**SQL Lesson 1: SELECT queries 101:**

****

1. **Find the title of each film?**

Answer: SELECT title FROM movies;

1. **Find the director of each film?**

Answer: SELECT director FROM movies;

1. **Find the title and director of each film?**

Answer: SELECT director, title FROM movies;

1. **Find the title and year of each film?**

Answer: SELECT year, title FROM movies;

1. **Find all the information about each film?**

Answer: SELECT \* FROM movies;

**SQL Lesson 2: Queries with constraints (Pt. 1):**

****

1. **Find the movie with a row id of 6?**

Answer: SELECT \* FROM movies where id=6;

1. **Find the movies released in the years between 2000 and 2010?**

Answer: SELECT \* FROM movies where year between 2000 and 2010;

1. **Find the movies not released in the years between 2000 and 2010?**

Answer: SELECT \* FROM movies where year not between 2000 and 2010;

1. **Find the first 5 Pixar movies and their release year?**

Answer: SELECT \* FROM movies limit 5;

**SQL Lesson 3: Queries with constraints (Pt. 2):**

****

1. **Find all the Toy Story movies?**

Answer: SELECT \* FROM movies where title like 'toy%';

1. **Find all the movies directed by John Lasseter?**

Answer: SELECT \* FROM movies where director like 'john%';

1. **Find all the movies (and director) not directed by John Lasseter?**

Answer: SELECT \* FROM movies where director not like 'john%';

1. **Find all the WALL-\* movies?**

Answer: SELECT \* FROM movies where title like 'wall-\_';

**SQL Lesson 4: Filtering and sorting Query results:** 

1. **List all directors of Pixar movies (alphabetically), without duplicates?**

Answer: SELECT DISTINCT director FROM movies order by director asc;

1. **List the last four Pixar movies released (ordered from most recent to least)?**

Answer: SELECT \* FROM movies order by year desc limit 4;

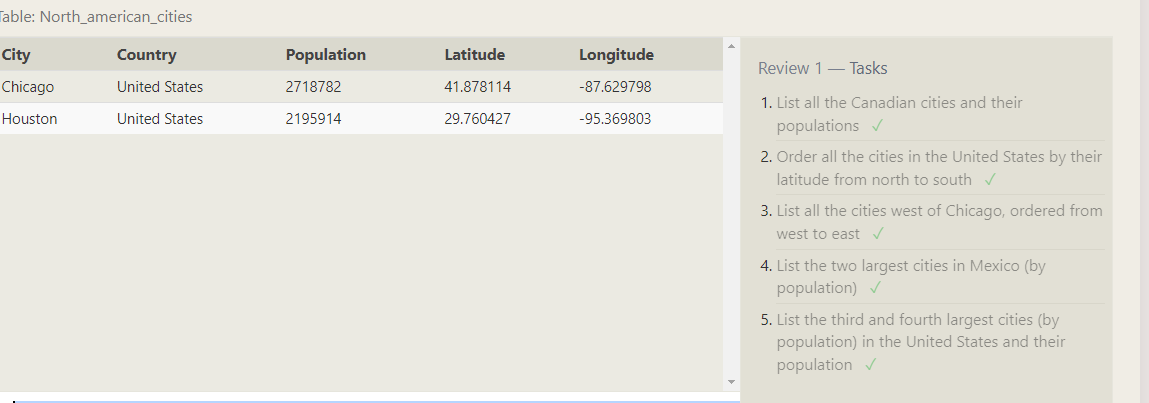
1. **List the first five Pixar movies sorted alphabetically?**

Answer: SELECT \* FROM movies order by title asc limit 5;

1. **List the next five Pixar movies sorted alphabetically?**

Answer: SELECT \* FROM movies order by title asc limit 5 offset 5;

**SQL Review: Simple SELECT Queries:**

****

1. **List all the Canadian cities and their populations?**

Answer: SELECT city, population FROM north\_american\_cities WHERE country = 'Canada'

ORDER BY city;

1. **Order all the cities in the United States by their latitude from north to south?**

Answer: SELECT city, population, latitude FROM north\_american\_cities where country='United States' order by latitude desc;

1. **List all the cities west of Chicago, ordered from west to east?**

Answer: SELECT \* FROM north\_american\_cities WHERE longitude < -87.629798

ORDER BY longitude ASC;

1. **List the two largest cities in Mexico (by population)?**

Answer: SELECT \* FROM north\_american\_cities WHERE country='Mexico'

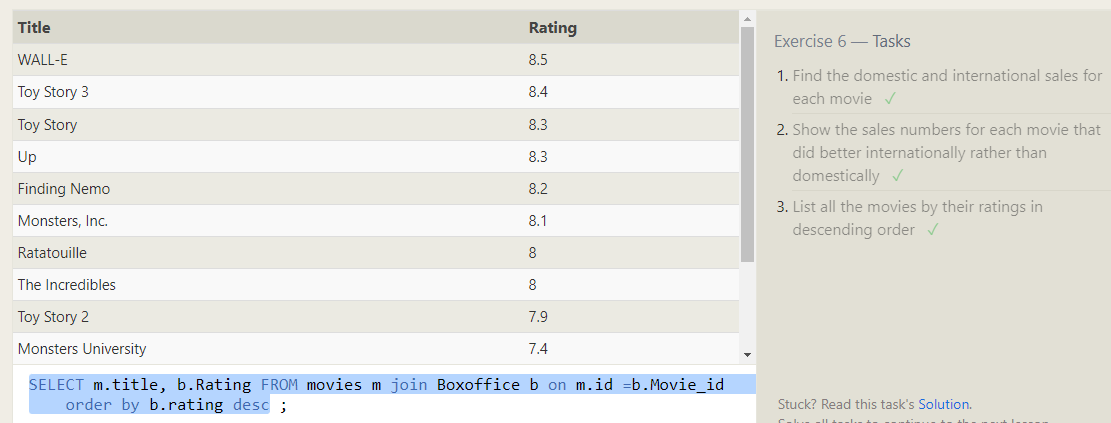
ORDER BY population desc limit 2;

1. **List the third and fourth largest cities (by population) in the United States and their population?**

Answer: SELECT \* FROM north\_american\_cities WHERE country='United States'

ORDER BY population desc limit 2 offset 2;

**SQL Lesson 6: Multi-table queries with JOINs:**

****

1. **Find the domestic and international sales for each movie?**

Answer: SELECT m.title, b.Domestic\_sales,b.International\_sales FROM movies m join Boxoffice b on m.id =b.Movie\_id;

1. **Show the sales numbers for each movie that did better internationally rather than domestically?**

Answer: SELECT m.title, b.Domestic\_sales,b.International\_sales FROM movies m join Boxoffice b on m.id =b.Movie\_id where b.International\_sales >

b.Domestic\_sales ;

1. **List all the movies by their ratings in descending order?**

Answer: SELECT m.title, b.Rating FROM movies m join Boxoffice b on m.id =b.Movie\_id order by b.rating desc?

**SQL Lesson 7: OUTER JOINs:**

****

1. **Find the list of all buildings that have employees?**

Answer: SELECT distinct building from employees;

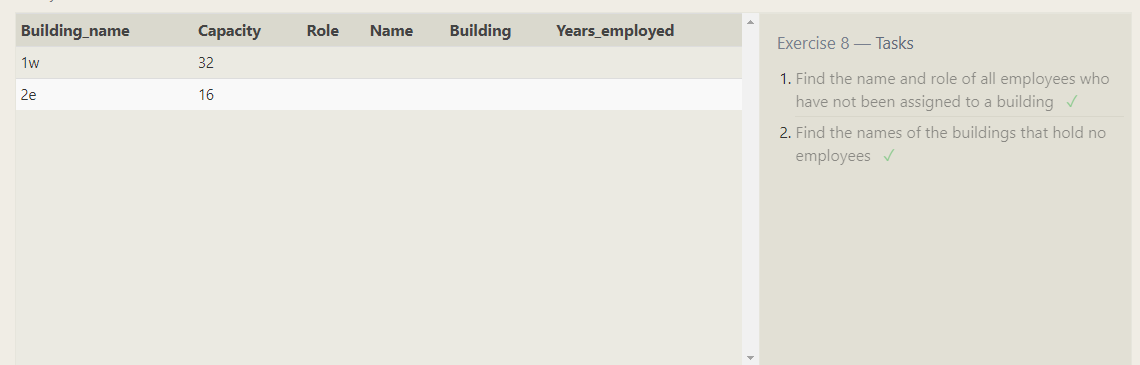
1. **Find the list of all buildings and their capacity?**

Anwer: SELECT Building\_name, capacity from Buildings;

1. **List all buildings and the distinct employee roles in each building (including empty buildings)?**

Answer: SELECT distinct Building\_name,Role from Buildings LEFT JOIN Employees ON Building\_name = Building;

**SQL Lesson 8: A short note on NULLs :**

****

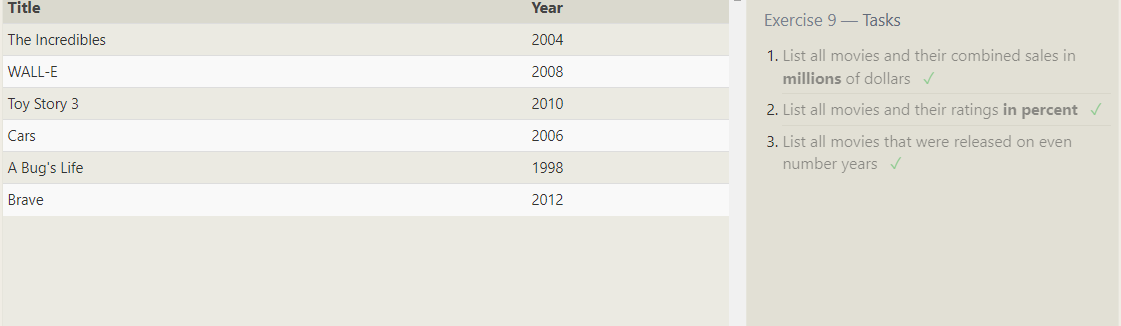
1. **Find the name and role of all employees who have not been assigned to a building?**

Answer: SELECT Role,name,building FROM employees where building is null;

1. **Find the names of the buildings that hold no employees?**

Answer: SELECT\* FROM Buildings left join Employees on Building\_name= Building where Building is null ;

**SQL Lesson 9: Queries with expressions:**

****

1. **List all movies and their combined sales in millions of dollars?**

Answer: SELECT Title,(Domestic\_sales + International\_sales)/1000000 as sale FROM movies join Boxoffice on id =Movie\_id;

1. **List all movies and their ratings in percent?**

Answer: SELECT Title,rating/10\*100 as rating FROM movies join Boxoffice on id =Movie\_id;

1. **List all movies that were released on even number years?**

Answer: SELECT Title, year FROM movies join Boxoffice on id =Movie\_id where year%2=0;

**SQL Lesson 10: Queries with aggregates (Pt. 1):**

****

1. **Find the longest time that an employee has been at the studio?**

Answer: SELECT max(Years\_employed)FROM employees;

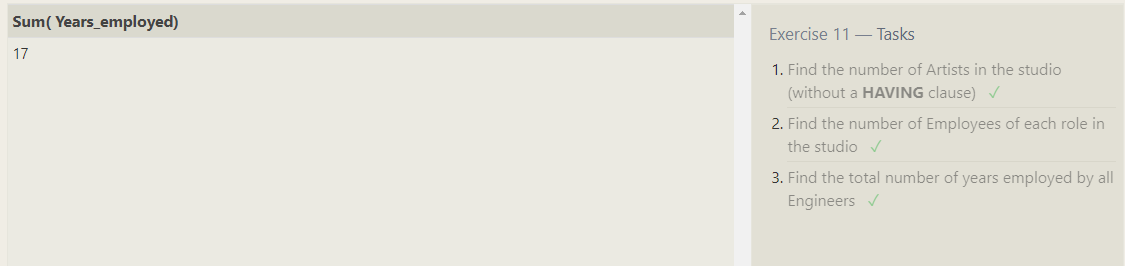
1. **For each role, find the average number of years employed by employees in that role?**

Answer: SELECT Role, avg(Years\_employed) FROM employees group by role;

1. **Find the total number of employee years worked in each building?**

Answer: SELECT building ,sum(years\_employed) FROM employees group by building ;

**SQL Lesson 11: Queries with aggregates (Pt. 2):**

****

1. **Find the number of Artists in the studio (without a HAVING clause)?**

Answer: SELECT count(role) FROM employees where role='Artist';

1. **Find the number of Employees of each role in the studio?**

Answer: SELECT count(role),role FROM employees group by role having role='Artist';

1. **Find the total number of years employed by all Engineers?**

Answer: SELECT sum(Years\_employed) FROM employees where role='Engineer' ;

**SQL Lesson 12: Order of execution of a Query:**

**3rew**

1. **Find the number of movies each director has directed?**

Answer: SELECT director , count(\*) as number\_of\_movie FROM movies group by director;

1. **Find the total domestic and international sales that can be attributed to each director?**

Answer: SELECT m.director , sum(b.Domestic\_sales + b.International\_sales)as total\_cummulative\_sales

FROM movies m join Boxoffice b on m.id=b.Movie\_id group by m.director;

**SQL Lesson 13: Inserting rows:**

****

1. **Add the studio's new production, Toy Story 4 to the list of movies (you can use any director)?**

Answer: INSERT INTO Movies

(Id,Title,Director,Year,Length\_minutes)

VALUES('11','Toy Story 4','John Lasster','2012','150');

1. **Toy Story 4 has been released to critical acclaim! It had a rating of 8.7, and made 340 million domestically and 270 million internationally. Add the record to the BoxOffice table?**

Answer: INSERT INTO Boxoffice

(Movie\_id,Rating,Domestic\_sales,International\_sales)

VALUES('11','8.7','34000000','27000000');

**SQL Lesson 14: Updating rows:**

****

1. **The director for A Bug's Life is incorrect, it was actually directed by John Lasseter?**

Answer: UPDATE Movies

SET Director='John Lasseter'

where id=2;

1. **The year that Toy Story 2 was released is incorrect, it was actually released in 1999?**

Answer: UPDATE Movies

SET Year='1999'

where id=3;

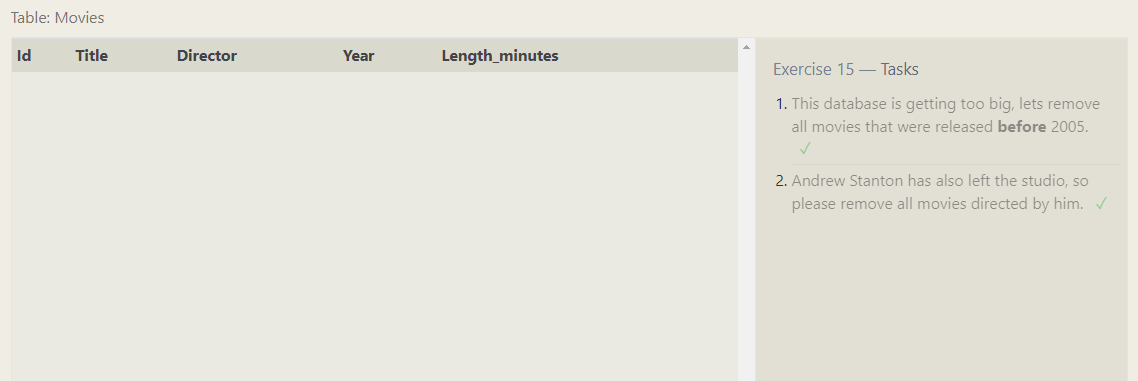
1. **Both the title and director for Toy Story 8 is incorrect! The title should be "Toy Story 3" and it was directed by Lee Unkrich?**

Answer: UPDATE Movies

SET Title='Toy Story 3', Director='Lee Unkrich'

where id=11;

**SQL Lesson 15: Deleting rows:**

****

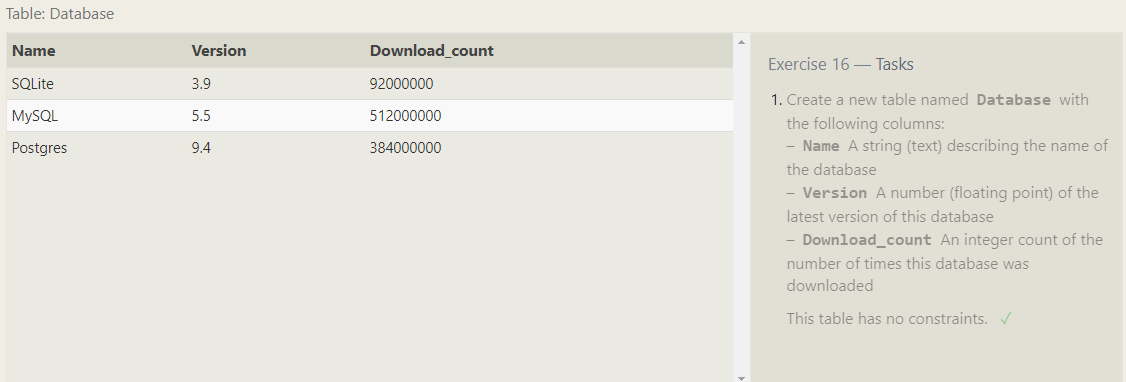
1. **This database is getting too big, lets remove all movies that were released before 2005?**

Answer: DELETE FROM movies where year < 2005;

1. **Andrew Stanton has also left the studio, so please remove all movies directed by him?**

Answer: DELETE FROM movies where director = "Andrew Stanton";

**SQL Lesson 16: Creating tables:**

****

**1..** **Create a new table named Database with the following columns:**

**–Name A string (text) describing the name of the database**

**– Version A number (floating point) of the latest version of this database**

**– Download\_count An integer count of the number of times this database was downloaded**

**This table has no constraints.**

Answer: CREATE TABLE Database(

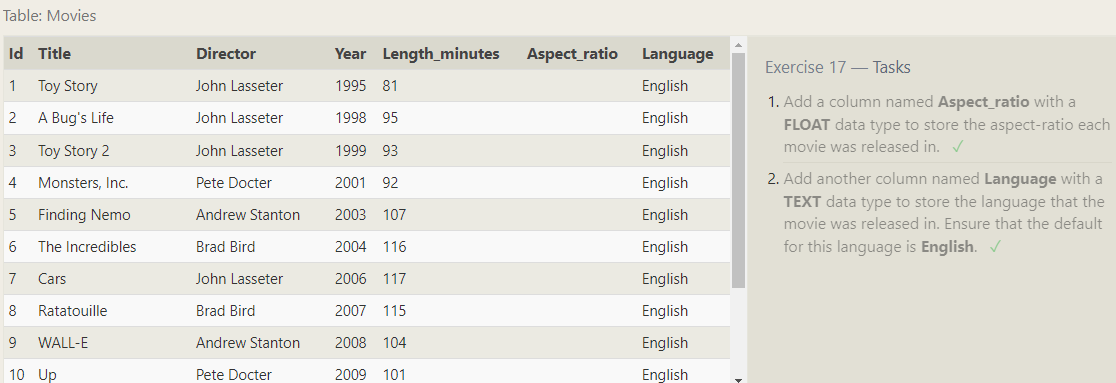
Name string,

Version float,

Download\_count interger

)

**SQL Lesson 17: Altering tables:**

****

1. **Add a column named Aspect\_ratio with a FLOAT data type to store the aspect-ratio each movie was released in?**

Answer: ALTER TABLE Movies

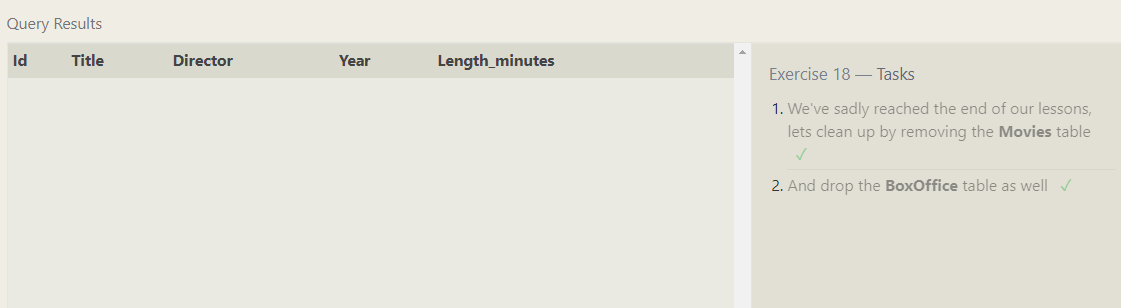
ADD COLUMN Aspect\_ratio FLOAT;

1. **Add another column named Language with a TEXT data type to store the language that the movie was released in. Ensure that the default for this language is English?**

Answer: ALTER TABLE Movies

ADD COLUMN Language TEXT Default English;

**SQL Lesson 18: Dropping tables:**

****

1. **We've sadly reached the end of our lessons, lets clean up by removing the Movies table?**

Answer: DROP TABLE Movies;

1. **And drop the BoxOffice table as well?**

Answer: DROP TABLE BoxOffice;