Homework 2 – SQL

CSCI 585 Fall 2018

Due Date: Thursday, October 4th, 2018 at 11:59 PM

In this assignment, we will use Google Cloud SQL to work with SQL queries. This will help us learn how to use cloud services as well as run code on SQL. The document is divided into several parts. Parts 1 through 3 go through the basics of setting up the platform. Part 4 is the assignment.

Good luck.

Contents

Part I: Setting up Google Cloud Platform

Google Cloud Platform helps you to run your work on Google Compute Engine (GCE) and to use its core infrastructure, data analytics and machine learning.

To set up GCP, follow the steps below to retrieve.

1. Go to the following link to request a Google Cloud Platform coupon.
   https://google.secure.force.com/GCPEDU/?cid=ZPjvOXdVQMeHCoEW6HWyQIukGevuuiwL6RtfA6YPW%2BC2g3pSGa2%2FeXF5%2Bz7iPF35/
   You will be asked to provide your school email address and name. (Figure 1.1).
   An email will be sent to you to confirm these details before a coupon is sent to you. (You can only request ONE code per unique email address).

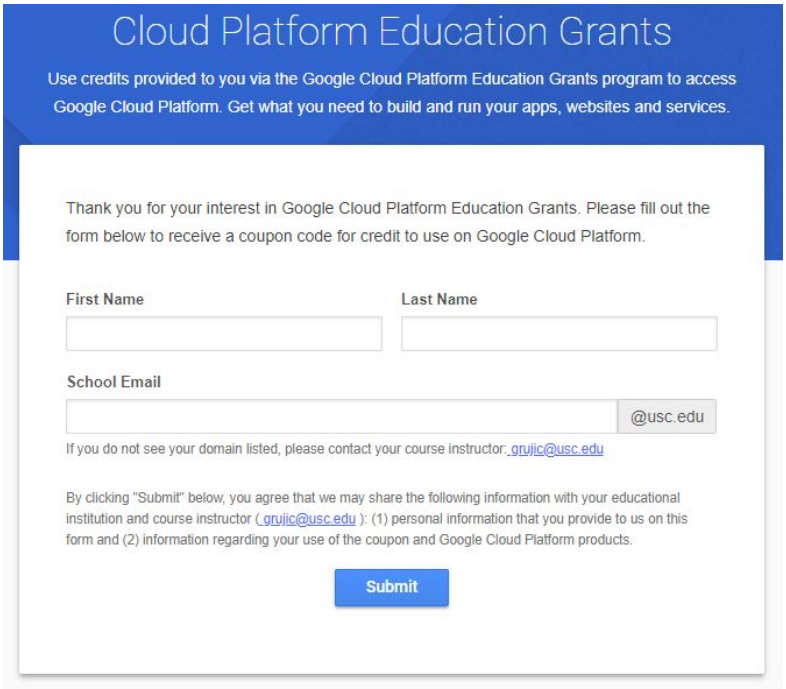


Figure 1.1 Request Form

You will receive an email to verify your USC email through a link (Figure 1.2)

Figure 1.2 Verification email

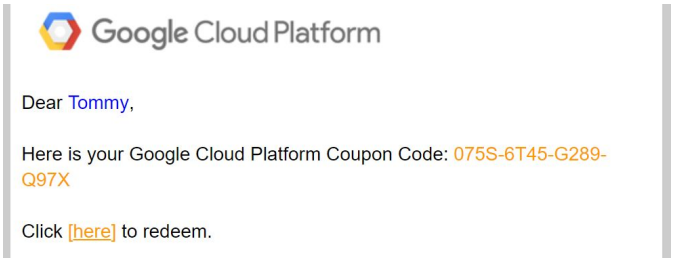


Figure 1.3 Coupon email

2. After verifying your USC email address, a code will be sent to you in a second email containing a unique promo code XXXX-XXXX-XXXX-XXXX. You can either (1) enter the promo code at this link: https://console.cloud.google.com/education, or (2) Click on the link in the second email (Figure 1.3) to redeem.
3. **IMPORTANT:** You have to use and be logged into your personal Gmail account to redeem this code. This is because USC accounts have GCP disabled. Please make sure which account your browser is currently logged into, since the promo code cannot be moved once applied.

   Please be aware if you end up messing up on this part, you would have to use the regular $300 free trial as normal.
4. If you chose to manually enter the promo code, you will enter it in the textbox shown in Figure 1.4. If you clicked the link, your code will be automatically filled out for you.

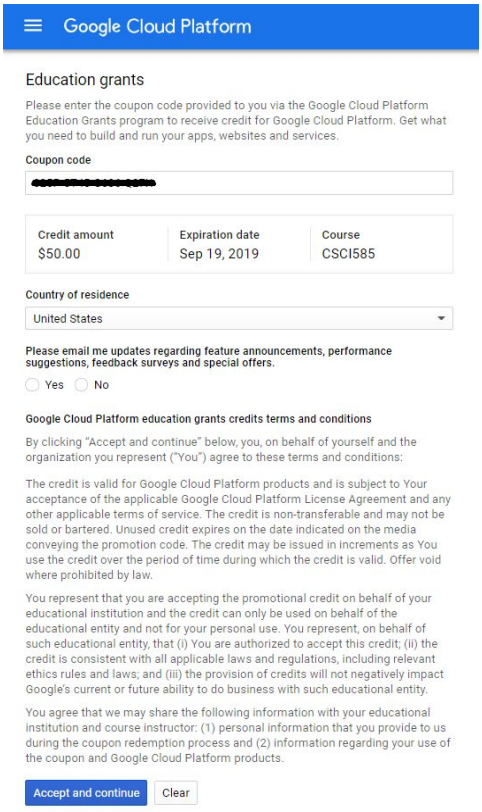


Figure 1.4 GCP Terms and conditions

5. "Accept and continue" the terms and conditions.

6. Now you should have a billing account called “CSCI585” listed with $50 credit inside.

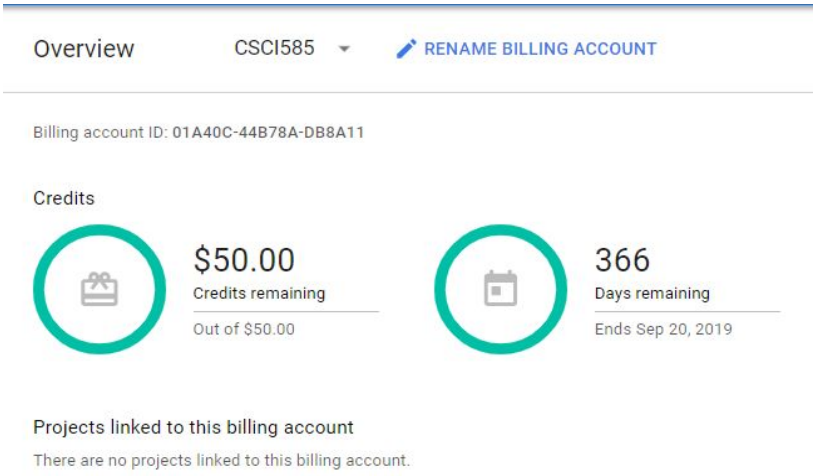


Figure 1.5: CSCI585 billing account

Congratulations! You just finished the first part of the assignment.

** If you happened to mess up, please enable your GCP free trial here

https://console.cloud.google.com/freetrial and accept the terms and conditions **

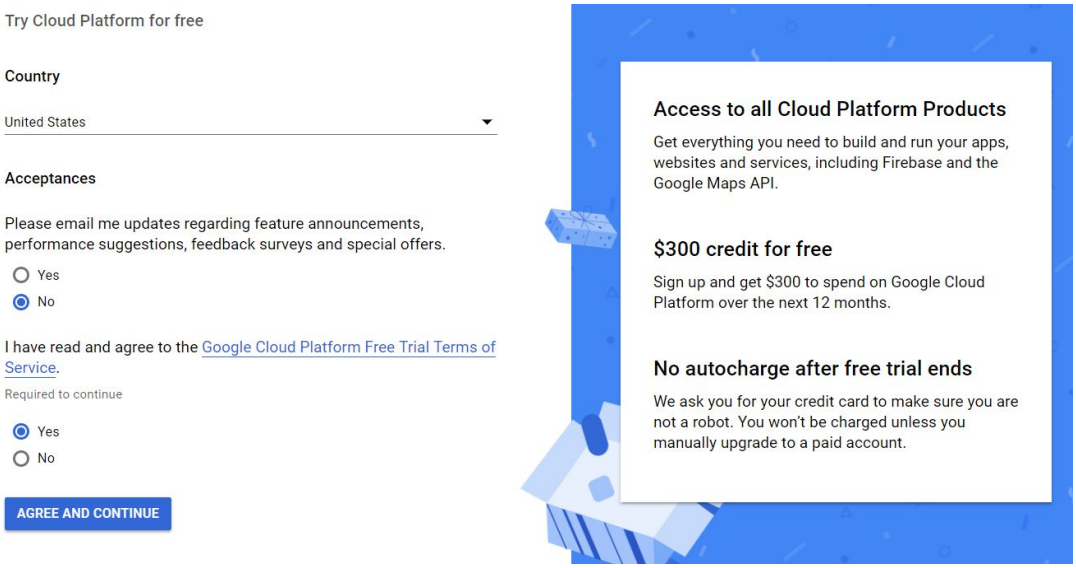


Figure 1.6 GCP Free Trial Terms and Conditions

This will create a “billing account” (just as what the promo code would do, but billing account name will not be CSCI585) for you to setup the SQL Cloud Service.

Part 2: Setting up Cloud SQL

Cloud SQL is a part of the GCE to run PostgreSQL and MySQL scripts.

Go to https://cloud.google.com/sql/.

If you prefer to use MySQL for this assignment, you can find the Quick Start guide at: https://cloud.google.com/sql/docs/mysql/quickstart.

If you prefer to use PostgreSQL instead, visit https://cloud.google.com/sql/docs/postgres/quickstart.

The pages are self-explanatory, and in case you do not face a problem, setting It up, feel free to skip the rest of Part 2. Below are detailed steps from the same page.

Before you begin

1. Select or create a Cloud platform project.

Go to: https://console.cloud.google.com/start. At the top, click on 'Select a project', and click on the NEW PROJECT sign.

2. In the next screen, (as in Figure 2.1), enter a project name.
   If it prompt you to select a billing account, you should be able to select the "CSCI585" billing account or the free trial billing account it created for you.
3. Click Create. Enable the Cloud SQL Administration API. Wait for API to be enabled and then click 'Continue'. You will be redirected to the dashboard
4. Enable the appropriate Cloud Service APIs (CloudSQL, AppEngine, etc) as normal.

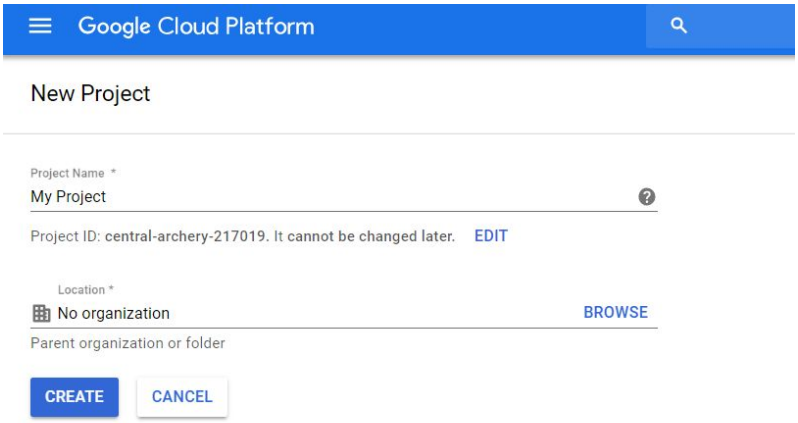


Figure 2.1: New Project Screen

Create a Cloud SQL Instance

1. Go to https://console.cloud.google.com/projectselector/sql/. You will get a screen like Fig. 2.3 (a). Click on 'Select', select the project and then click 'Open'. (Fig. 2.3 (b)).
2. Click on 'Create Instance' in the cloud Instances page. (Fig. 2.4).
3. Select one of MySQL or PostgreSQL and click 'Next'. *Note that PostgreSQL is in beta and might undergo changes which will not be backward compatible.*

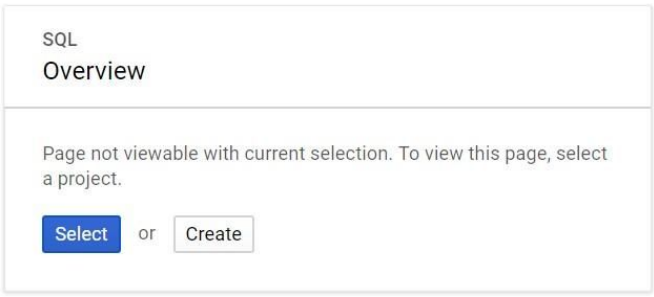


Figure 2.3 (a): SQL Overview Page.

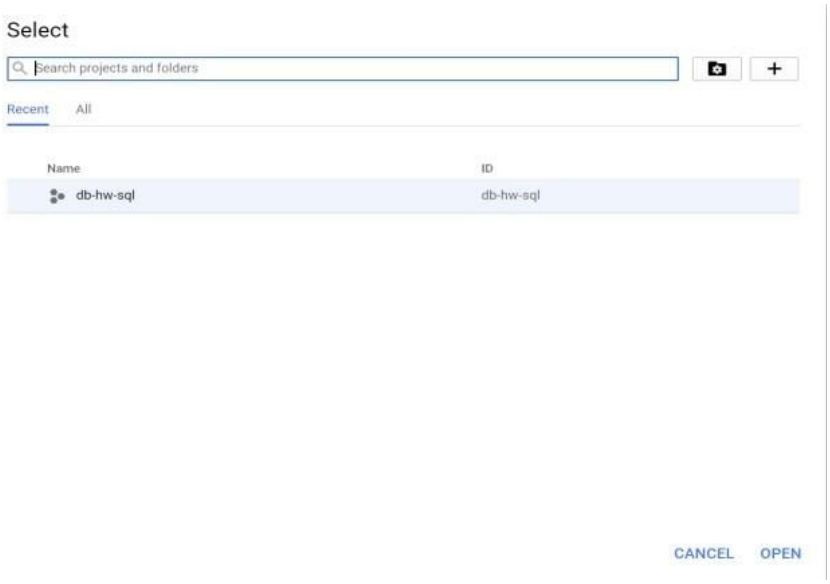


Figure 2.3 (b): Select the project

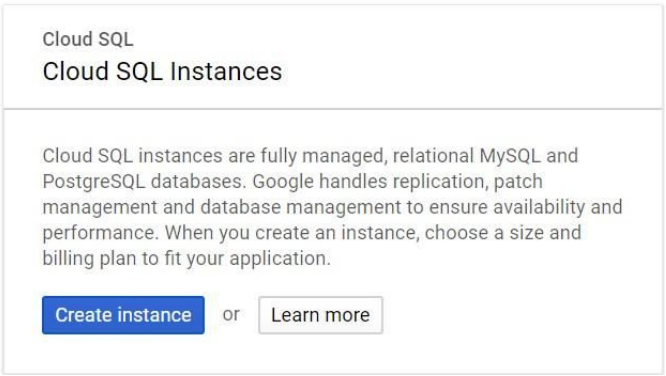


Fig 2.4: Create Instance

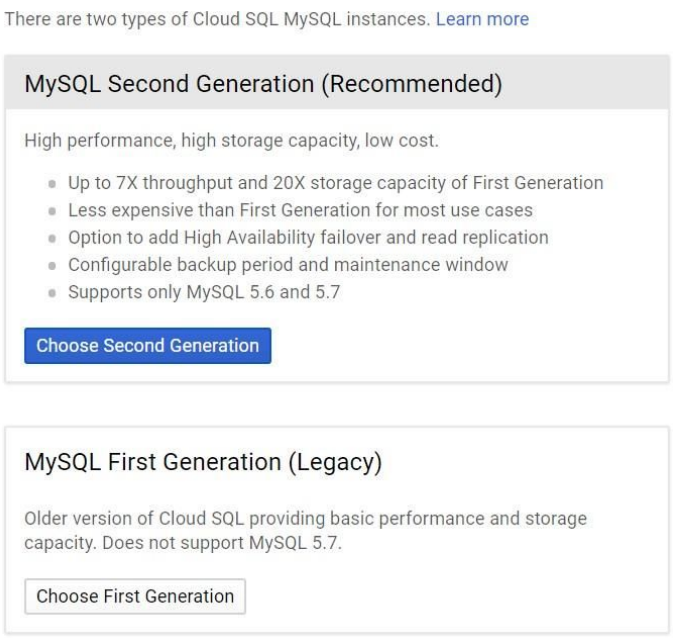
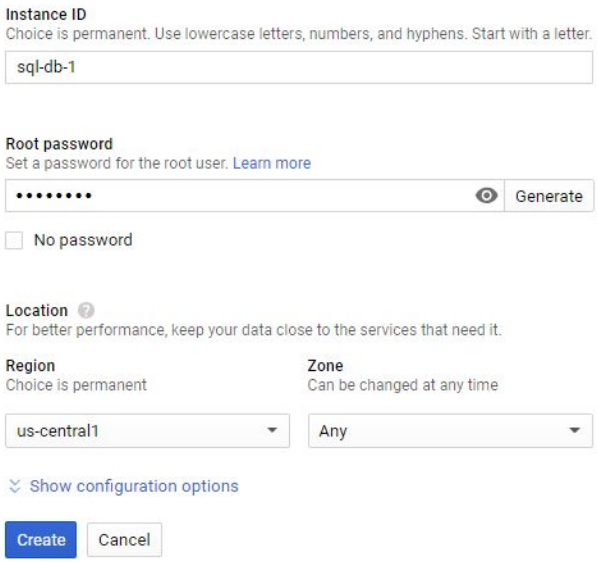


Figure 2.5: MySQL Second Generation page

The next steps are explained with MySQL.

4. Click on 'Choose second generation' in case you get the next screen as Figure 2.5.
5. In the Instance details page, provide an Instance ID name and a root password. Leave the rest as they are.

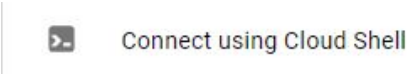


Click on 'Create'. You will see 'Instance is being created'. Wait until the left most wheel turns into a green tick.

Instance ID	Type	High availability	Location
sql-db-1	MySQL 2nd Gen 5.7	Add	us-central1-b

Note: On the right-hand side, the three-dot menu has a “Delete” option. Be sure to delete this instance once you are done with the homework to avoid extra charges on the instance.

6. Click on the instance ID name to open the ‘Instance details’ page, and then click on “Connect using Cloud Shell.”



At the Cloud Shell prompt, connect to your Cloud SQL instance. When the Cloud shell finishes initializing you should see:

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to central-archery-217019.
Use “gcloud config set project [PROJECT_ID]” to change to a different project.
```

7. At the Cloud Shell prompt, connect to your cloud SQL instance.

```
gcloud beta sql connect myinstance --user=root
```

Replace myinstance with the name of your instance, (in this example, sql-db-1.)

```
[redacted]@cloudshell:~ (central-archery-217019)$ gcloud beta sql connect sql-db-1 --user=root
Whitelisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
```

Enter your password (it is a linux terminal so you won’t see it being typed). You should now be able to see the mysql prompt.

```
Whitelisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 546
Server version: 5.7.14-google-log (Google)

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

Congratulations! You just finished the second part of the assignment.

Part 3: Working with SQL (Optional)

In this part of the assignment, we will build a database with one table and run queries to see if MySQL works as expected.

1. Create a SQL database on your Cloud SQL instance.

```
CREATE DATABASE test;
```

2. Insert sample data into the guestbook database:

```
~~USE test;~~  Not required in Postgres
CREATE TABLE entries (guestName VARCHAR(255), content VARCHAR(255),
entryID int not null AUTO_INCREMENT, PRIMARY KEY(entryID));
    INSERT intO entries (guestName, content) values ("first guest", "I got
here!");
    INSERT intO entries (guestName, content) values ("second guest", "Me
too!");
```

3. Retrieve the data.

```
SELECT * FROM entries;
```

You should see:

```
+--------------+------------------+---------+
| guestName    | content          | entryID |
+--------------+------------------+---------+
| first guest  | I got here!      |       1 |
| second guest | Me too!          |       2 |
+--------------+------------------+---------+
2 rows in set (0.00 sec)
mysql>
```

Congratulations! You are now ready to solve the assignment.

Part 4: Programming Assignment

A database for a movie review application consists of the following tables:

- users (**id**, name, date_of_birth).
- movies (**id**, name, genre, release_date)
- reviews (**user_id, movie_id**, rating, comment)
- actors (**id**, name, gender, date_of_birth)
- lead (**actor_id, movie_id**)

The primary key for each table is **bolded**. The user_id and movie_id of the reviews table are foreign keys referencing the users and movies tables, respectively. The same applies for the actor_id and movie_id of the lead table, which are foreign keys referencing the actors and movies tables respectively.

Notes:

- The comment column of the reviews table should allow 5000 characters.
- The format for the date_of_birth and release_date columns are 'YYYY-MM-DD'.
- The lead table contains a many-to-many relationship (multiple actors can lead in a movie and an actor can lead in multiple movies). The same applies for reviews table.
- We haven't provided any tables of data. You are responsible to make your own with the schema given above and do the query tests on them. We will have our own tables to test your queries.
- Assume the corresponding data for every query exists and that it must return some records.

Instructions:

- Please provide:
  - A working SQL query for each question.
  - Clear write-up explaining in details why each query works the way it does. Make sure to mention the database used to test the queries.
  - Table creation queries so we can test your answers.
- Make any assumptions that are not conflicting. Please only use the mentioned attributes and clarify any ambiguity. (Points might be deducted if the grader cannot make correlation between your SQL query and explanation so please be careful).

Questions:

1- List the name(s) of the user(s) born in April who rated at most 8 for the movie 'Notebook'. Output their names sorted in descending order.

2- Find user ‘John Doe’'s favorite type of movie genre(s) based on his movie review ratings. List the name(s) and genre(s) of all the movie(s) under this/these movie genre(s) sorted them based on the movie genre then movie name in the ascending order.

3- List the movie ID(s) with most male lead. Sort the IDs in descending order.

4- List the name(s) of all comedy movie(s) that were released before 2006 and have review rating better than average rating of all movies, sorted in ascending order.

   a. Note that you should compute the average of movie average ratings, not the average of all ratings. E.g. movie A got reviews 10, 10, and 10, and movie B got just one 6, the result should be ((10 + 10 + 10) / 3 + 6) / 2 = 8, instead of (10 + 10 + 10 + 6) / 4 = 9.

5- List the movie ID(s) and average review(s) where the average review higher than 9 and one of their leading actors is the actor 'Mark Clarkson'. Sort the output by average reviews and then movie IDs.

6- Find the actors who played the lead together the most. Display these their names and the number of times they played the lead together.

   a. Note: The resulting table must show both actors info in the same row (Actor1 and Actor2). This might result in duplicate data where two rows might have the same actors but in different columns. Here is an example of two actors that played the lead in two movies together:

Actor1	Actor2	count
Mark Clarkson	Jack Drake	2
Jack Drake	Mark Clarkson	2

NOTE: some of the questions may have a single or multiple results, please let the query return all of them.

Submission Guidelines:

The submission MUST be a pdf file named [Student First Name]_[Student Last Name]_HW2.pdf.
If you have any general questions about the homework, please post your questions on HW2 discussion on USC DEN course forum. Before asking, please check to see whether similar questions were asked and answered. Thank you!

Students can submit the assignment to USC DEN. Just go to the course MY TOOLS Assignments Homework 2. The deadline is firm, only submissions that make it to the system will be graded. Submit your assignment at the latest by 11:59 PM according to the clock on DEN (Dropbox) server. ***You will NOT be able to submit your homework after the deadline.*** *Also, please expect the network traffic around the deadline and network delay won't be treated as a valid reason for late submission. The system accepts multiple submissions and only the most recent submission will be graded. Therefore, we advise you to make the initial submission at least a day before the deadline, and overwrite it with a better version or more complete submission after you have it. Good Luck!*