

# Introduction to Databases

---

The purpose of this exercise is to practice the important skill of analyzing data in databases using Structured Query Language (SQL).

## Learning Objectives

After completing this exercise, students will understand:

- How to write **SELECT** statements.
- How to filter data using **WHERE** clauses.
- How to execute mathematical expressions in SQL statements.
- How to filter data for **NULL** values.

## Evaluation Criteria & Functional Requirements

- All of the queries run as expected.
- The number of results returned from your query is equal to the number of results specified in each question.
- Code is clean, concise, and readable.

To complete this exercise, you need to write SQL queries in the **intro-to-databases-exercises.sql** file. Below each commented out question, you'll write the query necessary to answer the question being asked using the world database as the source.

## Getting Started

- Open the **intro-to-databases-exercises.sql** file in DB Visualizer.
- If you have not done so already, create the world database. The script for this should be available in today's lecture code.
- In the "Database Connection" properties above the file, select the world database.
- You can run all of the database commands in the file at one time by pressing the command + enter key at the same time.
- You can run a single database command at a time by highlighting the command and then pressing the command + enter key at the same time.

## Tips and Tricks

- **SELECT** statements specify the columns of a table that you want to return from a query. While the values in the **SELECT** statement are usually directly mapped to a column name, they can also be used aliased using the **AS** keyword.
- **WHERE** clauses filter results. Some operators you can use for filtering out data include:
  - **=, <>, !=, >, >=, <, <=**
  - **IN(values), NOT IN(values)**
  - **BETWEEN value AND value**
  - **IS NULL, IS NOT NULL**
  - **LIKE, ILIKE** (with wildcard characters)

- Multiple filter conditions can be combined using **AND** and **OR**.
- The **DISTINCT** clause removes duplicate values from the results.
- The PostgreSQL documentation includes a [tutorial for querying database tables](#), as well as [documentation related to the \*\*SELECT\*\* statement](#).