

Theme Song: Star Wars (Main Title)

SQL is the language of relational databases. It's important to know how to query and manipulate structured data so that when you end up building your own database, you have an idea of how to use it. In this project, we'll practice writing SQL queries for the IMDb database.

1 Background Knowledge

To get started with SQL, check out our [Guide to SQL](#)! Even if you're already proficient in it, everybody benefits from a brief refresher.

Note that in this project, we are using **sqlite3 version 3.22** - this means that not all features you learn about in class are necessarily available. (*e.g.* window functions are not available for you to use) We assure you, however, that all of the problems are possible to do in sqlite3.

2 Assignment Spec

2.1 Overview

In this assignment, you will write SQL queries for the IMDb database. We have provided our version of this database with download instructions below.

2.2 Part 1:

1. Write a SQL query to find `primary_title` of top 10 movies which have received an average rating of more than or equal to 8.5, or have been voted by more than or equal to 1000000 people. The results should be sorted by the average rating in descending order.
2. Write a SQL query that gets the director's `primary_name` for the movie 'Poeta'.
3. Write a SQL query that returns the `primary_name` of directors and count of titles directed by each director. Results should be ordered by the count of titles in descending order
4. Write a SQL query that returns the different genres of all the movies and average runtime of all genres.
5. Write a SQL query that finds `primary_name` of directors who have directed movies that received an average rating less than or equal to 5.

2.3 Part 2:

1. Write a SQL query that returns `primary_title` and `start_year` of movies whose titles begin with 'The' and were released in a leap year (hint: leap years are divisible by 4).
2. Write a SQL query that returns the `primary_title` and `start_year` of movies that were released 19 years after the birth year of Walt Disney. (Hint: the `CAST` function may come in handy)

3. Write a SQL query that finds the `primary_title`, `start_year` and `runtime_minutes` of all shows whose `runtime_minutes` have exceeded the average runtime minutes of movies released in the same year. Results should be ordered by start year ascending and runtime minutes descending.

2.4 Part 3:

1. Write a SQL query that finds the `primary_title`, `avg_rating`, and `reviews` of 20 movies. The reviews depend on average rating. A movie with rating less than or equal to 3 should be reviewed as 'poor'. A movie with rating greater than 3 and less than or equal to 6 should be reviewed as 'okay' and a movie with rating greater than 6 should be reviewed as 'good'. The result should be sorted by the title (descending order) (Hint: The CASE function may come in handy).
2. Write a SQL query that returns the decades and percentage of titles which were released in the corresponding decade.
3. Write a SQL query that returns the name of director and a boolean value if the director has directed more than or equal to 5 movies. The boolean value will be 1 if true else 0.

2.5 Testing

To validate that your queries are working as intended, upload your submission to Gradescope and view the auto-grader output. If you notice a weird, fractional score, it is because we are calculating your grade using [F-score](#). This means that returning all rows or returning no rows won't get you *closer* to the answer; rather, only by returning more correct rows and returning fewer incorrect rows can you maximize your score.

2.6 Getting started

To get started, get your stencil repository [here](#). Then, download our database from [here](#), and place it in the stencil repo. A good first step is to confirm that the database has data in it, and that you can query it. You will write your queries for each subproblem in the provided sql file, and run it using a command like the following:

```
sqlite3 database.db < sql/1_1.sql.
```

If you don't have sqlite3, then install it from [here](#), or your package manager of choice. If you install it from the website, find the pre-compiled binary for your operating system. Note that sqlite3 is already installed on all department machines, so feel free to work from the CIT or SSH into a remote machine. Just be sure to save your work and keep it permissioned correctly so other students can't see what you've done!

Feedback

As this is a new course, we appreciate any feedback you have for us! If you enjoyed this assignment, hated this assignment, or have other thoughts to share, please do so [here](#)!