

Project 1, Part 2

- Assigned: 9/28
- **Due: 10/23 10:00AM** via Gradescope
- Value: 25% of Project 1 grade

In this part of the project, you will revise your design based on the staff's feedback on [part 1](#). You will implement the database tables (including all constraints), populate the database, and write some queries.

- [Project overview](#)
- If your teammate has dropped the class, [see the contingency plan](#)
- For any questions about collaboration, [see the Syllabus](#)
- If there are questions of general interest, please post to Piazza.

Logistics

Project Mentor

The TA that graded your part 1 will be your project mentor for the rest of your project -- this is likely the TA that you discussed part 1 with. He/she will be your main contact for the project, though the rest of the staff are of course available for questions or concerns.

Google Cloud Platform

- Go to <https://www.cs.columbia.edu/auth/cloud> and fill out your name to create a Google Cloud account. You will receive an invitation email from no-reply@cloud.cs.columbia.edu, go to <https://console.cloud.google.com> and log in with cloud.cs account and the temporary password. You will be prompted to change your password.
- We have sent your coupon code for Google Cloud by e-mail. **Make sure you are logged in with your @cloud.cs.columbia.edu account.** Sign out from other Google accounts, go to <https://console.cloud.google.com/education>, enter your code in the "coupon code" box, then click accept and continue.
- Right after redeeming your code, follow [these instructions](#) on how to configure your Google Cloud environment and create your instance.
- For Part 2, you will use your instance to connect to the course database server, and to get familiarized with the instance environment, which you'll use later on Part 3 to run your web application.

Procedures

Preliminaries

Check your graded Part 1 starting on **10/11**, and revise your design based on its feedback.

- You will be graded based on how well you addressed the project mentor's comments.
- In general, listen to your project mentor's suggestions.

Familiarize yourself with the [PostgreSQL documentation](#)! We will use PostgreSQL 10.

Connecting to the class database

- Navigate to your VM instance in the Cloud Platform Console, start it again, and click the SSH button that appears next to it. A terminal window will pop up.
- After the installation is completed, connect to our postgres database using `psql` command:

```
psql -h w4111.cisxo09blonu.us-east-1.rds.amazonaws.com -U YOUR_UNI w4111
```
- It will ask for your password, which is included in the e-mail we sent. If you didn't get the message, post a private question on Piazza. You may play with Postgres a little bit before the graded project 1 part 1 is returned to you.
- If the database cannot handle the number of connections, we may create a second database server (we will let you know!)
- **Don't forget to stop your VM instance after you're done using it or it will use up your credits!!!**

Creating your schema

Create the SQL tables based on your revised schema.

- Each student gets an individual account and environment (i.e. a PostgreSQL "schema") on the database server, so decide with your teammate which database account you will be using. You will share such account credentials and use it for submission.
- Include all key and type constraints.
- The PostgreSQL documentation for [CREATE TABLE](#) and [data types](#) may help.

Create the CHECK constraints that you need to express the rest of your real-world constraints.

- Note: PostgreSQL's CHECK constraints are limited ([see the documentation](#)), so do what you can.
- Note: PostgreSQL doesn't support CREATE ASSERTION statements, but does support triggers. However, you are not required to implement constraints that require triggers.

Populate the tables

Insert at least 10 realistic/real tuples into each table in your database.

- This should be based on your description in part 1

Run some queries

Create at least 3 interesting SELECT queries. The three queries, together, should include

- multi-table joins,
- WHERE clauses, and
- aggregation (e.g. COUNT, SUM, MIN, etc).

Each query does not need to include all of those SQL features.

Submission

Since you created the database on the course database server, we have access to your database and populated tables, so you are almost done!

Submit a PDF file **via Gradescope** containing:

- your UNIs;
- the UNI used to create the schema on the course database server (no need to send your password);
- the 3 interesting queries you created, along with a short description for each of them.
- descriptions of any changes to the application, data modeling, or schema that you have made since Part 1 of the project. Remember to have short explanations for why you made those changes.

If you perform any changes to your schema after your Gradescope submission, we will consider it as your new submission date/time. SELECT queries are allowed.

Grading

Grading will be based on the following:

- How well you incorporated your mentor's feedback (important)
- Quality of the SQL schema and implementation: how well it conforms with the ER diagram and constraints
- Your SQL statements: are they reasonable application queries and do they use the SQL features as requested?
- Quality of the data: is it realistic?