

LAB 1

Lab Overview

Start by reading and interpreting the case study to find the customer's requirements for this project. In this lab, you are expected to create schemas with all the attributes and domains using the case study, populate it, and then query it. Ensure you save all Create table statements and Insert statements so you can show these during the presentation, and alter these after receiving feedback on your design.

Rules

1. You must follow the rules of the [honor code](#). The labs must be done in groups of exactly two people. No larger groups are allowed, and if you have extraordinary extenuating circumstances that force you to do the labs alone, you must obtain permission to do so from the course leader.
2. Presenting P+ assignments are optional for a higher grade if the given tasks are completed and passed.
3. This is a PSQL lab. No other programming languages, either embedded in the database or external to it, are allowed.
4. Please refrain from creating any [functions](#) since this lab is designed to assess query programming languages

Lab Presentation

Course related terminology is expected during the presentation. Make sure you have the following ready to be presented to the TA:

- Your schemas and your code. Show how you have translated them into the database. Specifically show your keys and domains.
- The contents of the database.
- Optional: Present your P+ solution.

Before you start

Read through the “PostgreSQL Installation Guide” and have the terminal ready for the tasks. Make sure you are saving all your insert statements. PostgreSQL uses the “.psql” file format. You can use your preferred text editing software.

***Note:** You are allowed to skip using the server and instead set up the database locally. In this case it is your responsibility that everything works as expected.*

LAB 1

Tasks

0. Schemas

Analyze the case study and create schemas. The schemas do not have to be exactly correct for this lab, however they will have to be right to pass the homework.

Note: If you need guidance on creating schemas look at homework 1.

1. Creating the database

Create all the tables that are necessary for the Social Media Platform based on your solutions.

Make sure to save all code for the create table statements, insert statements and queries for the presentation.

The whole database must have:

1. Well thought out names for each table and attribute.
2. Reasonable domains for each attribute.
3. Sensible primary key(s) for each table.
4. Correctly implemented foreign keys. (Check this [link](#)).

Finally, check that the tables have been properly added to the database.

For each table, write in the prompt:

```
SELECT * FROM <Insert table name here>;
```

Example:

```
SELECT * FROM countries;
```

This can be read as: **Select** all **attributes** from **the schema countries**

LAB 1

2. Adding constraints to the database

A vital aspect of database design is the proper use of constraints. Constraints impose certain requirements on the data inserted, to ensure that certain conditions hold. Understanding the capabilities of the different kinds of constraints is vital.

The following constraints should be implemented in your database:

1. Foreign key constraints, everywhere applicable
2. Not null constraints, everywhere applicable
3. Check constraints:

Check constraints allow for more complex logic and come in two main varieties: attribute-based and tuple-based. You should know and be prepared to explain the differences between the two varieties. Then, implement the following as either attribute- or tuple-based checks:

- a. PostID should be a non-negative integer
- b. StartDate should always be on the same day or earlier than EndDate.
- c. All tags should be “crypto”, “studying”, “question” or “social”

Note: You should be able to motivate for 3.a-3.c why the constraints are implemented as attribute-based or tuple-based.

Additional note:

In PSQL, while it is technically possible to use queries in check constraints, it is not recommended if the conditions are mutable. This is due to the operating principles of PSQL and most other SQL dialects, as the queries are not re-run upon every update to the database. If the conditions are mutable, the constraint is not guaranteed to hold. For more information, see this [link](#).

Further, database integrity can be maintained by triggers. During the presentation, you should be able to answer the following:

1. What is a trigger?
2. Name 3 events that can cause a trigger to activate
3. What can be done with triggers?

Note: You do not need to implement any triggers in your database.

LAB 1

3. Populating the database

Insert data into each table fulfilling (a-e) and check the relations thoroughly to make sure the data is inserted correctly.

The database must contain at least:

- a. 6 users
- b. At least 4 friend relationships
- c. 3 posts, one of each type (Text, Video & Image)

Each post must have:

- between 0-2 tags, where at least one must have 2 or more tags.
- between 1-6 likes

- d. 1 Event
 - created/hosted by 1 user
 - only friends of the host will attend
- e. 6 subscriptions

4. Querying the database

The following statements for querying a database are to be executed and should show a satisfactory result.

Note: All queries only require information from one table at a time. Joins are not necessary.

1. Find and display all users' full names.
2. Display all friend relationships
3. Display all posts in 3 separate tables for each type (Text, Video & Image)
4. Display all created events
5. Display all Subscriptions

LAB 1

P +

Make the CREATE TABLE statements in PSQL for the following ER Diagram.

