:--:--    0
100 7232k 100 7232k    0     0 3635k    0  0:00:01  0:00:01 --:--:-- 6245k
```

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq load babynames.names2010 yob2010.txt name:string,gender:string,count:integer
Upload complete.
Waiting on bqjob_r35d1d6ca90ebba93_0000018dcf6c2b7a_1 ... (1s) Current status: DONE
```

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq ls
datasetId
-----
babynames
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq show babynames:names2010
BigQuery error in show operation: Project babynames is not found. Make sure it references valid GCP project that hasn't been deleted.; Project id: babynames
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq ls
datasetId
-----
babynames
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq ls babynames
tableId    Type    Labels    Time Partitioning    Clustered Fields
-----
names2010  TABLE
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq show babynames.names2010
Table cnguyen4-cpsc324-hw2-415105:babynames.names2010
s      Last modified      Schema      Total Rows    Total Bytes    Expiration    Time Partitioning    Clustered Field
s      Total Logical Bytes    Total Physical Bytes    Labels
-----
21 Feb 22:06:31  |- name: string    34098        654967
654967          |- gender: string
                |- count: integer
```

c.

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq query "SELECT SUM(COUNT) AS total_count FROM babynames.names2010 WHERE gender='F' AND count=(SELECT MIN(COUNT) FROM babynames.names2010 WHERE gender='F')"
```

total_count
14245

d.

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq query "SELECT SUM(COUNT) AS total_count FROM babynames.names2010 WHERE gender='M' AND count=(SELECT MAX(COUNT) FROM babynames.names2010 WHERE gender='M')"
```

total_count
22144

e.

- Write a query to find the total count of baby names for each gender.

```
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsm % bq query "SELECT gender, SUM(count) AS total_count FROM babynames.names2010 GROUP BY gender"
+-----+-----+
| gender | total_count |
+-----+-----+
| F      | 1776636    |
| M      | 1917792    |
+-----+-----+
```

4. Step 4

a.

Untitled
 RUN
 SAVE
 DOWNLOAD
 SHARE
 SCHEDULE
 MORE

```

1 #standardSQL
2 # Find the total unique visitors
3 SELECT
4   COUNT(*) AS product_views,
5   COUNT(DISTINCT fullVisitorId) AS unique_visitors
6 FROM `data-to-insights.ecommerce.all_sessions`;
  
```

Query results
 SAVE RESULTS
 EXPLORE DATA

OB INFORMATION
 RESULTS
 CHART
 JSON
 EXECUTION DETAILS
 EXECUTION

Row	product_views	unique_visitors
1	21493109	389934

1 #standardSQL
 2 #Find the total unique visitors by the referring site
 3 SELECT
 4 COUNT(DISTINCT fullVisitorId) AS unique_visitors,
 5 channelGrouping
 6 FROM `data-to-insights.ecommerce.all_sessions`
 7 GROUP BY channelGrouping
 8 ORDER BY channelGrouping DESC;

Query results
 SAVE RESULTS
 EXPLORE DATA

OB INFORMATION
 RESULTS
 CHART
 JSON
 EXECUTION DETAILS
 EXECUTION

Row	unique_visitors	channelGrouping
1	38101	Social
2	57308	Referral
3	11865	Paid Search
4	211993	Organic Search
5	3067	Display
6	75688	Direct

Open menu

Untitled

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

```
1 #standardSQL
2 #List all the unique product names (v2ProductName) alphabetically:
3 SELECT
4   (v2ProductName) AS ProductName
5 FROM `data-to-insights.ecommerce.all_sessions`
6 GROUP BY ProductName
7 ORDER BY ProductName
```

Press Option+F1 for Accessibility Options.

Query results

SAVE RESULTSEXPLORE DATA

<

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

>

Row	ProductName
1	1 oz Hand Sanitizer
2	14oz Ceramic Google Mug
3	15 oz Ceramic Mug
4	15" Android Squishable - O...
5	16 oz. Hot and Cold Tumbler

Results per page: 501 - 50 of 633<>>>

Job history

REFRESH

Untitled

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

Query completed.

```
1 SELECT v2ProductName as ProductName, COUNT(*) as ProductCount
2 FROM `data-to-insights.ecommerce.all_sessions`
3 WHERE type = 'PAGE'
4 GROUP BY ProductName
5 ORDER BY ProductCount DESC
6 LIMIT 5;
```

Press Option+F1 for Accessibility Options.

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	ProductName	ProductCount
1	Google Men's 100% Cotton	316482
2	22 oz YouTube Bottle Infu...	221558
3	YouTube Men's Short Slee...	210700
4	Google Men's 100% Cotton	202205
5	YouTube Custom Decals	200789

Untitled

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

Query completed.

```
1 WITH unique_product_views_by_person AS (  
2 SELECT  
3   fullVisitorId,  
4   (v2ProductName) AS ProductName  
5 FROM `data-to-insights.ecommerce.all_sessions`  
6 WHERE type = 'PAGE'  
7 GROUP BY fullVisitorId, v2ProductName )  
8  
9 SELECT  
10  COUNT(*) AS unique_view_count,  
11  ProductName  
12 FROM unique_product_views_by_person  
13 GROUP BY ProductName  
14 ORDER BY unique_view_count DESC  
15 LIMIT 5
```

Press Option+F1 for Accessibility Options.

Query results

SAVE RESULTS

EXPLORE DATA

Job Information	Results	Chart	JSON	Execution Details	Execution Graph
Row	unique_view_cour	ProductName			
1	152358	Google Men's 100% Cotto...			
2	143770	22 oz YouTube Bottle Infu...			
3	127904	YouTube Men's Short Slee...			
4	122051	YouTube Twill Cap			
5	121288	YouTube Custom Decals			

Untitled

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

Query completed.

```
1 SELECT  
2   COUNT(*) AS product_views,  
3   COUNT(productQuantity) AS orders,  
4   SUM(productQuantity) AS quantity_product_ordered,  
5   v2ProductName  
6 FROM `data-to-insights.ecommerce.all_sessions`  
7 WHERE type = 'PAGE'  
8 GROUP BY v2ProductName  
9 ORDER BY product_views DESC  
10 LIMIT 5;
```

Press Option+F1 for Accessibility Options.

Query results

SAVE RESULTS

EXPLORE DATA

Job Information	Results	Chart	JSON	Execution Details	Execution Graph
Row	product_views	orders	quantity_product	v2ProductName	
1	316482	3158	6352	Google Men's 100% Cotto...	
2	221558	508	4769	22 oz YouTube Bottle Infu...	
3	210700	949	1114	YouTube Men's Short Slee...	
4	202205	2713	8072	Google Men's 100% Cotto...	
5	200789	1703	11336	YouTube Custom Decals	

Untitled

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

Query completed.

```
1 SELECT  
2   COUNT(*) AS product_views,  
3   COUNT(productQuantity) AS orders,  
4   SUM(productQuantity) AS quantity_product_ordered,  
5   SUM(productQuantity) / COUNT(productQuantity) AS avg_per_order,  
6   (v2ProductName) AS ProductName  
7 FROM `data-to-insights.ecommerce.all_sessions`  
8 WHERE type = 'PAGE'  
9 GROUP BY v2ProductName  
10 ORDER BY product_views DESC  
11 LIMIT 5;
```

Press Option+F1 for Accessibility Options.

Query results

SAVE RESULTS

EXPLORE DATA

Job Information	Results	Chart	JSON	Execution Details	Execution Graph
Row	product_views	orders	quantity_product	avg_per_order	ProductName
1	316482	3158	6352	2.011399620...	Google Men's 100% Cotto...
2	221558	508	4769	9.387795275...	22 oz YouTube Bottle Infu...
3	210700	949	1114	1.173867228...	YouTube Men's Short Slee...
4	202205	2713	8072	2.975304091...	Google Men's 100% Cotto...
5	200789	1703	11336	6.656488549...	YouTube Custom Decals

b.

Untitled

Query completed.

```
1 WITH session_counts AS (  
2   SELECT  
3     COUNT(*) AS product_views,  
4     COUNT(productQuantity) AS orders,  
5     SUM(productQuantity) AS quantity_product_ordered,  
6     SUM(productQuantity) / COUNT(productQuantity) AS avg_per_order,  
7     v2ProductName  
8   FROM 'data-to-insights.ecommerce.all_sessions'  
9   WHERE type = 'PAGE'  
10  GROUP BY v2ProductName  
11 )  
12  
13 SELECT  
14   product_views,  
15   orders,  
16   quantity_product_ordered,  
17   avg_per_order,  
18   (v2ProductName) AS ProductName  
19 FROM session_counts  
20 WHERE avg_per_order >= (SELECT MAX(avg_per_order)  
21                        FROM session_counts  
22                        );
```

Query results

Row	product_views	orders	quantity_product	avg_per_order	ProductName
1	138471	1670	702478	420.6455089...	Google Men's Vintage Bad...

c.

- Find the top 5 products with the highest average quantity ordered per session that have been viewed less than 100 times.

Untitled

Query completed.

```
1 WITH session_counts AS (  
2   SELECT  
3     COUNT(*) AS session_count,  
4     COUNT(productQuantity) AS orders,  
5     SUM(productQuantity) AS quantity_product_ordered,  
6     SUM(productQuantity) / COUNT(*) AS avg_per_session,  
7     v2ProductName  
8   FROM 'data-to-insights.ecommerce.all_sessions'  
9   WHERE type = 'PAGE'  
10  GROUP BY v2ProductName  
11  HAVING session_count < 100  
12 )  
13  
14 SELECT  
15   session_count,  
16   orders,  
17   quantity_product_ordered,  
18   avg_per_session,  
19   (v2ProductName) AS ProductName  
20 FROM session_counts  
21 ORDER BY session_counts.avg_per_session DESC  
22 LIMIT 5;
```

Row	session_count	orders	quantity_product	avg_per_session	ProductName
1	41	16	194	4.731707317...	Eco-Aware Large Recycled...
2	36	19	107	2.972222222...	Google Sunglasses Red
3	19	11	39	2.052631578...	Google Sunglasses Green
4	24	14	38	1.583333333...	Google Sunglasses Yellow
5	31	18	49	1.580645161...	Google Sunglasses Blue

5. Step 5

a.


```

1 SELECT
2   -- Create a timestamp from the date components.
3   stn,
4   TIMESTAMP(CONCAT(year, "-", mo, "-", da)) AS timestamp,
5   -- Replace numerical null values with actual null
6   AVG(IF (temp=9999.9,
7           null,
8           temp)) AS temperature,
9   AVG(IF (wdsp="999.9",
10          null,
11          CAST(wdsp AS Float64))) AS wind_speed,
12   AVG(IF (prcp=99.99,
13          0,
14          prcp)) AS precipitation
15 FROM
16   `bigquery-public-data.noaa_gsod.gsod20*`
17 WHERE
18   CAST(YEAR AS INT64) > 2010
19   AND CAST(MO AS INT64) = 6
20   AND CAST(DA AS INT64) = 12
21   AND (stn="725030" OR -- La Guardia
22        stn="744860")   -- JFK
23 GROUP BY
24   stn,
25   timestamp
26 ORDER BY
27   timestamp DESC,
28   stn ASC

```

```

1 -- Query the level of complaints
2 SELECT
3   EXTRACT(YEAR
4   FROM
5   | created_date) AS year,
6   complaint_type,
7   COUNT(1) AS num_complaints
8 FROM
9   `bigquery-public-data.new_york.311_service_requests`
10 GROUP BY
11   year,
12   complaint_type
13 ORDER BY
14   num_complaints DESC

```

b.

- Part 3 is trying to save a query result as a table in a dataset. By default, BigQuery returns a temporary table to show the query result. Now, such results can be saved for future uses.
- There is the IF statement, something that I have not used before
- The FROM clause seems to have a special construct of “.gsod20*” at the end,
- I guess what the FROM clause tries to do is that I query all of the files that have the beginning filename pattern to be “gsod20...”.

c.

- The CORR function measures the correlation between two variables.
- Strong correlation is above 0.7 absolute value

```

1  -- compare the number of complaints received and daily temperature
2  SELECT
3      descriptor,
4      sum(complaint_count) as total_complaint_count,
5      count(temperature) as data_count,
6      ROUND(corr(temperature, avg_count),3) AS corr_count,
7      ROUND(corr(temperature, avg_pct_count),3) AS corr_pct
8  FROM (
9      SELECT
10         avg(pct_count) as avg_pct_count,
11         avg(day_count) as avg_count,
12         sum(day_count) as complaint_count,
13         descriptor,
14         temperature
15     FROM (
16         SELECT
17             DATE(timestamp) AS date,
18             temperature
19         FROM
20             demos.nyc_weather) a
21     JOIN (
22         SELECT x.date, descriptor, day_count, day_count / all_calls_count as pct_count
23     FROM
24         (SELECT
25             DATE(created_date) AS date,
26             concat(complaint_type, ': ', descriptor) as descriptor,
27             COUNT(*) AS day_count
28         FROM
29             `bigquery-public-data.new_york.311_service_requests`
30         GROUP BY
31             date,
32             descriptor)x
33     JOIN (
34         SELECT
35             DATE(timestamp) AS date,
36             COUNT(*) AS all_calls_count
37         FROM `demos.nyc_weather`
38         GROUP BY date
39     )y
40
41 )b
42 ON
43     a.date = b.date
44 GROUP BY
45     descriptor,
46     temperature
47 )
48 GROUP BY descriptor
49 HAVING
50     total_complaint_count > 5000 AND
51     ABS(corr_pct) > 0.5 AND
52     data_count > 5
53 ORDER BY
54     ABS(corr_pct) DESC

```

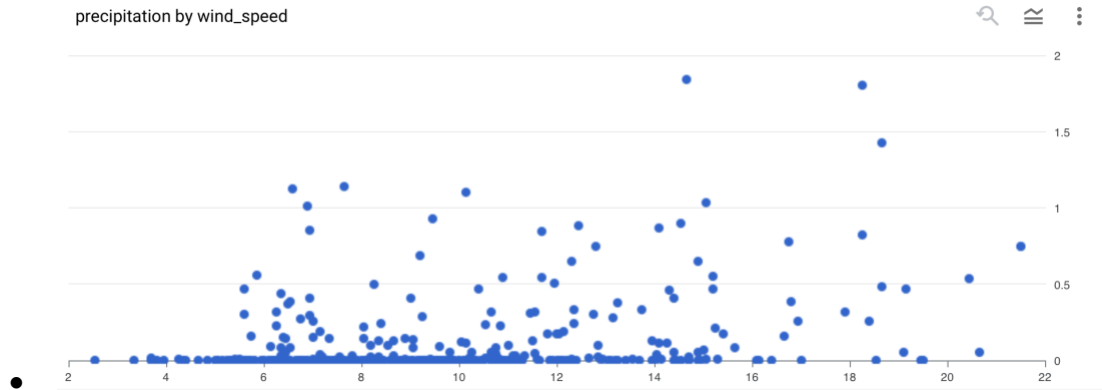
d.

- Create a query that shows the correlation between wind_speed and precipitation in 2022

```

1  SELECT wind_speed, precipitation
2  FROM `demos.nyc_weather`
3  WHERE FORMAT_TIMESTAMP('%Y-%m-%d', timestamp) BETWEEN '2022-01-01' AND '2022-12-31';

```



6. Step 6

- Command to append data to a current table
 - Note that it needs to be run within the same project

bq load

--source_format=CSV

--autodetect //for schema

--noreplace //append mode

nyctaxi.2018trips //destination table

gs://cloud-training/OCBL013/nyc_tlc_yellow_trips_2018_subset_2.csv

//location in bucket

- Create a new table from a query

CREATE TABLE

nyctaxi.january_trips AS

SELECT *

FROM nyctaxi.2018trips

WHERE EXTRACT(Month FROM pickup_datetime)=1;

•

7. Step 7

a.

- To create a dataset
 - `bq --location=US mk -d hw2_dataset`

b.

- ```
((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq --location=us mk -d hw2 datasets
Dataset 'cnguyen4-cpsc324-hw2-415105:hw2' successfully created.
((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % gcloud storage buckets create gs://cnguyen4-cpsc324-hw2-bucket
Creating gs://cnguyen4-cpsc324-hw2-bucket/...

((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % gcloud storage cp jobs_in_data.csv gs://cnguyen4-cpsc324-hw2-bucket
Copying file://jobs_in_data.csv to gs://cnguyen4-cpsc324-hw2-bucket/jobs_in_data.csv
Completed files 1/1 | 1.1MiB/1.1MiB
((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % gcloud storage cp ufo_sightings.csv gs://cnguyen4-cpsc324-hw2-bucket
Copying file://ufo_sightings.csv to gs://cnguyen4-cpsc324-hw2-bucket/ufo_sightings.csv
Completed files 1/1 | 14.6MiB/14.6MiB

((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % gcloud storage ls gs://cnguyen4-cpsc324-hw2-bucket
gs://cnguyen4-cpsc324-hw2-bucket/jobs_in_data.csv
gs://cnguyen4-cpsc324-hw2-bucket/ufo_sightings.csv
```

c.

- ```
((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq load --source_format=CSV --autodetect hw2_datasets.jobs_in_data
gs://cnguyen4-cpsc324-hw2-bucket/jobs_in_data.csv

Waiting on bqjob_r514c5a359f850748_0000018dd7b5ffdc_1 ... (1s) Current status: DONE
((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq load --source_format=CSV --autodetect hw2_datasets.ufo_sightings
gs://cnguyen4-cpsc324-hw2-bucket/ufo_sightings.csv

Waiting on bqjob_r738971ec55aa7d59_0000018dd7b69e6c_1 ... (2s) Current status: DONE
((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq ls
datasetId
-----
babynames
hw2_datasets
((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq ls hw2_datasets
tableId      Type      Labels      Time Partitioning  Clustered Fields
-----
jobs_in_data  TABLE
ufo_sightings TABLE
```

- ```
((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq show hw2_datasets.jobs_in_data
Table cnguyen4-cpsc324-hw2-415105:hw2_datasets.jobs_in_data

Last modified Schema Total Rows Total Bytes Expiration Time Partitioning Clu
stored Fields Total Logical Bytes Total Physical Bytes Labels

23 Feb 12:44:07 |- work_year: integer 9355 1230158
1230158
|- job_title: string
|- job_category: string
|- salary_currency: string
|- salary: integer
|- salary_in_usd: integer
|- employee_residence: string
|- experience_level: string
|- employment_type: string
|- work_setting: string
|- company_location: string
|- company_size: string
```

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq show hw2_datasets.ufo_sightings
Table cnguyen4-cpsc324-hw2-415105:hw2_datasets.ufo_sightings
```

| Last modified    | Schema                         | Total Rows           | Total Bytes | Expiration | Time Partitioning | C |
|------------------|--------------------------------|----------------------|-------------|------------|-------------------|---|
| Clustered Fields | Total Logical Bytes            | Total Physical Bytes | Labels      |            |                   |   |
| 23 Feb 12:44:50  | 15979657                       | 88875                | 15979657    |            |                   |   |
|                  | - datetime: string             |                      |             |            |                   |   |
|                  | - city: string                 |                      |             |            |                   |   |
|                  | - state: string                |                      |             |            |                   |   |
|                  | - country: string              |                      |             |            |                   |   |
|                  | - shape: string                |                      |             |            |                   |   |
|                  | - duration__seconds_: float    |                      |             |            |                   |   |
|                  | - duration__hours_min_: string |                      |             |            |                   |   |
|                  | - comments: string             |                      |             |            |                   |   |
|                  | - date_posted: string          |                      |             |            |                   |   |
|                  | - latitude: string             |                      |             |            |                   |   |
|                  | - longitude: float             |                      |             |            |                   |   |
|                  | - int64_field_11: integer      |                      |             |            |                   |   |

d.

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq query "select salary_in_usd from hw2_datasets.jobs_in_data where employee_residence = 'United States' and experience_level = 'Senior' order by salary_in_usd DESC limit 5"
```

| salary_in_usd |
|---------------|
| 412000        |
| 405000        |
| 392000        |
| 385000        |
| 385000        |

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq query "select datetime, city, shape from hw2_datasets.ufo_sightings where country = 'de' and duration__seconds_>= 60.0 limit 10"
```

| datetime         | city                       | shape    |
|------------------|----------------------------|----------|
| 1/23/2001 15:30  | ansbach (germany)          | other    |
| 7/15/2007 00:45  | ansbach (germany)          | triangle |
| 12/21/2008 20:31 | neuss (germany)            | other    |
| 8/9/2008 22:50   | buchholz (germany)         | light    |
| 1/10/2011 18:38  | ottersberg (germany)       | triangle |
| 1/1/2009 00:15   | ramstein (germany)         | light    |
| 7/30/2008 01:30  | ramstein (germany)         | light    |
| 6/24/1974 14:40  | schweinfurt (west germany) | disk     |
| 5/7/1995 18:00   | emmelshausen (germany)     | unknown  |
| 6/17/2008 22:00  | dresden (germany)          | light    |

## 8. Step 8

a.

- Traditional database tries to achieve the highest normalization possible.
- However, with BigQuery, we try to denormalize it to make the operation more efficient.
- Array data type
  - SELECT ['raspberry', 'blackberry', 'strawberry', 'cherry'] AS fruit\_array

- Should have the same data type for all array items
- ARRAY\_AGG(): Function to aggregate our string values into an array
- SELECT fullVisitorId, date, v2ProductName, pageTitle

FROM `data-to-insights.ecommerce.all\_sessions`

WHERE visitId = 1501570398

ORDER BY date

| Row | fullVisitorId       | date     | v2ProductName                 | pageTitle                    |
|-----|---------------------|----------|-------------------------------|------------------------------|
| 1   | 5710379250208908569 | 20170731 | Google Snapback Hat Black     | Google RFID Journal          |
| 2   | 5710379250208908569 | 20170731 | Google Women's Lightwei...    | Google Snapback Hat Black    |
| 3   | 5710379250208908569 | 20170801 | Android Sticker Sheet Ultr... | Office   Google Merchandi... |
| 4   | 5710379250208908569 | 20170801 | 1 oz Hand Sanitizer           | Office   Google Merchandi... |
| 5   | 5710379250208908569 | 20170801 | Windup Android                | Accessories   Google Mer...  |

- SELECT fullVisitorId, date, ARRAY\_AGG(v2ProductName) AS

products\_viewed, ARRAY\_AGG(pageTitle) AS pages\_viewed

FROM `data-to-insights.ecommerce.all\_sessions`

WHERE visitId = 1501570398

GROUP BY fullVisitorId, date

ORDER BY date

| Row | fullVisitorId       | date     | products_viewed            | pages_viewed                 |
|-----|---------------------|----------|----------------------------|------------------------------|
| 1   | 5710379250208908569 | 20170731 | Google Snapback Hat Black  | Google RFID Journal          |
|     |                     |          | Google Women's Lightwei... | Google Snapback Hat Black    |
| 2   | 5710379250208908569 | 20170801 | Android Hard Cover Journal | Office   Google Merchandi... |
|     |                     |          | Suitcase Organizer Cubes   | Office   Google Merchandi... |
|     |                     |          | Waze Dress Socks           | Apparel   Google Merchan...  |
|     |                     |          | Google Flashlight          | Accessories   Google Mer...  |
|     |                     |          | Plastic Sliding Flashlight | Accessories   Google Mer...  |
|     |                     |          | Android Hard Cover Journal | Shop by Brand   Google M...  |

- ARRAY\_LENGTH(): counts the number of pages and products that were viewed
- SELECT

fullVisitorId,

date,

ARRAY\_AGG(v2ProductName) AS products\_viewed,

ARRAY\_LENGTH(ARRAY\_AGG(v2ProductName)) AS

num\_products\_viewed,

ARRAY\_AGG(pageTitle) AS pages\_viewed,

ARRAY\_LENGTH(ARRAY\_AGG(pageTitle)) AS num\_pages\_viewed

FROM `data-to-insights.ecommerce.all\_sessions`

WHERE visitId = 1501570398

GROUP BY fullVisitorId, date

ORDER BY date

| Row | fullVisitorId       | date     | products_viewed               | num_products_viewed | pages_viewed                 |
|-----|---------------------|----------|-------------------------------|---------------------|------------------------------|
| 1   | 5710379250208908569 | 20170731 | Google Snapback Hat Black     | 2                   | Google RFID Journal          |
|     |                     |          | Google Women's Lightwei...    |                     | Google Snapback Hat Black    |
| 2   | 5710379250208908569 | 20170801 | Android Sticker Sheet Ultr... | 109                 | Office   Google Merchandi... |
|     |                     |          | 1 oz Hand Sanitizer           |                     | Office   Google Merchandi... |
|     |                     |          | Windup Android                |                     | Accessories   Google Mer...  |
|     |                     |          | Compact Selfie Stick          |                     | Accessories   Google Mer...  |

- finding the number of elements with ARRAY\_LENGTH(<array>)
- deduplicating elements with ARRAY\_AGG(DISTINCT <field>)
- ordering elements with ARRAY\_AGG(<field> ORDER BY <field>)
- limiting ARRAY\_AGG(<field> LIMIT 5)

•

- Before we can access a repeated (type) array normally, we need to break it into

rows

For example, the array for `hits.page.pageTitle` is stored currently as a single row like:

```
['homepage', 'product page', 'checkout']
```



and it needs to be:

```
['homepage',
 'product page',
 'checkout']
```



- 
- UNNEST(): Bring the array element back into rows
  - Always follow the table name in the FROM clause
- STRUCT (RECORD data type): think of it as a separate table that is pre-joined into your main table
  - Can have:
    - One or many fields
    - Same or different data types
    - Its own alias
  - A struct can have another STRUCT as one of its field
  - totals.\*,: Return all fields for that STRUCT
- SELECT STRUCT("Rudisha" as name, 23.4 as split) as runner

| Row | runner.name | runner.split |
|-----|-------------|--------------|
| 1   | Rudisha     | 23.4         |

- 
- SELECT STRUCT("Rudisha" as name, [23.4, 26.3, 26.4, 26.1] as splits) AS runner



| Row | runner.name | runner.splits |
|-----|-------------|---------------|
| 1   | Rudisha     | 23.4          |
|     |             | 26.3          |
|     |             | 26.4          |
|     |             | 26.1          |

- Assuming, in this case, we need to perform a CROSS JOIN to populate the data across all fields.

| Row | race | participants.name |
|-----|------|-------------------|
| 1   | 800M | Rudisha           |
| 2   | ???  | Makhloufi         |
| 3   | ???  | Murphy            |

...to this:

| Row | race | participants.name |
|-----|------|-------------------|
| 1   | 800M | Rudisha           |
| 2   | 800M | Makhloufi         |
| 3   | 800M | Murphy            |

```

1 -- correct sql
2 SELECT race, participants.name
3 FROM racing.race_results
4 CROSS JOIN
5 race_results.participants # must use the referenced name

```

- You can also use comma join

Recap of STRUCTs:

- A SQL [STRUCT](#) is simply a container of other data fields which can be of different data types. The word struct means data structure. Recall the example from earlier: `STRUCT(`"Rudisha" as name, [23.4, 26.3, 26.4, 26.1] as splits`)` AS runner
- STRUCTs are given an alias (like runner above) and can conceptually be thought of as a table inside of your main table.
- STRUCTs (and ARRAYs) must be unpacked before you can operate over their elements. Wrap an UNNEST() around the name of the struct itself or the struct field that is an array in order to unpack and flatten it.

•

- `SELECT COUNT(p.name) AS racer_count`

`FROM racing.race_results AS r, UNNEST(r.participants) AS p`

- Note how we need to UNNEST participant first before doing a comma join

- `SELECT`

`p.name,`

`SUM(split_times) as total_race_time`

`FROM racing.race_results AS r, UNNEST(r.participants) AS p,`

`UNNEST(p.splits) AS split_times`

`WHERE p.name LIKE 'R%'`

`GROUP BY p.name`

`ORDER BY total_race_time ASC;`

- Note the LIKE construct: find words that start with an R

c.

- Positive: arrays and structs allow us to represent more flexible data modeling since we can now represent more information within a single table. I also think it

will give the user a friendlier look at the table since everything is grouped nicely.

At the same time, it also enables us to perform more complicated queries that are faster and more efficient since fewer joint operations need to be done, which is less expensive.

- Negative: Though faster, querying a table with arrays and structs can be more challenging since we now have to deal with nested data structure, which is not initially designed to be handled by SQL. We probably have to perform additional join statements to propagate grouped data labels across their own instances. Also it also needs to factor in cost consideration when more computing powers may be needed to process the nested structure.

d.

- The NULL values mean undefined data, not necessarily missing data or empty values. In this case, an array column or its elements can be NULL.
- NULL requires special handling through functions. Because of that, developing and maintaining data structures that have NULL values is more tricky and expensive.
- NULL values in a data structure make it harder to debug in general. We have to deal with more complicated issues, not to mention data integrity.
- Besides, there is a high possibility of unexpected behaviors when performing a query, as NULL values are often treated differently by functions.

## 9. Step 9

a.

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % unzip bikedata_10000.zip
Archive: bikedata_10000.zip
 inflating: bikedata_10000.json
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % unzip Video_Games_10000.zip
Archive: Video_Games_10000.zip
 inflating: Video_Games_10000.json
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % gcloud storage cp bikedata_10000.json gs://cnguyen4-cpsc324-hw2-bucket
Copying file://bikedata_10000.json to gs://cnguyen4-cpsc324-hw2-bucket/bikedata_10000.json
Completed files 1/1 | 6.4MiB/6.4MiB
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % gcloud storage cp Video_Games_10000.json gs://cnguyen4-cpsc324-hw2-bucket
Copying file://Video_Games_10000.json to gs://cnguyen4-cpsc324-hw2-bucket/Video_Games_10000.json
Completed files 1/1 | 10.0MiB/10.0MiB
```

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % gcloud storage ls gs://cnguyen4-cpsc324-hw2-bucket
gs://cnguyen4-cpsc324-hw2-bucket/Video_Games_10000.json
gs://cnguyen4-cpsc324-hw2-bucket/bikedata_10000.json
gs://cnguyen4-cpsc324-hw2-bucket/jobs_in_data.csv
gs://cnguyen4-cpsc324-hw2-bucket/ufo_sightings.csv
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq load --source_format=NEWLINE_DELIMITED_JSON --autodetect hw2_data
sets.bikedata gs://cnguyen4-cpsc324-hw2-bucket/bikedata_10000.json

Waiting on bqjob_r6b361ddf5a5c4226_0000018dd93918c5_1 ... (2s) Current status: DONE
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq load --source_format=NEWLINE_DELIMITED_JSON --autodetect hw2_data
sets.Video_Games gs://cnguyen4-cpsc324-hw2-bucket/Video_Games_10000.json

Waiting on bqjob_r6d11b43ad6044f63_0000018dd9397d89_1 ... (2s) Current status: DONE
```

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq ls
datasetId

babynames
hw2_datasets
(cpsc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq ls hw2_datasets
tableId Type Labels Time Partitioning Clustered Fields

Video_Games TABLE
bikedata TABLE
jobs_in_data TABLE
ufo_sightings TABLE
```

d.

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq show hw2_datasets.bikedata
Table cnguyen4-cpsc324-hw2-415105:hw2_datasets.bikedata
```

| Last modified   | Schema                                                                                                                                                                                                                                                                                                                                                                                                                   | Total Rows | Total Bytes | Expiration | Time Partitioning |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|-------------|------------|-------------------|
| 23 Feb 19:46:57 | <pre> cycle_type_slug: string year: integer url: string stolen_location: string stolen_coordinates: float (repeated) stolen: boolean registry_url: string registry_name: string propulsion_type_slug: string is_stock_img: boolean manufacturer_name: string title: string large_img: string external_id: string serial: string id: integer frame_colors: string (repeated) status: string location_found: string </pre> | 10000      | 3240506     |            |                   |

```

|- date_stolen: integer
|- frame_model: string
|- thumb: string
|- description: string

```

```

((cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq show hw2_datasets.Video_Games
Table cnguyen4-cpsc324-hw2-415105:hw2_datasets.Video_Games

 Last modified Schema Total Rows Total Bytes Expiration Time Partitioning Clust
ered Fields Total Logical Bytes Total Physical Bytes Labels

23 Feb 19:47:23 |- image: string (repeated) 10000 8952695
8952695 3456774
|- unixReviewTime: integer
|- vote: integer
+- style: record
| |- Color: string
| |- Edition: string
| |- Platform: string
| |- Format: string
|- summary: string
|- reviewText: string
|- reviewerName: string
|- overall: float
|- asin: string
|- reviewTime: string
|- reviewerID: string
|- verified: boolean

```

- Fields that are non-1NF (repeated or records):

- bikedata Dataset
  - stolen\_coordinates
  - frame\_colors
- Videos\_Games dataset
  - image
  - style

e.

- Bikedata Dataset

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq query --use_legacy_sql=false "select f, count(*) as bike_count from hw2_datasets.bikedata as b, UNNEST (b.frame_colors) as f group by f"
```

| f                               | bike_count |
|---------------------------------|------------|
| Blue                            | 1721       |
| Yellow or Gold                  | 352        |
| Silver, gray or bare metal      | 1780       |
| Black                           | 3813       |
| Orange                          | 473        |
| Teal                            | 302        |
| Green                           | 941        |
| White                           | 999        |
| Red                             | 970        |
| Brown                           | 176        |
| Purple                          | 309        |
| Stickers tape or other cover-up | 51         |
| Pink                            | 145        |

○

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq query --use_legacy_sql=false "select count(*) as NumOfBike from hw2_datasets.bikedata where ARRAY_LENGTH(frame_colors) > 1"
```

| NumOfBike |
|-----------|
| 1760      |

○

- For some reason, it needs the `--use_legacy_sql=false` flag

- Video\_Games dataset

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq query --use_legacy_sql=false "select style.Platform, style.Edition, COUNT(*) as r_count from hw2_datasets.Video_Games where style.Platform is not null and style.Edition is not null group by style.Platform, style.Edition order by r_count DESC"
```

| Platform     | Edition          | r_count |
|--------------|------------------|---------|
| PC           | Standard         | 91      |
| PC/Mac       | Standard         | 36      |
| Sega Genesis | Special Champion | 22      |
| Mac          | Standard         | 5       |
| Mac          | Gold             | 4       |
| PlayStation  | Standard         | 4       |
| PC           | DVD-Rom          | 3       |
| PC Download  | Standard         | 3       |

○

```
(cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % bq query --use_legacy_sql=false "select style.Color, AVG(overall) as avg_rating from hw2_datasets.Video_Games where style.Color is not null group by style.Color order by avg_rating DESC"
```

| Color            | avg_rating        |
|------------------|-------------------|
| Yellow           | 5.0               |
| Green            | 4.8               |
| Kiwi             | 4.705882352941177 |
| Teal             | 4.666666666666668 |
| Grape            | 4.571428571428571 |
| Atomic Purple    | 4.468750000000001 |
| Dandelion        | 4.333333333333332 |
| Translucent Blue | 3.333333333333333 |

○

g.

- Bikedata dataset

```
cnguyen4-cpsc324-hw2-415105> query --use_legacy_sql=false 'select count(*) as NumOfBike from hw2_datasets.bikedata where ARRAY_LENGTH(frame_colors) > 1'
```

| NumOfBike |
|-----------|
| 1760      |

○

```
cnguyen4-cpsc324-hw2-415105> query --use_legacy_sql=false 'select f, count(*) as bike_count from hw2_datasets.bikedata as b, UNNEST (b.frame_colors) as f group by f'
```

| f                               | bike_count |
|---------------------------------|------------|
| Blue                            | 1721       |
| Yellow or Gold                  | 352        |
| Silver, gray or bare metal      | 1780       |
| Black                           | 3813       |
| Orange                          | 473        |
| Teal                            | 302        |
| Green                           | 941        |
| White                           | 999        |
| Red                             | 970        |
| Brown                           | 176        |
| Purple                          | 309        |
| Stickers tape or other cover-up | 51         |
| Pink                            | 145        |

- Video\_Games dataset

```
cnguyen4-cpsc324-hw2-415105> query --use_legacy_sql=false "select style.Platform, style.Edition, COUNT(*) as r_count from hw2_datasets.Video_Games where style.Platform is not null and style.Edition is not null group by style.Platform, style.Edition order by r_count DESC"
```

| Platform     | Edition          | r_count |
|--------------|------------------|---------|
| PC           | Standard         | 91      |
| PC/Mac       | Standard         | 36      |
| Sega Genesis | Special Champion | 22      |
| Mac          | Standard         | 5       |
| Mac          | Gold             | 4       |
| PlayStation  | Standard         | 4       |
| PC           | DVD-Rom          | 3       |
| PC Download  | Standard         | 3       |

```
cnguyen4-cpsc324-hw2-415105> query --use_legacy_sql=false "select style.Color, AVG(overall) as avg_rating from hw2_datasets.Video_Games where style.Color is not null group by style.Color order by avg_rating DESC"
```

| Color            | avg_rating        |
|------------------|-------------------|
| Yellow           | 5.0               |
| Green            | 4.8               |
| Kiwi             | 4.705882352941177 |
| Teal             | 4.666666666666668 |
| Grape            | 4.571428571428571 |
| Atomic Purple    | 4.468750000000001 |
| Dandelion        | 4.333333333333332 |
| Translucent Blue | 3.333333333333333 |

## 10. Step 10

a.

Untitled RUN SAVE DOWNLOAD SHARE SCHEDULE MORE Query completed.

```

1 #standardSQL
2 SELECT *
3 FROM `data-to-insights.ecommerce.partition_by_day`
4 WHERE date_formatted = '2016-08-01'

```

Press Option+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

| JOB INFORMATION | RESULTS        | CHART               | JSON | EXECUTION DETAILS | EXECUTION GRAPH |
|-----------------|----------------|---------------------|------|-------------------|-----------------|
| Row             | date_formatted | fullvisitorid       |      |                   |                 |
| 1               | 2016-08-01     | 8346614539128137085 |      |                   |                 |
| 2               | 2016-08-01     | 1856237131266550302 |      |                   |                 |
| 3               | 2016-08-01     | 8422029627538180622 |      |                   |                 |
| 4               | 2016-08-01     | 7298538238612725446 |      |                   |                 |
| 5               | 2016-08-01     | 8271170844108113200 |      |                   |                 |
| 6               | 2016-08-01     | 6030957980134486247 |      |                   |                 |

Results per page: 50 1 - 50 of 888

Untitled RUN SAVE DOWNLOAD SHARE SCHEDULE MORE Query completed.

```

1 #standardSQL
2 SELECT *
3 FROM `data-to-insights.ecommerce.partition_by_day`
4 WHERE date_formatted = '2018-07-08'

```

Press Option+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

| JOB INFORMATION              | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH |
|------------------------------|---------|-------|------|-------------------|-----------------|
| There is no data to display. |         |       |      |                   |                 |

## b. Notes

- We create a partition table in order to divide our original data into smaller partitions, which can be used to query against. It makes it easier, faster, and more efficient to query as we don't have to scan through the whole table anymore. It also reduces costs.



```
#standardSQL
CREATE OR REPLACE TABLE ecommerce.partition_by_day
PARTITION BY date_formatted
OPTIONS(
 description="a table partitioned by date"
) AS

SELECT DISTINCT
PARSE_DATE("%Y%m%d", date) AS date_formatted,
fullvisitorId
FROM `data-to-insights.ecommerce.all_sessions_raw`
```

- - Note that the Partition By function has two options
    - Date\_formatted
    - Timestamp
  - PARSE\_DATE function: get the proper DATE type for partitioning

```
#standardSQL
CREATE or REPLACE TABLE ecommerce.days_with_rain
PARTITION BY date
OPTIONS (
 partition_expiration_days=60,
 description="weather stations with precipitation, partitioned by day"
) AS

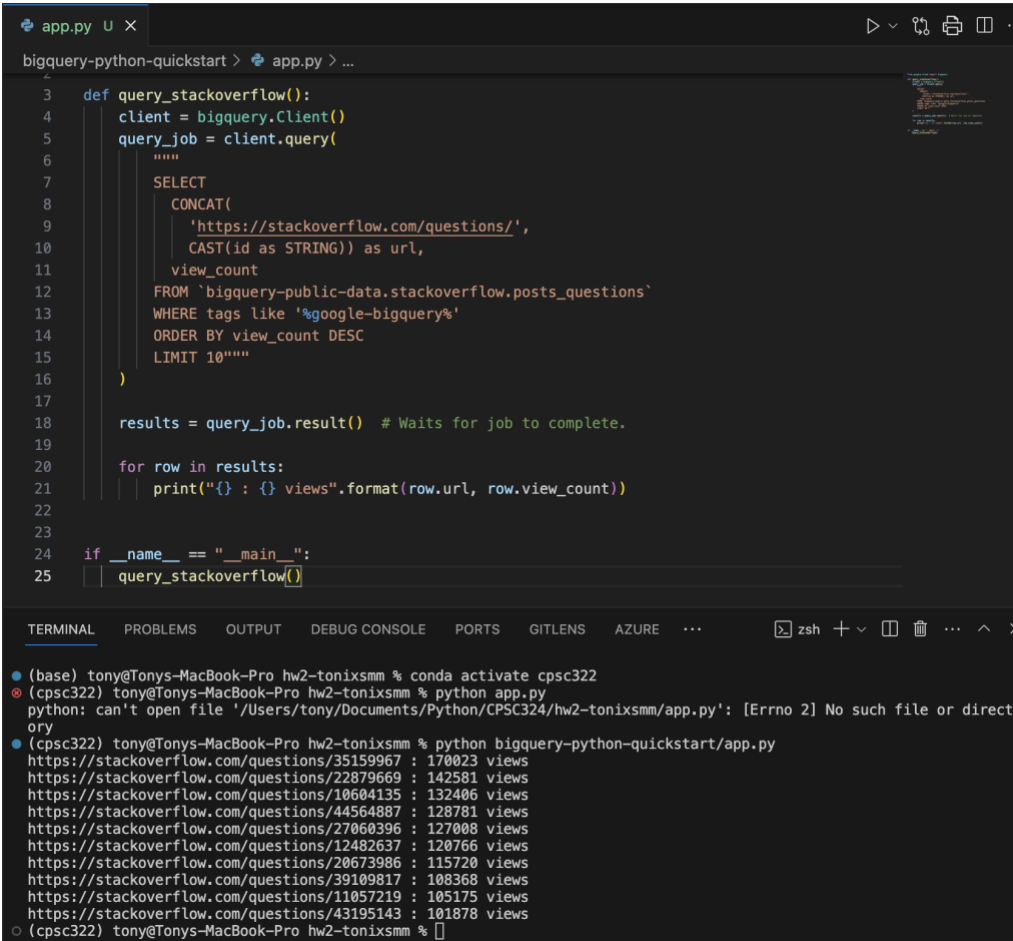
SELECT
 DATE(CAST(year AS INT64), CAST(mo AS INT64), CAST(da AS INT64)) AS date,
 (SELECT ANY_VALUE(name) FROM `bigquery-public-data.noaa_gsod.stations` AS stations
 WHERE stations.usaf = stn) AS station_name, -- Stations may have multiple names
 prcp
FROM `bigquery-public-data.noaa_gsod.gsod*` AS weather
WHERE prcp < 99.9 -- Filter unknown values
 AND prcp > 0 -- Filter stations/days with no precipitation
 AND _TABLE_SUFFIX >= '2018'
```

- Query to check when did this data table is stored

```
#standardSQL
avg monthly precipitation
SELECT
 AVG(prcp) AS average,
 station_name,
 date,
 CURRENT_DATE() AS today,
 DATE_DIFF(CURRENT_DATE(), date, DAY) AS partition_age,
 EXTRACT(MONTH FROM date) AS month
FROM ecommerce.days_with_rain
WHERE station_name = 'WAKAYAMA' #Japan
GROUP BY station_name, date, today, month, partition_age
ORDER BY date DESC; # most recent days first
```

○

## 11. Step 11



```
app.py U X
bigquery-python-quickstart > app.py > ...

3 def query_stackoverflow():
4 client = bigquery.Client()
5 query_job = client.query(
6 """
7 SELECT
8 CONCAT(
9 'https://stackoverflow.com/questions/',
10 CAST(id as STRING)) as url,
11 view_count
12 FROM `bigquery-public-data.stackoverflow.posts_questions`
13 WHERE tags like '%google-bigquery%'
14 ORDER BY view_count DESC
15 LIMIT 10"""
16)
17
18 results = query_job.result() # Waits for job to complete.
19
20 for row in results:
21 print("{} : {} views".format(row.url, row.view_count))
22
23
24 if __name__ == "__main__":
25 query_stackoverflow()
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE PORTS GITLENS AZURE ... zsh + - □ □ ... ^ X

```

• (base) tony@Tonys-MacBook-Pro hw2-tonixsmm % conda activate cp322
• (cp322) tony@Tonys-MacBook-Pro hw2-tonixsmm % python app.py
python: can't open file '/Users/tony/Documents/Python/CP324/hw2-tonixsmm/app.py': [Errno 2] No such file or direct
ory
• (cp322) tony@Tonys-MacBook-Pro hw2-tonixsmm % python bigquery-python-quickstart/app.py
https://stackoverflow.com/questions/35159967 : 170023 views
https://stackoverflow.com/questions/22879669 : 142581 views
https://stackoverflow.com/questions/10604135 : 132406 views
https://stackoverflow.com/questions/44564887 : 128781 views
https://stackoverflow.com/questions/27060396 : 127008 views
https://stackoverflow.com/questions/12482637 : 120766 views
https://stackoverflow.com/questions/20673986 : 115720 views
https://stackoverflow.com/questions/39109817 : 108368 views
https://stackoverflow.com/questions/11057219 : 105175 views
https://stackoverflow.com/questions/43195143 : 101878 views
• (cp322) tony@Tonys-MacBook-Pro hw2-tonixsmm %
```

## 12. Step 12

```

hw2_bq_params.py
1 import (module) google
2 from google.cloud import bigquery
3
4 # Construct a BigQuery client object.
5 client = bigquery.Client()
6
7 county_param = input("Enter the county: ")
8 state_param = input("Enter the state: ")
9 year_param = input("Enter the year: ")
10 month_param = input("Enter the month: ")
11 day_param = input("Enter the day: ")
12
13 query = """
14 SELECT TIMESTAMP_ADD(@ts_value, INTERVAL 0 SECOND) as ts_value, county, state_name, confirmed_cases, death
15 FROM `bigquery-public-data.covid19_nyt.us_counties`
16 WHERE TIMESTAMP(date) = @ts_value AND county = @county AND state_name = @state_name;
17 """
18
19 job_config = bigquery.QueryJobConfig(
20 query_parameters=[
21 bigquery.ScalarQueryParameter("ts_value", "TIMESTAMP", datetime.datetime(int(year_param), int(month_param),
22 bigquery.ScalarQueryParameter("county", "STRING", county_param),
23 bigquery.ScalarQueryParameter("state_name", "STRING", state_param),
24]
25)
26 query_job = client.query(query, job_config=job_config) # Make an API request.
27
28 for row in query_job:
29 print("Day: {} \n County: {} \n State Name: {} \n Confirmed case: {} \n Deaths: {}".format(row.ts_value, row.c

```

```

• (cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % python hw2_bq_params.py
Enter the county: King
Enter the state: Washington
Enter the year: 2022
Enter the month: 01
Enter the day: 10
Day: 2022-01-10 00:00:00+00:00
County: King
State Name: Washington
Confirmed case: 248777
Deaths: 2198
• (cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % python hw2_bq_params.py
Enter the county: Spokane
Enter the state: Washington
Enter the year: 2021
Enter the month: 12
Enter the day: 31
Day: 2021-12-31 00:00:00+00:00
County: Spokane
State Name: Washington
Confirmed case: 81271
Deaths: 1152
• (cpssc322) tony@Tonys-MacBook-Pro hw2-tonixsmm % python hw2_bq_params.py
Enter the county: Harris
Enter the state: Texas
Enter the year: 2021
Enter the month: 12
Enter the day: 29
Day: 2021-12-29 00:00:00+00:00
County: Harris
State Name: Texas
Confirmed case: 644647
Deaths: 9731

```