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, you'll 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 and 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.
- The unit tests pass as expected.
- Code is clean, concise, and readable.

To complete this exercise, you need to write SQL queries in the files that are in the `Exercises` folder. You'll use the `MovieDB` database as a source for all queries.

In each file, there's a commented out problem statement. Below it, write the query needed to solve the problem. The value immediately after the problem statement is the expected number of rows that must be returned by the query.

> Note: remember that this database is a random sampling of movies and actors and doesn't have every single movie and actor that appeared in them.

Getting started

1. If you haven't done so already, create the `MovieDB` database. The script for this is available in today's lecture code.
2. Open the `Exercises` folder. Each file is numbered in suggested order of completion, but you can do them in any order you wish.
3. Launch pgAdmin and open each numbered exercise file one-by-one. Write and run the query for the individual exercise. Save the file before moving onto the next exercise
4. The unit tests project is in the same directory as this README. You can open it in IntelliJ and run the tests as you did in earlier exercises.

> Note: Make sure to save your changes to the SQL file before running the unit tests.

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.
- Remember that `DISTINCT` removes duplicate values in the results. If your result set has more than the expected number of rows, check your results for duplicates.
- Some exercises ask you to return the results in a specific order and others don't. If no sort order is requested, you don't need to include an `ORDER BY` clause.
- 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. Look at both explanations to determine which one is best for your understanding.
