WEEK 4 - SQL BASICS

Links:

\*[Movies and Actors Playground](https://teamtreehouse.com/sql_playgrounds/32#/queries/8a72e7cb-7d11-4f51-bb9b-abb283daa11f)

\*[Books Playground](https://teamtreehouse.com/sql_playgrounds/112#/queries/5e90e962-c20f-4850-b561-eef1855c95d2)

Welcome to the SQL Programming Challenges! These activities have been designed to test and enhance your SQL querying skills. In each challenge, you'll work with the provided database schema, focusing on crafting precise queries to retrieve specific data. The goal is to explore the potential of the essential clauses learned during the G-code track, while solving real-world scenarios.

The working environment for these activities will be the treehouse playgrounds at the top of this document. *Please check that you have access to them before starting.*

**To start please create five new queries in each playground and go to the activities.**

**\*\*Note:** Don't forget that as soon as you close the SQL interface you will lose everything you have written, so consider storing what you write in a text editor such as Google Docs.

**Movies and Actors Playground**

1. From the actors dataset, retrieve the names of actors whose names start with the letter 'A' or end with the letter 'n'.

Select name from actors where name like “a%” or name like “%n”;

1. Retrieve the usernames and reviews where the review contains the word 'scary' or 'cute'.

Select username, review from reviews where review like “%scary%” or review like “%cute%”;

1. Find the titles of movies that do **not** have the word 'Sci fi' in their genre AND were released in the year 1985 or later.

Select title from movies where genre not like “%sci fi%” and year\_released >= 1985;

1. Retrieve the "id" and "name" of actors whose names contain 'cole,' renaming the "id" column to 'Actor ID'.

Select id as “Actor ID”, name from actors where name like “%cole%”;

1. A. Return the id and rating of movies that were reviewed by the user 'chalkers'.

Select movie\_id, rating from reviews where username = “chalkers”;

B. Find the name and year of release for these movies

Select title, year\_released from movies where id in (6, 3, 7);

**Books Playground**

1. Find all books whose author begins with the letter **J** and which were published before 2002.

*\*\*Return the columns ‘title’ and ‘first\_published’ as: 'Title' and 'Release Date '.*

Select title as “Title”, first\_published as “Release Date” from books where author like “j%” and first\_published < 2002;

1. Retrieve the title of all books whose genre does not contain the word 'fiction'.

Select title from books where genre not like “%fiction%”;

1. A. Find the ID of all people who have not yet returned their borrowed books.

Select patron\_id from loans where returned\_on is null;

B. Get the name, email and address of those who have not returned the books.

Select first\_name, last\_name, email, address from patrons where id in (2, 1, 3);

1. A. Find the ID of all books borrowed after December 11, 2015.

Select book\_id from loans where loaned\_on > “2015-12-11”;

B. Retrieve the name and author of these books.

Select title, author from books where id in (8, 9, 11);

1. This challenge is a bit more complicated than the previous ones as it requires information from each dataset. To solve it try to break the problem into shorter instructions by thinking about what information you need from each dataset.

Also think that there is no single valid solution and you can do it in the way that seems most reasonable to you.

The challenge is to find the ID of all the libraries where **NON FICTION** books have been loaned.

Select id from books where genre = “Non Fiction”;

Select patron\_id from loans where book\_id in (8, 9);

Select library\_id from patrons where id in (1, 3);