

Databases

Lab 03

Andrés Calderón, Ph.D.

February 19, 2025

1 Introduction

We will work in three parts today. First, we will reinforce what we learn in class watching a couple of videos from [w3schools.com](https://www.w3schools.com) about basic SQL. Then we will explore some of the lessons in the interactive tutorial from the same website. Finally, we will finish running some SQL code in the `university` database we will download from the book website.

Let's go!

2 W3Schools SQL YouTube Videos

We will watch a YouTube playlist from the w3shools.com channel about introductory SQL. You can access the playlist [here](#). You can activate the Spanish subtitle in the Settings button if you want. There are 11 short videos. Just watch them, there are not any particular submission to do for now.

3 W3Schools Interactive SQL Tutorial

Now, you will go through the interactive SQL tutorial at the W3Schools website, so first you need to create an account. The tutorial start [here](#). You will see several lessons in the left panel (red square in Figure 1). You only have to cover the followings:

- Intro
- Syntax
- Select
- Select Distinct
- Where
- Order By
- And
- Or
- Not
- Null Values
- Aggregate Functions

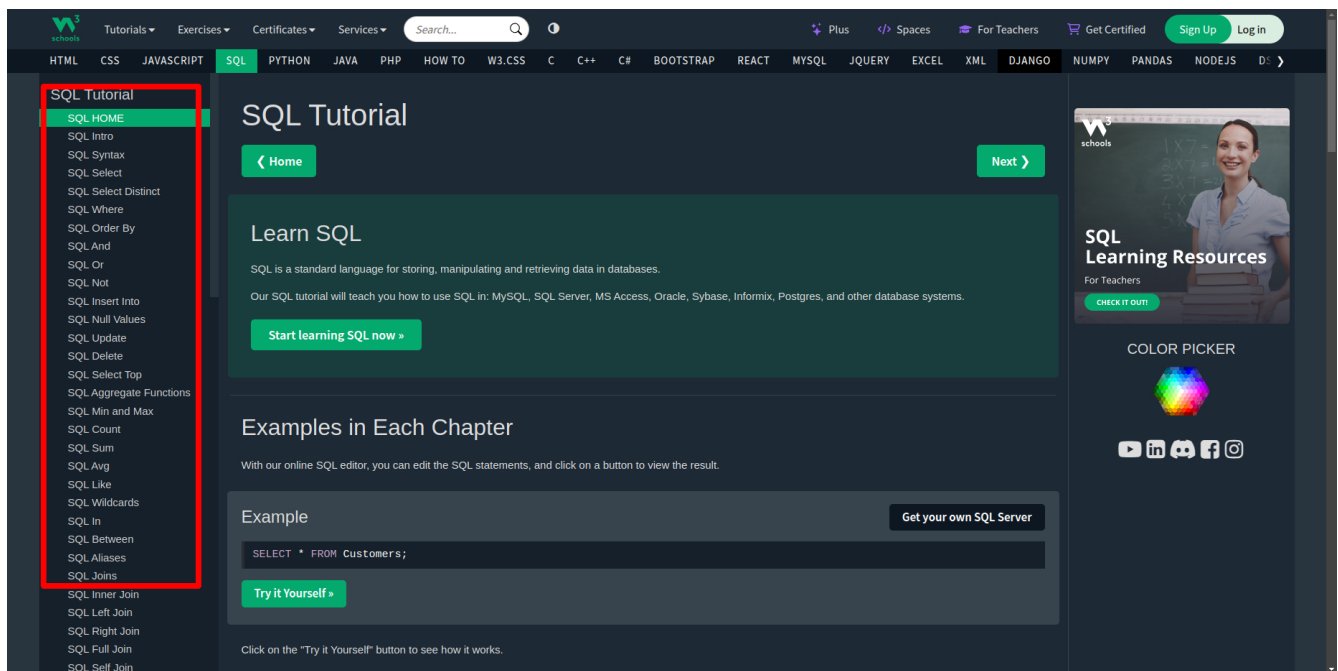


Figure 1: Lessons panel at W3Schools SQL Tutorial.

- Min and Max
- Count
- Sum
- Avg
- Like
- Wildcards
- In
- Between
- Aliases

Remember capture and screenshot when you execute each of the SQL statement so you can attach them later on your report. Try to capture your username if possible.

4 The university Database

The website of the book *“Database System Concepts”* provides backups of the **university** database we have been following in classes. We will load a small version in PostgreSQL so we can practice a bit the SQL statements we already saw. You will need to download two files:

1. The **DDL.sql** file with the code to create the database structure.

2. The `smallRelationsInsertFile.sql` file with the code to populate the database with the same records we can find in the book examples.

Download both files and put them in the same folder.

We will assume that you have already a functional instance of PostgreSQL running in your machine. We expect you have access to a PostgreSQL console similar that Figure 2.

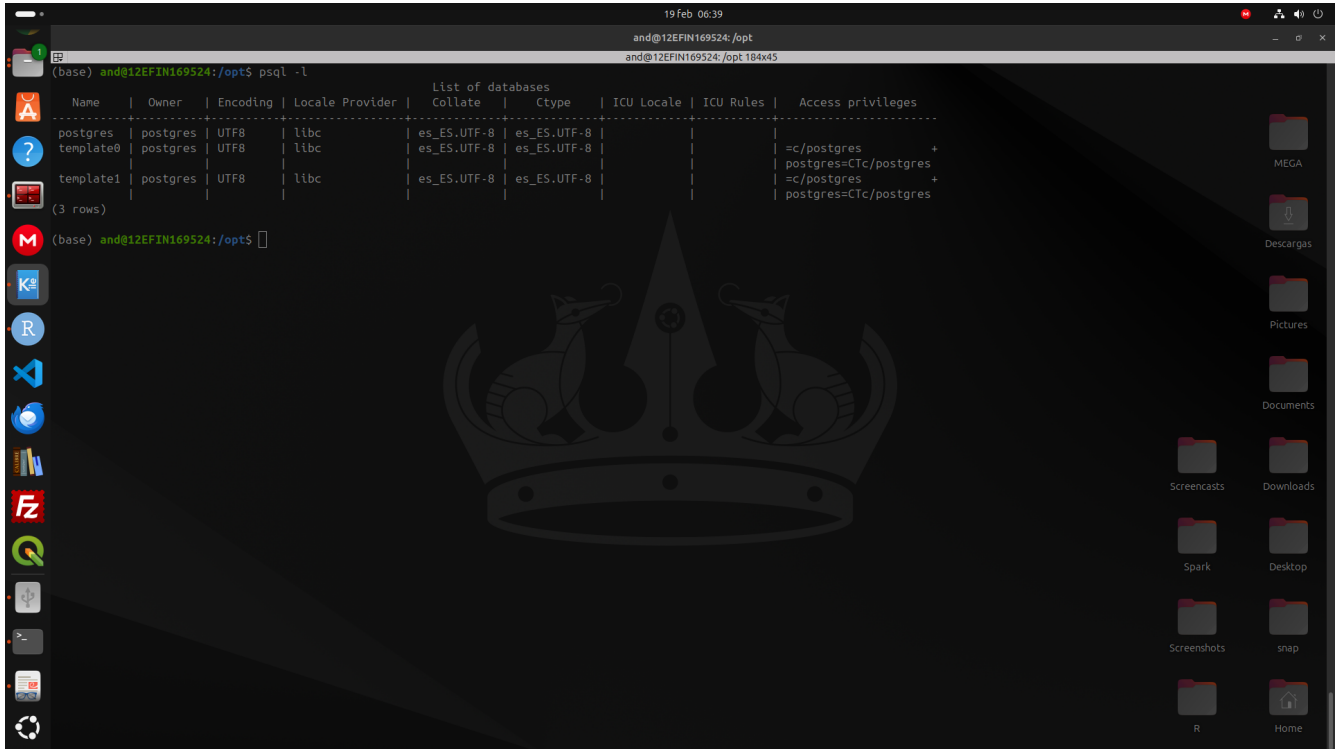


Figure 2: PostgreSQL installed in your system.

First, we will create a database named `university`:

```
createdb university
```

Next, we will connect to that database:

```
psql university
```

Now we are in the PostgreSQL prompt. There we will use the `\i` command to load the schema and data for our database. You will need to change the paths of the downloaded files accordingly.

First, we load the `DDL.sql` file:

```
\i /path_to_files/DDL.sql
```

Then, the `smallRelationsInsertFile.sql` file:

```
\i /path_to_files/smallRelationsInsertFile.file
```

And you are done! Let's try some SQL:

```
SELECT * FROM instructor;
```

Now you are ready to explore the SQL we have already seen in class.

5 Independent Work

Now that we have the **university** database ready to go, you are asked to answer the following queries:

1. Retrieve the names of all instructors, along with their department names and department building name.
2. For all instructors in the university who have taught some course, find their names and the course ID of all courses they taught.
3. Find the names of all departments whose building name includes the substring 'Watson'.
4. Find the set of all courses taught in the Spring 2018 semester and in the Fall 2017 semester.
5. Find all instructors who appear in the instructor relation with null values for salary.
6. Find the average salary in each department.
7. Show those departments where the average salary of the instructors is more than \$42,000.
8. Find the total number of (distinct) students who have taken course sections taught by the instructor with ID 110011.
9. Find the names of all instructors that have a salary value greater than that of each instructor in the Biology department.
10. Finally you will propose your own query and solve it using SQL by you own.

For each query you have to present its corresponding Relational Algebra Expression, the SQL code, and the result of the query (copy-paste or screenshot). We expect you to submit a well-structured report in PDF format, containing the requested information in the previous sections. Additionally, include the SQL code with the answers to the queries in a ZIP file. The submission deadline is **March 5th, 2025**.

Happy Hacking ☺!