

# Databases

## Lab 02: A ‘gentle’ Introduction to Basic SQL.

Andrés Oswaldo Calderón Romero, Ph.D.

August 12, 2025

### 1 Introduction

In this lab, you will strengthen your understanding of basic SQL through a combination of guided resources and hands-on practice. We will begin by revisiting the core concepts discussed in class using short video tutorials and interactive exercises from W3Schools. These activities will provide a structured review of SQL syntax, selection, filtering, aggregation, and other foundational operations, ensuring you are comfortable with the essential commands before moving on to more complex tasks.

Once familiar with the basics, you will apply your skills using the **university** database from the Database System Concepts book. You will load a smaller version of this database into PostgreSQL and run a series of queries that combine SQL coding, relational algebra expressions, and output verification. This practical component will help you bridge the gap between theoretical knowledge and actual database manipulation, preparing you for more advanced SQL topics in future labs.

Let’s go!

### 2 W3Schools SQL YouTube Videos

We will watch a YouTube playlist from the [W3Schools.com](#) channel covering introductory SQL. You can access the playlist [here](#). If you prefer, you can enable Spanish subtitles via the Settings button. The playlist consists of 11 short videos. Simply watch them—there is no specific assignment or submission required for now.

### 3 W3Schools Interactive SQL Tutorial

Now, you will go through the interactive SQL tutorial on the W3Schools website. First, create a free account. The tutorial begins [here](#). You will find several lessons listed in the left panel. You only need to complete the following:

- Intro
- Syntax
- Select
- Select Distinct
- Where
- Order By
- And
- Or

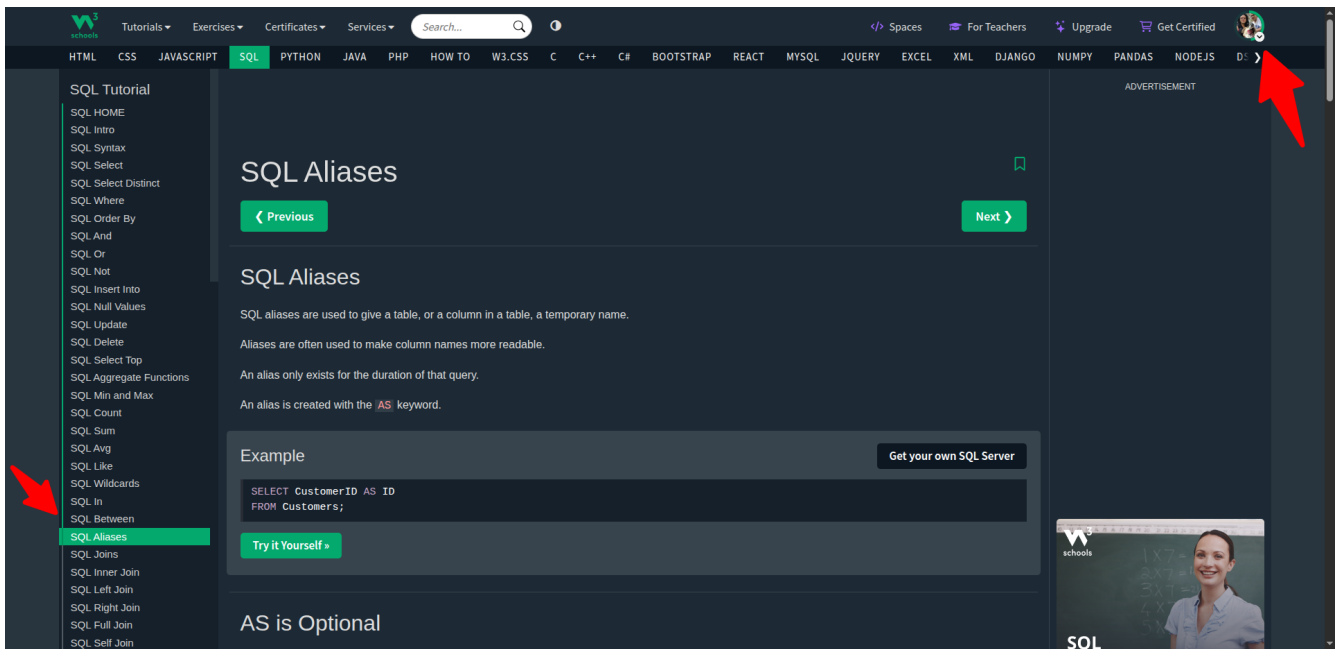


Figure 1: Lessons panel displayed in the W3Schools SQL Tutorial, showing progress tracking in the left sidebar.

- Not
- Null Values
- Aggregate Functions
- Min and Max
- Count
- Sum
- Avg
- Like
- Wildcards
- In
- Between
- Aliases

Once you have completed all the lessons, take a screenshot similar to Figure 1. Make sure your profile picture and your progress (green line in the left panel) are clearly visible. Each team member must provide their own screenshot.

```
vagrant@vagrant-ubuntu: ~  
vagrant@vagrant-ubuntu: ~ 190x45  
vagrant@vagrant-ubuntu: $ psql -l  
List of databases  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
Name | Owner | Encoding | Locale Provider | Collate | Ctype | ICU Locale | ICU Rules | Access privileges  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
postgres | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | =c/postgres +  
template0 | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | postgres=CTc/postgres +  
template1 | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | =c/postgres +  
vagrant | vagrant | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | postgres=CTc/postgres  
(4 rows)  
vagrant@vagrant-ubuntu: $
```

Figure 2: Ensure that PostgreSQL is installed on your system.

## 4 The university Database

The website for the book *Database System Concepts* provides backups of the **university** database we have been using in class. We will load a smaller version into PostgreSQL so we can practice the SQL statements we have already learned. You will need to download the following two files:

1. The **DDL.sql** file, which contains the code to create the database structure.
2. The **smallRelationsInsertFile.sql** file, which contains the code to populate the database with the same records used in the examples from the book.

Download both files and place them in the same folder. We will assume that you already have a functional PostgreSQL instance running on your machine. You should be able to access a PostgreSQL console similar to Figure 2.

First, create a database named **university**:

```
createdb university
```

Next, connect to the newly created database:

```
psql university
```

You are now in the PostgreSQL prompt. From here, use the `\i` command to load the schema and data for the database. Remember to update the file paths to match the location where you saved the downloaded files.

First, load the `DDL.sql` file:

---

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

---

Then, load the `smallRelationsInsertFile.sql` file:

---

```
\i /path_to_files/smallRelationsInsertFile.sql
```

---

And that's it! You're all set. Let's run a quick SQL query to test:

---

```
SELECT * FROM instructor;
```

---

You are now ready to explore the SQL queries we have already covered in class. If you encounter any issues, watch this [video](#), which provides a detailed walkthrough of how to load the schema and data for the **university** database.

## 5 Independent Work

With the **university** database prepared, proceed to execute and answer the following queries:

1. Retrieve the names of all instructors, along with their department names and the corresponding department building names.
2. For all instructors in the university who have taught at least one course, find their names and the course IDs of all courses they taught.
3. Find the names of all departments whose building name contains the substring **Watson**.
4. Find the set of all courses taught in both the Spring 2018 semester and the Fall 2017 semester.
5. Find all instructors who appear in the **instructor** relation with a **NULL** value for salary.
6. Find the average salary in each department.
7. Show the departments where the average salary of instructors is greater than \$42,000.
8. Find the titles of courses in the **Comp. Sci.** department that have 3 credits.
9. Find the enrollment of each section that was offered in Fall 2017.
10. Find the names of all instructors whose salary is greater than that of every instructor in the Biology department.
11. Propose your own query and solve it using SQL.

For each query, you must provide its corresponding *relational algebra* expression, the **SQL** code, and the query result (either a copy-paste of the output or a screenshot).

We expect a well-structured report in **PDF** format containing the requested information from the previous sections. Additionally, include a plain-text file with extension **.sql** that contains all the statements from Section 5. Package both files (**.pdf** and **.sql**) into a single **ZIP** file and submit it via Brightspace. The submission deadline is **August 27, 2025**.

Happy Hacking! 😊