Assignment 1

PSTAT 135/235

Name: Selin Karabulut

Perm Number: 6075253

MovieLens Dataset

In this assignment, we will be working on a new dataset. To download it paste the following URL into your laptop's browser: `http://files.grouplens.org/datasets/movielens/ml-latest.zip`. Alternatively, you can also go to `https://grouplens.org/datasets/movielens/` and download `ml-latest.zip`.

This dataset has around 27 million ratings on about 58,000 movies done by over 280,000 users and last updated on 9/2018. Unzip this 288 MB file. For the purpose of this assignment we will be using only two of the files that are included:

1. `movies.csv` (2.9 MB)
2. `ratings.csv` (760 MB).

Question 1: Uploading Data to BigQuery

Upload these two files into a dataset in BigQuery and call it `movie_ratings`.

Create a new dataset and call it `movie_ratings`. We will load these two files into the newly created dataset two ways: using the web interface and agian using cloud shell.

Question 1a: `movies` table

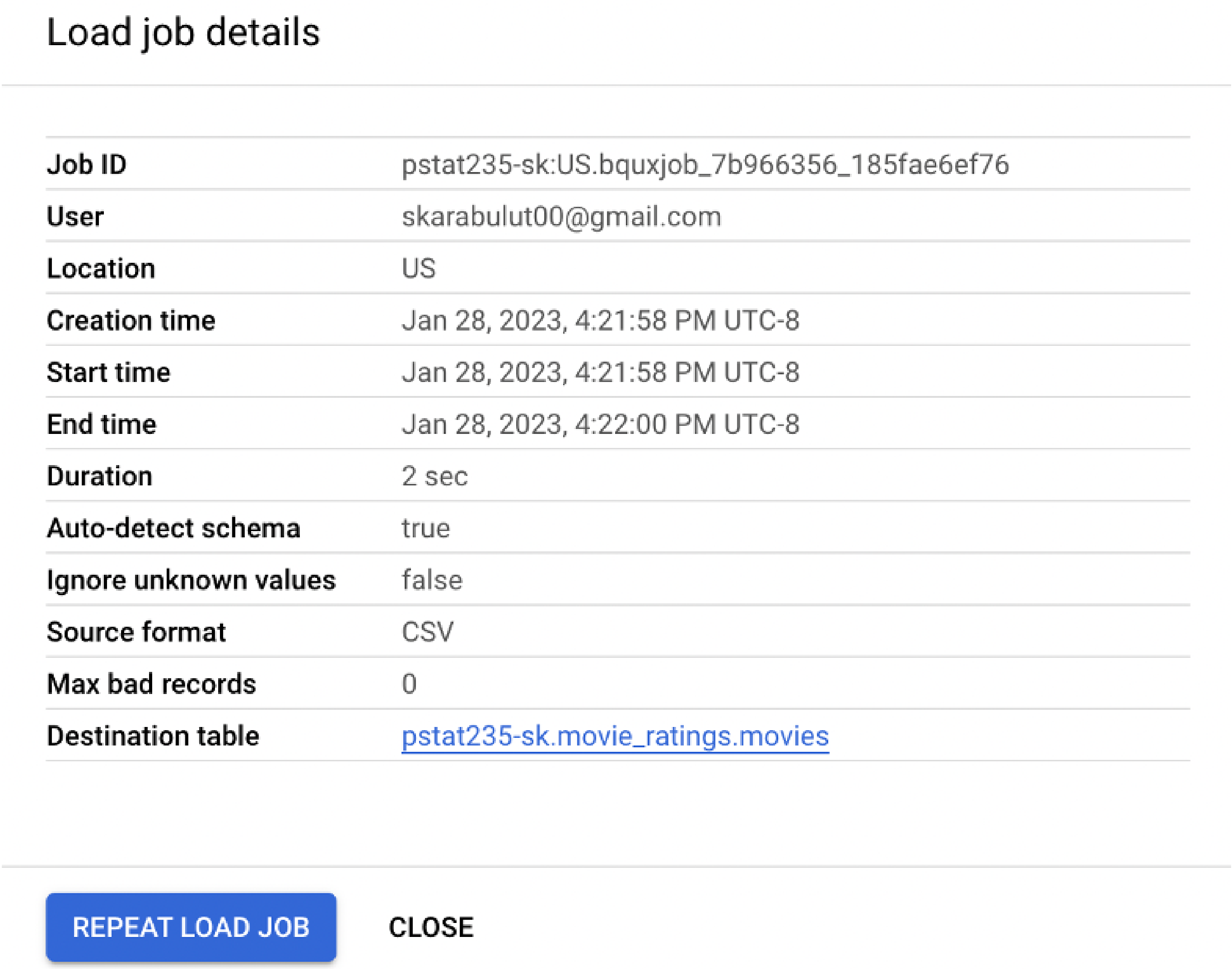
To create `movies` table from `movies.csv` file,

1. Download the zipped file
2. Unzip the archive
3. In your BigQuery interface, select in the resources list `<YOUR-PROJECT-ID>` > `movie_ratings` > click **"CREATE TABLE"** button
4. `Create table from`: `Upload`
   `Select file`: BROWSE and find `movies.csv` from your computer
   `Table`: `movies`
   Schema `Auto detect`: check

Find your LOAD job information from `PROJECT HISTORY` (next to `PERSONAL HISTORY`) at the bottom. Mine looks like @fig-job-info

Post screenshot of your LOAD job information here:

Answer



Load job details

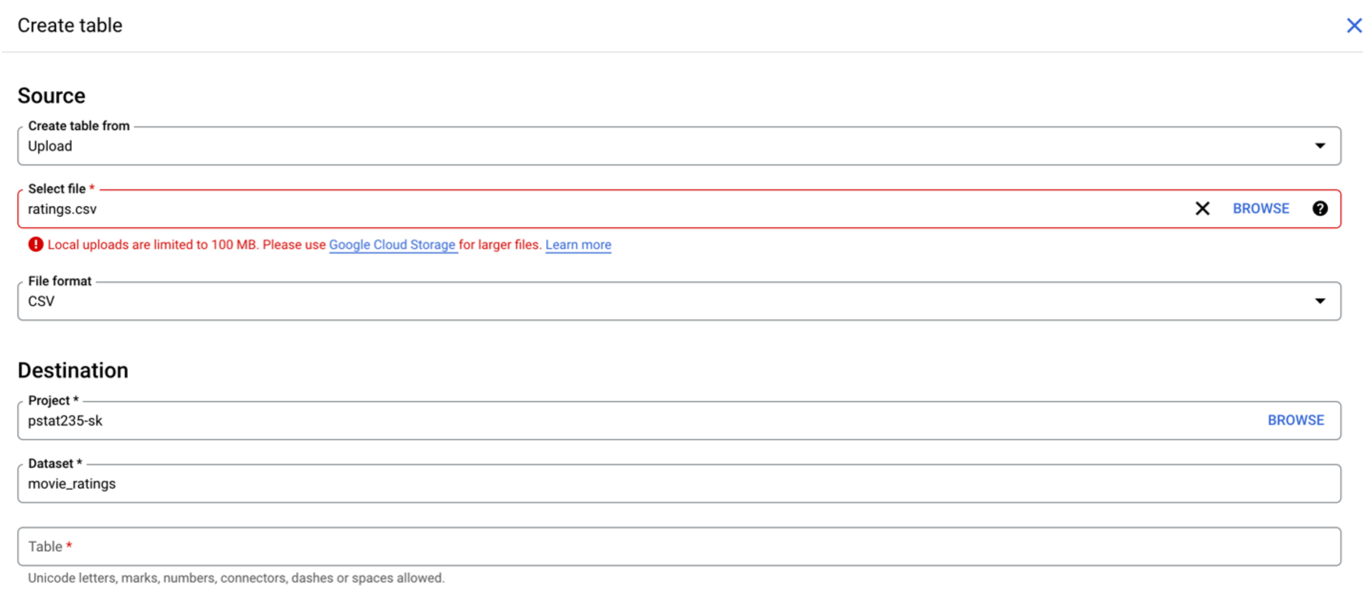
Job ID	pstat235-sk:US.bquxjob_7b966356_185fae6ef76
User	skarabulut00@gmail.com
Location	US
Creation time	Jan 28, 2023, 4:21:58 PM UTC-8
Start time	Jan 28, 2023, 4:21:58 PM UTC-8
End time	Jan 28, 2023, 4:22:00 PM UTC-8
Duration	2 sec
Auto-detect schema	true
Ignore unknown values	false
Source format	CSV
Max bad records	0
Destination table	pstat235-sk.movie_ratings.movies

Question 1b: `ratings` table

Follow the same procedure as Question 1a to crate `ratings` table from `ratings.csv`. What happens?

Answer

It didn't let us to upload `ratings.csv` (below is the screenshot of the warning generated by the system) because local uploads are limited to 100 MB and this file (760 MB) is larger than that.



Create table

Source

Create table from: Upload

Select file *: ratings.csv

Local uploads are limited to 100 MB. Please use Google Cloud Storage for larger files. Learn more

File format: CSV

Destination

Project *: pstat235-sk

Dataset *: movie_ratings

Table *

Unicode letters, marks, numbers, connectors, dashes or spaces allowed.

PSTAT 135 Students: Upload `ratings.csv` file to Cloud Storage and create `ratings` table from it using the web interface. Then, post the screenshot of your LOAD job information here:

Replace this text with your screenshot image

PSTAT 235 Students: Upload `ratings.csv` file to Cloud Storage and create `ratings` table using the commmand line tools: `bq` and `gsutil`.

1. Verify the location of `ratings.csv` file using Cloud Storage command:

   ```
   gsutil ls gs://<YOUR-BUCKET-NAME>
   ```

   Note your the path to your `ratings.csv` file (referred to as `<RATINGS-FILE-LOCATION>` below).

2. Create an empty table with `bq`. Read the documentation, `bq mk --help` to fill-in the blanks in the code below:

   ```
   bq mk _______
   ```

3. Using `bq` command to load `movie_ratings.ratings` table with contents from `<RATINGS-FILE-LOCATION>`. Read the documentation, `bq load --help` to fill-in the blanks in the code below:

   ```
   bq load --autodetect _______ _______
   ```

Replace the section below with your own commands:

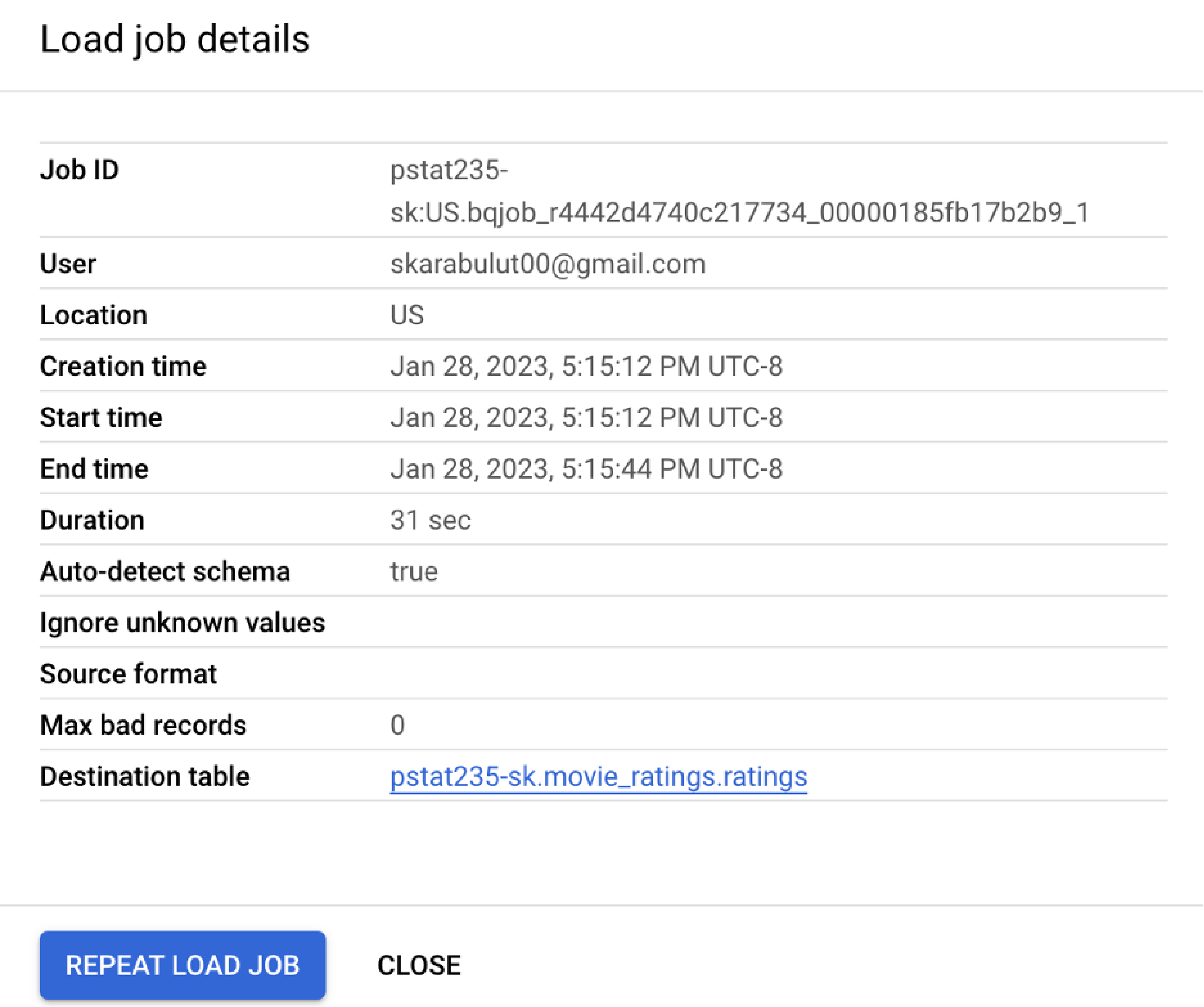
```
gsutil ls gs://<YOUR-BUCKET-NAME>
bq mk _______
bq load --autodetect _______ _______
```

Answer

```
gsutil ls gs://pstat235-sk
bq mk --table movie_ratings.ratings
bq load --autodetect movie_ratings.ratings  gs://pstat235-sk/ratings.csv
```

Also, post screenshot of your LOAD job information here:

Answer



Load job details

Job ID	pstat235-sk:US.bqjob_r4442d4740c217734_00000185fb17b2b9_1
User	skarabulut00@gmail.com
Location	US
Creation time	Jan 28, 2023, 5:15:12 PM UTC-8
Start time	Jan 28, 2023, 5:15:12 PM UTC-8
End time	Jan 28, 2023, 5:15:44 PM UTC-8
Duration	31 sec
Auto-detect schema	true
Ignore unknown values	
Source format	
Max bad records	0
Destination table	pstat235-sk.movie_ratings.ratings

Question 2: `ratings` table number of rows

How many rows are there in `ratings` table?

A. 27753445
B. 27000001
C. 27753444
D. 27000000

Answer

C.27753444

- SQL CODE

```
SELECT COUNT(*)
FROM `pstat235-sk.movie_ratings.ratings`;
```

Question 3: `movies` table number of rows

How many rows are there in the `movies` table?

A. 57999
B. 58000
C. 58097
D. 58098

Answer

D.58098

- SQL CODE

```
SELECT COUNT(*)
FROM `pstat235-sk.movie_ratings.movies`;
```

Question 4: number of unique movies

How many unique `movieId`'s are in `ratings` table?

A. 52019
B. Around 27 million
C. 53889
D. 58097

Answer

C. 53889

What is your SQL code to obtain the info?

```
SELECT COUNT(DISTINCT movieId) AS count_unique_movieId
FROM `pstat235-sk.movie_ratings.ratings`;
```

Question 5: highly rated movies

Which one of these movies are among top 10 highly rated movies, with at least 10,000 reviews? (select all that apply)

A. Star Wars: Episode IV - A New Hope (1977)
B. Chinatown (1974)
C. Godfather
D. Casablanca (1942)

Answer

C. Godfather

What is your SQL code to obtain the info?

```
SELECT m.movieId, m.title, temp.avg_rating
FROM `pstat235-sk.movie_ratings.movies` AS m
INNER JOIN (SELECT AVG(rating) as avg_rating, movieId
FROM `pstat235-sk.movie_ratings.ratings`
GROUP BY movieId
HAVING COUNT(rating)>=10000
ORDER BY AVG(rating) DESC
LIMIT 10) temp
ON m.movieId=temp.movieId;
```

Question 6: most watched movies

Which movie is the most watched? Make an assumption that number of ratings is strongly correlated with number of people watching it.

A. Shawshank Redemption
B. Forrest Gump (1994)
C. Matrix
D. Toy Story (1995)

Answer

A. Shawshank Redemption

What is your SQL code to obtain the info?

```
SELECT m.title, COUNT(r.rating)
FROM `pstat235-sk.movie_ratings.movies` AS m
JOIN `pstat235-sk.movie_ratings.ratings` AS r
ON m.movieId = r.movieId
GROUP BY m.title
ORDER BY COUNT(r.rating) DESC
LIMIT 5;
```