

Assignment 2 – SQL

CSCI 585 20181

Due Date: Thursday, February 22nd, 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 1: Setting up Google Cloud Platform	2
Part 2: Setting up Cloud SQL	3
Create a Cloud SQL Instance	4
Part 3: Working with SQL (Optional)	7
Part 4: Programming Assignment	8
Submission Guidelines	9

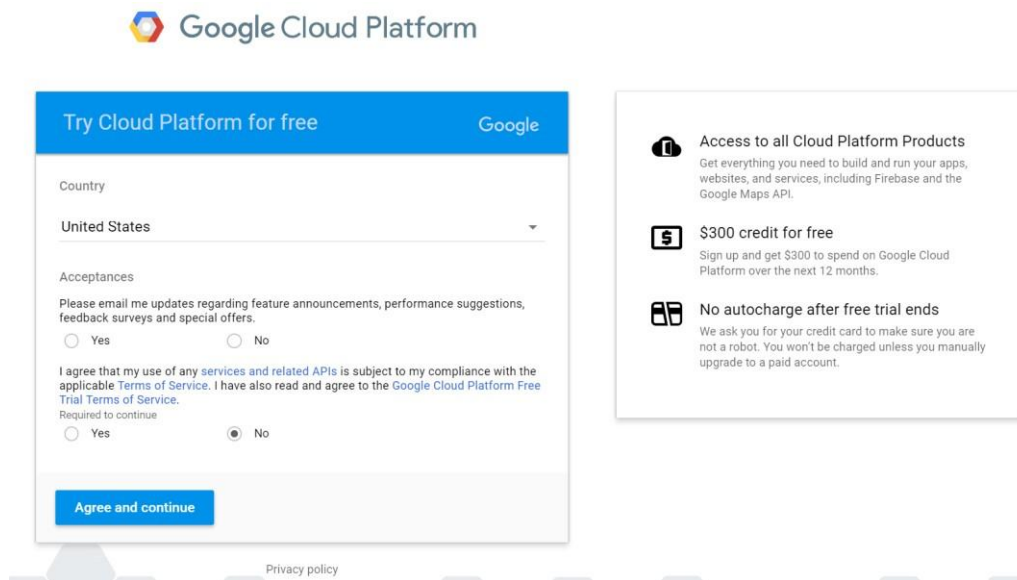
Part I: Setting up Google Cloud Platform

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.

1. Go to <https://console.cloud.google.com/freetrial> using your personal Google account. (Do not use the USC account.)
This gives you a \$300 credit to spend on the GCP for the next 12 months. Agree to the acceptances and click on 'Agree and Continue', as in Figure 1.1.
2. In the Customer info screen, select Account Type: Individual and fill in all the details. (You will be required to enter credit/debit card details, but you will not be charged as long as you don't exceed the \$300, which is sufficient for all the database assignments.)
3. Click on 'Start my Free Trial' and wait for Google Cloud Platform to set up.

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



The image shows the Google Cloud Platform sign-up page. At the top, the Google Cloud Platform logo is displayed. Below it, a blue header bar reads "Try Cloud Platform for free" with the Google logo on the right. The main form area is divided into two sections. The left section contains a "Country" dropdown menu set to "United States", an "Acceptances" section with checkboxes for "Yes" and "No" (the "No" option is selected), and a blue "Agree and continue" button at the bottom. The right section is a white box with a grey border containing three bullet points: "Access to all Cloud Platform Products", "\$300 credit for free", and "No autocharge after free trial ends". At the bottom of the page, there is a "Privacy policy" link.

Figure 1.1: Google Cloud Platform Sign-up page

Notes on Google Cloud Pricing If you go to <https://cloud.google.com/free/docs/always-free-usage-limits> you will see that there are certain usage items that are always free. For example, under the Google App Engine 28 instance hours/day and 5GB Cloud Storage are just two of several items. The Google Cloud Datastore offers 1GB storage and 50,000 reads, 20,000 writes and 20,000 deletes for free. There are many other aspects of the Google Cloud that include free elements. Unfortunately, the MySQL and PostgreSQL are not one of them. As a result, the \$300 free credit will kick in immediately.

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  sign.
2. In the next screen, (as in Figure 2.1), enter a project name.
3. Enable billing for your project.
In the navigation drawer on the left, click on Billing and add a billing account. Follow the steps and make a billing profile. The details would already be pre-filled as you had entered card details before. Click on 'Submit and enable billing'.
4. Enable the Cloud SQL Administration API. (Select the project that you made earlier in the screen like Figure 2.2.) Wait for API to be enabled and then click 'Continue'. You will be redirected to the dashboard

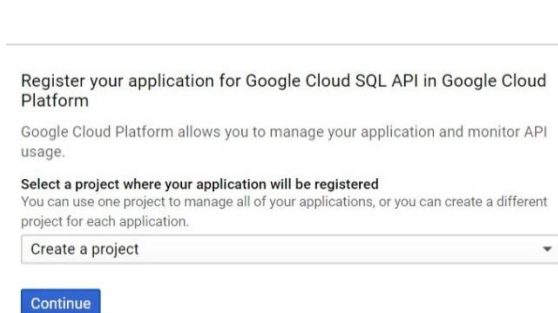


Figure 2.1: New Project Screen

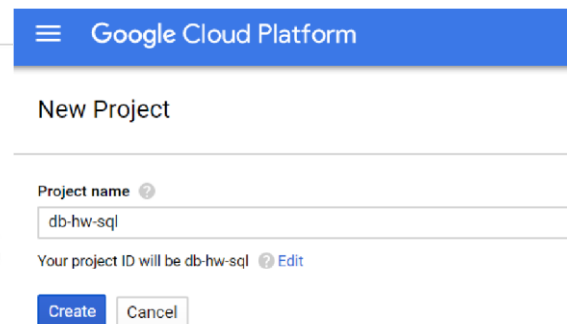


Figure 2.2: Enable the Cloud API.

Create a Cloud SQL Instance

1. Click on the 'Go to the Cloud SQL instances page' button on the Quick-start page. 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.

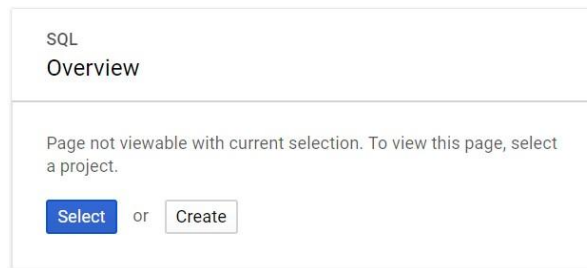


Figure 2.3 (a): SQL Overview Page.

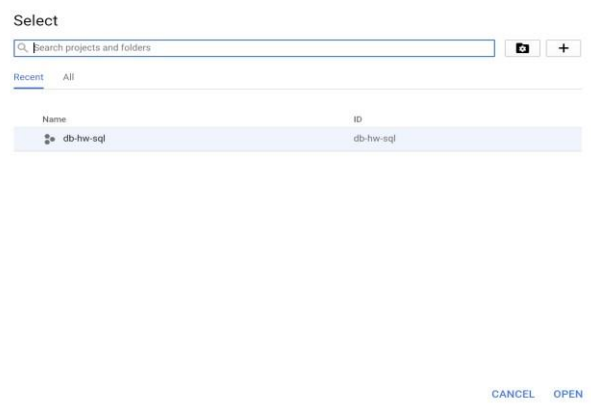


Figure 2.3 (b): Select the project

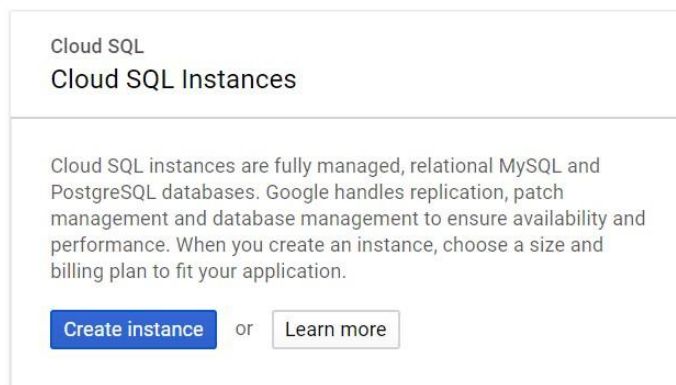


Fig 2.4: Create Instance

There are two types of Cloud SQL MySQL instances. [Learn more](#)

MySQL Second Generation (Recommended)

High performance, high storage capacity, low cost.

- Up to 7X throughput and 20X storage capacity of First Generation
- Less expensive than First Generation for most use cases
- Option to add High Availability failover and read replication
- Configurable backup period and maintenance window
- Supports only MySQL 5.6 and 5.7

[Choose Second Generation](#)

MySQL First Generation (Legacy)

Older version of Cloud SQL providing basic performance and storage capacity. Does not support MySQL 5.7.

[Choose First Generation](#)

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. Update this:

Instance ID
Cannot be changed later. Use lowercase letters, numbers, and hyphens. Start with a letter.

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

Instance ID 	Type	IP address	Failover	Storage used 	Location	
 sql-db-1	MySQL Second Generation		–	–	us-central1	
Instance ID 	Type	IP address	Failover	Storage used 	Location	
 sql-db-1	MySQL Second Generation		Add Failover	–	us-central1	

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.  
anindya_dutta22@db-hw-sql:~$
```

db-hw-sql would be replaced with the name of your 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.)

```
Welcome to Cloud Shell! Type "help" to get started.  
anindya_dutta22@db-hw-sql:~$ gcloud beta sql connect sql-db-1 --user=root
```

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.

```
anindya_dutta22@db-hw-sql:~$ gcloud beta sql connect sql-db-1 --user=root  
Whitelisting your IP for incoming connection for 5 minutes...done.  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 101  
Server version: 5.7.14-google-log (Google)  
  
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

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;
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 social networking application consists of the following tables:

- Users (**user_id**, name, gender, date_of_birth).
- Friendships (**user_id**, **friend_id**).
- Posts (**post_id**, user_id, text).
- Comments (**comment_id**, post_id, commenter_user_id, text).

The primary key for each table is bolded. The user_id and friend_id of the Friendships table are foreign keys referencing the Users table. The same applies for the user_id of the Posts table. The post_id and commenter_user_id columns of the Comments table are foreign keys referencing the Posts and Users tables respectively. Everyone can make a comment on any post.

Friendship relationships are symmetric. If user 1 is a friend of user 2, user 2 is a friend of user 1 as well. Consequently, a friendship relationship is represented as two records in the Friendships table. For example, if user 1 is a friend of user 2, the Friendships table must contain two records as follows:

userid	friendid
2	1
1	2

Notes:

- The gender column of the Users is represented as 'F' for female users and 'M' for male users.
- The format for the date of birth column is 'YYYY-MM-DD'.
- 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.
- Don't use views. They are **not** allowed in this assignment.

Instructions:

- For each query provide the SQL query and an explanation of why the query works the way it does. Make any assumptions that are not conflicting. Only **use** the **mentioned** attributes (we will make queries only with the **exact** given names of tables and attributes).

Questions:

- 1- List the ids and names of users who have no posts and have one or more comments on post_id=5.
- 2- List the ids of female mutual friends between users 1 and 2.
- 3- List the ids of users who have more than 2 friends whom have at least one post.
- 4- List unique ids of female users who were born after '1990-12-20' and commented on posts of user_id=10. Show their friends count in a separate column.
- 5- List the ids of users who commented on post_id=7 and are friends with the post creator.
- 6- List the ids and names of the 3 most female commenters, who are friends with user_id=20, with at least 3 comments on all the posts combined, excluding the comments under ones posted by user_id=10 and themselves. Show the augmented count of comments in a separate column. Also, show their total number of comments (non-augmented count) in another separate column. (Note 1: user_id=20 should not be among the 3 answers. Note 2: **Augmented count of comments** is the count of the set of all comments not under "posts by user_id=10 and their closed (3 person) group posts" (so exclude all the comments under these 4 people posts when counting the number of comments for each female commenter). Note 3: ranking **should be** based on the augmented count of comments. Note 4: for tie cases, make a simple choice and write it down (your personal choice).

Submission Guidelines:

Some notes for your reference:

- 1) The submission MUST be a pdf file named [Student First Name]_[Student Last Name]_HW2.pdf.
- 2) If you have any general questions about the homework, please post your questions on HW2 discussion on USC DEN course forum only.
- 3) Please don't email the TAs directly with questions. Let other students in the class also benefit from answers and questions. This will also ensure, all the TAs can contribute to class.
- 4) Before asking, please check the forum to see whether similar questions were asked and answered.
- 5) Note that your programs will be automatically checked against other codes. This will be in the amount of percentage similarities. Any percentage can risk your code credibility. So don't show/share your code with anyone.

*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!*