

# Joins

---

The purpose of this exercise is to practice using [joins](#) to combine data from multiple tables in a single query result using Structured Query Language (SQL).

## Learning Objectives

After completing this exercise, students will understand:

- How to interpret database diagrams to determine how tables are related.
- How to use SQL [JOIN](#) clauses to combine data from multiple tables in a database query.
- The difference between [INNER](#) AND [OUTER](#) joins and when to use one or the other.
- The difference between a [LEFT](#) and [RIGHT](#) join.

## 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 [joins-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 [joins-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 yesterday'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

- The results of an [INNER JOIN](#) between tables A and B consist of the intersection of A and B. That is, only the records that exist in both tables based on the [JOIN](#) condition are returned.
- The results of an [OUTER JOIN](#) between tables A and B are all of the records that exist in the [LEFT](#) table (A) and any matches that exist on the [RIGHT](#).
- You can specify which table ([LEFT](#) or [RIGHT](#)) all of the results should be returned from explicitly.
- See the [PostgreSQL documentation](#) for more information on how table joins work in PostgreSQL.
- [Some people find value in using Venn diagrams](#) to understand and explain SQL joins, while [others argue that this is wrong and join diagrams are easier to understand](#). Which one is the correct way to

explain table joins? The way that makes the most sense to you is the correct way. Try taking a look at both explanations to determine which one is best for your understanding.

---