

CIS 9440  
Section Number: UWA 32009  
Homework #1

**Student name:** Jason Jiang

**Student ID:** 23435668


**Date:** 9/18/2022

1. Follow the instructions on Blackboard to create a new project in Google BigQuery  
Once complete, take a screenshot of the results of any query you write.



**ANSWER:**

```
1 SELECT
2 |   DISTINCT(Champion)
3 FROM `cis9440-361100.nba_data.nba_finals_teams`;
```

Press Alt+F1 for Access

Query results [SAVE RESULTS](#) 

JOB INFORMATION		RESULTS	JSON	EXPORT
Row	Champion			
1	Miami Heat			
2	Chicago Bulls			
3	Boston Celtics			
4	Detroit Pistons			
5	Houston Rockets			
6	Toronto Raptors			
7	Dallas Mavericks			
8	San Antonio Spurs			
9	Los Angeles Lakers			
10	Cleveland Cavaliers			
11	Golden State Warriors			

Results per page: 50 ▼ 1 – 11 of 11  

2. Take a screenshot of your results and post them as the answer to this question.

ANSWER:

```
1 SELECT
2 | address
3 FROM `bigquery-public-data.new_york_trees.tree_census_2015`
4 LIMIT 100;
```

Press Alt+F1 for Acc

### Query results

[SAVE RESULTS](#) [EXPLORE DATA](#)


JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	address			
1	63 CRANBERRY STREET			
2	28 OLD FULTON STREET			
3	45 CRANBERRY STREET			
4	87 COLUMBIA HEIGHTS			
5	97 COLUMBIA HEIGHTS			
6	87 COLUMBIA HEIGHTS			
7	79 COLUMBIA HEIGHTS			
8	97 COLUMBIA HEIGHTS			
9	73 COLUMBIA HEIGHTS			
10	87 COLUMBIA HEIGHTS			
11	91 COLUMBIA HEIGHTS			
12	81 COLUMBIA HEIGHTS			

Results per page: 50 1 – 50 of 100

3. Continuing to use the “new\_york\_trees” dataset, write a query to find the top 5 most common trees in the “tree\_census\_2015” table. More specifically, you are looking for the top 5 most common “spc\_common” in the table.

ANSWER:

```
1 SELECT
2   spc_common,
3   COUNT(spc_common) AS total_occurance
4 FROM `bigquery-public-data.new_york_trees.tree_census_2015`
5 GROUP BY spc_common
6 ORDER BY COUNT(spc_common) DESC
7 LIMIT 5;
```

Query results [SAVE RESULTS](#) 

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	spc_common	total_occura...		
1	London planetree	87014		
2	honeylocust	64264		
3	Callery pear	58931		
4	pin oak	53185		
5	Norway maple	34189		

4. Continuing to use the “new\_york\_trees” dataset, write a query to find the average tree diameter of trees in “Good” health by Borough in the “tree\_census\_2015” table.

For more details, in the “tree\_census\_2015” table the tree diameter is in column “tree\_dbh”, tree health is in column “health”, and Boroughs are in column “boroname”.

ANSWER:

```
1 SELECT
2   boroname,
3   AVG(tree_dbh) AS avg_tree_diameter
4 FROM `bigquery-public-data.new_york_trees.tree_census_2015`
5 WHERE health = "Good"
6 GROUP BY boroname;
```

Press A

Query results [SAVE RESULTS](#) [EX](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	boroname	avg_tree_diameter		
1	Brooklyn	12.48599253320965		
2	Queens	13.299946393963...		
3	Bronx	10.34126090416345		
4	Manhattan	9.061066768022302		
5	Staten Island	11.005757901994...		

5. Continuing to use the “new\_york\_trees” dataset and the “tree\_census\_2015” table, write a query to find the common name of the tree with the largest tree diameter in the Borough of “Brooklyn”.

For more details, in the “tree\_census\_2015” table the tree diameter is in column “tree\_dbh”, tree common name is in column “spc\_common”, and Boroughs are in column “boroname”.

ANSWER:

```
1 SELECT
2   spc_common,
3   MAX(tree_dbh) AS largest_tree_diameter
4 FROM `bigquery-public-data.new_york_trees.tree_census_2015`
5 WHERE boroname = "Brooklyn"
6 GROUP BY spc_common
7 ORDER BY MAX(tree_dbh) DESC
8 LIMIT 1;
```

Query results

 SAVE RESULTS ▾

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	spc_common	largest_tree...		
1	swamp white oak	425		

6. Continuing to use the “new\_york\_trees” dataset and the “tree\_census\_2015” table, write a query to determine which “curb\_loc” has the largest average “tree\_dbh”.

For more details, in the “tree\_census\_2015” table the tree diameter is in column “tree\_dbh”, tree common name is in column “spc\_common”, and Boroughs are in column “boroname”.

ANSWER:

```
1 SELECT
2   curb_loc,
3   AVG(tree_dbh) AS average_tree_diameter
4 FROM `bigquery-public-data.new_york_trees.tree_census_2015`
5 GROUP BY curb_loc
6 ORDER BY AVG(tree_dbh) DESC
7 LIMIT 1;
```

## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	curb_loc	average_tree_diameter		
1	OffsetFromCurb	13.392793395805501		

7. Continuing to use the “new\_york\_trees” dataset and the “tree\_census\_2015” table, write a query to determine the zip code with the most trees in “Good” health.

For more details, in the “tree\_census\_2015” table the tree diameter is in column “tree\_dbh”, tree common name is in column “spc\_common”, and Boroughs are in column “boroname”.

ANSWER:

```
1 SELECT
2   |   zipcode,
3   |   COUNT(health) AS total_trees_in_good_health
4 FROM `bigquery-public-data.new_york_trees.tree_census_2015`
5 WHERE health = "Good"
6 GROUP BY zipcode
7 ORDER BY COUNT(health) DESC
8 LIMIT 1;
```

## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	zipcode	total_trees_in_good_health		
1	10312	16691		

For more details, in the “tree\_census\_2015” table the tree diameter is in column “tree\_dbh”, tree common name is in column “spc\_common”, and Boroughs are in column “boroname”.

RUN
 SAVE ▾
 SHARE ▾
 SCHEDULE ▾
 MORE ▾

```

1 SELECT
2   guards,
3   health,
4   COUNT(health) AS total_counts_by_health
5 FROM `bigquery-public-data.new_york_trees.tree_census_2015`
6 WHERE guards IN ("Helpful","None","Harmful","Unsure")
7    AND health <> "null"
8 GROUP BY guards, health
9 ORDER BY guards DESC;
```

Pres

---

### Query results SAVE RESULTS ▾

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS
Row	guards	health	total_counts...
1	Unsure	Good	5912
2	Unsure	Fair	1376
3	Unsure	Poor	460
4	None	Fair	84123
5	None	Good	464978
6	None	Poor	23204
7	Helpful	Good	42638
8	Helpful	Fair	7166
9	Helpful	Poor	2062
10	Harmful	Good	15322
11	Harmful	Fair	3839
12	Harmful	Poor	1091

Results per page: 50 ▾      1 – 12 of 12



Below is to help with the analysis in question #8 to make sure the counts add up

RUN

SAVE

SHARE

SCHEDULE

This

```
1 SELECT
2   health,
3   COUNT(health) AS total_counts_by_health
4 FROM `bigquery-public-data.new_york_trees.tree_census_2015`
5 WHERE guards IN ("Helpful", "None", "Harmful", "Unsure")
6    AND health <> "null"
7 GROUP BY health;
```

Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTS

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

Row	health	total_health	
1	Good	528850	
2	Fair	96504	
3	Poor	26817	

### Explanation to Question #8

Guards **do not** help improve the tree's health because the majority - about 87% ( $\frac{464,978}{528,850}$ ) of trees with good health do not have guards at all.

9. Continuing to use the “new\_york\_trees” dataset and the “tree\_census\_2015” table, write a query to determine the most common “user\_type” for trees that are “London planetree”.

For more details, in the “tree\_census\_2015” table the tree diameter is in column “tree\_dbh”, tree common name is in column “spc\_common”, and Boroughs are in column “boroname”.

ANSWER:

▶ RUN 📄 SAVE ▼ 👤 SHARE ▼ 🕒 SCHEDULE ▼ ⚙️ MORE ▼

```
1 SELECT
2   user_type,
3   spc_common,
4   COUNT(user_type) AS total_count
5 FROM `bigquery-public-data.new_york_trees.tree_census_2015`
6 WHERE spc_common = "London planetree"
7 GROUP BY user_type, spc_common
8 ORDER BY COUNT(user_type) DESC
9 LIMIT 1;
```

## Query results






JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	
Row	user_type	spc_common	total_count		
1	TreesCount Staff	London planetree	37406		

10. Now, we will move onto a different Google Public Dataset. Find the “chicago\_taxi\_trips” dataset in the “bigquery-public-data” project.

In the table “taxi\_trips”, write a query to find the average tip left by taxi riders that paid with a “Credit Card” for rides that were longer than 15 minutes.

For more details, the tip is in the “tips” column, the payment type is in the “payment\_type” column, and the trip duration is in the “trip\_seconds” column.

ANSWER:

 RUN  SAVE  SHARE  SCHEDULE  MORE

```
1 SELECT
2   payment_type,
3   AVG(tips) AS average_tip
4 FROM `bigquery-public-data.chicago_taxi_trips.taxi_trips`
5 WHERE payment_type = "Credit Card"
6   AND trip_seconds > 900
7 GROUP BY payment_type;
```

## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	payment_type	average_tip		
1	Credit Card	6.0177381559671161		

11. Still using the “taxi\_trips” table in the “chicago\_taxi\_trips” dataset,

Write a query to find the payment type that resulted in the largest average tip for rides that were longer than 10 minutes and between 5 and 10 miles.

For more details, the tip is in the “tips” column, the payment type is in the “payment\_type” column, the trip duration is in the “trip\_seconds” column, and the trip distance is in the “trip\_miles” column.

ANSWER:

```
1 SELECT
2   payment_type,
3   AVG(tips) AS average_tips
4 FROM `bigquery-public-data.chicago_taxi_trips.taxi_trips`
5 /*Assuming non-inclusive of 5 and 10
6 If we want to be inclusive I would use BETWEEN here*/
7 WHERE (trip_miles > 5 AND trip_miles < 10)
8   AND trip_seconds > 600
9 GROUP BY payment_type
10 ORDER BY AVG(tips) DESC
11 LIMIT 1;
```

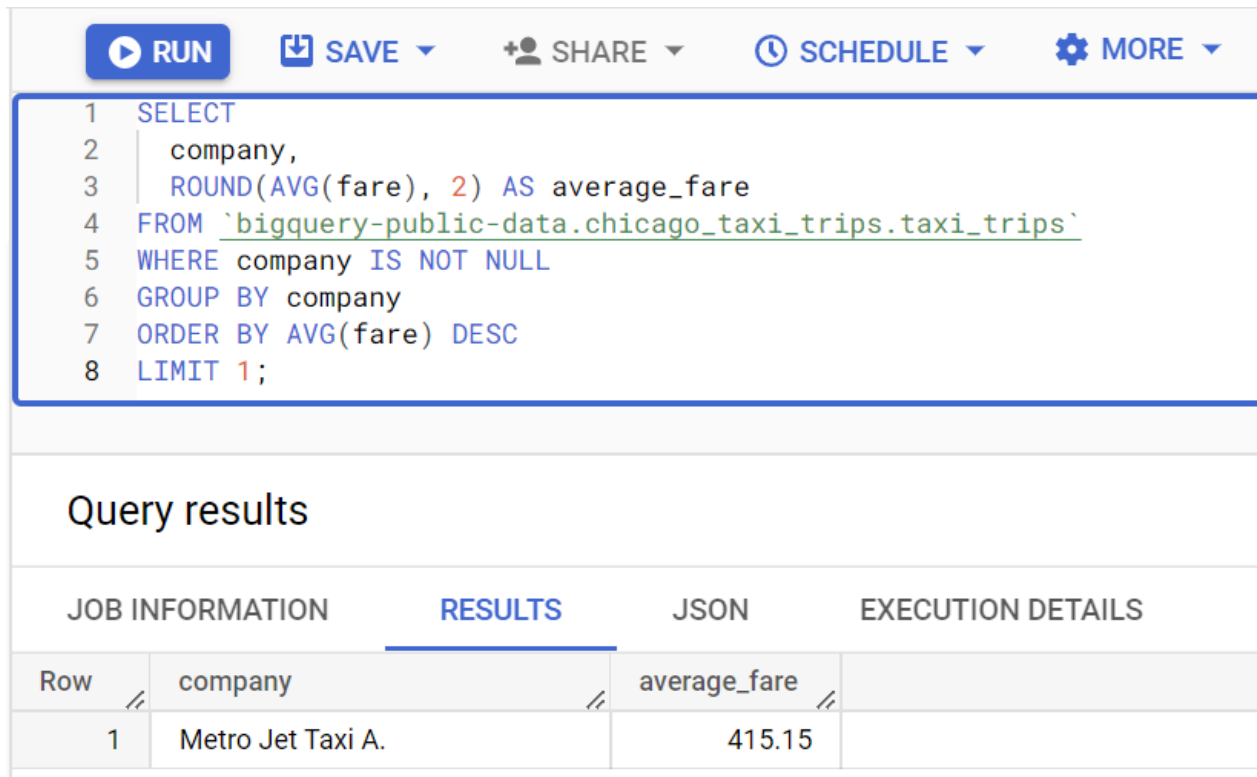
## Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	payment_type	average_tips		
1	Mobile	3.9795541636739995		

12. Still using the “taxi\_trips” table in the “chicago\_taxi\_trips” dataset,

Use a SQL query to find the most expensive taxi “company”. You choose how to define “expensive”. Please paste a screenshot of your query, your query results, and your explanation of “expensive” as your answer.

ANSWER:



The screenshot shows a BigQuery interface with a SQL query editor at the top and a results table below. The query is designed to find the company with the highest average fare from the 'chicago\_taxi\_trips' dataset. The results table shows one row: 'Metro Jet Taxi A.' with an average fare of 415.15.

```
1 SELECT
2   company,
3   ROUND(AVG(fare), 2) AS average_fare
4 FROM `bigquery-public-data.chicago_taxi_trips.taxi_trips`
5 WHERE company IS NOT NULL
6 GROUP BY company
7 ORDER BY AVG(fare) DESC
8 LIMIT 1;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	company	average_fare		
1	Metro Jet Taxi A.	415.15		

### Explanation to Question #12

Metro Jet Taxi A. is the most expensive taxi company because it has an average fare cost of about \$415.15. I decided to use “fares” instead of the other metrics (tips, tolls, extras, and trip\_total) because these other metrics are influenced by other factors outside of the company’s control. Fares gives us just the amount charged by the company.

- Write a SQL query to find the “title”, “tags”, “view\_count”, and “score” of the 5 posts with the highest “favorite\_count” in the table “stackoverflow\_posts”.**

```
1 SELECT
2     title,
3     tags,
4     view_count,
5     score,
6     favorite_count
7 FROM `bigquery-public-data.stackoverflow.stackoverflow_posts`
8 ORDER BY favorite_count DESC
9 LIMIT 5;
```


JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		
Row	title	tags	view_count	score	favorite_cou...	
1	Why is processing a sorted array faster than an unsorted array?	java c++ performance optimiza...	805490	14772	7317	
2	The Definitive C++ Book Guide ...	c++ c++-faq	1275671	4265	6824	
3	What is the single most influential book every programmer should read?	resources	831805	1440	6258	
4	Thinking in AngularJS if I have ...	javascript jquery angularjs desi...	636707	4536	5262	
5	What is the best comment in source code you have ever encountered?	comments	2376668	360	5114	


14. Continue in the “stackoverflow” dataset in the “bigquery-public-data” project.


Write a SQL query to find the 10 most viewed posts regarding BigQuery.


Hint: leverage the LIKE operator.


ANSWER:

 RUN

 MORE ▾

 SAVE ▾

 SHARE ▾

 SCHEDULE ▾

```
1 SELECT
2   title,
3   view_count
4 FROM `bigquery-public-data.stackoverflow.stackoverflow_posts`
5 WHERE LOWER(title) LIKE "%bigquery%"
6 ORDER BY view_count DESC
7 LIMIT 10;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	title	view_count		
1	The OAuth Client was not found - Google Apps Script - BigQuery	14074		
2	BigQuery datatypes	7429		
3	Support UNION function in Big...	7405		
4	BigQuery COUNT(DISTINCT val...	6958		
5	Joins on Google Bigquery	6523		
6	Pros & cons of BigQuery vs. A...	5666		
7	Can we cast the type in BigQue...	5268		
8	Bigquery + PHP examples	4905		
9	Bigquery add columns to table ...	4886		
10	BigQuery Wildcard using TABL...	4730		

15. Continue in the “stackoverflow” dataset in the “bigquery-public-data” project.

Write a SQL query that joins the “stackoverflow\_posts” and “users” tables to return the “title”, “view\_count”, “owner\_display\_name”, and “reputation” of the 10 titles with the most comments (“comment\_count”). Please add a WHERE clause to filter out NULLs in the “title” column.

Hint: “reputation” is from the “users” table.

ANSWER:

```
1 SELECT
2   Posts.title,
3   Posts.view_count,
4   Posts.owner_display_name,
5   Users.reputation
6 FROM `bigquery-public-data.stackoverflow.stackoverflow_posts` AS Posts
7 JOIN `bigquery-public-data.stackoverflow.users` AS Users
8 ON Posts.id = Users.id
9 WHERE Posts.title IS NOT NULL
10 ORDER BY Posts.comment_count DESC
11 LIMIT 10;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	
row	title	view_count	owner_display_name	reputation	
1	How to create a new web character symbol recognizable by html/javascript?	1310	null	1	
2	Connect ffmpeg to Visual Studio 2008	3854	null	11	
3	File I/O in Every Programming Language	74530	null	1	
4	What is the best Battleship AI?	42952	null	5251	
5	what is the fastest way to notify another thread that data is available? any alternatives to spinning?	1845	null	1	
6	Does the C++ standard mandate poor performance for iostreams or am I just dealing with a poor implementation?	12706	null	21	
7	PHP session not working in Safari Webkit Nightly	5023	null	1	
8	What is the worst programming language you ever worked with?	96574	null	21	
9	Why do browsers match CSS selectors from right to left?	48139	null	1	
10	Eclipse consuming a web service gives java.lang.reflect.InvocationTargetException	3524	null	1	



### Question 15 ANSWER (for those whose “owner\_display\_name” is not null):

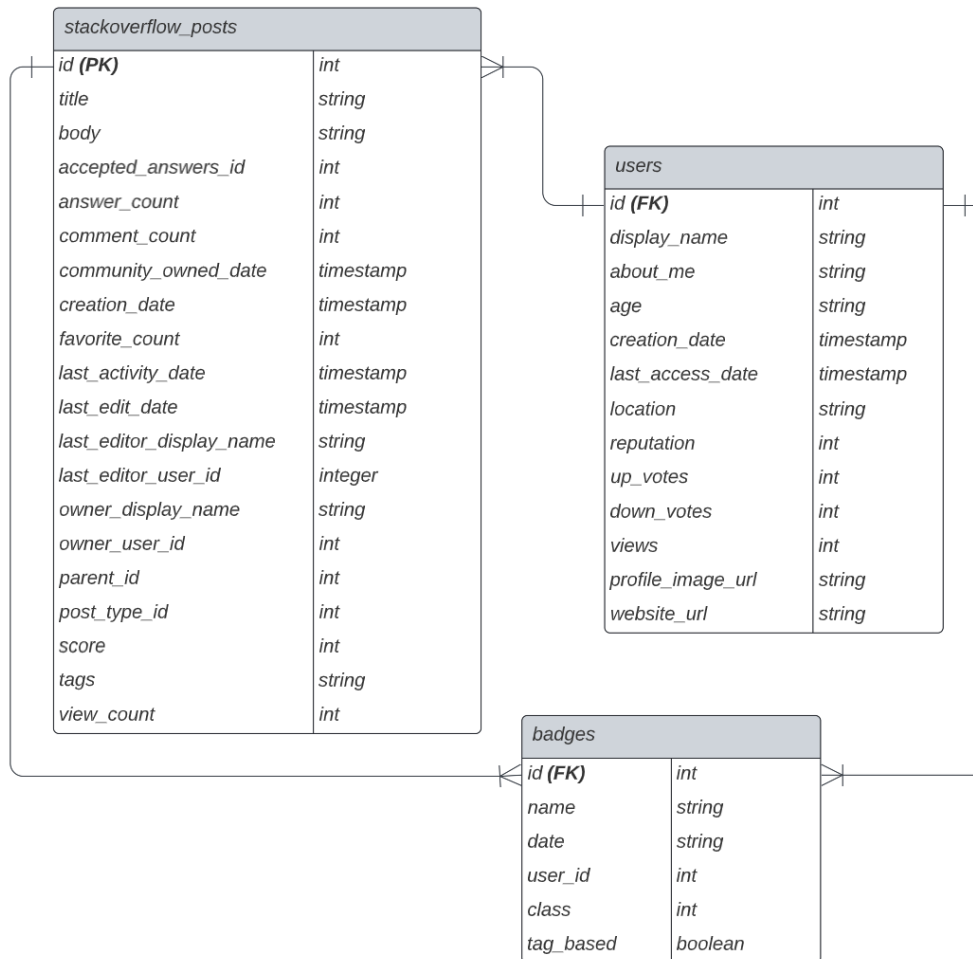
```
1 SELECT
2     Posts.title,
3     Posts.view_count,
4     Posts.owner_display_name,
5     Users.reputation
6 FROM `bigquery-public-data.stackoverflow.stackoverflow_posts` AS Posts
7 JOIN `bigquery-public-data.stackoverflow.users` AS Users
8 ON Posts.id = Users.id
9 WHERE Posts.title IS NOT NULL AND Posts.owner_display_name IS NOT NULL # Having a display name
10 ORDER BY Posts.comment_count DESC
11 LIMIT 10;
```

### Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		
row	title	view_count	owner_display_name	reputation		
1	What's the best name for a non-mutating add method on an immutable collection?	12934	Jon Skeet	437		
2	SQLite database not saving data	378	user2568107	1		
3	What is your opinion of the Enti...	2597	Shaun	1190		
4	Does anyone beside me just N...	18313	Jonathan Holland	1		
5	C++ String Interview Question	3381	Matthieu N.	1		
6	Is Multiplying the Inverse Bette...	1435	anon	9		
7	In .NET which loop runs faster '...	89386	Binoj Antony	1		
8	No success using Header Redir...	101	user2036852	1		
9	Trouble with constant objects ...	193	user1581100	1		
10	Avoiding != null statements	737912	Goran Martinic	499		

16. Paste your ERD from Lucidchart as your answer.

ANSWER:



Academic Integrity Statement - This needs to be signed by you.

*The work in this assignment is my own. I have not used outside help when answering the questions and have not used sites to procure answers (e.g., Chegg.com). Any outside sources have been properly cited.*

Jason Jiang  
(Signature)