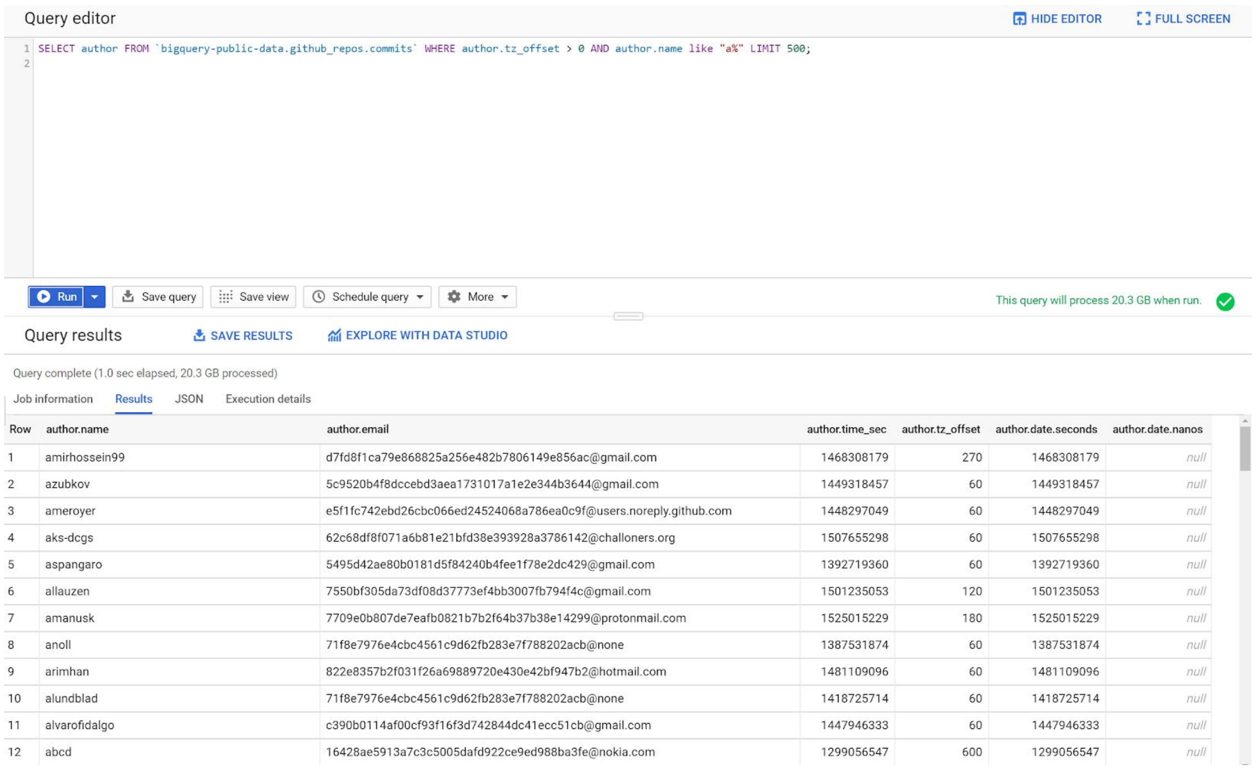
CSC 370
Assignment 4
Database Jumpers

Paige Loffler	ID: paigeloffler
Bradon Horcoff	ID: braydonh
Oliver Tonnesen	ID: otonnesen
Nat Dring	ID: ndring
Parm Johal	ID: parmj

1. **What are the Github Authors names that start with an A, that also have an offset greater than 0?**

SELECT author FROM `bigquery-public-data.github_repos.commits` WHERE author.tz_offset > 0 AND author.name like "a%" LIMIT 500;



```
SELECT author FROM `bigquery-public-data.github_repos.commits` WHERE author.tz_offset > 0 AND author.name like "a%" LIMIT 500;
```

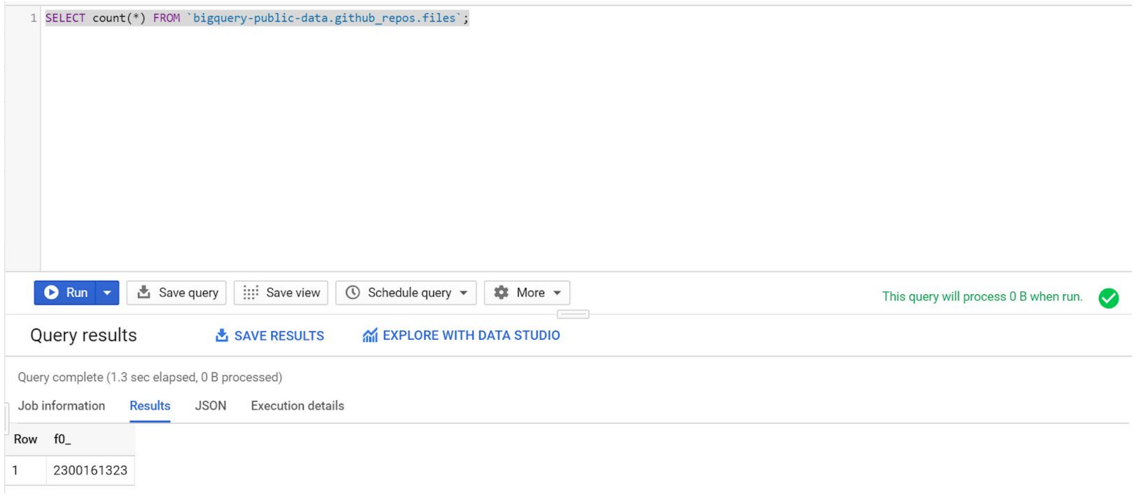
This query will process 20.3 GB when run.

Query complete (1.0 sec elapsed, 20.3 GB processed)

Row	author.name	author.email	author.time_sec	author.tz_offset	author.date.seconds	author.date.nanos
1	amirhossein99	d7fd8f1ca79e868825a256e482b7806149e856ac@gmail.com	1468308179	270	1468308179	null
2	azubkov	5c9520b4f8dccebd3aea1731017a1e2e344b3644@gmail.com	1449318457	60	1449318457	null
3	ameroyer	e5f1fc742ebd26cbc066ed24524068a786ea0c9f@users.noreply.github.com	1448297049	60	1448297049	null
4	aks-dcgs	62c68df8f071a6b81e21bfd38e393928a3786142@challoners.org	1507655298	60	1507655298	null
5	aspangaro	5495d42ae80b0181d5f84240b4fee1f78e2dc429@gmail.com	1392719360	60	1392719360	null
6	allauzen	7550bf305da73df08d37773ef4bb3007fb794f4c@gmail.com	1501235053	120	1501235053	null
7	amanusk	7709e0b807de7eafb0821b7b2f64b37b38e14299@protonmail.com	1525015229	180	1525015229	null
8	anoll	71f8e7976e4cbc4561c9d62fb283e7f788202acb@none	1387531874	60	1387531874	null
9	arimhan	822e8357b2f031f26a69889720e430e42bf947b2@hotmail.com	1481109096	60	1481109096	null
10	alundblad	71f8e7976e4cbc4561c9d62fb283e7f788202acb@none	1418725714	60	1418725714	null
11	alvarofidalgo	c390b0114af00cf93f16f3d742844dc41ecc51cb@gmail.com	1447946333	60	1447946333	null
12	abcd	16428ae5913a7c3c5005dafd922ce9ed988ba3fe@nokia.com	1299056547	600	1299056547	null

2. **How many different IDs are there in the files table?**

SELECT count(*) FROM `bigquery-public-data.github_repos.files`;



```
SELECT count(*) FROM `bigquery-public-data.github_repos.files`;
```

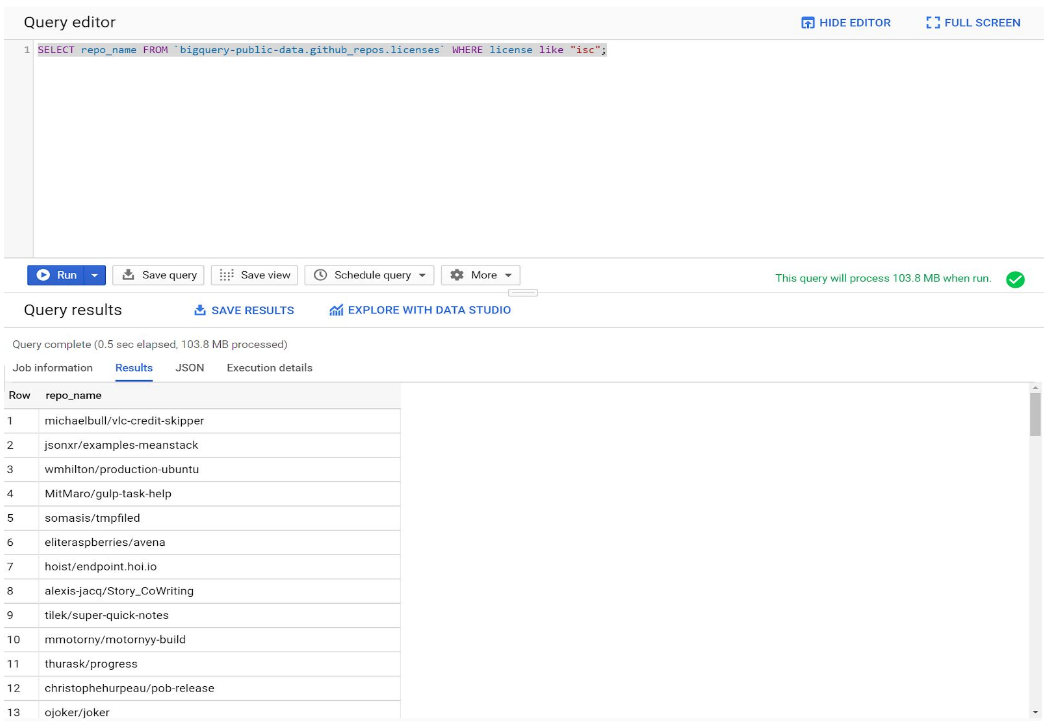
This query will process 0 B when run.

Query complete (1.3 sec elapsed, 0 B processed)

Row	f0_
1	2300161323

3. **What are the names of every repo with an isc license?**

SELECT repo_name FROM `bigquery-public-data.github_repos.licenses` WHERE license like "isc";



Query editor

HIDE EDITOR | FULL SCREEN

```
SELECT repo_name FROM `bigquery-public-data.github_repos.licenses` WHERE license like "isc";
```

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

This query will process 103.8 MB when run.

Query results

SAVE RESULTS | EXPLORE WITH DATA STUDIO

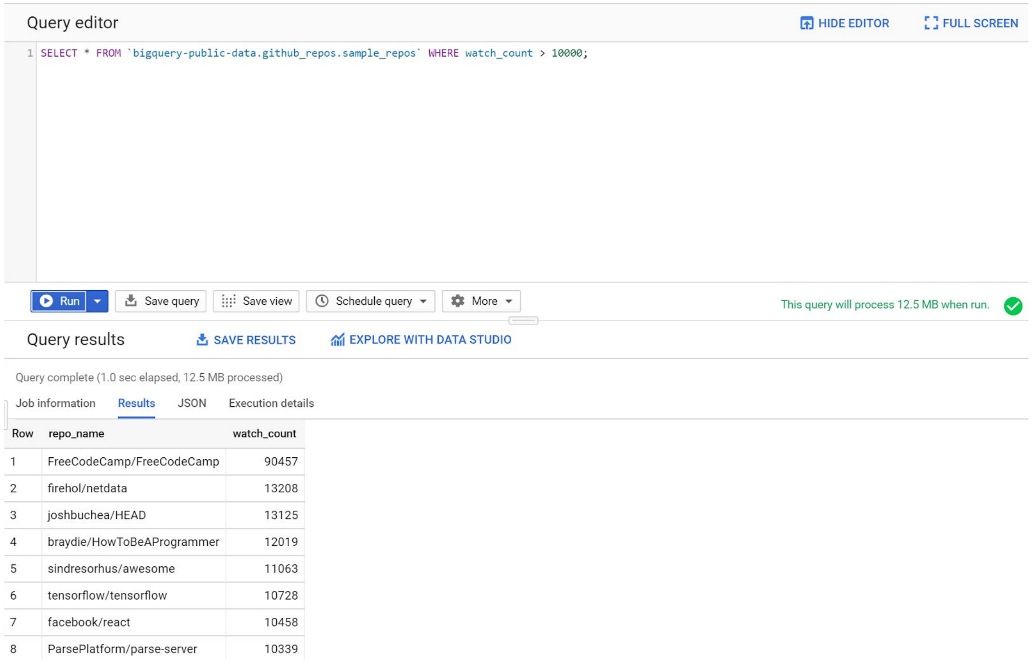
Query complete (0.5 sec elapsed, 103.8 MB processed)

Job information | Results | JSON | Execution details

Row	repo_name
1	michaelbull/vlc-credit-skipper
2	jsonxr/examples-meanstack
3	wmhilton/production-ubuntu
4	MitMaro/gulp-task-help
5	somasis/tmpfiled
6	eliteraspberries/avena
7	hoist/endpoint.hoi.io
8	alexis-jacq/Story_CoWriting
9	tilek/super-quick-notes
10	mmotorny/motornyy-build
11	thurask/progress
12	christophehurpeau/pob-release
13	ojoker/joker

4. What are all the repos with watch counts larger than ten thousand?

SELECT * FROM `bigquery-public-data.github_repos.sample_repos` WHERE watch_count > 10000;



Query editor

HIDE EDITOR | FULL SCREEN

```
SELECT * FROM `bigquery-public-data.github_repos.sample_repos` WHERE watch_count > 10000;
```

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

This query will process 12.5 MB when run.

Query results

SAVE RESULTS | EXPLORE WITH DATA STUDIO

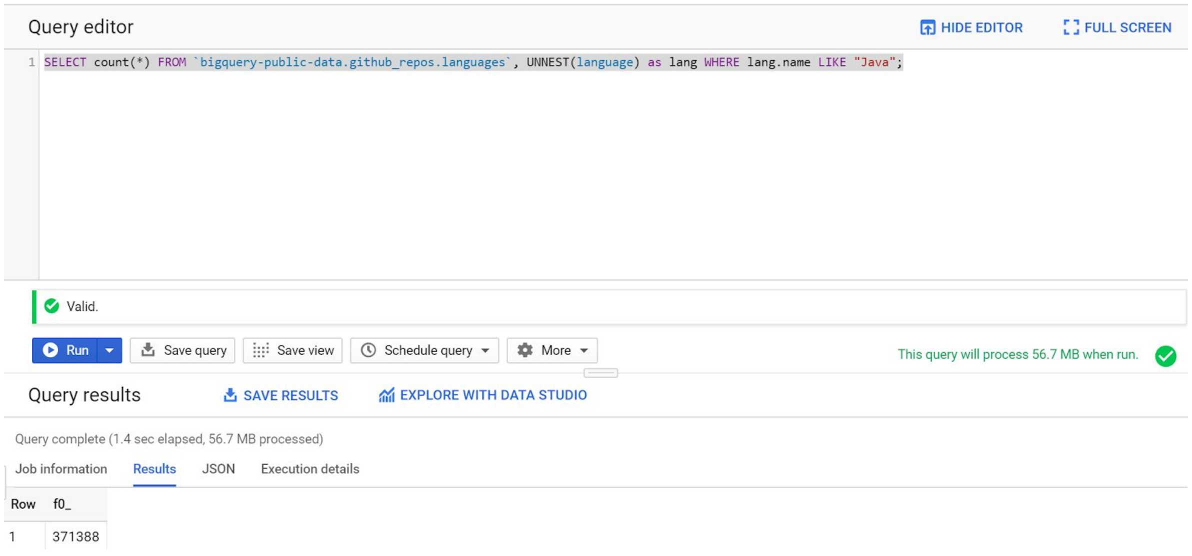
Query complete (1.0 sec elapsed, 12.5 MB processed)

Job information | Results | JSON | Execution details

Row	repo_name	watch_count
1	FreeCodeCamp/FreeCodeCamp	90457
2	firehol/netdata	13208
3	joshbuchea/HEAD	13125
4	braydie/HowToBeAProgrammer	12019
5	sindresorhus/awesome	11063
6	tensorflow/tensorflow	10728
7	facebook/react	10458
8	ParsePlatform/parse-server	10339

5. How many Github repos have projects written in Java?

SELECT count(*) FROM `bigquery-public-data.github_repos.languages`, UNNEST(language) as lang WHERE lang.name LIKE "Java";



```
SELECT count(*) FROM `bigquery-public-data.github_repos.languages`, UNNEST(language) as lang WHERE lang.name LIKE "Java";
```

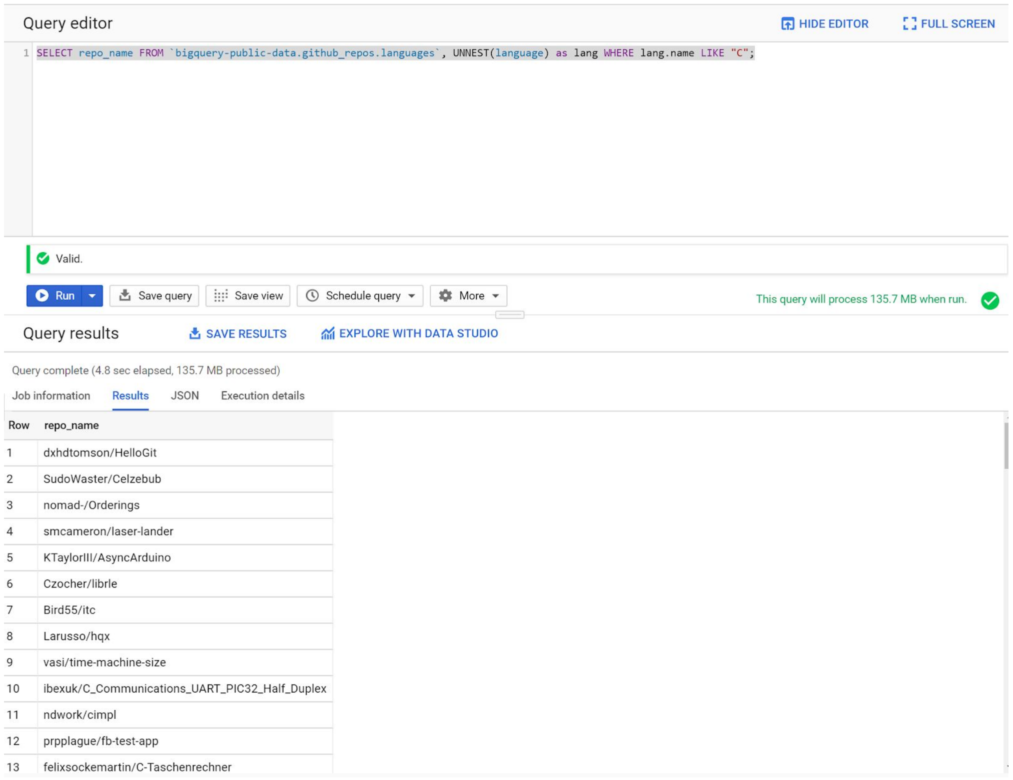
This query will process 56.7 MB when run.

Query complete (1.4 sec elapsed, 56.7 MB processed)

Row	f0_
1	371388

6. **What are the repo names that have projects written in C?**

SELECT repo_name FROM `bigquery-public-data.github_repos.languages`, UNNEST(language) as lang WHERE lang.name LIKE "C";



```
SELECT repo_name FROM `bigquery-public-data.github_repos.languages`, UNNEST(language) as lang WHERE lang.name LIKE "C";
```

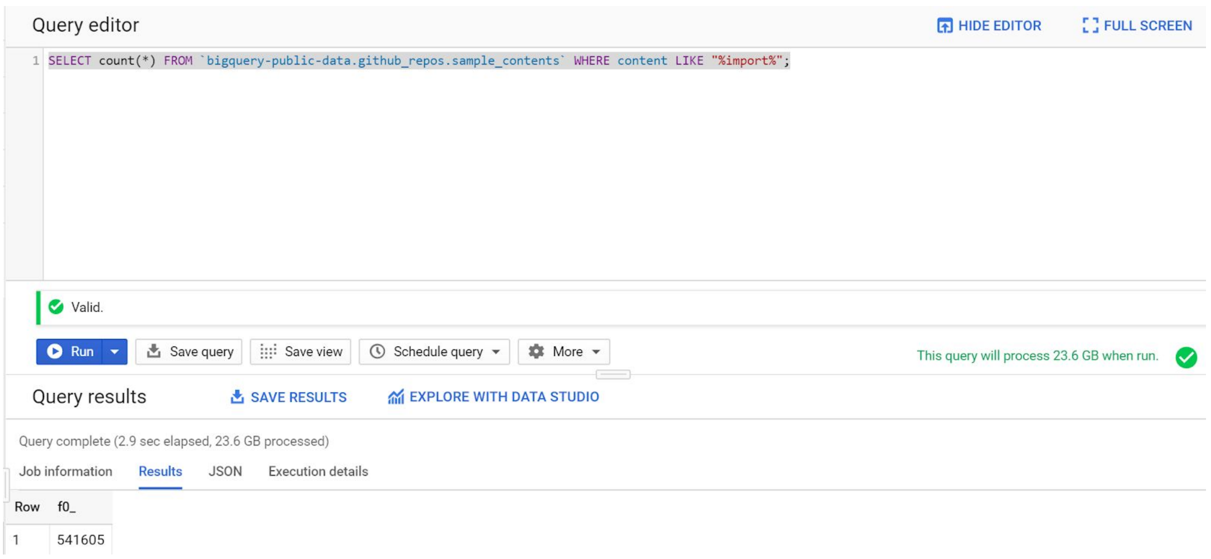
This query will process 135.7 MB when run.

Query complete (4.8 sec elapsed, 135.7 MB processed)

Row	repo_name
1	dxhdtomson/HelloGit
2	SudoWaster/Celzebub
3	nomad-/Orderings
4	smcameron/laser-lander
5	KTaylorIII/AsyncArduino
6	Czocher/librle
7	Bird55/itc
8	Larusso/hqx
9	vasi/time-machine-size
10	ibexuk/C_Communications_UART_PIC32_Half_Duplex
11	ndwork/cimpl
12	prpplague/fb-test-app
13	felixsockemartin/C-Taschenrechner

7. **How many users use the import statement within their projects?**

SELECT count(*) FROM `bigquery-public-data.github_repos.sample_contents` WHERE content LIKE "%import%";



```
SELECT count(*) FROM `bigquery-public-data.github_repos.sample_contents` WHERE content LIKE "%import%";
```

Valid.

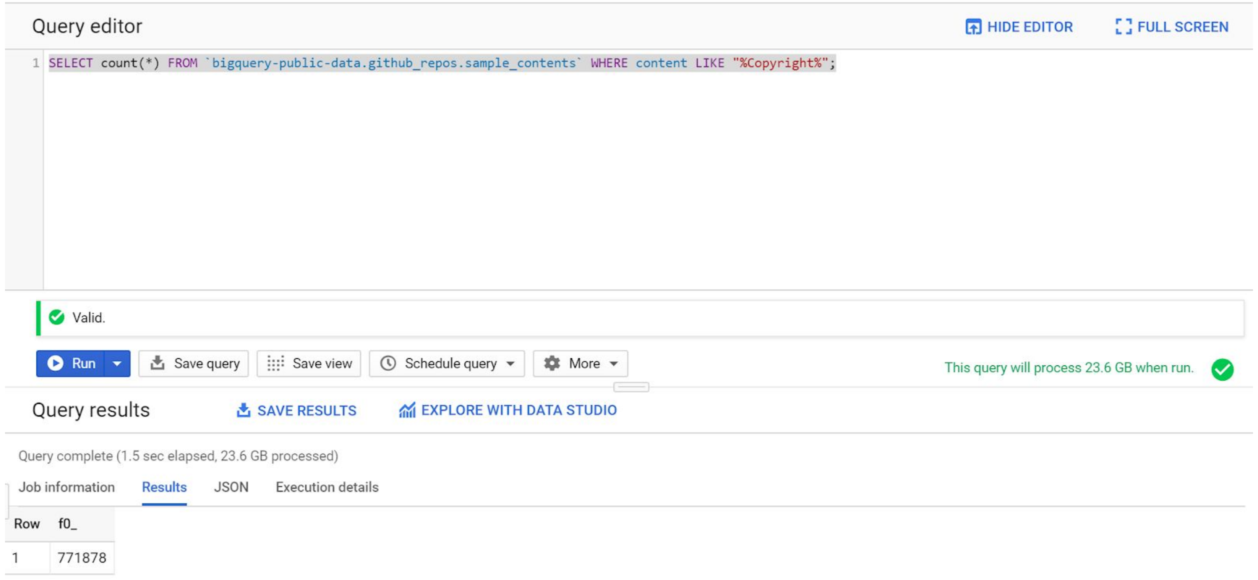
This query will process 23.6 GB when run.

Query complete (2.9 sec elapsed, 23.6 GB processed)

Row	f0_
1	541605

8. How many users have a project that is contains a copyright?

SELECT count(*) FROM `bigquery-public-data.github_repos.sample_contents` WHERE content LIKE "%Copyright%";



```
SELECT count(*) FROM `bigquery-public-data.github_repos.sample_contents` WHERE content LIKE "%Copyright%";
```

Valid.

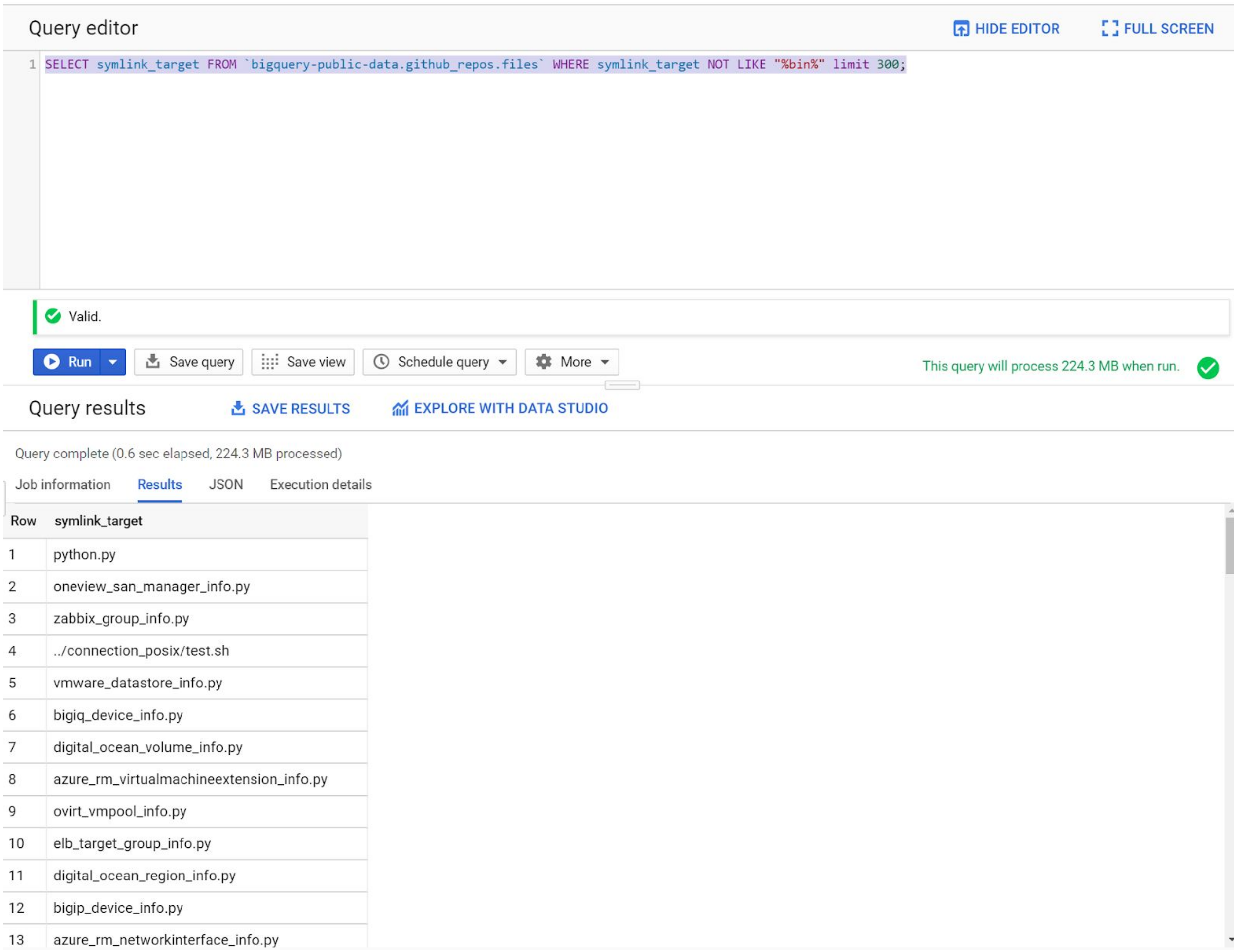
This query will process 23.6 GB when run.

Query complete (1.5 sec elapsed, 23.6 GB processed)

Row	f0_
1	771878

9. What are the first 300 symbolic link targets that do not contain the bin path?

SELECT symlink_target FROM `bigquery-public-data.github_repos.files` WHERE symlink_target NOT LIKE "%bin%" limit 300;



```
SELECT symlink_target FROM `bigquery-public-data.github_repos.files` WHERE symlink_target NOT LIKE "%bin%" limit 300;
```

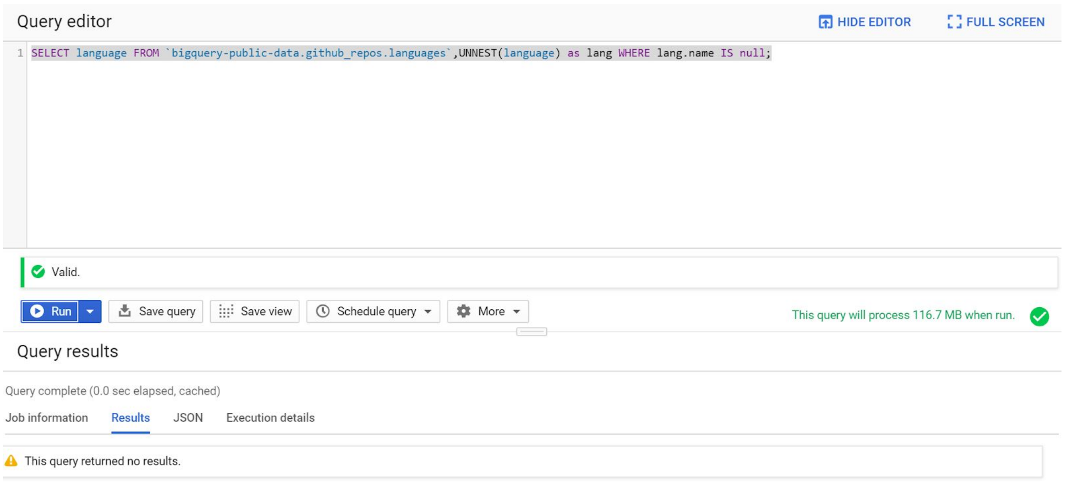
This query will process 224.3 MB when run.

Query complete (0.6 sec elapsed, 224.3 MB processed)

Row	symlink_target
1	python.py
2	oneview_san_manager_info.py
3	zabbix_group_info.py
4	../connection_posix/test.sh
5	vmware_datastore_info.py
6	bigiq_device_info.py
7	digital_ocean_volume_info.py
8	azure_rm_virtualmachineextension_info.py
9	ovirt_vmpool_info.py
10	elb_target_group_info.py
11	digital_ocean_region_info.py
12	bigip_device_info.py
13	azure_rm_networkinterface_info.py

10. Are there any broken repos whose language is defaulted to null?

SELECT language FROM `bigquery-public-data.github_repos.languages`,UNNEST(language) as lang WHERE lang.name IS null;



```
SELECT language FROM `bigquery-public-data.github_repos.languages`,UNNEST(language) as lang WHERE lang.name IS null;
```

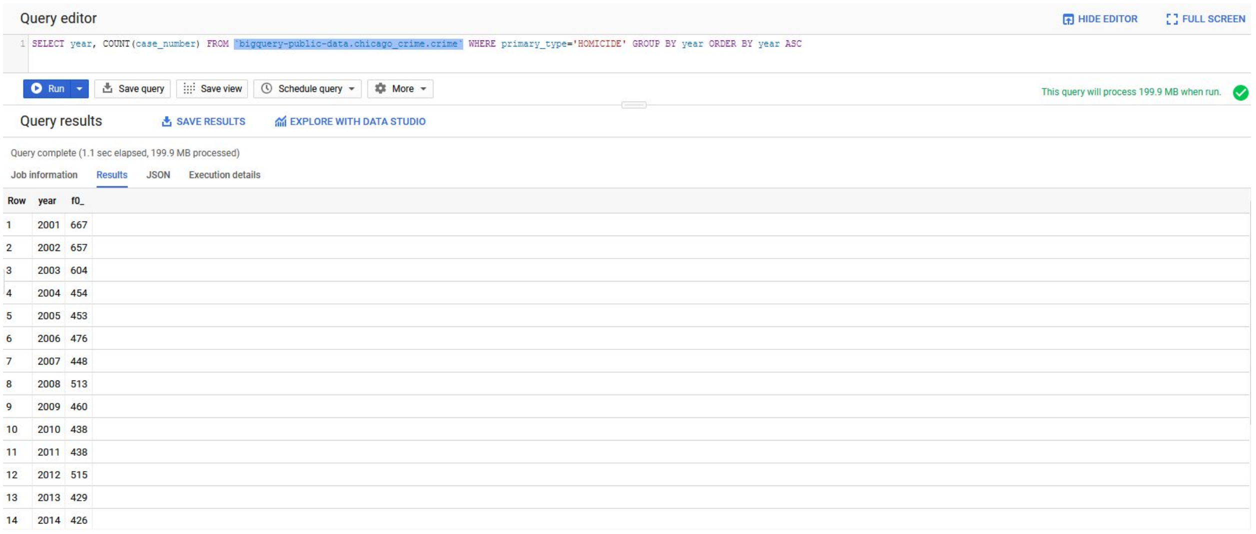
This query will process 116.7 MB when run.

Query complete (0.0 sec elapsed, cached)

This query returned no results.

11. Total number of homicides in Chicago each year

SELECT year, COUNT(case_number) FROM `bigquery-public-data.chicago_crime.crime` WHERE primary_type='HOMICIDE' GROUP BY year ORDER BY year ASC



```
SELECT year, COUNT(case_number) FROM `bigquery-public-data.chicago_crime.crime` WHERE primary_type='HOMICIDE' GROUP BY year ORDER BY year ASC
```

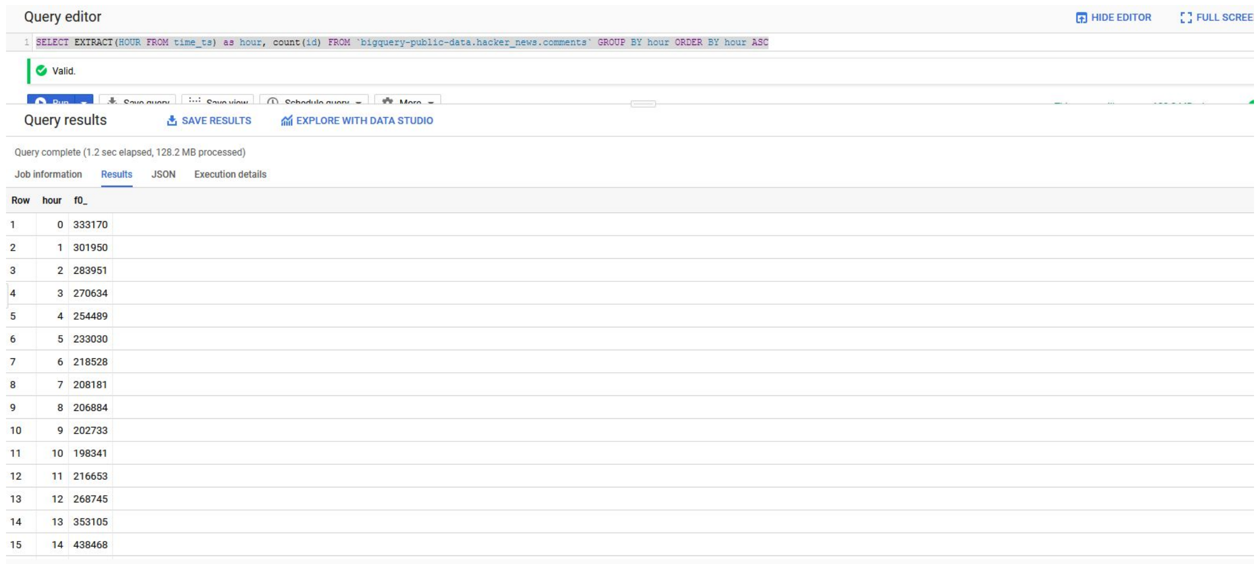
This query will process 199.9 MB when run.

Query complete (1.1 sec elapsed, 199.9 MB processed)

Row	year	f0_
1	2001	667
2	2002	657
3	2003	604
4	2004	454
5	2005	453
6	2006	476
7	2007	448
8	2008	513
9	2009	460
10	2010	438
11	2011	438
12	2012	515
13	2013	429
14	2014	426

12. Total number of Hacker News comments made during each hour of the day

SELECT EXTRACT(HOUR FROM time_ts) as hour, count(id) FROM \`bigquery-public-data.hacker_news.comments\` GROUP BY hour ORDER BY hour ASC



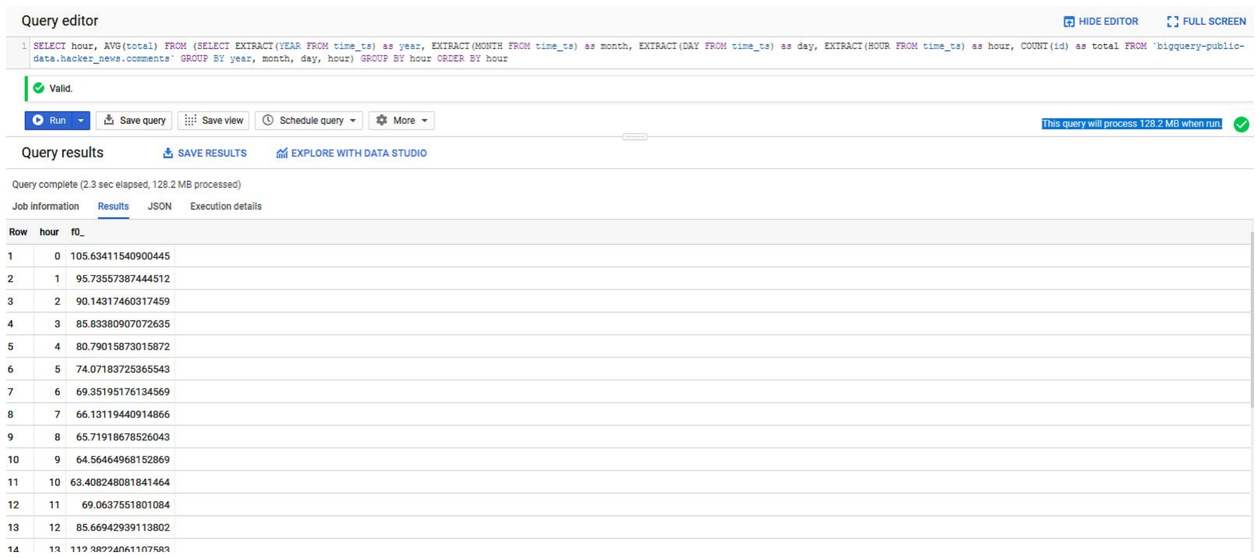
```
SELECT EXTRACT(HOUR FROM time_ts) as hour, count(id) FROM `bigquery-public-data.hacker_news.comments` GROUP BY hour ORDER BY hour ASC
```

Query complete (1.2 sec elapsed, 128.2 MB processed)

Row	hour	f0_
1	0	333170
2	1	301950
3	2	283951
4	3	270634
5	4	254489
6	5	233030
7	6	218528
8	7	208181
9	8	206884
10	9	202733
11	10	198341
12	11	216653
13	12	268745
14	13	353105
15	14	438468

13. Average number of comments submitted to Hacker News per hour

SELECT hour, AVG(total) FROM (SELECT EXTRACT(YEAR FROM time_ts) as year, EXTRACT(MONTH FROM time_ts) as month, EXTRACT(DAY FROM time_ts) as day, EXTRACT(HOUR FROM time_ts) as hour, COUNT(id) as total FROM `bigquery-public-data.hacker_news.comments` GROUP BY year, month, day, hour) GROUP BY hour



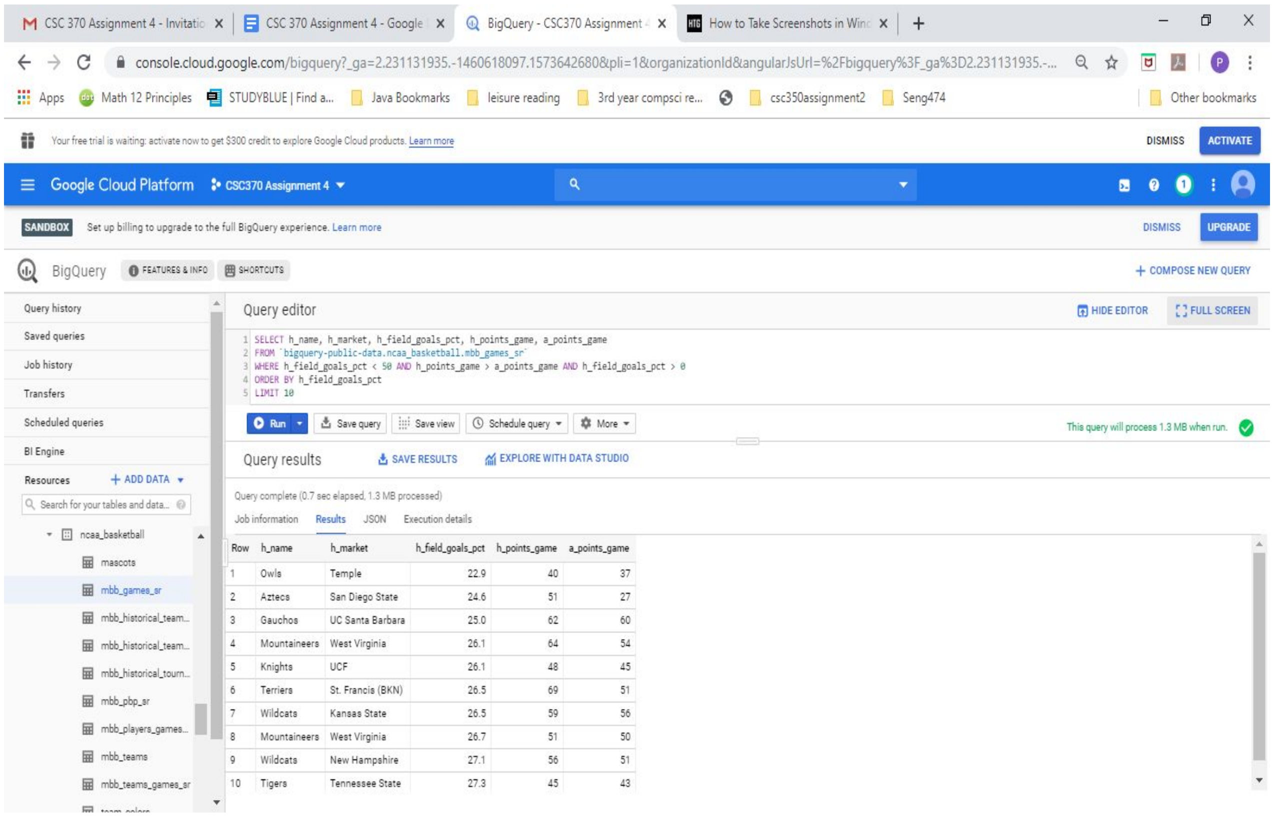
```
SELECT hour, AVG(total) FROM (SELECT EXTRACT(YEAR FROM time_ts) as year, EXTRACT(MONTH FROM time_ts) as month, EXTRACT(DAY FROM time_ts) as day, EXTRACT(HOUR FROM time_ts) as hour, COUNT(id) as total FROM `bigquery-public-
data.hacker_news.comments` GROUP BY year, month, day, hour) GROUP BY hour ORDER BY hour
```

Query complete (2.3 sec elapsed, 128.2 MB processed)

Row	hour	f0_
1	0	105.63411540900445
2	1	95.73557387444512
3	2	90.14317460317459
4	3	85.83380907072635
5	4	80.79015873015872
6	5	74.07183725365543
7	6	69.35195176134569
8	7	66.13119440914866
9	8	65.71918678526043
10	9	64.56464968152869
11	10	63.408248081841464
12	11	69.0637551801084
13	12	85.66942939113802
14	13	112.38224061107583

14. **Who are the top 10 teams in the country who have won a game with the lowest field goal percentage?**

```
SELECT h_name, h_market, h_field_goals_pct, h_points_game, a_points_game
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
WHERE h_field_goals_pct < 50 AND h_points_game > a_points_game AND
h_field_goals_pct > 0
ORDER BY h_field_goals_pct
LIMIT 10
```



```
SELECT h_name, h_market, h_field_goals_pct, h_points_game, a_points_game
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
WHERE h_field_goals_pct < 50 AND h_points_game > a_points_game AND h_field_goals_pct > 0
ORDER BY h_field_goals_pct
LIMIT 10
```

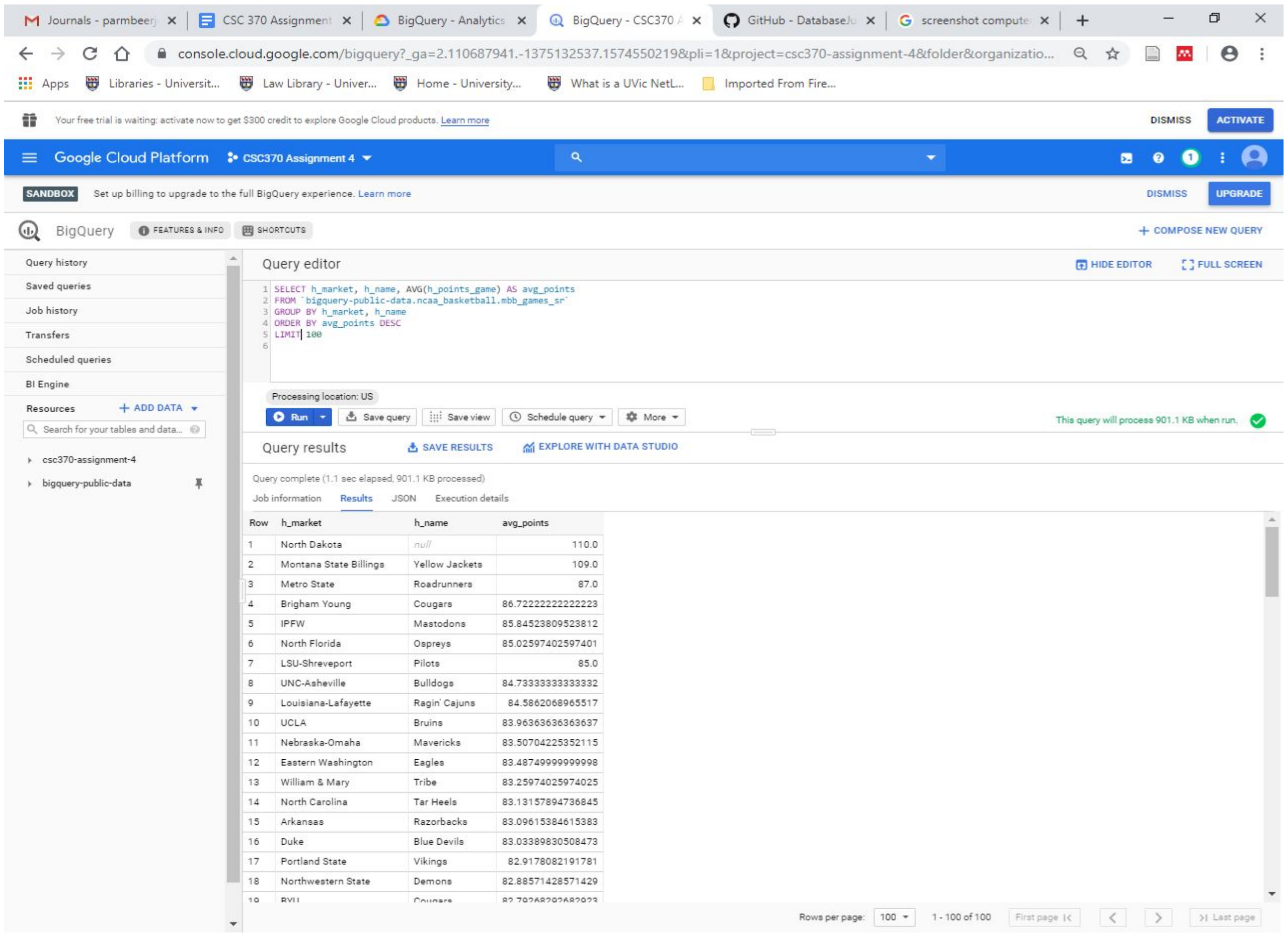
This query will process 1.3 MB when run.

Query complete (0.7 sec elapsed, 1.3 MB processed)

Row	h_name	h_market	h_field_goals_pct	h_points_game	a_points_game
1	Owls	Temple	22.9	40	37
2	Aztecs	San Diego State	24.6	51	27
3	Gauchos	UC Santa Barbara	25.0	62	60
4	Mountaineers	West Virginia	26.1	64	54
5	Knights	UCF	26.1	48	45
6	Terriers	St. Francis (BKN)	26.5	69	51
7	Wildcats	Kansas State	26.5	59	56
8	Mountaineers	West Virginia	26.7	51	50
9	Wildcats	New Hampshire	27.1	56	51
10	Tigers	Tennessee State	27.3	45	43

15. Rankings of teams with the highest average points per game.

```
SELECT h_market, h_name, AVG(h_points_game) AS avg_points
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
GROUP BY h_market, h_name
ORDER BY avg_points DESC
LIMIT 100
```



```
SELECT h_market, h_name, AVG(h_points_game) AS avg_points
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
GROUP BY h_market, h_name
ORDER BY avg_points DESC
LIMIT 100
```

Processing location: US

This query will process 901.1 KB when run.

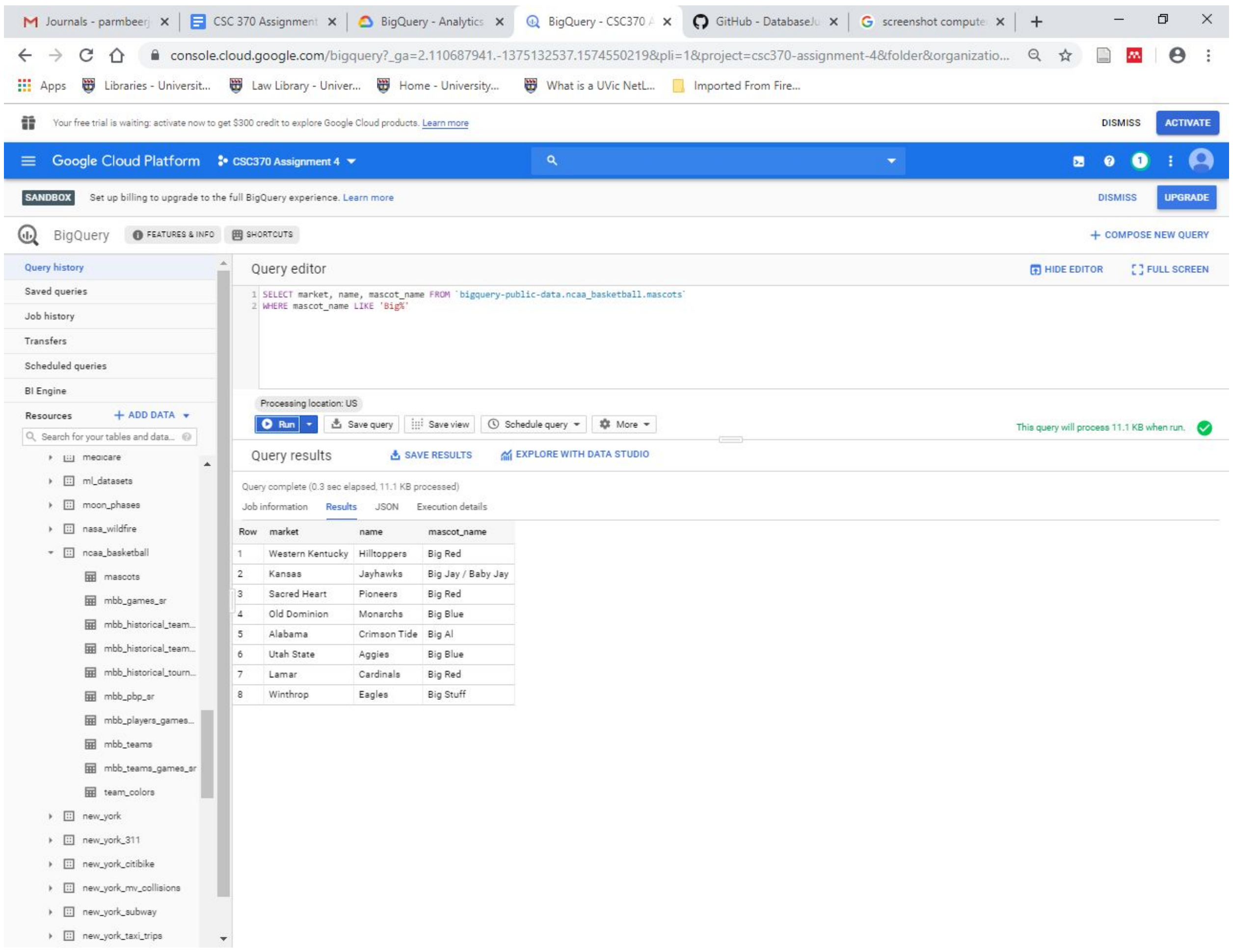
Query complete (1.1 sec elapsed, 901.1 KB processed)

Row	h_market	h_name	avg_points
1	North Dakota	null	110.0
2	Montana State Billings	Yellow Jackets	109.0
3	Metro State	Roadrunners	87.0
4	Brigham Young	Cougars	86.72222222222223
5	IPFW	Mastodons	85.84523809523812
6	North Florida	Ospreys	85.02597402597401
7	LSU-Shreveport	Pilots	85.0
8	UNC-Asheville	Bulldogs	84.73333333333332
9	Louisiana-Lafayette	Ragin' Cajuns	84.5862068965517
10	UCLA	Bruins	83.96363636363637
11	Nebraska-Omaha	Mavericks	83.50704225352115
12	Eastern Washington	Eagles	83.48749999999998
13	William & Mary	Tribe	83.25974025974025
14	North Carolina	Tar Heels	83.13157894736845
15	Arkansas	Razorbacks	83.09615384615383
16	Duke	Blue Devils	83.03389830508473
17	Portland State	Vikings	82.9178082191781
18	Northwestern State	Demons	82.88571428571429
19	BYU	Cougars	82.79268292682923

Rows per page: 100 1 - 100 of 100

16. Find all the teams whose mascot names begin with the word ‘Big’.

```
SELECT market, name, mascot_name
FROM `bigquery-public-data.ncaa_basketball.mascots`
WHERE mascot_name LIKE 'Big%'
```



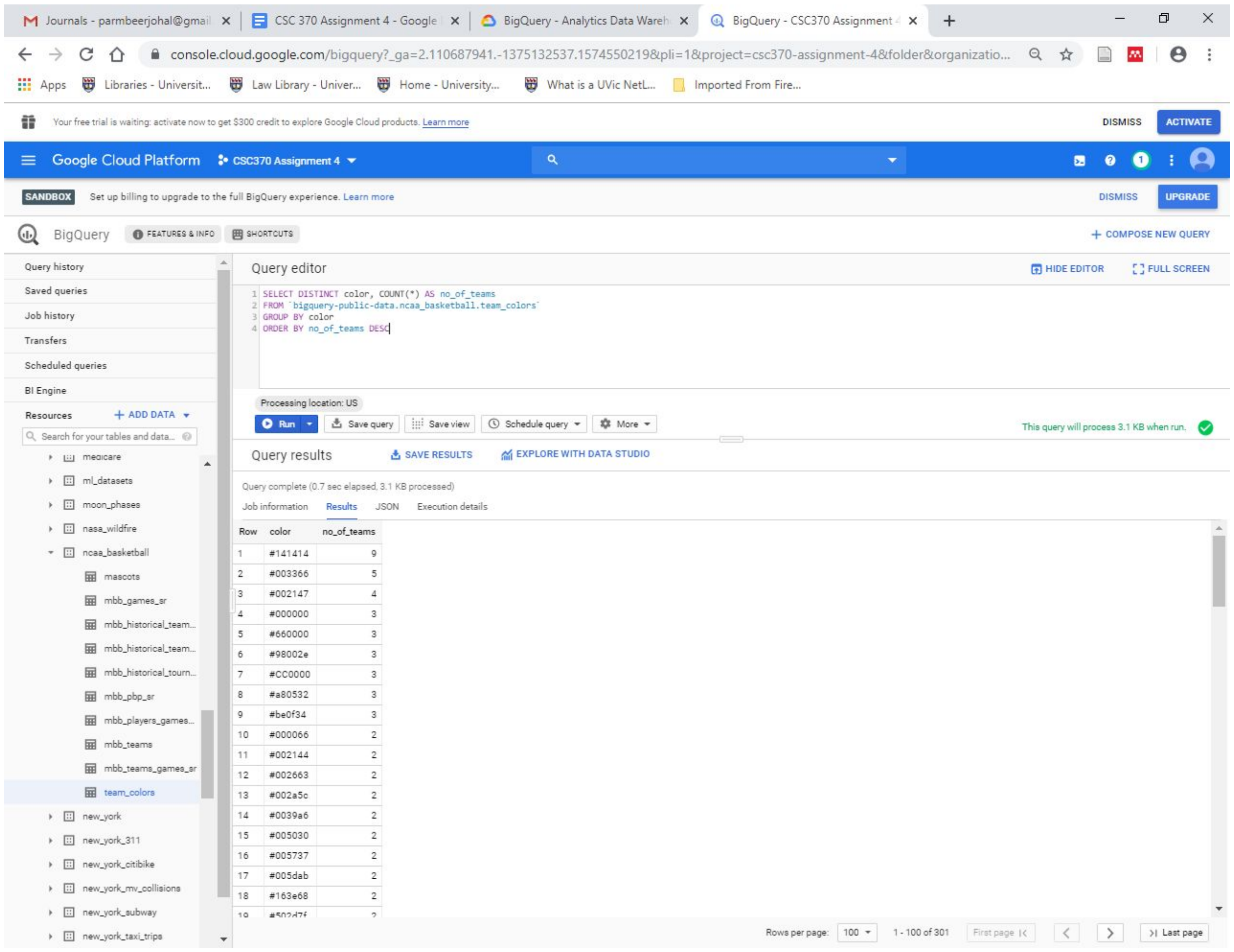
```
SELECT market, name, mascot_name FROM `bigquery-public-data.ncaa_basketball.mascots`
WHERE mascot_name LIKE 'Big%'
```

Query complete (0.3 sec elapsed, 11.1 KB processed)

Row	market	name	mascot_name
1	Western Kentucky	Hilltoppers	Big Red
2	Kansas	Jayhawks	Big Jay / Baby Jay
3	Sacred Heart	Pioneers	Big Red
4	Old Dominion	Monarchs	Big Blue
5	Alabama	Crimson Tide	Big Al
6	Utah State	Aggies	Big Blue
7	Lamar	Cardinals	Big Red
8	Winthrop	Eagles	Big Stuff

17. Which color worn the most by teams?

```
SELECT DISTINCT color, COUNT(*) AS no_of_teams
FROM `bigquery-public-data.ncaa_basketball.team_colors`
GROUP BY color
ORDER BY no_of_teams DESC
```



```
SELECT DISTINCT color, COUNT(*) AS no_of_teams
FROM `bigquery-public-data.ncaa_basketball.team_colors`
GROUP BY color
ORDER BY no_of_teams DESC
```

Processing location: US

This query will process 3.1 KB when run.

Query results

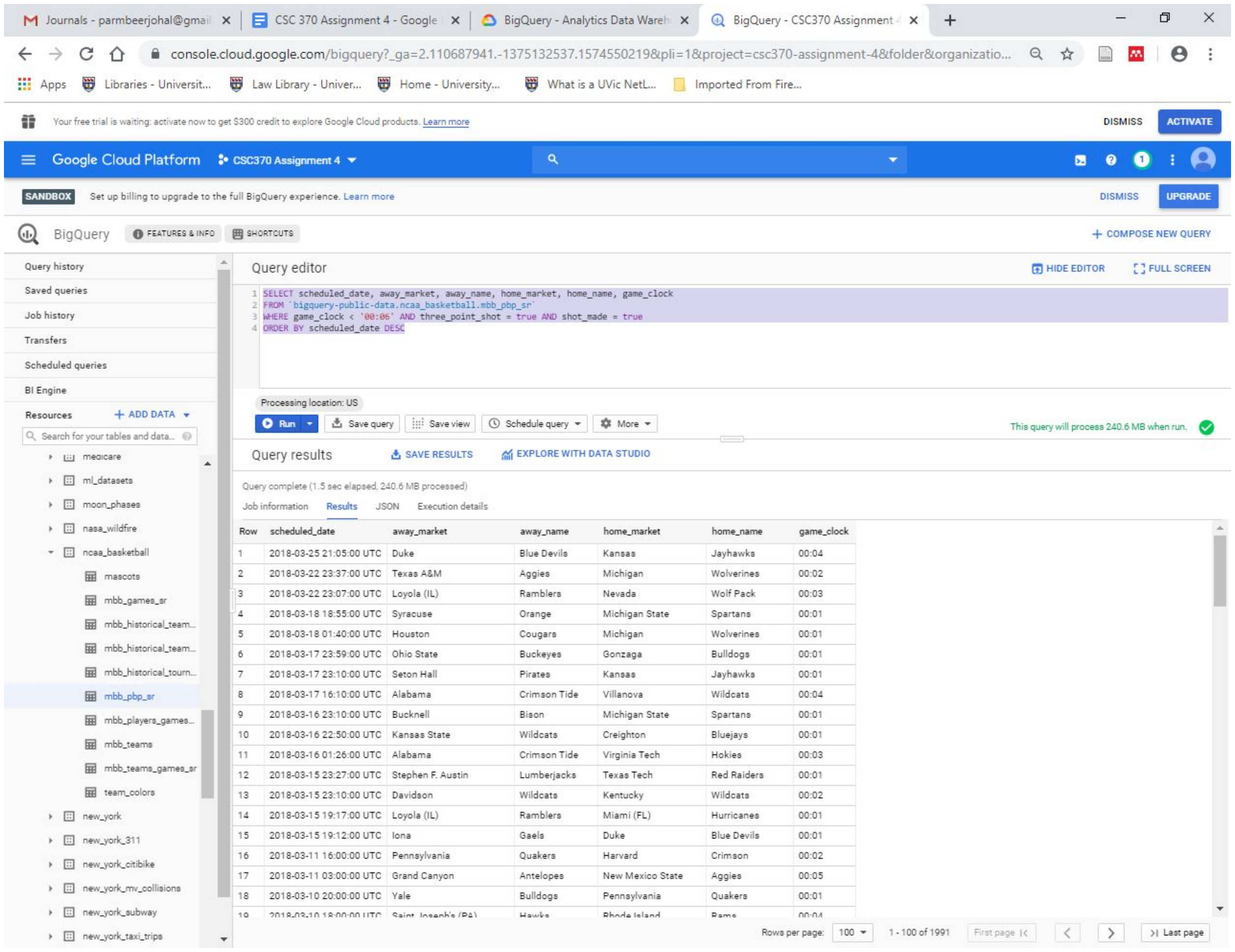
Query complete (0.7 sec elapsed, 3.1 KB processed)

Row	color	no_of_teams
1	#141414	9
2	#003366	5
3	#002147	4
4	#000000	3
5	#660000	3
6	#98002e	3
7	#CC0000	3
8	#a80532	3
9	#be0f34	3
10	#000066	2
11	#002144	2
12	#002663	2
13	#002a5c	2
14	#0039a6	2
15	#005030	2
16	#005737	2
17	#005dab	2
18	#163e68	2
19	#502d7f	2

Rows per page: 100 1 - 100 of 301

18. All the games (scheduled date, teams involved) where a 3-point shot was made in the final 5 seconds.

SELECT scheduled_date, away_market, away_name, home_market, home_name, game_clock
FROM `bigquery-public-data.ncaa_basketball.mbb_pbp_sr`
WHERE game_clock < '00:06' AND three_point_shot = true AND shot_made = true
ORDER BY scheduled_date DESC



```
SELECT scheduled_date, away_market, away_name, home_market, home_name, game_clock
FROM `bigquery-public-data.ncaa_basketball.mbb_pbp_sr`
WHERE game_clock < '00:06' AND three_point_shot = true AND shot_made = true
ORDER BY scheduled_date DESC
```

Processing location: US

This query will process 240.6 MB when run.

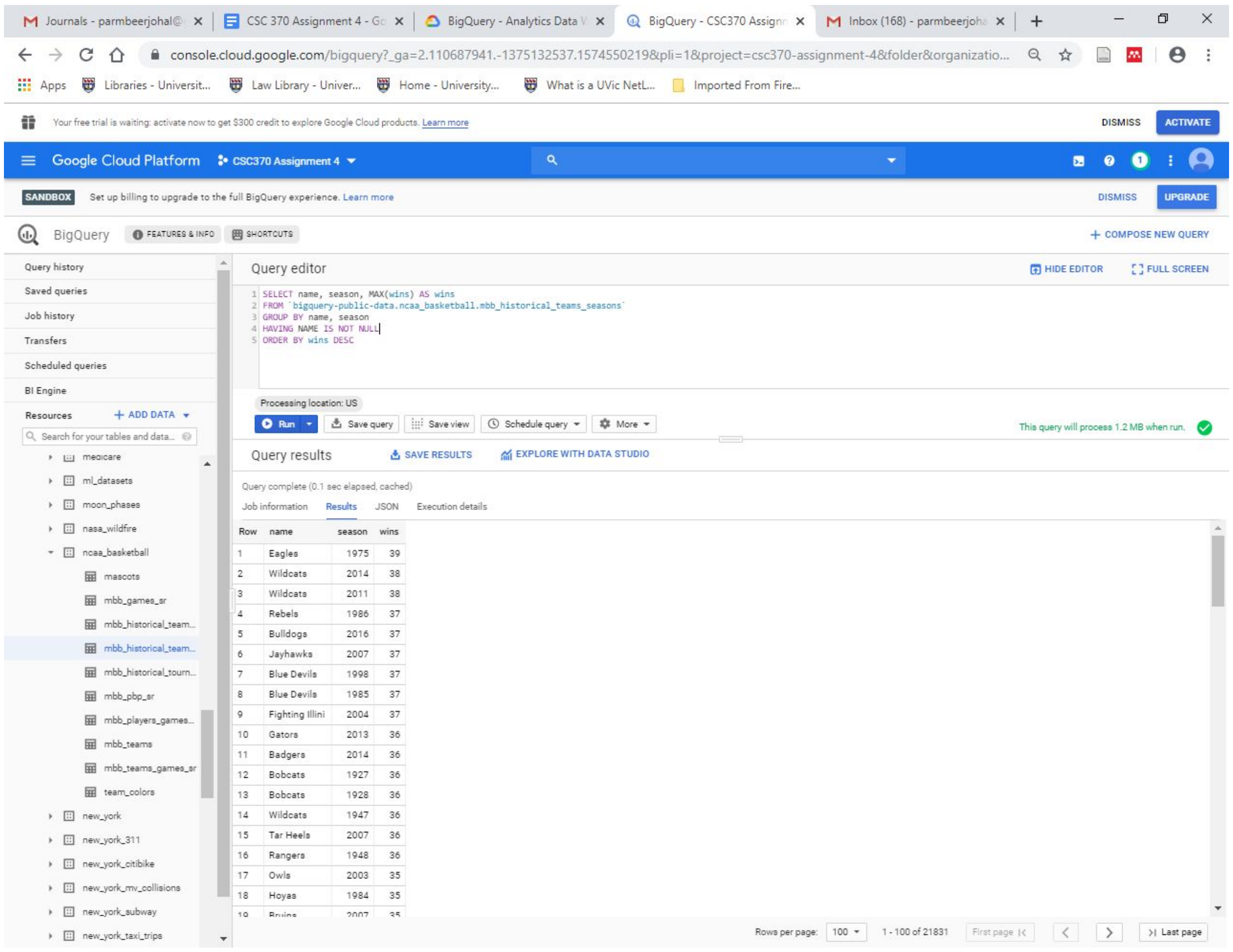
Query complete (1.5 sec elapsed, 240.6 MB processed)

Row	scheduled_date	away_market	away_name	home_market	home_name	game_clock
1	2018-03-25 21:05:00 UTC	Duke	Blue Devils	Kansas	Jayhawks	00:04
2	2018-03-22 23:37:00 UTC	Texas A&M	Aggies	Michigan	Wolverines	00:02
3	2018-03-22 23:07:00 UTC	Loyola (IL)	Ramblers	Nevada	Wolf Pack	00:03
4	2018-03-18 18:55:00 UTC	Syracuse	Orange	Michigan State	Spartans	00:01
5	2018-03-18 01:40:00 UTC	Houston	Cougars	Michigan	Wolverines	00:01
6	2018-03-17 23:59:00 UTC	Ohio State	Buckeyes	Gonzaga	Bulldogs	00:01
7	2018-03-17 23:10:00 UTC	Seton Hall	Pirates	Kansas	Jayhawks	00:01
8	2018-03-17 16:10:00 UTC	Alabama	Crimson Tide	Villanova	Wildcats	00:04
9	2018-03-16 23:10:00 UTC	Bucknell	Bison	Michigan State	Spartans	00:01
10	2018-03-16 22:50:00 UTC	Kansas State	Wildcats	Creighton	Bluejays	00:01
11	2018-03-16 01:26:00 UTC	Alabama	Crimson Tide	Virginia Tech	Hokies	00:03
12	2018-03-15 23:27:00 UTC	Stephen F. Austin	Lumberjacks	Texas Tech	Red Raiders	00:01
13	2018-03-15 23:10:00 UTC	Davidson	Wildcats	Kentucky	Wildcats	00:02
14	2018-03-15 19:17:00 UTC	Loyola (IL)	Ramblers	Miami (FL)	Hurricanes	00:01
15	2018-03-15 19:12:00 UTC	Iona	Gaels	Duke	Blue Devils	00:01
16	2018-03-11 16:00:00 UTC	Pennsylvania	Quakers	Harvard	Crimson	00:02
17	2018-03-11 03:00:00 UTC	Grand Canyon	Antelopes	New Mexico State	Aggies	00:05
18	2018-03-10 20:00:00 UTC	Yale	Bulldogs	Pennsylvania	Quakers	00:01
19	2018-03-10 18:00:00 UTC	Saint Joseph's (PA)	Hawks	Rhode Island	Rams	00:04

Rows per page: 100 1 - 100 of 1991

19. What are the rankings of the most wins by a team in a season in history?

```
SELECT name, season, MAX(wins) AS wins
FROM `bigquery-public-data.ncaa_basketball.mbb_historical_teams_seasons`
GROUP BY name, season
HAVING NAME IS NOT NULL
ORDER BY wins DESC
```



```
SELECT name, season, MAX(wins) AS wins
FROM `bigquery-public-data.ncaa_basketball.mbb_historical_teams_seasons`
GROUP BY name, season
HAVING NAME IS NOT NULL
ORDER BY wins DESC
```

Processing location: US

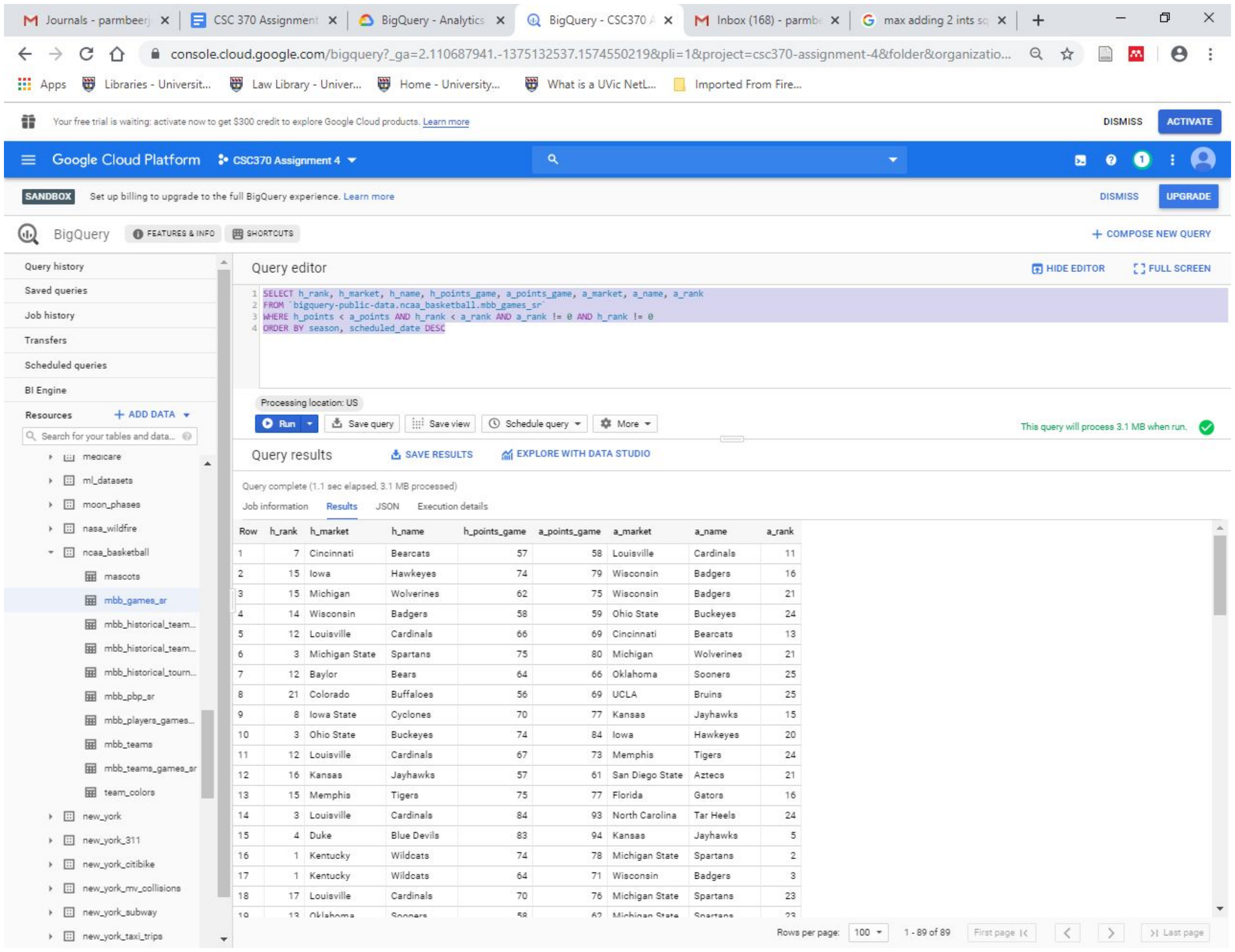
This query will process 1.2 MB when run.

Query complete (0.1 sec elapsed, cached)

Row	name	season	wins
1	Eagles	1975	39
2	Wildcats	2014	38
3	Wildcats	2011	38
4	Rebels	1986	37
5	Bulldogs	2016	37
6	Jayhawks	2007	37
7	Blue Devils	1998	37
8	Blue Devils	1985	37
9	Fighting Illini	2004	37
10	Gators	2013	36
11	Badgers	2014	36
12	Bobcats	1927	36
13	Bobcats	1928	36
14	Wildcats	1947	36
15	Tar Heels	2007	36
16	Rangers	1948	36
17	Owls	2003	35
18	Hoyas	1984	35
19	Bruins	2007	35

Rows per page: 100 1 - 100 of 21831

20. **Which games had home team upsets starting with the most recent ones? (upsets are when a higher ranked team loses to a lower ranked one)**
SELECT h_rank, h_market, h_name, h_points_game, a_points_game, a_market, a_name, a_rank
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
WHERE h_points < a_points AND h_rank < a_rank AND a_rank != 0 AND h_rank != 0
ORDER BY season, scheduled_date DESC



Query editor

```
SELECT h_rank, h_market, h_name, h_points_game, a_points_game, a_market, a_name, a_rank
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
WHERE h_points < a_points AND h_rank < a_rank AND a_rank != 0 AND h_rank != 0
ORDER BY season, scheduled_date DESC
```

Processing location: US

This query will process 3.1 MB when run.

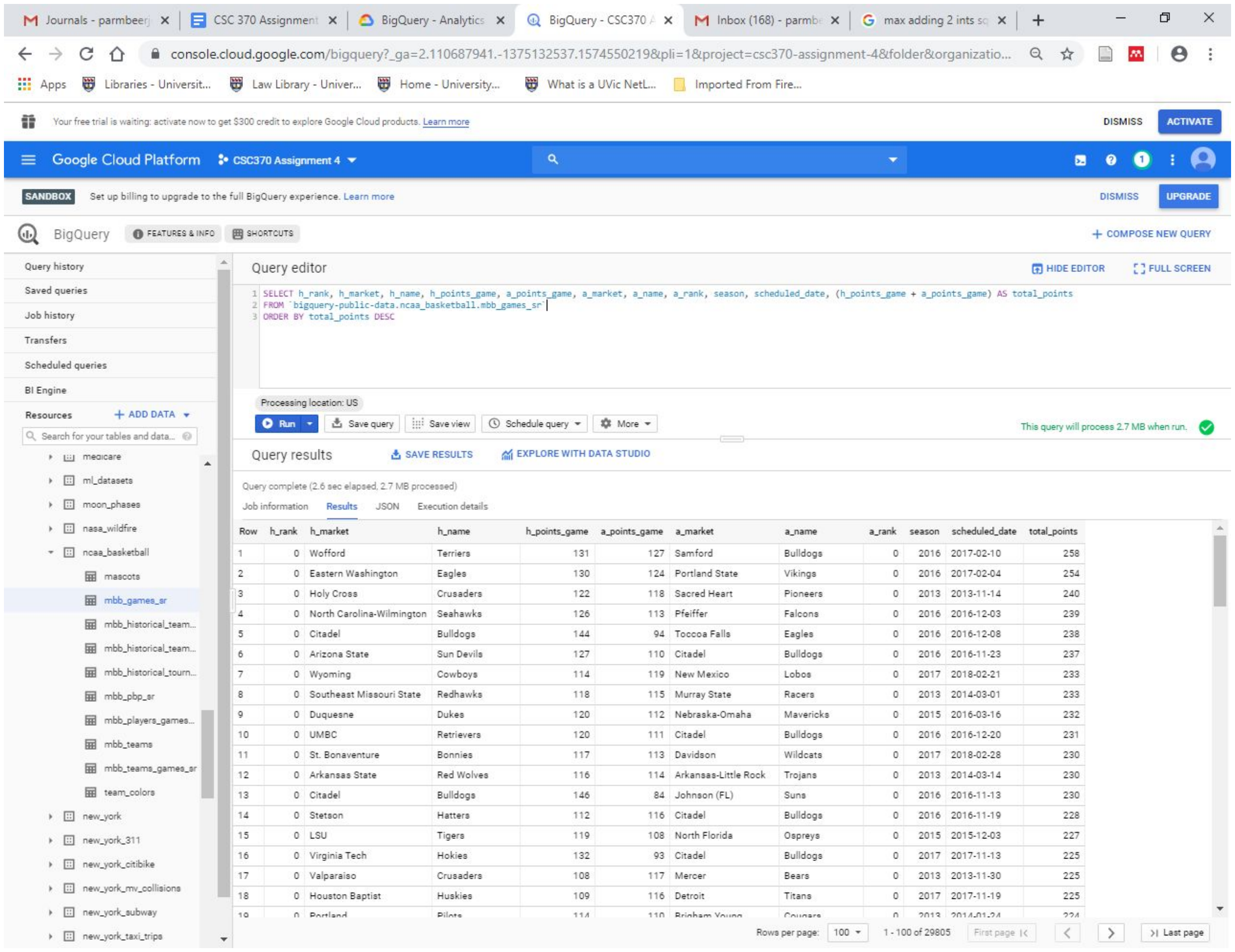
Query results

Query complete (1.1 sec elapsed, 3.1 MB processed)

Row	h_rank	h_market	h_name	h_points_game	a_points_game	a_market	a_name	a_rank
1	7	Cincinnati	Bearcats	57	58	Louisville	Cardinals	11
2	15	Iowa	Hawkeyes	74	79	Wisconsin	Badgers	16
3	15	Michigan	Wolverines	62	75	Wisconsin	Badgers	21
4	14	Wisconsin	Badgers	58	59	Ohio State	Buckeyes	24
5	12	Louisville	Cardinals	66	69	Cincinnati	Bearcats	13
6	3	Michigan State	Spartans	75	80	Michigan	Wolverines	21
7	12	Baylor	Bears	64	66	Oklahoma	Sooners	25
8	21	Colorado	Buffaloes	56	69	UCLA	Bruins	25
9	8	Iowa State	Cyclones	70	77	Kansas	Jayhawks	15
10	3	Ohio State	Buckeyes	74	84	Iowa	Hawkeyes	20
11	12	Louisville	Cardinals	67	73	Memphis	Tigers	24
12	16	Kansas	Jayhawks	57	61	San Diego State	Aztecs	21
13	15	Memphis	Tigers	75	77	Florida	Gators	16
14	3	Louisville	Cardinals	84	93	North Carolina	Tar Heels	24
15	4	Duke	Blue Devils	83	94	Kansas	Jayhawks	5
16	1	Kentucky	Wildcats	74	78	Michigan State	Spartans	2
17	1	Kentucky	Wildcats	64	71	Wisconsin	Badgers	3
18	17	Louisville	Cardinals	70	76	Michigan State	Spartans	23
19	13	Oklahoma	Sooners	58	62	Michigan State	Spartans	23

Rows per page: 100 1 - 89 of 89

21. **What are the rankings of the most amount of points scored in a game since the 2013-2014 season?**
SELECT h_rank, h_market, h_name, h_points_game, a_points_game, a_market, a_name, a_rank, season, scheduled_date, (h_points_game + a_points_game) AS total_points
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
ORDER BY total_points DESC



```
SELECT h_rank, h_market, h_name, h_points_game, a_points_game, a_market, a_name, a_rank, season, scheduled_date, (h_points_game + a_points_game) AS total_points
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
ORDER BY total_points DESC
```

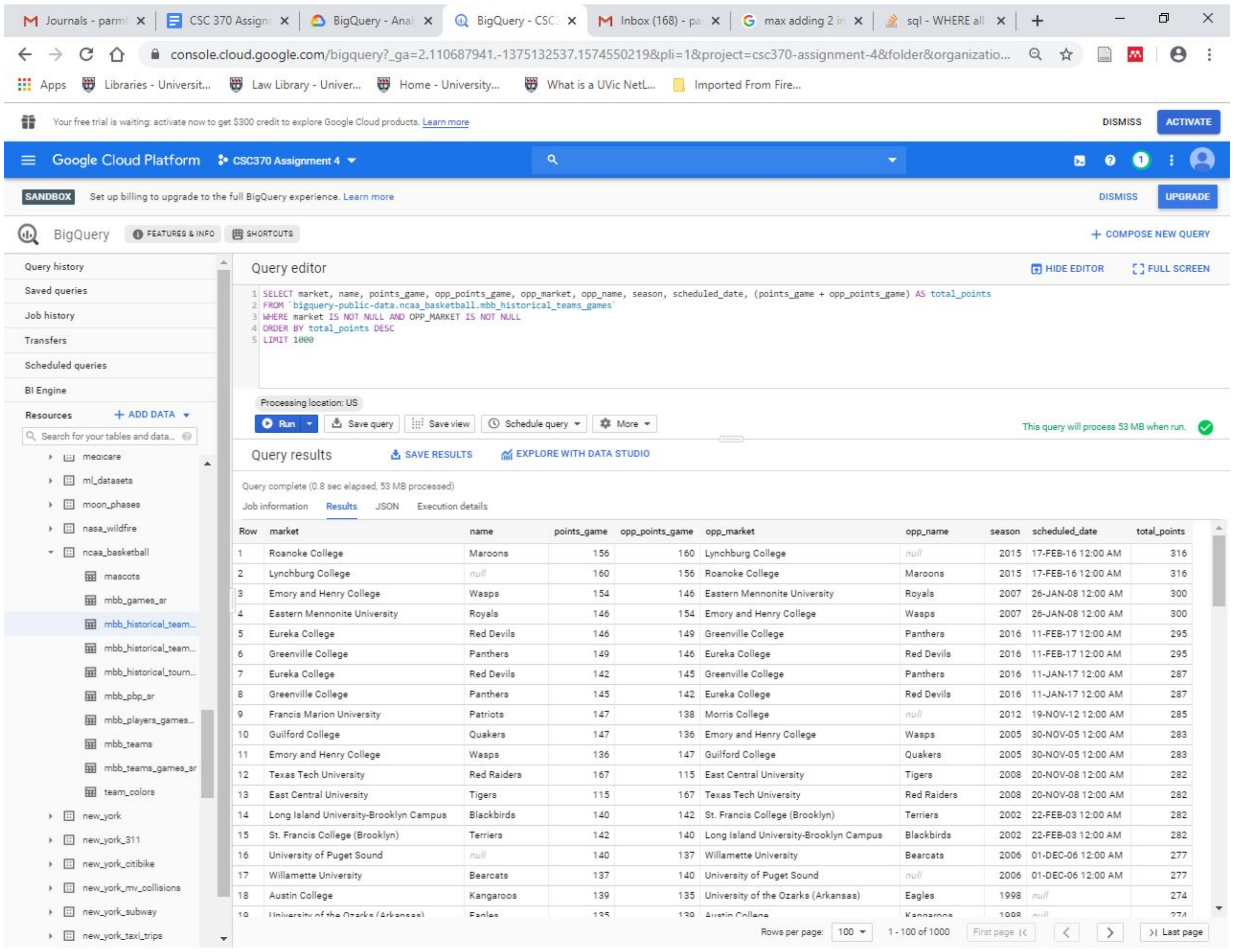
Query complete (2.6 sec elapsed, 2.7 MB processed)

Row	h_rank	h_market	h_name	h_points_game	a_points_game	a_market	a_name	a_rank	season	scheduled_date	total_points
1	0	Wofford	Terriers	131	127	Samford	Bulldogs	0	2016	2017-02-10	258
2	0	Eastern Washington	Eagles	130	124	Portland State	Vikings	0	2016	2017-02-04	254
3	0	Holy Cross	Crusaders	122	118	Sacred Heart	Pioneers	0	2013	2013-11-14	240
4	0	North Carolina-Wilmington	Seahawks	126	113	Pfeiffer	Falcons	0	2016	2016-12-03	239
5	0	Citadel	Bulldogs	144	94	Toccoa Falls	Eagles	0	2016	2016-12-08	238
6	0	Arizona State	Sun Devils	127	110	Citadel	Bulldogs	0	2016	2016-11-23	237
7	0	Wyoming	Cowboys	114	119	New Mexico	Lobos	0	2017	2018-02-21	233
8	0	Southeast Missouri State	Redhawks	118	115	Murray State	Racers	0	2013	2014-03-01	233
9	0	Duquesne	Dukes	120	112	Nebraska-Omaha	Mavericks	0	2015	2016-03-16	232
10	0	UMBC	Retrievers	120	111	Citadel	Bulldogs	0	2016	2016-12-20	231
11	0	St. Bonaventure	Bonnies	117	113	Davidson	Wildcats	0	2017	2018-02-28	230
12	0	Arkansas State	Red Wolves	116	114	Arkansas-Little Rock	Trojans	0	2013	2014-03-14	230
13	0	Citadel	Bulldogs	146	84	Johnson (FL)	Suns	0	2016	2016-11-13	230
14	0	Stetson	Hatters	112	116	Citadel	Bulldogs	0	2016	2016-11-19	228
15	0	LSU	Tigers	119	108	North Florida	Ospreys	0	2015	2015-12-03	227
16	0	Virginia Tech	Hokies	132	93	Citadel	Bulldogs	0	2017	2017-11-13	225
17	0	Valparaiso	Crusaders	108	117	Mercer	Bears	0	2013	2013-11-30	225
18	0	Houston Baptist	Huskies	109	116	Detroit	Titans	0	2017	2017-11-19	225
19	0	Portland	Pilots	114	110	Brigham Young	Cougars	0	2013	2014-01-24	224

Rows per page: 100 1 - 100 of 29805

22. What are the rankings of the most amount of points scored in a game in history? (limit is 1000 teams)

SELECT market, name, points_game, opp_points_game, opp_market, opp_name, season, scheduled_date, (points_game + opp_points_game) AS total_points
FROM `bigquery-public-data.ncaa_basketball.mbb_historical_teams_games`
WHERE market IS NOT NULL AND OPP_MARKET IS NOT NULL
ORDER BY total_points DESC
LIMIT 1000



```
SELECT market, name, points_game, opp_points_game, opp_market, opp_name, season, scheduled_date, (points_game + opp_points_game) AS total_points
FROM `bigquery-public-data.ncaa_basketball.mbb_historical_teams_games`
WHERE market IS NOT NULL AND OPP_MARKET IS NOT NULL
ORDER BY total_points DESC
LIMIT 1000
```

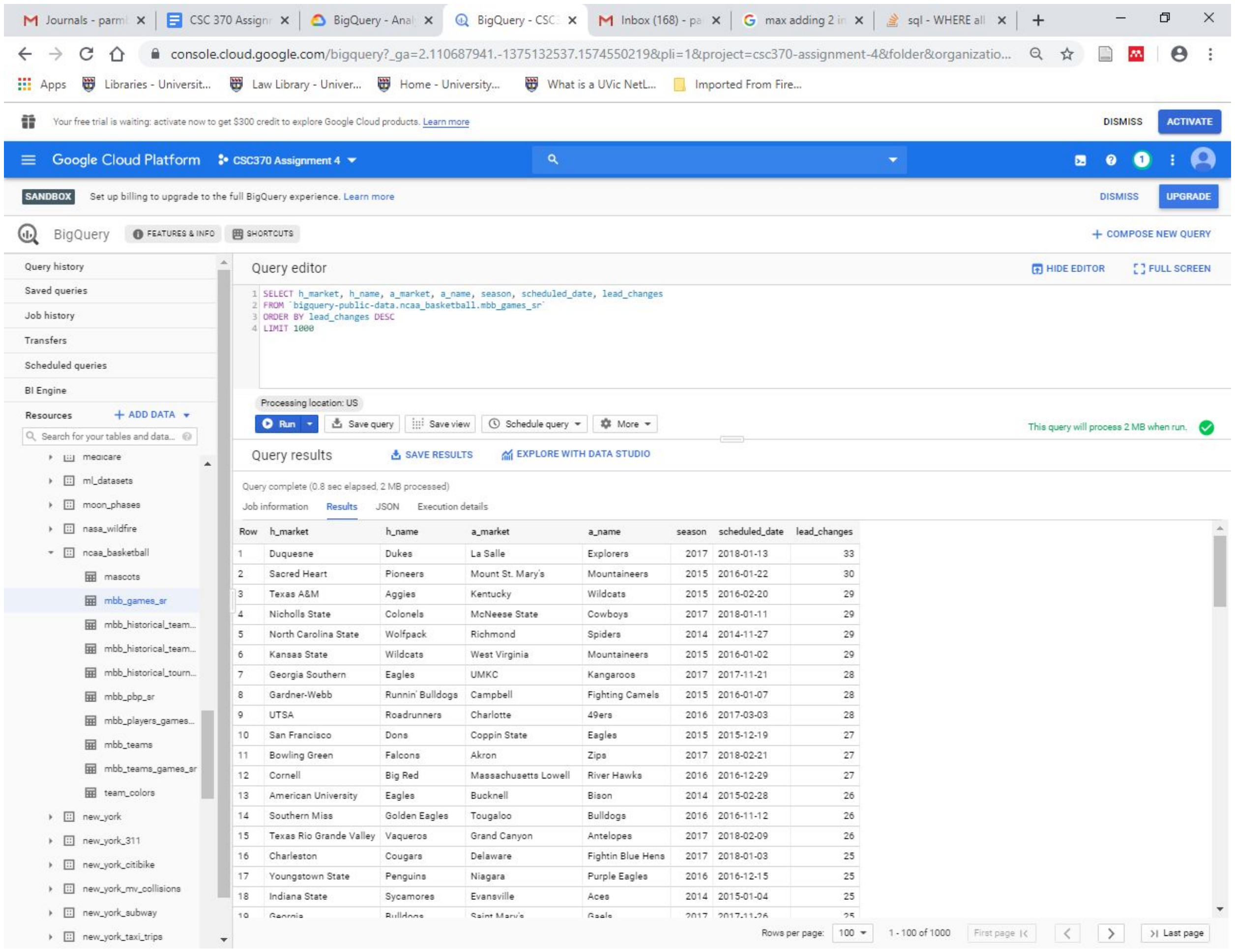
Query complete (0.8 sec elapsed, 53 MB processed)

Row	market	name	points_game	opp_points_game	opp_market	opp_name	season	scheduled_date	total_points
1	Roanoke College	Maroons	156	160	Lynchburg College	null	2015	17-FEB-16 12:00 AM	316
2	Lynchburg College	null	160	156	Roanoke College	Maroons	2015	17-FEB-16 12:00 AM	316
3	Emory and Henry College	Wasps	154	146	Eastern Mennonite University	Royals	2007	26-JAN-08 12:00 AM	300
4	Eastern Mennonite University	Royals	146	154	Emory and Henry College	Wasps	2007	26-JAN-08 12:00 AM	300
5	Eureka College	Red Devils	146	149	Greenville College	Panthers	2016	11-FEB-17 12:00 AM	295
6	Greenville College	Panthers	149	146	Eureka College	Red Devils	2016	11-FEB-17 12:00 AM	295
7	Eureka College	Red Devils	142	145	Greenville College	Panthers	2016	11-JAN-17 12:00 AM	287
8	Greenville College	Panthers	145	142	Eureka College	Red Devils	2016	11-JAN-17 12:00 AM	287
9	Francis Marion University	Patriots	147	138	Morris College	null	2012	19-NOV-12 12:00 AM	285
10	Guilford College	Quakers	147	136	Emory and Henry College	Wasps	2005	30-NOV-05 12:00 AM	283
11	Emory and Henry College	Wasps	136	147	Guilford College	Quakers	2005	30-NOV-05 12:00 AM	283
12	Texas Tech University	Red Raiders	167	115	East Central University	Tigers	2008	20-NOV-08 12:00 AM	282
13	East Central University	Tigers	115	167	Texas Tech University	Red Raiders	2008	20-NOV-08 12:00 AM	282
14	Long Island University-Brooklyn Campus	Blackbirds	140	142	St. Francis College (Brooklyn)	Terriers	2002	22-FEB-03 12:00 AM	282
15	St. Francis College (Brooklyn)	Terriers	142	140	Long Island University-Brooklyn Campus	Blackbirds	2002	22-FEB-03 12:00 AM	282
16	University of Puget Sound	null	140	137	Willamette University	Bearcats	2006	01-DEC-06 12:00 AM	277
17	Willamette University	Bearcats	137	140	University of Puget Sound	null	2006	01-DEC-06 12:00 AM	277
18	Austin College	Kangaroos	139	135	University of the Ozarks (Arkansas)	Eagles	1998	null	274
19	University of the Ozarks (Arkansas)	Eagles	135	139	Austin College	Kangaroos	1998	null	274

Rows per page: 100 1 - 100 of 1000

23. What are the most lead changes in a game since the 2013-2014 season?

SELECT h_market, h_name, a_market, a_name, season, scheduled_date, lead_changes
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
ORDER BY lead_changes DESC
LIMIT 1000



```
SELECT h_market, h_name, a_market, a_name, season, scheduled_date, lead_changes
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
ORDER BY lead_changes DESC
LIMIT 1000
```

Processing location: US

This query will process 2 MB when run.

Query complete (0.8 sec elapsed, 2 MB processed)

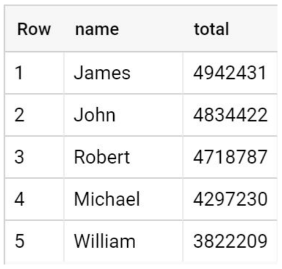
Row	h_market	h_name	a_market	a_name	season	scheduled_date	lead_changes
1	Duquesne	Dukes	La Salle	Explorers	2017	2018-01-13	33
2	Sacred Heart	Pioneers	Mount St. Mary's	Mountaineers	2015	2016-01-22	30
3	Texas A&M	Aggies	Kentucky	Wildcats	2015	2016-02-20	29
4	Nicholls State	Colonels	McNeese State	Cowboys	2017	2018-01-11	29
5	North Carolina State	Wolfpack	Richmond	Spiders	2014	2014-11-27	29
6	Kansas State	Wildcats	West Virginia	Mountaineers	2015	2016-01-02	29
7	Georgia Southern	Eagles	UMKC	Kangaroos	2017	2017-11-21	28
8	Gardner-Webb	Runnin' Bulldogs	Campbell	Fighting Camels	2015	2016-01-07	28
9	UTSA	Roadrunners	Charlotte	49ers	2016	2017-03-03	28
10	San Francisco	Dons	Coppin State	Eagles	2015	2015-12-19	27
11	Bowling Green	Falcons	Akron	Zips	2017	2018-02-21	27
12	Cornell	Big Red	Massachusetts Lowell	River Hawks	2016	2016-12-29	27
13	American University	Eagles	Bucknell	Bison	2014	2015-02-28	26
14	Southern Miss	Golden Eagles	Tougaloo	Bulldogs	2016	2016-11-12	26
15	Texas Rio Grande Valley	Vaqueros	Grand Canyon	Antelopes	2017	2018-02-09	26
16	Charleston	Cougars	Delaware	Fightin Blue Hens	2017	2018-01-03	25
17	Youngstown State	Penguins	Niagara	Purple Eagles	2016	2016-12-15	25
18	Indiana State	Sycamores	Evansville	Aces	2014	2015-01-04	25
19	Georgia	Bulldogs	Saint Mary's	Gaels	2017	2017-11-26	25

Rows per page: 100 1 - 100 of 1000

24.

SELECT name, sum(number) as total
FROM `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY name
ORDER BY sum(number) DESC
LIMIT 1000

-I wanted to see the total number of people sharing the same name in the USA. The name James was the most popular with 4942431 people sharing the name in 2013.

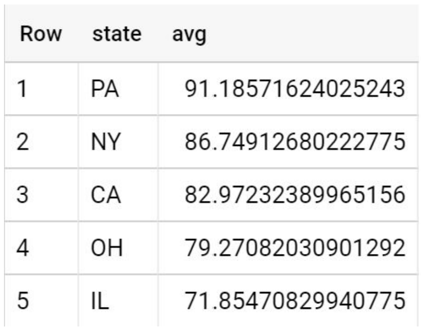


Row	name	total
1	James	4942431
2	John	4834422
3	Robert	4718787
4	Michael	4297230
5	William	3822209

25.

```
SELECT state, avg(number) as avg
FROM `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY state
ORDER BY avg desc
LIMIT 1000
```

-Here I wanted to display the average number of people who share one name in each state. The query takes the average of the number column when grouped by the state column and displays a row for every state with an average as the output.

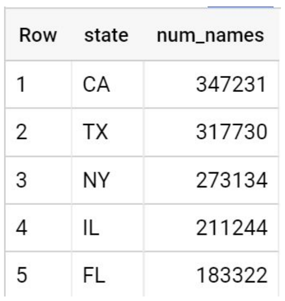


Row	state	avg
1	PA	91.18571624025243
2	NY	86.74912680222775
3	CA	82.97232389965156
4	OH	79.27082030901292
5	IL	71.85470829940775

26.

```
SELECT state, COUNT(*) as num_names
FROM `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY state
ORDER BY num_names DESC
LIMIT 1000
```

-This outputs a row for each state and a total count of the unique names which applied for a social security card in 2013.

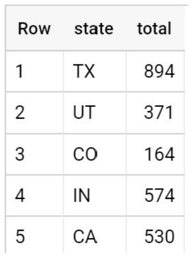


Row	state	num_names
1	CA	347231
2	TX	317730
3	NY	273134
4	IL	211244
5	FL	183322

27.

```
SELECT state, sum(number) as total
FROM `bigquery-public-data.usa_names.usa_1910_2013`
WHERE name = 'Braydon'
GROUP BY state
LIMIT 1000
```

- I wanted to see how many people with my name applied for a social security card in 2013. According to the results, Texas has the most legends with 894 people named Braydon who applied.

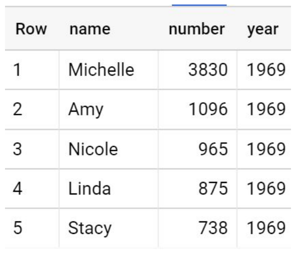


Row	state	total
1	TX	894
2	UT	371
3	CO	164
4	IN	574
5	CA	530

28.

```
SELECT name, number, year
FROM `bigquery-public-data.usa_names.usa_1910_2013`
WHERE state = 'CA' AND year = 1969
LIMIT 1000
```

-I wanted to see the highest number of names of SS card applicants born in the year 1969 from California. Surprising the top result was MIchelle.

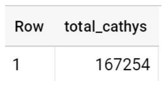


Row	name	number	year
1	Michelle	3830	1969
2	Amy	1096	1969
3	Nicole	965	1969
4	Linda	875	1969
5	Stacy	738	1969

29.

```
SELECT sum(number) as total_cathys
FROM `bigquery-public-data.usa_names.usa_1910_2013`
WHERE name = 'Cathy'
LIMIT 1000
```

-Query to see the total number of people named Cathy who applied for a SS card from 1910-2013. I chose this because Cathy is my Mom's name.

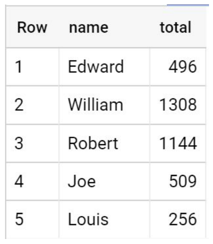


Row	total_cathys
1	167254

30.

```
SELECT name, sum(number) as total
FROM (SELECT state, name , number
    FROM `bigquery-public-data.usa_names.usa_1910_2013`
    WHERE gender = 'M'
    LIMIT 1000)
GROUP BY name;
```

-Tried a nested query where I first select all the rows of males and then do a query to display the number of people with the same name who applied for a SS card from 1910-2013.



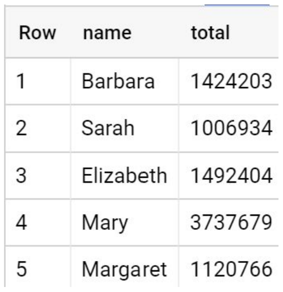
Row	name	total
1	Edward	496
2	William	1308
3	Robert	1144
4	Joe	509
5	Louis	256

31.

```
SELECT name, total
```

```
FROM (SELECT name, sum(number) as total
    FROM `bigquery-public-data.usa_names.usa_1910_2013`
    GROUP BY name
    LIMIT 1000)
WHERE total >= 1000000
```

-Selected the total number of names and then re-queried on the table to select only those rows where the total exceeds 1000000 names.

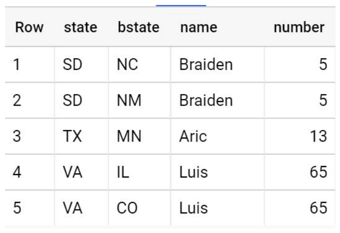


Row	name	total
1	Barbara	1424203
2	Sarah	1006934
3	Elizabeth	1492404
4	Mary	3737679
5	Margaret	1120766

32.

```
SELECT DISTINCT A.state, B.state as bstate, A.name, B.number
FROM `bigquery-public-data.usa_names.usa_1910_2013` A,
`bigquery-public-data.usa_names.usa_1910_2013` B
WHERE A.name = B.name AND
    A.state > B.state AND
    A.number = B.number
LIMIT 1000
```

-Wanted to select the states and names that share the same number of applicants for any name.

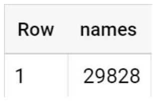


Row	state	bstate	name	number
1	SD	NC	Braiden	5
2	SD	NM	Braiden	5
3	TX	MN	Aric	13
4	VA	IL	Luis	65
5	VA	CO	Luis	65

33.

```
SELECT COUNT(DISTINCT name) as names
FROM (SELECT name
    FROM `bigquery-public-data.usa_names.usa_1910_2013`)
LIMIT 1000
```

-Count the distinct names in the whole table. By the pigeonhole principle we know we need at least 29829 people in one room to guarantee that two people share a name.



Row	names
1	29828