FINAL PROJECT PHASE II - Databases

Team Members:

Gabby Kang (gkang9@jhu.edu) - 415

Vijay Murari Tiyyala (vtiyyal1@jhu.edu) - 615

Target Domain:

In our target domain, we focus on entertainment statistics, which include statistics about movies, television shows, ratings, views, watch hours, as well as data about various games and Twitch streams.

Accessing Project:

The project can be accessed at https://github.com/gabbykang02/DBFinalProject

3. Phase 1 Submission Changes

Overall, we have changed some parts of the DDL structure, particularly with the keys. We have also reduced focus on querying intra-table relationships in favor of focusing on time dimension changes.

4. Populating data

The database can be populated by running table_creation.sql in a mysql environment. Afterwards, the database tables can be populated by running each of the remaining sql files, in which each sql file corresponds to one table; for example imdb_basics.sql populated the Basics table. Currently, the repo has a smaller subset of the larger imdb databases due to file size restrictions (ie: imdb_basics_small.sql is 1% of the data in the imdb_basics file). If you would like to input the full database, you may go to the imdb database, download the specific file locally, and run tsv_to_sql.ipynb file (changing parameters and comments as needed) to generate a sql file of INSERT statements.

Source	Summary	Method of Extraction
IMDB: IMDB	7 databases including info on title, people involved, ratings, type of media, etc.	Thread tsv file through a separate script, read each entry value and convert into SQL 'INSERT INTO . . . " statement
Twitch: longtermstats	Long term statistics on twitch statistics: hours watched, affiliates, partners, average viewers etc.	Thread csv file through a separate script, read each entry value and convert into SQL 'INSERT INTO . . . " statement
Twitch Game Streams: peakviewers	Long term stats on what games have been streamed	Thread csv file through a separate script, read each entry value and convert into SQL 'INSERT INTO . . . " statement
Twitch Streamer Stats (Most Watched): watched	Long term stats on what individual streamers have done	Thread csv file through a separate script, read each entry value and convert into SQL 'INSERT INTO . . . " statement
Metacritic reviews: top-video-games	Info on game title, review score, platform, date, etc.	Thread CSV into insert file

Source	Summary	Method of Extraction
VGChartz: vgchartz	Databases including title console, genres, and sales estimates	We used this crawler to generate the csv file vgchartz-crawler
Covid Data jhu covid data jhu covid master	Database includes data like date, country cases, population, new cases etc	Thread csv file through a separate script, read each entry value and convert into SQL 'INSERT INTO ..." statement

5. Platform

The database was hosted on dbase.cs.jhu.edu.

6. User Guide

Clone the repo and open up a database connection. Run all of the sql files in the insert_files folder in order to populate the database. You may run any of the available stored procedure listed below in the database terminal:

Stored Procedure Name	**Inputs**	**Purpose**
GetMaxCovid	country VARCHAR	Returns the month, year, and number of cases that occurred in @country
getAverageRatings	year INT	Returns type, average rating, number of pieces of media, and average run time of all pieces of film media released in @year
GetCovidPlatforms	year INT month INT	Return platform, number of games relased, average metascore rating of all games released in @month, @year

Stored Procedure Name	**Inputs**	**Purpose**
GetStatCumulative	country VARCHAR startMonth INT startYear INT stopMonth INT stopYear INT	Returns all TwitchStats as well as CovidData stats for @country between @startMonth, @startYear and @stopMonth, @stopYear. When paired with the UI, plots all statistics on the same plot over time as a line plot and plots each statistics as an interactive sunburst plot.
GetTwitchStats	command VARCHAR	Returns the month, year, and twitch stat as specified by @command. When paired with the UI, plots the specified twitch stat as an interactive sunburst plot.
GetGameScoresPerGenre	genre VARCHAR year VARCHAR	Returns the month, year, average scores, and average number of gamesales for @genre games released after @year
getmovieoninputgenre	p__year INT p__genre VARCHAR	Returns the originalTitle, averageRating, startyear, genres, and type of the top 10 rated pieces of film media released after @p__year that have @p__genre

Stored Procedure Name	**Inputs**	**Purpose**
getmoviesofall	p__year INT	Returns the originalTitle, averageRating, startYear, genres, and type of the top 2 film media with the highest rating split up by all genre combinations that were released after @p__year
gettopchannelinlanguage	p__language VARCHAR	Returns the number of followers, name, and language of the channels with longest watch time for @p__language
GETTOPCHANNELSBY LANGUAGE		Returns the watchtime, name, and language the channel with the longest watch time for each language
getmostwatchedgame	p__year INT p__continent VARCHAR p__country VARCHAR	Returns the most viewed game on twitch during the peak covid cases month in @p__year given @p__country in @p__continent
gettopwatchedgames	p__year INT	Returns the game with the top view on Twitch during the peak covid case month of @p__year

For table based queries, we offer a continuous terminal based access. After populating the database, setup a python virtual environment in the DBFinal-Project folder. Import python packages for matplotlib, pymsql, pandas, and os. Adjust the pymsql.connect command to link to the database in which you've installed all the data, and then run the report__generation.py script to open up the interface. Use "q" to stop the script and "h" to receive a list of all available stored procedures. From there you may call any of the stored procedures listed in the h command to interact with the database.

If you are able to connect to a database on an environment compatible with python notebooks, you may run report__generation.ipynb to have access to sunburst charts and line graphs that are used in the GetTwitchStats and Get-

StatCumulative procedures. The setup and use for the .ipynb is equivalent to the .py report_generation setup.

7. Areas of Specialization

We focused on UI/report generation and usage of Advanced SQL - cursors for our main topic and web scraping and resolving complex data extraction issues from online data sources as minor topics. Our project utilizes Python plots (ex: sunburst plots) such that, given a few input parameters, you can query for a particular subset of data and freely interact with further subsets of the data without submitting another query. This allows us to have an interactive visualization of the data for extrapolation of further parameters.

Additionally, we used cursors to get some key insights from our huge database by implementing in some of our stored procedures. Our minor focus was on resolving complex data extraction issues from online resources.

8. Strengths

- All entertainment tables have some kind of temporal linkage. This makes it easy to associate entertainment tables with COVID data.
- Temporal linkage also allows examination of trends on other entertainment data. (ex: find the year that released the most movies in translated languages and then compare that to global game sales to see if that trend in film entertainment is mimicked in the game industry)
- Entertainment tables have many attributes and contain data on global ratings/views/sales.
- Joins on imdb tables are easy due to the maintenance of tconst and nconst ids.
- We implemented cursors in both the terminal print statements, and more in depth within the getmoviesofall and GETTOPCHANNELSBYLANGUAGE stored procedures.

9. Limitations

- A lot of interesting information, but not necessarily useful if joined to another table. Thus it becomes difficult to identify what the core functions we should focus on, and the selection of stored procedures we developed is not representative of what queries we can answer.
- Slight variations in video game title make it ineffective to join GameSales with Metacritic or TwitchGames on game name, thus making it very difficult to draw relations between these tables.

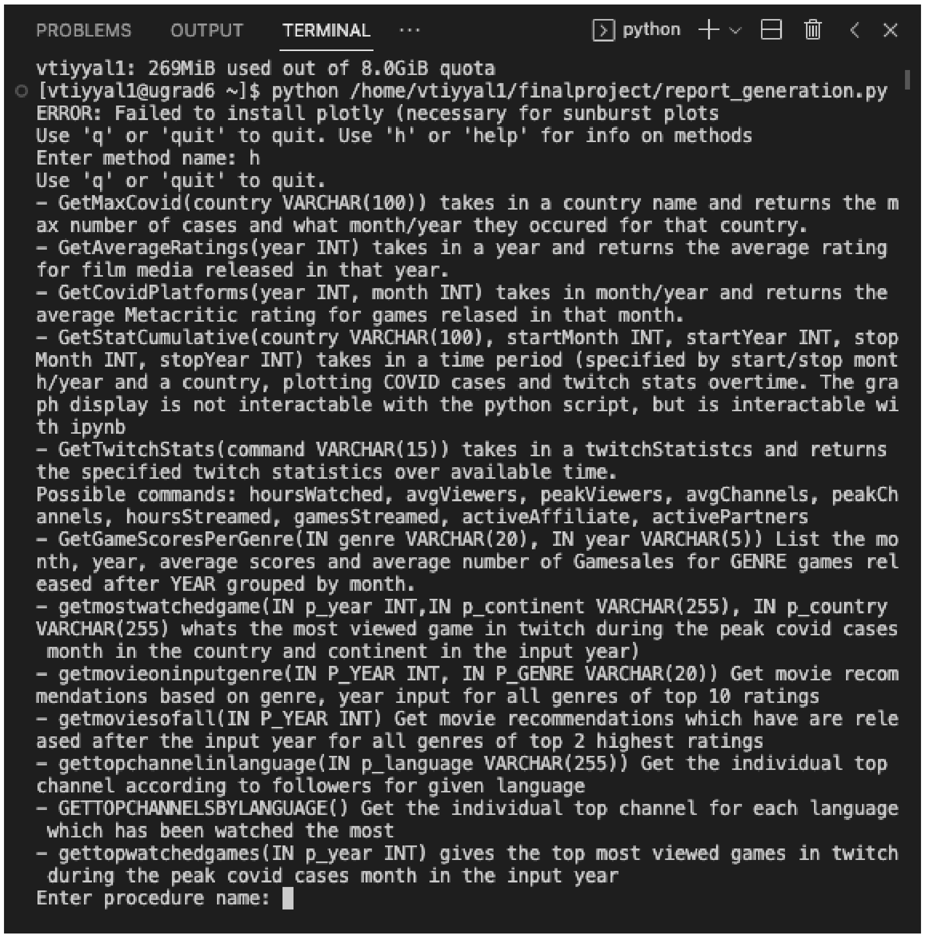
- Due to file size restrictions, we weren't able to run the full imdb databases in ugrad/db.cs.jhu.edu. This means that we weren't able to make tconst and nconst foreign keys, since we couldn't guarantee that the same small subsets of imdb were being inserted across all tables.
- Unfortunately, our cursor based sp's are incompatible with our .py and .pynb result statements and can thus only be run from the database directly.

10. External sources

We used user baynebrannen's web scraper to get the latest VGChartz data. For printing out tables to terminal, we used user Le Droid's answer in this Stack Overflow question.

11. Sample output

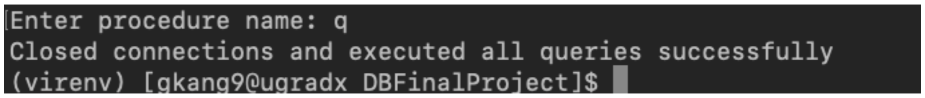
- Result of using the help command, this displays the list of procedures you can choose from to query the database.



PROBLEMS OUTPUT TERMINAL ··· python

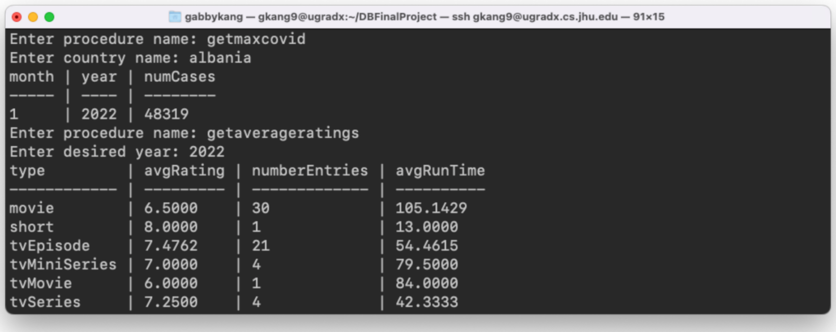
```
vtiyyal1: 269MiB used out of 8.0GiB quota
[vtiyyal1@ugrad6 ~]$ python /home/vtiyyal1/finalproject/report_generation.py
ERROR: Failed to install plotly (necessary for sunburst plots
Use 'q' or 'quit' to quit. Use 'h' or 'help' for info on methods
Enter method name: h
Use 'q' or 'quit' to quit.
- GetMaxCovid(country VARCHAR(100)) takes in a country name and returns the m
ax number of cases and what month/year they occured for that country.
- GetAverageRatings(year INT) takes in a year and returns the average rating
for film media released in that year.
- GetCovidPlatforms(year INT, month INT) takes in month/year and returns the
average Metacritic rating for games relased in that month.
- GetStatCumulative(country VARCHAR(100), startMonth INT, startYear INT, stop
Month INT, stopYear INT) takes in a time period (specified by start/stop mont
h/year and a country, plotting COVID cases and twitch stats overtime. The gra
ph display is not interactable with the python script, but is interactable wi
th ipynb
- GetTwitchStats(command VARCHAR(15)) takes in a twitchStatistcs and returns
the specified twitch statistics over available time.
Possible commands: hoursWatched, avgViewers, peakViewers, avgChannels, peakCh
annels, hoursStreamed, gamesStreamed, activeAffiliate, activePartners
- GetGameScoresPerGenre(IN genre VARCHAR(20), IN year VARCHAR(5)) List the mo
nth, year, average scores and average number of Gamesales for GENRE games rel
eased after YEAR grouped by month.
- getmostwatchedgame(IN p_year INT,IN p_continent VARCHAR(255), IN p_country
VARCHAR(255) whats the most viewed game in twitch during the peak covid cases
 month in the country and continent in the input year)
- getmovieoninputgenre(IN P_YEAR INT, IN P_GENRE VARCHAR(20)) Get movie recom
mendations based on genre, year input for all genres of top 10 ratings
- getmoviesofall(IN P_YEAR INT) Get movie recommendations which have are rele
ased after the input year for all genres of top 2 highest ratings
- gettopchannelinlanguage(IN p_language VARCHAR(255)) Get the individual top
channel according to followers for given language
- GETTOPCHANNELSBYLANGUAGE() Get the individual top channel for each language
 which has been watched the most
- gettopwatchedgames(IN p_year INT) gives the top most viewed games in twitch
 during the peak covid cases month in the input year
Enter procedure name:
```

- Sample output for closing the command on terminal



```
[Enter procedure name: q
Closed connections and executed all queries successfully
(virenv) [gkang9@ugradx DBFinalProject]$
```

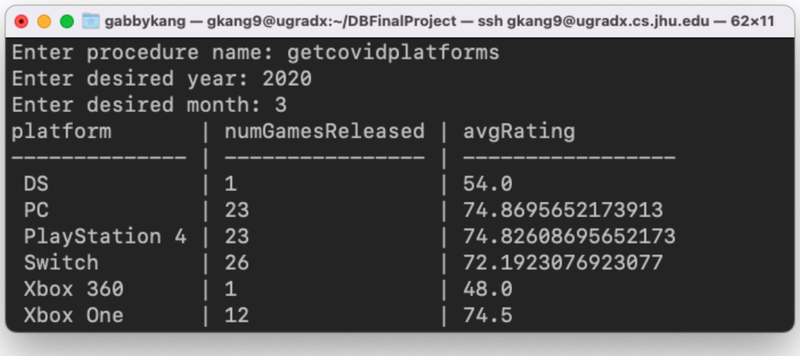
- Output of GetMaxCovid and GetAverageRatings methods



gabbykang — gkang9@ugradx:~/DBFinalProject — ssh gkang9@ugradx.cs.jhu.edu — 91×15

```
Enter procedure name: getmaxcovid
Enter country name: albania
month | year | numCases
----- | ---- | --------
1     | 2022 | 48319
Enter procedure name: getaverageratings
Enter desired year: 2022
type         | avgRating | numberEntries | avgRunTime
------------ | --------- | ------------- | ----------
movie        | 6.5000    | 30            | 105.1429
short        | 8.0000    | 1             | 13.0000
tvEpisode    | 7.4762    | 21            | 54.4615
tvMiniSeries | 7.0000    | 4             | 79.5000
tvMovie      | 6.0000    | 1             | 84.0000
tvSeries     | 7.2500    | 4             | 42.3333
```

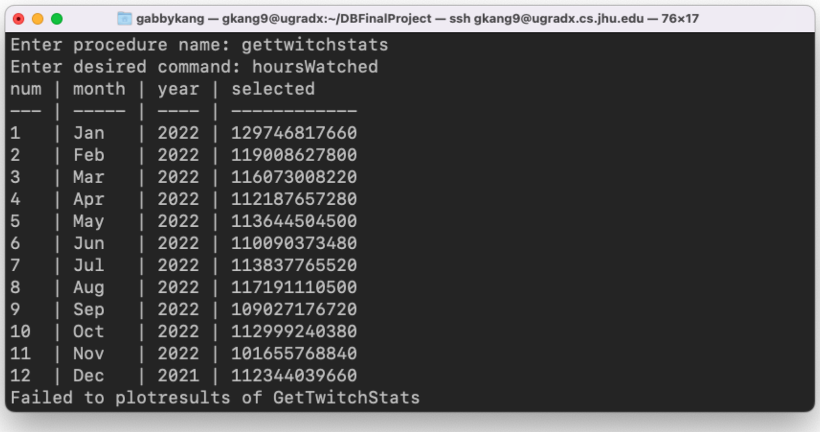
- Sample output for GetCovidPlatforms



gabbykang — gkang9@ugradx:~/DBFinalProject — ssh gkang9@ugradx.cs.jhu.edu — 62×11

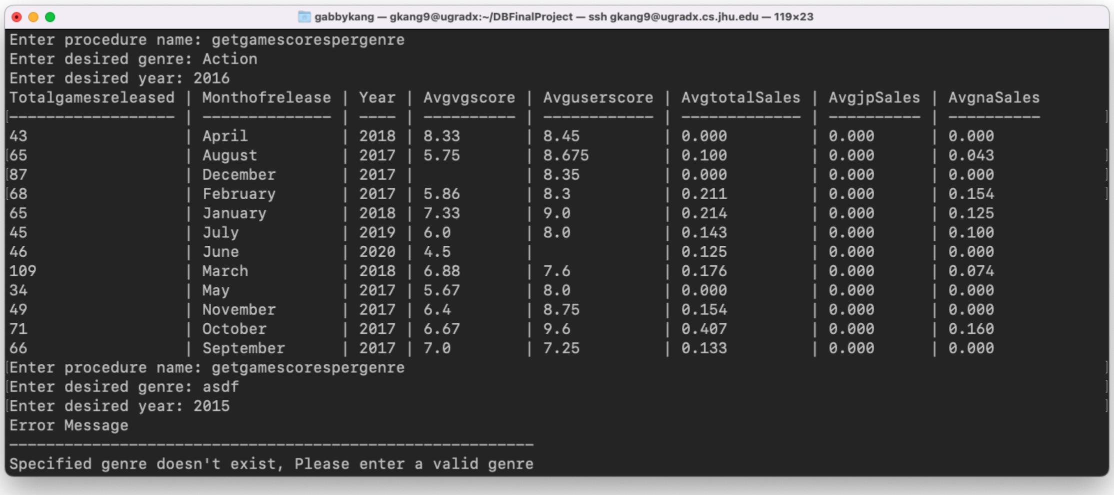
```
Enter procedure name: getcovidplatforms
Enter desired year: 2020
Enter desired month: 3
platform        | numGamesReleased | avgRating
--------------- | ---------------- | -----------------
 DS             | 1                | 54.0
 PC             | 23               | 74.8695652173913
 PlayStation 4  | 23               | 74.82608695652173
 Switch         | 26               | 72.1923076923077
 Xbox 360       | 1                | 48.0
 Xbox One       | 12               | 74.5
```

- Sample output for GetTwitchStats. Includes error message for being unable to plot display of results (since we are using the terminal).


gabbykang — gkang9@ugradx:~/DBFinalProject — ssh gkang9@ugradx.cs.jhu.edu — 76×17

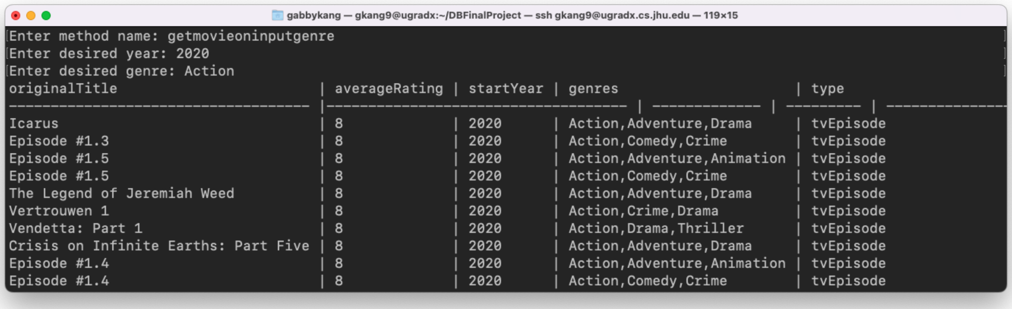
```
Enter procedure name: gettwitchstats
Enter desired command: hoursWatched
num | month | year | selected
--- | ----- | ---- | ------------
1   | Jan   | 2022 | 129746817660
2   | Feb   | 2022 | 119008627800
3   | Mar   | 2022 | 116073008220
4   | Apr   | 2022 | 112187657280
5   | May   | 2022 | 113644504500
6   | Jun   | 2022 | 110090373480
7   | Jul   | 2022 | 113837765520
8   | Aug   | 2022 | 117191110500
9   | Sep   | 2022 | 109027176720
10  | Oct   | 2022 | 112999240380
11  | Nov   | 2022 | 101655768840
12  | Dec   | 2021 | 112344039660
Failed to plotresults of GetTwitchStats
```

- Sample output for GetGameScoresPerGenre where based on the genre of game we display the number of games released and their sales and scores average. In real world this kind of data is very useful to analyze and take decisions


gabbykang — gkang9@ugradx:~/DBFinalProject — ssh gkang9@ugradx.cs.jhu.edu — 119×23

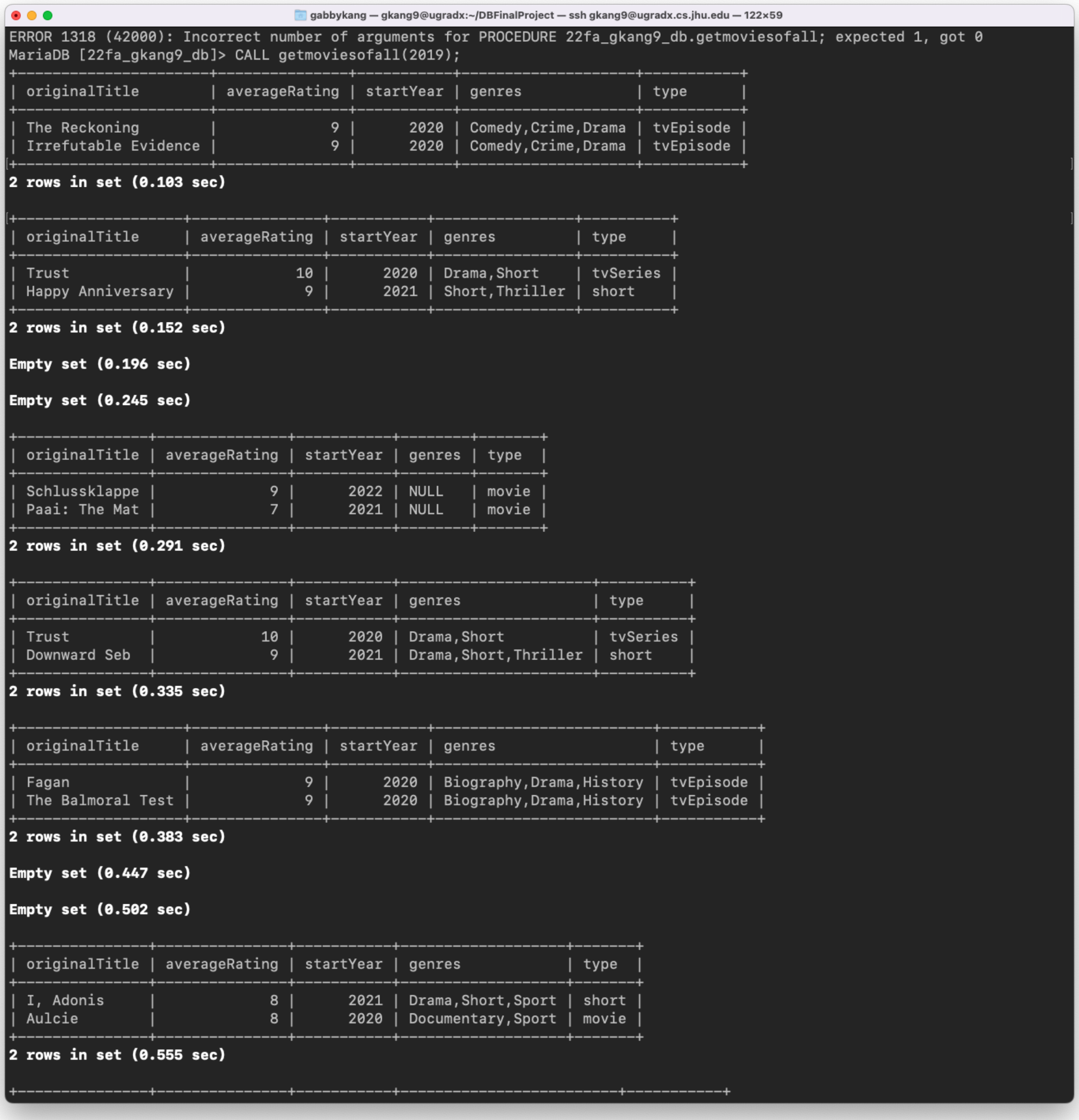
```
Enter procedure name: getgamescorespergenre
Enter desired genre: Action
Enter desired year: 2016
Totalgamesreleased | Monthofrelease | Year | Avgvgscore | Avguserscore | AvgtotalSales | AvgjpSales | AvgnaSales
------------------ | -------------- | ---- | ---------- | ------------ | ------------- | ---------- | ----------
43                 | April          | 2018 | 8.33       | 8.45         | 0.000         | 0.000      | 0.000
65                 | August         | 2017 | 5.75       | 8.675        | 0.100         | 0.000      | 0.043
87                 | December       | 2017 |            | 8.35         | 0.000         | 0.000      | 0.000
68                 | February       | 2017 | 5.86       | 8.3          | 0.211         | 0.000      | 0.154
65                 | January        | 2018 | 7.33       | 9.0          | 0.214         | 0.000      | 0.125
45                 | July           | 2019 | 6.0        | 8.0          | 0.143         | 0.000      | 0.100
46                 | June           | 2020 | 4.5        |              | 0.125         | 0.000      | 0.000
109                | March          | 2018 | 6.88       | 7.6          | 0.176         | 0.000      | 0.074
34                 | May            | 2017 | 5.67       | 8.0          | 0.000         | 0.000      | 0.000
49                 | November       | 2017 | 6.4        | 8.75         | 0.154         | 0.000      | 0.000
71                 | October        | 2017 | 6.67       | 9.6          | 0.407         | 0.000      | 0.160
66                 | September      | 2017 | 7.0        | 7.25         | 0.133         | 0.000      | 0.000
Enter procedure name: getgamescorespergenre
Enter desired genre: asdf
Enter desired year: 2015
Error Message
-----------------------------------------------------------
Specified genre doesn't exist, Please enter a valid genre
```

- Sample output for GetMovieInputGenre, where we get movie recommendations based on the desired genre which are released in the desired year. The genres are constituting of multiple types so we selected the tuple even if it has other genres along with it.


gabbykang — gkang9@ugradx:~/DBFinalProject — ssh gkang9@ugradx.cs.jhu.edu — 119×15

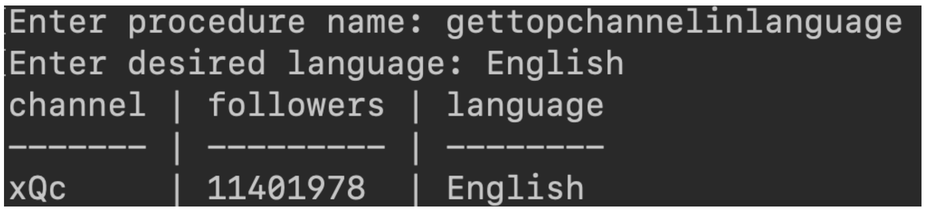
```
Enter method name: getmovieoninputgenre
Enter desired year: 2020
Enter desired genre: Action
originalTitle                      | averageRating | startYear | genres                     | type
---------------------------------- |------------------------------------ | ------------ | --------- | ---------------
Icarus                             | 8             | 2020      | Action,Adventure,Drama     | tvEpisode
Episode #1.3                       | 8             | 2020      | Action,Comedy,Crime        | tvEpisode
Episode #1.5                       | 8             | 2020      | Action,Adventure,Animation | tvEpisode
Episode #1.5                       | 8             | 2020      | Action,Comedy,Crime        | tvEpisode
The Legend of Jeremiah Weed        | 8             | 2020      | Action,Adventure,Drama     | tvEpisode
Vertrouwen 1                       | 8             | 2020      | Action,Crime,Drama         | tvEpisode
Vendetta: Part 1                   | 8             | 2020      | Action,Drama,Thriller      | tvEpisode
Crisis on Infinite Earths: Part Five | 8           | 2020      | Action,Adventure,Drama     | tvEpisode
Episode #1.4                       | 8             | 2020      | Action,Adventure,Animation | tvEpisode
Episode #1.4                       | 8             | 2020      | Action,Comedy,Crime        | tvEpisode
```

- Sample output for GetMoviesOfAll. It produces the top 2 movies according to the rating in each category of genre present in the database. We used Advanced sql topic cursor in order to achieve this task.



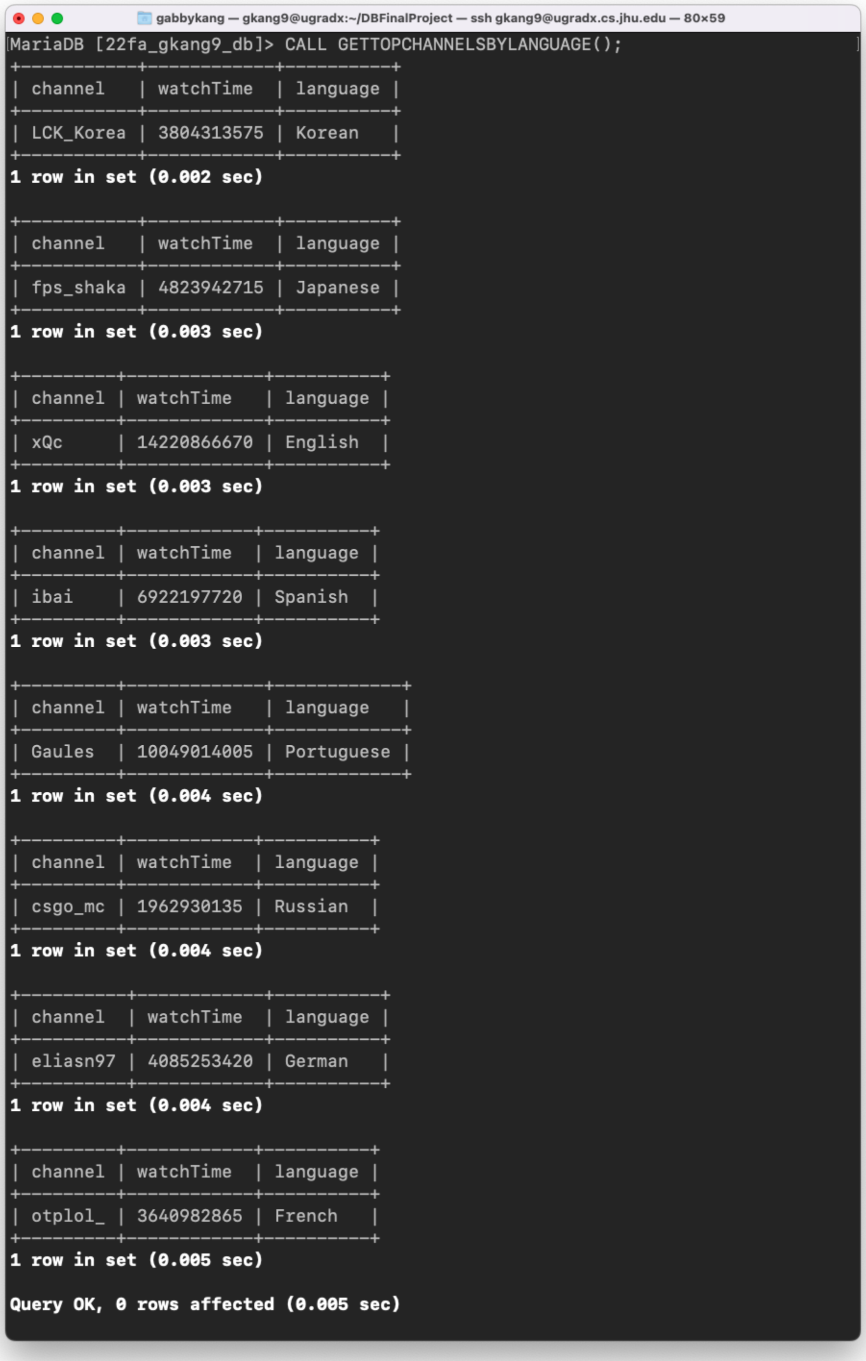
```
ERROR 1318 (42000): Incorrect number of arguments for PROCEDURE 22fa_gkang9_db.getmoviesofall; expected 1, got 0
MariaDB [22fa_gkang9_db]> CALL getmoviesofall(2019);
+---------------------+---------------+-----------+--------------------+-----------+
| originalTitle       | averageRating | startYear | genres             | type      |
+---------------------+---------------+-----------+--------------------+-----------+
| The Reckoning       |             9 |      2020 | Comedy,Crime,Drama | tvEpisode |
| Irrefutable Evidence |            9 |      2020 | Comedy,Crime,Drama | tvEpisode |
+---------------------+---------------+-----------+--------------------+-----------+
2 rows in set (0.103 sec)

+-------------------+---------------+-----------+----------------+----------+
| originalTitle     | averageRating | startYear | genres         | type     |
+-------------------+---------------+-----------+----------------+----------+
| Trust             |            10 |      2020 | Drama,Short    | tvSeries |
| Happy Anniversary |             9 |      2021 | Short,Thriller | short    |
+-------------------+---------------+-----------+----------------+----------+
2 rows in set (0.152 sec)

Empty set (0.196 sec)

Empty set (0.245 sec)

+---------------+---------------+-----------+--------+-------+
| originalTitle | averageRating | startYear | genres | type  |
+---------------+---------------+-----------+--------+-------+
| Schlussklappe |             9 |      2022 | NULL   | movie |
| Paai: The Mat |             7 |      2021 | NULL   | movie |
+---------------+---------------+-----------+--------+-------+
2 rows in set (0.291 sec)

+---------------+---------------+-----------+----------------------+----------+
| originalTitle | averageRating | startYear | genres               | type     |
+---------------+---------------+-----------+----------------------+----------+
| Trust         |            10 |      2020 | Drama,Short          | tvSeries |
| Downward Seb  |             9 |      2021 | Drama,Short,Thriller | short    |
+---------------+---------------+-----------+----------------------+----------+
2 rows in set (0.335 sec)

+-------------------+---------------+-----------+--------------------------+-----------+
| originalTitle     | averageRating | startYear | genres                   | type      |
+-------------------+---------------+-----------+--------------------------+-----------+
| Fagan             |             9 |      2020 | Biography,Drama,History  | tvEpisode |
| The Balmoral Test |             9 |      2020 | Biography,Drama,History  | tvEpisode |
+-------------------+---------------+-----------+--------------------------+-----------+
2 rows in set (0.383 sec)

Empty set (0.447 sec)

Empty set (0.502 sec)

+---------------+---------------+-----------+-------------------+-------+
| originalTitle | averageRating | startYear | genres            | type  |
+---------------+---------------+-----------+-------------------+-------+
| I, Adonis     |             8 |      2021 | Drama,Short,Sport | short |
| Aulcie        |             8 |      2020 | Documentary,Sport | movie |
+---------------+---------------+-----------+-------------------+-------+
2 rows in set (0.555 sec)

+---------------+---------------+-----------+-------------------------+-----------+
```

- Sample output for GetTopChannelsInLanguage



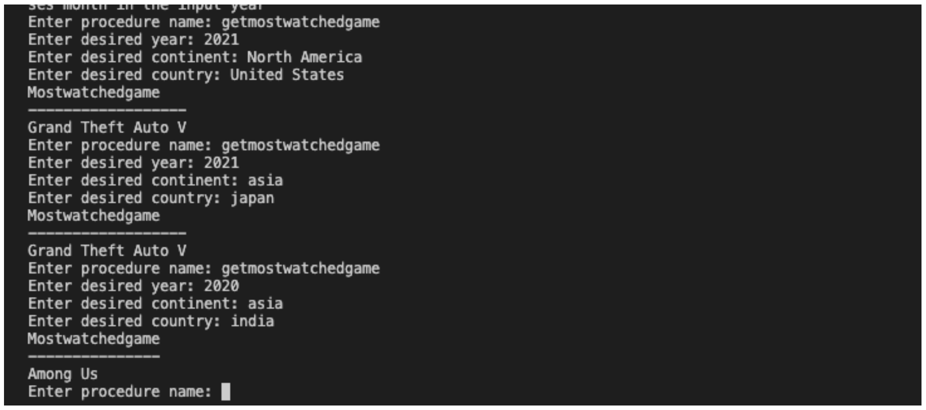
```
Enter procedure name: gettopchannelinlanguage
Enter desired language: English
channel | followers | language
------- | --------- | --------
xQc     | 11401978  | English
```

- Sample output for GetTopChannelsByLanguage gets all the top channels in various languages based on the watch time, we used Cursors to achieve this task.


gabbykang — gkang9@ugradx:~/DBFinalProject — ssh gkang9@ugradx.cs.jhu.edu — 80×59

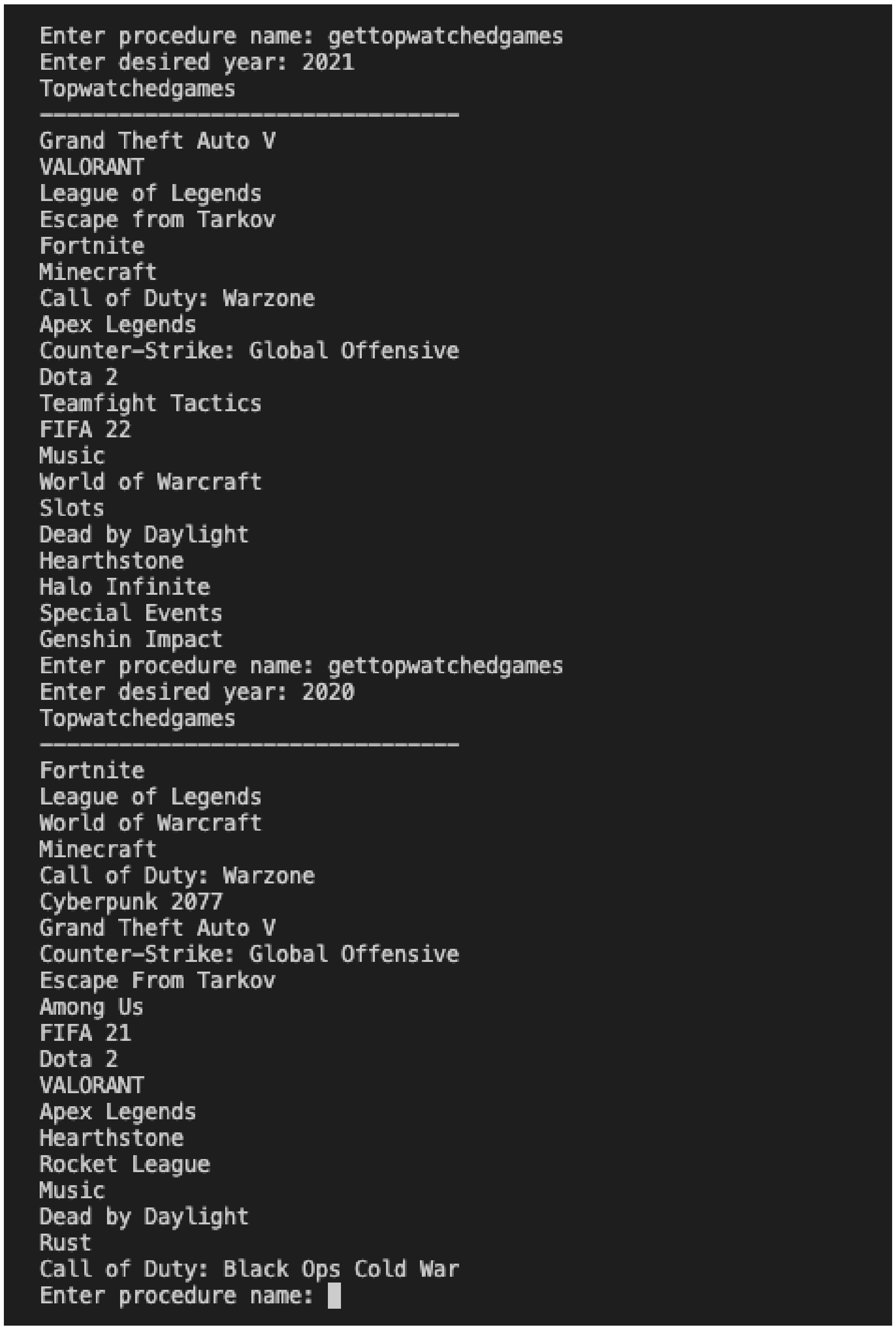
```
MariaDB [22fa_gkang9_db]> CALL GETTOPCHANNELSBYLANGUAGE();
+-----------+------------+----------+
| channel   | watchTime  | language |
+-----------+------------+----------+
| LCK_Korea | 3804313575 | Korean   |
+-----------+------------+----------+
1 row in set (0.002 sec)

+-----------+------------+----------+
| channel   | watchTime  | language |
+-----------+------------+----------+
| fps_shaka | 4823942715 | Japanese |
+-----------+------------+----------+
1 row in set (0.003 sec)

+---------+-------------+----------+
| channel | watchTime   | language |
+---------+-------------+----------+
| xQc     | 14220866670 | English  |
+---------+-------------+----------+
1 row in set (0.003 sec)

+---------+------------+----------+
| channel | watchTime  | language |
+---------+------------+----------+
| ibai    | 6922197720 | Spanish  |
+---------+------------+----------+
1 row in set (0.003 sec)

+---------+-------------+------------+
| channel | watchTime   | language   |
+---------+-------------+------------+
| Gaules  | 10049014005 | Portuguese |
+---------+-------------+------------+
1 row in set (0.004 sec)

+---------+------------+----------+
| channel | watchTime  | language |
+---------+------------+----------+
| csgo_mc | 1962930135 | Russian  |
+---------+------------+----------+
1 row in set (0.004 sec)

+----------+------------+----------+
| channel  | watchTime  | language |
+----------+------------+----------+
| eliasn97 | 4085253420 | German   |
+----------+------------+----------+
1 row in set (0.004 sec)

+---------+------------+----------+
| channel | watchTime  | language |
+---------+------------+----------+
| otplol_ | 3640982865 | French   |
+---------+------------+----------+
1 row in set (0.005 sec)

Query OK, 0 rows affected (0.005 sec)
```

- Sample output for GetMostWatchedGame, this is a good insight of getting details about various games and their stats during the time of pandemic. This was a bit tricky to get the month with highest new cases registered as the covid data is very huge and has a lot of countries and its covid data per day.



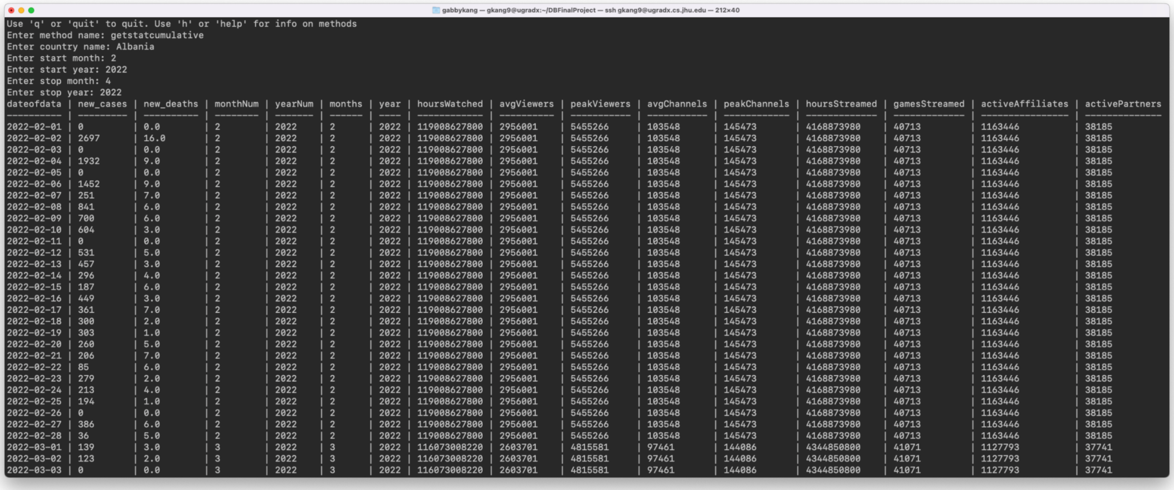
```
ses month in the input year
Enter procedure name: getmostwatchedgame
Enter desired year: 2021
Enter desired continent: North America
Enter desired country: United States
Mostwatchedgame
------------------
Grand Theft Auto V
Enter procedure name: getmostwatchedgame
Enter desired year: 2021
Enter desired continent: asia
Enter desired country: japan
Mostwatchedgame
------------------
Grand Theft Auto V
Enter procedure name: getmostwatchedgame
Enter desired year: 2020
Enter desired continent: asia
Enter desired country: india
Mostwatchedgame
---------------
Among Us
Enter procedure name: 
```

- Sample output for gettopwatchedgames, which returns the list of top 20 viewed games during the month of peak covid cases in the given year.



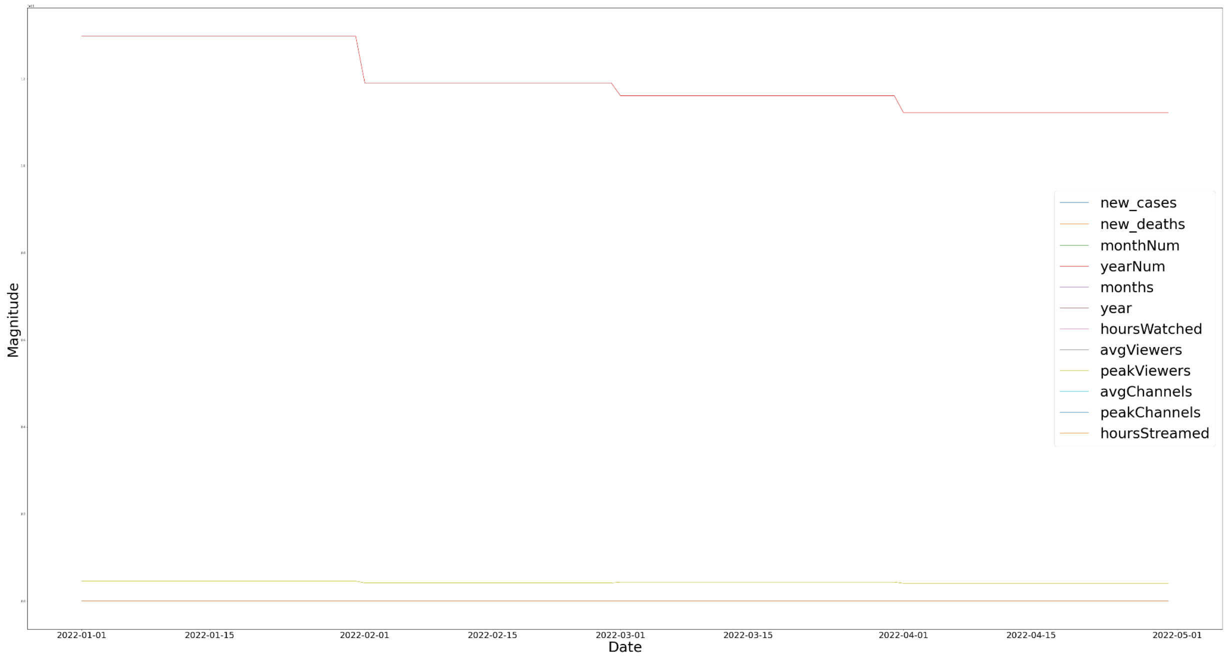
```
Enter procedure name: gettopwatchedgames
Enter desired year: 2021
Topwatchedgames
--------------------------------
Grand Theft Auto V
VALORANT
League of Legends
Escape from Tarkov
Fortnite
Minecraft
Call of Duty: Warzone
Apex Legends
Counter-Strike: Global Offensive
Dota 2
Teamfight Tactics
FIFA 22
Music
World of Warcraft
Slots
Dead by Daylight
Hearthstone
Halo Infinite
Special Events
Genshin Impact
Enter procedure name: gettopwatchedgames
Enter desired year: 2020
Topwatchedgames
--------------------------------
Fortnite
League of Legends
World of Warcraft
Minecraft
Call of Duty: Warzone
Cyberpunk 2077
Grand Theft Auto V
Counter-Strike: Global Offensive
Escape From Tarkov
Among Us
FIFA 21
Dota 2
VALORANT
Apex Legends
Hearthstone
Rocket League
Music
Dead by Daylight
Rust
Call of Duty: Black Ops Cold War
Enter procedure name: 
```

- Sample output of GetStatCumulative (truncated view)

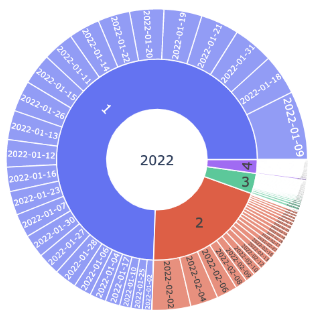


```
Use 'q' or 'quit' to quit. Use 'h' or 'help' for info on methods
Enter method name: getstatcumulative
Enter country name: Albania
Enter start month: 2
Enter start year: 2022
Enter stop month: 4
Enter stop year: 2022
dateofdata | new_cases | new_deaths | monthNum | yearNum | months | year | hoursWatched | avgViewers | peakViewers | avgChannels | peakChannels | hoursStreamed | gamesStreamed | activeAffiliates | activePartners
---------- | --------- | ---------- | -------- | ------- | ------ | ---- | ------------ | ---------- | ----------- | ----------- | ------------ | ------------- | ------------- | ---------------- | --------------
2022-02-01 | 0         | 0.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-02 | 2697      | 16.0       | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-03 | 0         | 0.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-04 | 1932      | 9.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-05 | 0         | 0.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-06 | 1452      | 9.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-07 | 251       | 7.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-08 | 841       | 6.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-09 | 700       | 6.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-10 | 604       | 3.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-11 | 0         | 0.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-12 | 531       | 5.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-13 | 457       | 3.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-14 | 296       | 4.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-15 | 187       | 6.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-16 | 449       | 3.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-17 | 361       | 7.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-18 | 300       | 2.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-19 | 303       | 1.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-20 | 260       | 5.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-21 | 206       | 7.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-22 | 85        | 6.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-23 | 279       | 2.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-24 | 213       | 4.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-25 | 194       | 1.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-26 | 0         | 0.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-27 | 386       | 6.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-02-28 | 36        | 5.0        | 2        | 2022    | 2      | 2022 | 119008627800 | 2956001    | 5455266     | 103548      | 145473       | 4168873980    | 40713         | 1163446          | 38185
2022-03-01 | 139       | 3.0        | 3        | 2022    | 3      | 2022 | 116073008220 | 2603701    | 4815581     | 97461       | 144086       | 4344850800    | 41071         | 1127793          | 37741
2022-03-02 | 123       | 2.0        | 3        | 2022    | 3      | 2022 | 116073008220 | 2603701    | 4815581     | 97461       | 144086       | 4344850800    | 41071         | 1127793          | 37741
2022-03-03 | 0         | 0.0        | 3        | 2022    | 3      | 2022 | 116073008220 | 2603701    | 4815581     | 97461       | 144086       | 4344850800    | 41071         | 1127793          | 37741
```

- Sample graphs for GetStatCumulative (same parameters as above)

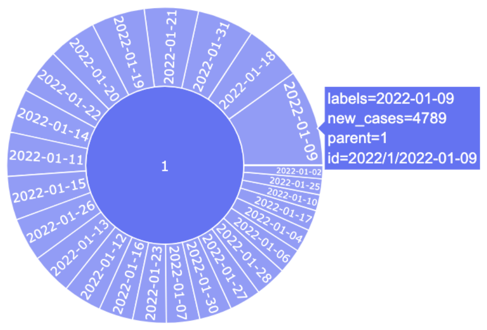


new_cases



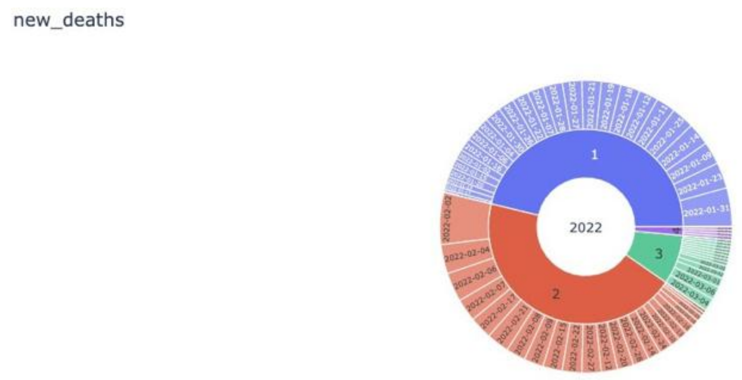
- By clicking on the portion labeled 1 (for January), we are able to "zoom in" on that portion of the graph and get a better distribution of new cases

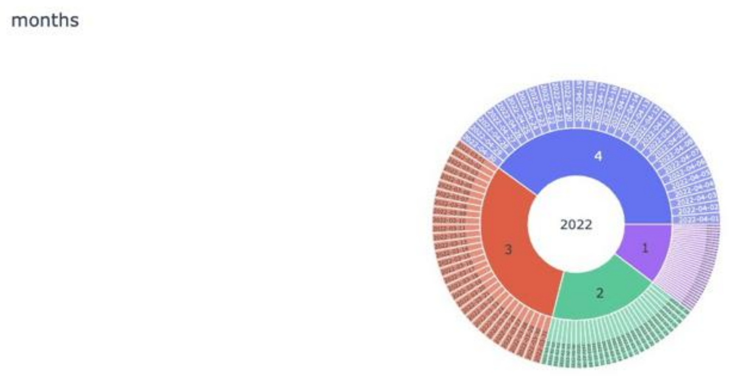
new_cases



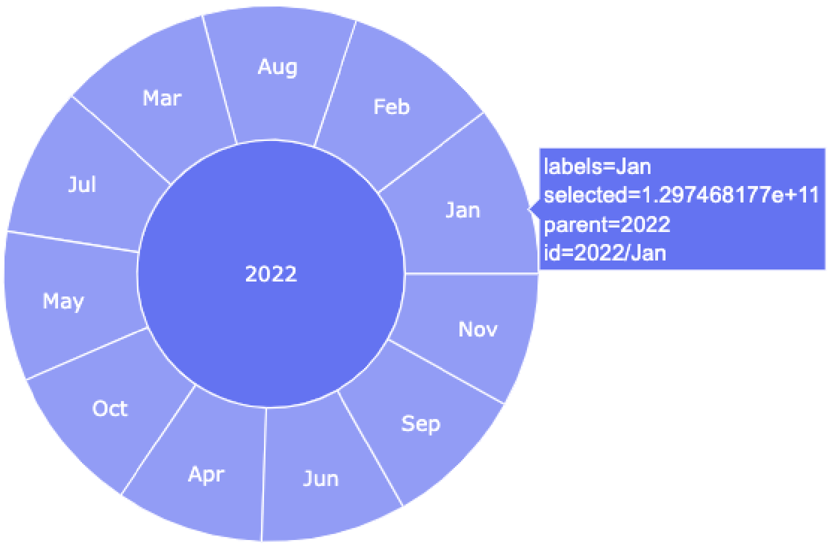
labels=2022-01-09
new_cases=4789
parent=1
id=2022/1/2022-01-09

- Additional graphs (Note: GetStatCumulative with display graphs for 12 different variables):





- Sample graph output for GetTwitchStats



12. Relational Database Structure

Please see DBFinalProject/insert_files/table_creation.sql for the full relational database structure.