B. Imel
Practicing SQL with Google BigQuery

1. **Please find out which Github repositories Facebook has. Exclude those that are forks. Only return the top 5 repositories.**

   Dataset: bigquery-public-data.samples.github_nested
   a. **"Top 5" measured by number of open issues**
   b. **"Top 5" measured by the number of users that have created a pull request.**

Query A:

```
SELECT repository.owner, repository.open_issues, repository.fork
FROM [bigquery-public-data:samples.github_nested]
WHERE repository.owner = 'facebook' AND repository.fork = false
ORDER BY repository.open_issues DESC LIMIT 5;
```

Returns:

Row	repository_owner	repository_open_issues	repository_fork
1	facebook	241	false
2	facebook	241	false
3	facebook	241	false
4	facebook	241	false
5	facebook	241	false

Query B:

```
SELECT payload.pull_request.user.login, repository.owner, repository.fork,
COUNT (payload.pull_request.user.login) AS num_occurrences
FROM [bigquery-public-data:samples.github_nested]
WHERE repository.owner = 'facebook' AND repository.fork = false
GROUP BY payload.pull_request.user.login, repository.owner, repository.fork
ORDER BY num_occurrences DESC
LIMIT 5;
```

Returns:

Row	payload_pull_request_user_login	repository_owner	repository_fork	num_occurrences
1	jeffmo	facebook	false	4
2	wsantos	facebook	false	3
3	DmitrySoshnikov	facebook	false	2
4	jsamuel	facebook	false	2
5	adamvduke	facebook	false	2

B. Imel
Practicing SQL with Google BigQuery

2. Please find out the Top 25 Wikipedia pages, measured by the number of revisions. Dataset: bigquery-public-data.samples.wikipedia

Query:

```
SELECT id, title,
COUNT(id) AS num_occurrences
FROM [bigquery-public-data:samples.wikipedia],
GROUP BY id, title
ORDER BY num_occurrences DESC
LIMIT 25;
```

Returns:

id	title	num_occurrences
1952670	Wikipedia:Administrator intervention against vandalism	643271
5137507	Wikipedia:Administrators' noticeboard/Incidents	419695
2189161	Wikipedia:Sandbox/Archive	326337
16283969	Wikipedia:Sandbox	257893
13784401	User:Cyde/List of candidates for speedy deletion/Subpage	226802
2535910	Wikipedia:Reference desk/Science	204469
16927404	Wikipedia:WikiProject Spam/LinkReports	191679
40297	Wikipedia:Reference desk/Miscellaneous	186715
972034	Template talk:Did you know	184508
564696	Wikipedia:Help desk	169952
352651	Wikipedia:Requests for page protection	164620
5149102	Wikipedia:Administrators' noticeboard	155706
2535875	Wikipedia:Reference desk/Humanities	149093
1226609	Wikipedia:Introduction	142421
11022716	Wikipedia:Usernames for administrator attention	92554
5030553	Talk:Main Page	91719
6041086	Wikipedia:Reference desk/Computing	83628
2515121	Wikipedia:Reference desk/Language	81268
10701605	User:COIBot/LinkReports	77993
11005908	Wikipedia:Tutorial (Editing)/sandbox	71514
8979574	User:Cyde/List of candidates for speedy deletion	70776
5964327	Wikipedia:Suspected copyright violations	69007
1052128	Wikipedia:Requested moves	65778
11238105	Wikipedia:Usernames for administrator attention/Bot	65222
8993207	User:Cyde/List of current proposed deletions	64210

3. Please find out which were the latest 5 Wikipedia contributions. For each of those contributions, please find out to which page it belonged to and which user made the contribution.
Dataset: bigquery-public-data.samples.wikipedia

Query:

```
SELECT id, title, timestamp
FROM [bigquery-public-data:samples.wikipedia]
ORDER BY timestamp DESC
LIMIT 5;
```

Returns:

Row	title	timestamp	contributor_username
1	List of NBC Saturday Night at the Movies Titles	1265098303	Cricket2QXR3
2	HIStory: Past, Present and Future, Book I	1265098302	Crystal Clear x3
3	User talk:Extremepro	1265098302	KrebMarkt
4	Appraisal theory	1265098302	OfriRaviv
5	Motet	1265098302	Bdiscoe

4. Please find out which are the top 5 most used words by Shakespeare.
Dataset: bigquery-public-data.samples.shakespeare

Query:

```
SELECT SUM(word_count) AS count_sum, word
FROM [bigquery-public-data:samples.shakespeare]
GROUP BY word
ORDER BY count_sum DESC LIMIT 5;
```

Returns:

Row	count_sum	word
1	25568	the
2	21028	I
3	19649	and
4	17361	to
5	16438	of

5. **Please find out the 3 works by Shakespeare that have the highest number of words in them.**
   Dataset: bigquery-public-data.samples.shakespeare
   a. **Total number of words**
   b. **Number of different words**

Query A:

```
SELECT SUM(word_count) AS count_sum, corpus
FROM [bigquery-public-data:samples.shakespeare]
GROUP BY corpus
ORDER BY count_sum DESC LIMIT 3;
```

Returns:

Row	count_sum	corpus
1	32446	hamlet
2	31868	kingrichardiii
3	29535	coriolanus

Query B:

```
SELECT COUNT(word) AS distinct_sum, corpus
FROM [bigquery-public-data:samples.shakespeare]
GROUP BY corpus
ORDER BY distinct_sum DESC LIMIT 3
```

Returns:

Row	distinct_sum	corpus
1	5318	hamlet
2	5104	kinghenryv
3	4875	cymbeline

6. **Please find out the min, max, and average number of total words in Shakespeare's works. In addition, please also include the standard deviation**
   Dataset: bigquery-public-data.samples.shakespeare

Maximum Query:

```
SELECT MAX(count_sum) AS max_words
FROM (SELECT SUM(word_count) AS count_sum, corpus
FROM [bigquery-public-data:samples.shakespeare]
GROUP BY corpus);
```

B. Imel
Practicing SQL with Google BigQuery

Returns:

Row	max_words
1	32446

Minimum Query:

```
SELECT MIN(count_sum) AS min_words
FROM (SELECT SUM(word_count) AS count_sum, corpus
FROM [bigquery-public-data:samples.shakespeare]
GROUP BY corpus);
```

Returns:

Row	min_words
1	2586

Average Query:

```
SELECT AVG(count_sum) AS average_words
FROM (SELECT SUM(word_count) AS count_sum, corpus
FROM [bigquery-public-data:samples.shakespeare]
GROUP BY corpus);
```

Returns:

Row	average_words
1	22520.119047619046

Standard Deviation Query:

```
SELECT STDDEV(count_sum) AS stddev_words
FROM (SELECT SUM(word_count) AS count_sum, corpus
FROM [bigquery-public-data:samples.shakespeare]
GROUP BY corpus);
```

Returns:

Row	stddev_words
1	6411.872540318224

7. **Please calculate the correlation coefficient between the maximum temperature in a year and the average weight of a newborn child in a year.**
   Datasets: bigquery-public-data.samples.gsod
   bigquery-public-data.samples.natality

Query:

```
SELECT CORR(temp_table.max_year_temp, birth_table.avg_birth_weight) AS corr_coeff FROM

(SELECT year, MAX(max_temperature) AS max_year_temp,
FROM [bigquery-public-data:samples.gsod] GROUP BY year) AS temp_table

INNER JOIN

(SELECT year, AVG(weight_pounds) AS avg_birth_weight,
FROM [bigquery-public-data:samples.natality] GROUP BY year) AS birth_table

ON temp_table.year = birth_table.year
```

Returns:

Row	corr_coeff
1	0.3374072098385498

8. **Please find out the maximum weight of a newborn baby.**
   Dataset: bigquery-public-data.samples.natality

Query:

```
SELECT MAX(weight_pounds) AS max_birth_weight
FROM [bigquery-public-data:samples.natality];
```

Returns:

Row	max_birth_weight
1	18.0007436923

9. We want to know how many authors of Hackernews stories are also Wikipedia contributors. Assume that if the usernames are the same, those are the same users.

Query:

```
SELECT Count(*) AS dupl_users FROM

(SELECT [by] AS hs_user,
FROM [bigquery-public-data:hacker_news.full] GROUP BY hs_user) AS hs_table

INNER JOIN

(SELECT contributor_username AS wiki_user,
FROM [bigquery-public-data:samples.wikipedia] GROUP BY wiki_user) AS wiki_table

ON hs_table.hs_user = wiki_table.wiki_user;
```

Returns:

Row	dupl_users
1	5809

10. Please find out the titles of the top 10 Hackernews stories measured by their score. Dataset: bigquery-public-data.hacker_news.full

Query:

```
SELECT title, type, score FROM [bigquery-public-data:hacker_news.full]
WHERE type = 'story'
ORDER BY score DESC
LIMIT 10;
```

Returns:

Row	title	type	score
1	Stephen Hawking has died	story	6015
2	A Message to Our Customers	story	5771
3	Steve Jobs has passed away.	story	4338
4	Reflecting on one very, very strange year at Uber	story	4107
5	Show HN: This up votes itself	story	3531
6	F.C.C. Repeals Net Neutrality Rules	story	3384
7	Cloudflare Reverse Proxies Are Dumping Uninitialized Memory	story	3238
8	UK votes to leave EU	story	3125
9	Tim Cook Speaks Up	story	3086
10	Announcing the first SHA-1 collision	story	3030