EmailID:Vishnuravichandran.28@gmail.com

SQL BOLT All Exercise

Exercise-1

Problem 1: Find the title of each film

SELECT title FROM Movies;

Problem 2: Find the director of each film

SELECT director FROM Movies;

Problem 3: Find the title and director of each film

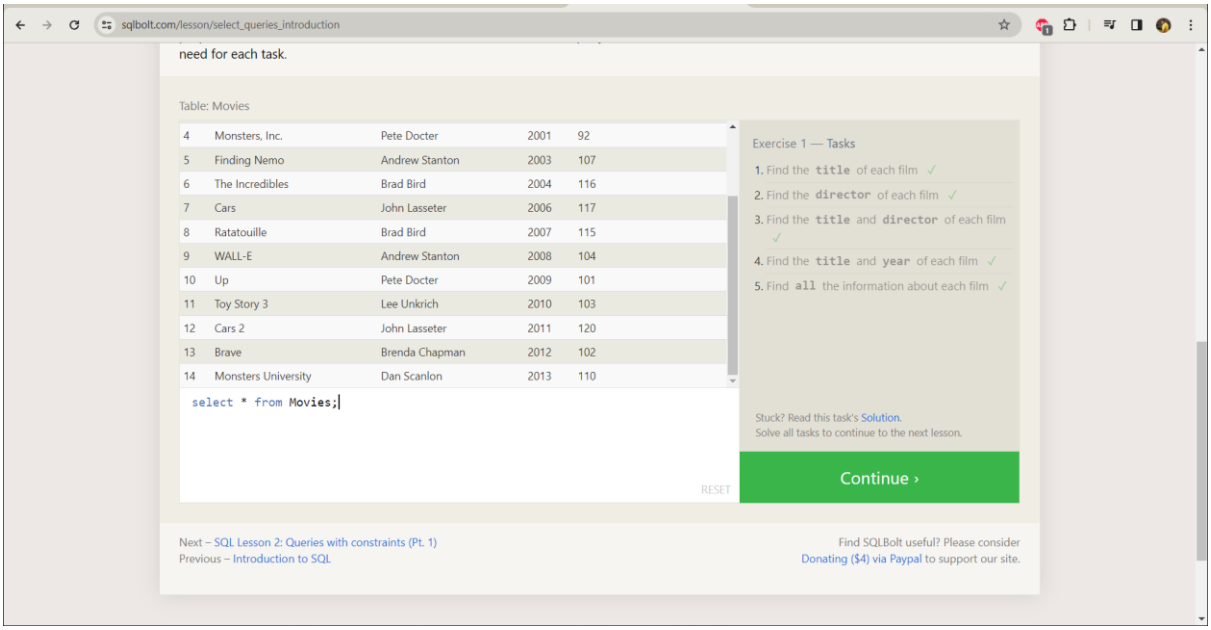
SELECT title, director FROM Movies;

Problem 4: Find the title and year of each film

SELECT title, year FROM Movies;

Problem 5: Find all the information about each film

SELECT * FROM Movies;



EmailID:Vishnuravichandran.28@gmail.com

Exercise-2

Problem 1: Find the movie with a row id of 6

SELECT title FROM Movies WHERE id = 6;

Problem 2: Find the movies released in the years between 2000 and 2010

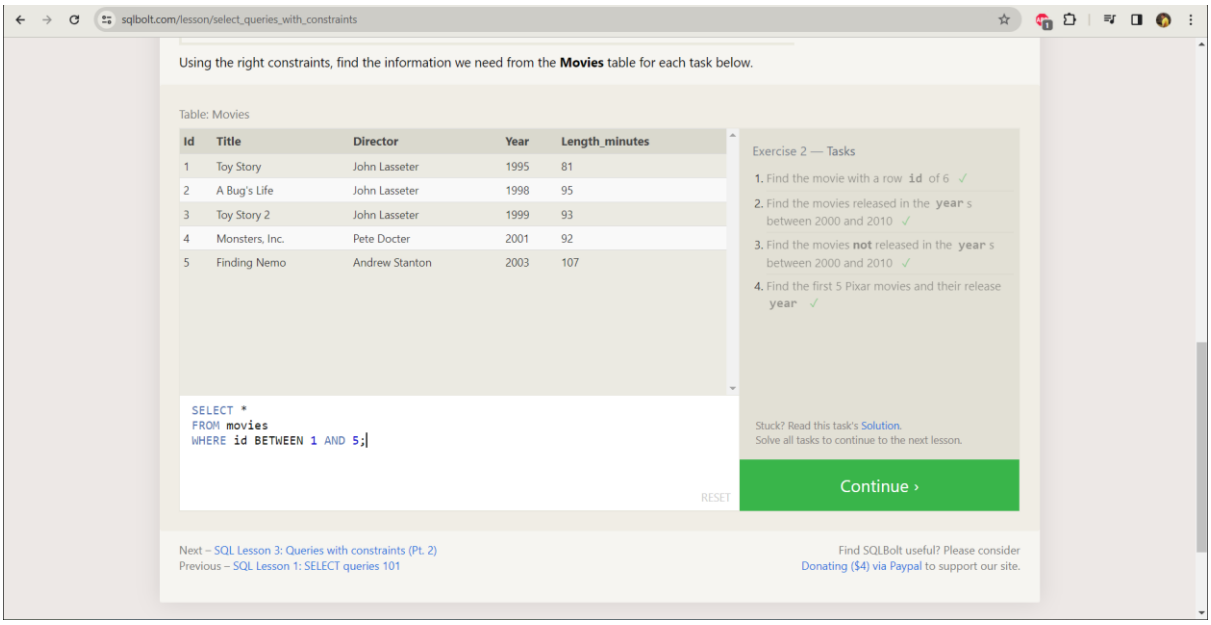
SELECT title FROM Movies WHERE year BETWEEN 2000 AND 2010;

Problem 3: Find the movies not released in the years between 2000 and 2010

SELECT title FROM Movies WHERE year NOT BETWEEN 2000 AND 2010;

Problem 4: Find the first 5 Pixar movies and their release year

SELECT * FROM movies WHERE id BETWEEN 1 AND 5;



EmailID:Vishnuravichandran.28@gmail.com

Exercise-3

Problem 1: Find all the Toy Story movies

SELECT title

FROM Movies

WHERE title LIKE 'Toy Story%';

Problem 2: Find all the movies directed by John Lasseter

SELECT title

FROM Movies

WHERE director = 'John Lasseter';

Problem 3: Find all the movies (and directors) not directed by John Lasseter

SELECT title, director

FROM Movies

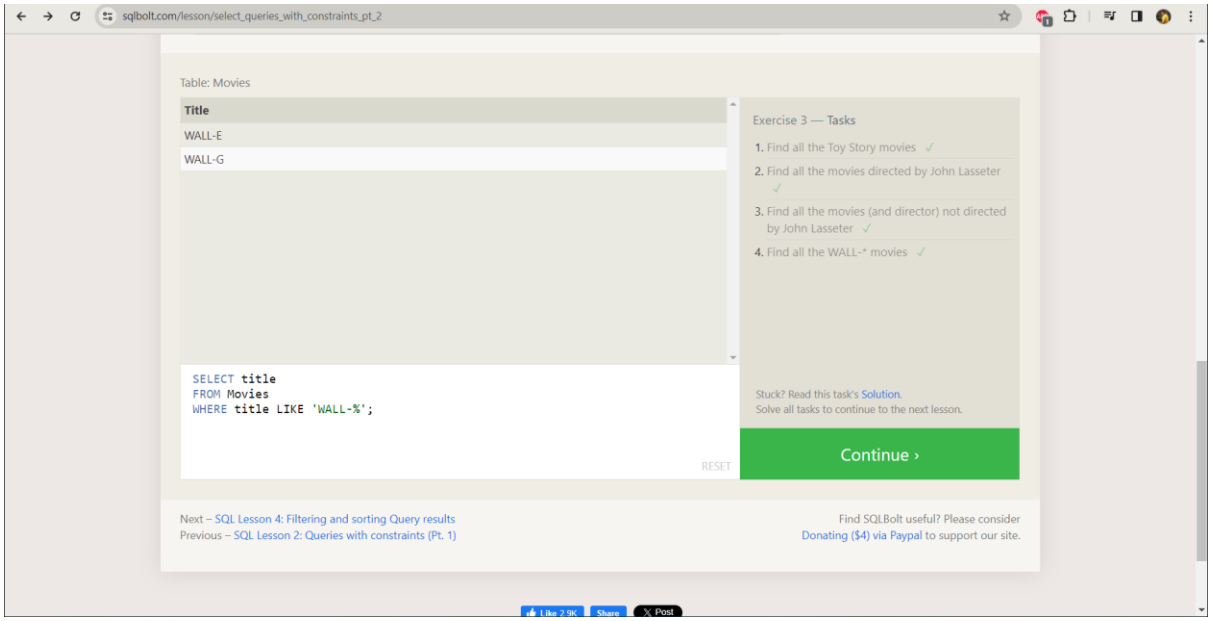
WHERE director != 'John Lasseter';

Problem 4: Find all the WALL-* movies

SELECT title

FROM Movies

WHERE title LIKE 'WALL-%';



EmailID:Vishnuravichandran.28@gmail.com

Exercise-4

Problem 1: List all directors of Pixar movies (alphabetically), without duplicates

SELECT DISTINCT director FROM movies WHERE director IS NOT NULL ORDER BY director ASC;

Problem 2: List the last four Pixar movies released (ordered from most recent to least)

SELECT title FROM movies WHERE director = 'Pixar' ORDER BY year DESC LIMIT 4;

List the first five Pixar movies sorted alphabetically:

SELECT title FROM movies WHERE director = 'Pixar' ORDER BY title ASC LIMIT 5;

List the next five Pixar movies sorted alphabetically:

SELECT title FROM movies WHERE director = 'Pixar' ORDER BY title ASC LIMIT 5 OFFSET 5;

EmailID:Vishnuravichandran.28@gmail.com

Exercise-5

Problem 1: List all the Canadian cities and their populations

SELECT city, Population FROM north_american_cities WHERE country = 'Canada';

Problem 2: Order all the cities in the United States by their latitude from north to south

SELECT city FROM North_american_cities WHERE Country = 'United States' ORDER BY Latitude DESC;

Problem 3: List all the cities west of Chicago, ordered from west to east

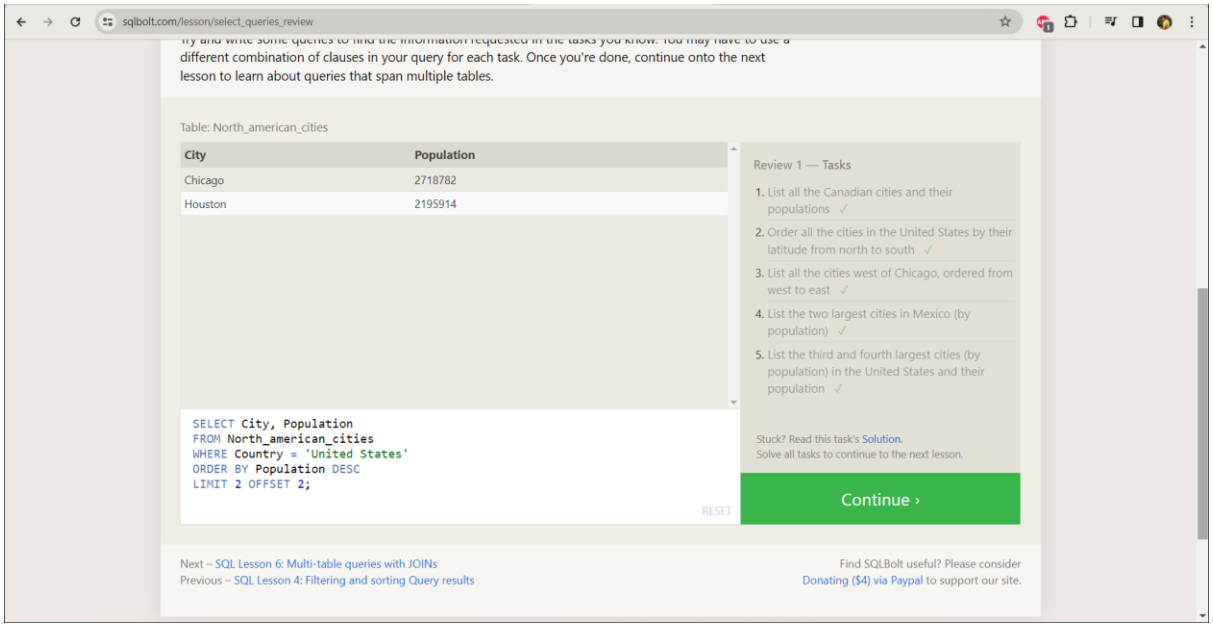
SELECT city FROM North_american_cities WHERE Longitude < (SELECT Longitude FROM North_american_cities WHERE city = 'Chicago') ORDER BY Longitude ASC;

Problem 4: List the two largest cities in Mexico (by population)

SELECT City FROM North_american_cities WHERE Country = 'Mexico' ORDER BY Population DESC LIMIT 2;

Problem 5: List the third and fourth largest cities (by population) in the United States and their population

SELECT City, Population FROM North_american_cities WHERE Country = 'United States ORDER BY Population DESC LIMIT 2 OFFSET 2;



EmailID:Vishnuravichandran.28@gmail.com

Exercise-6

Problem 1: Find the domestic and international sales for each movie

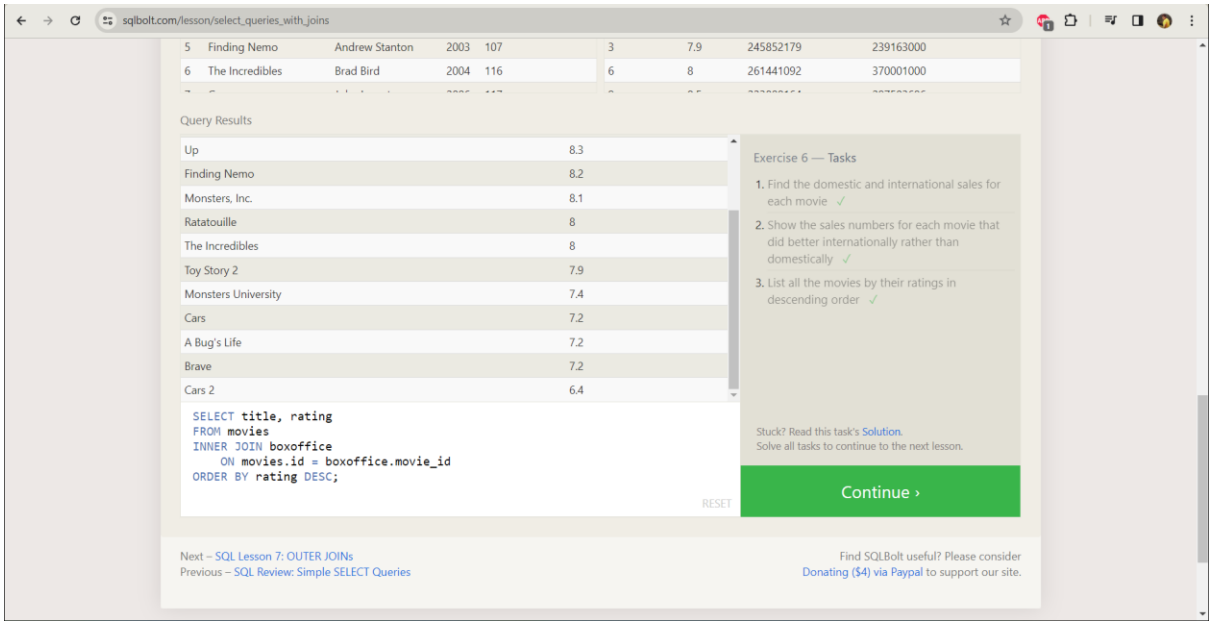
SELECT title, domestic_sales, international_sales FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id;

Problem 2: Show the sales numbers for each movie that did better internationally rather than domestically

SELECT * FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id WHERE international_sales > domestic_sales;

Problem 3: List all the movies by their ratings in descending order

SELECT title, rating FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id ORDER BY rating DESC;



EmailID:Vishnuravichandran.28@gmail.com

Exercise-7

Problem 1: Find the list of all buildings that have employees

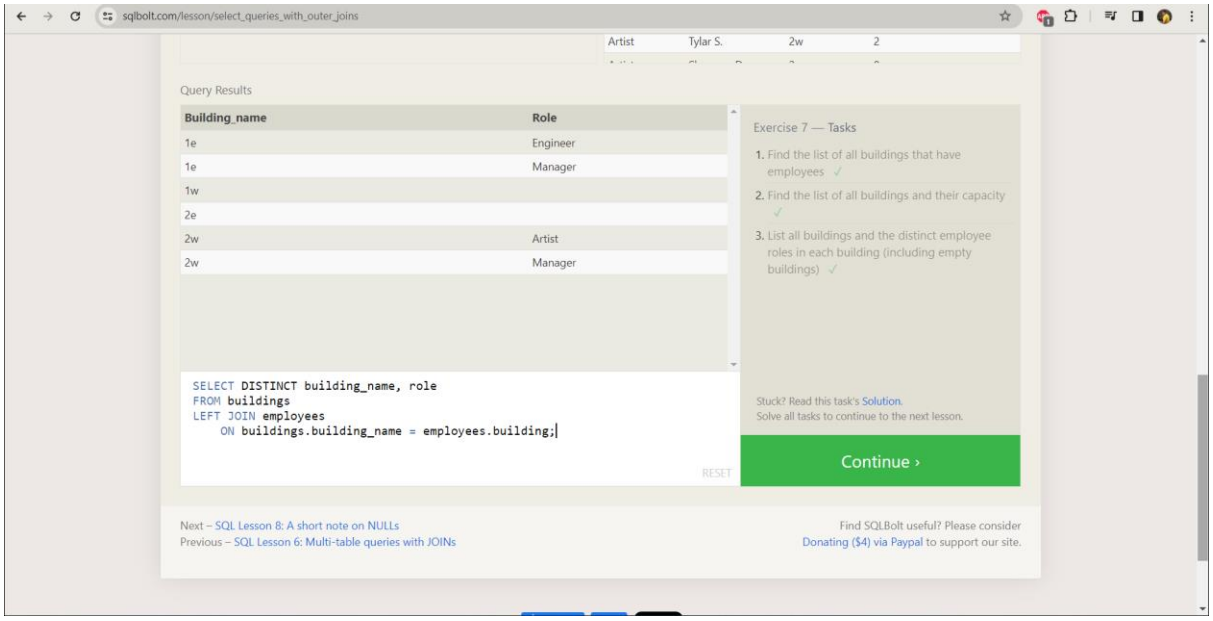
SELECT DISTINCT building_name FROM buildings LEFT JOIN employees ON buildings.building_name = employees.building WHERE building IS NOT NULL;

Problem 2: Find the list of all buildings and their capacity

SELECT * FROM buildings;

Problem 3: List all buildings and the distinct employee roles in each building (including empty buildings)

SELECT DISTINCT building_name, role FROM buildings LEFT JOIN employees ON buildings.building_name = employees.building;



EmailID:Vishnuravichandran.28@gmail.com

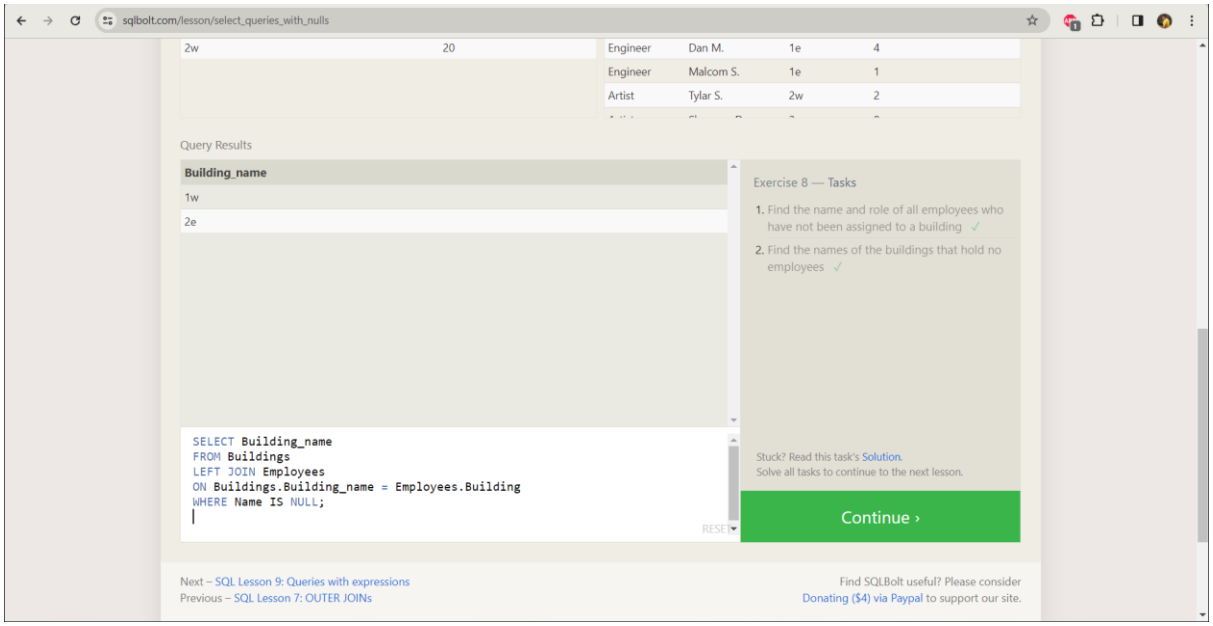
Exercise-8

Find the Name and Role of Employees Not Assigned to a Building:

SELECT Name, Role FROM Employees WHERE Building IS NULL;

Find the Names of Buildings That Hold No Employees:

SELECT Building_name FROM Buildings LEFT JOIN Employees ON Buildings.Building_name = Employees.Building WHERE Name IS NULL;



EmailID:Vishnuravichandran.28@gmail.com

Exercise-9

List all movies and their combined sales in millions of dollars:

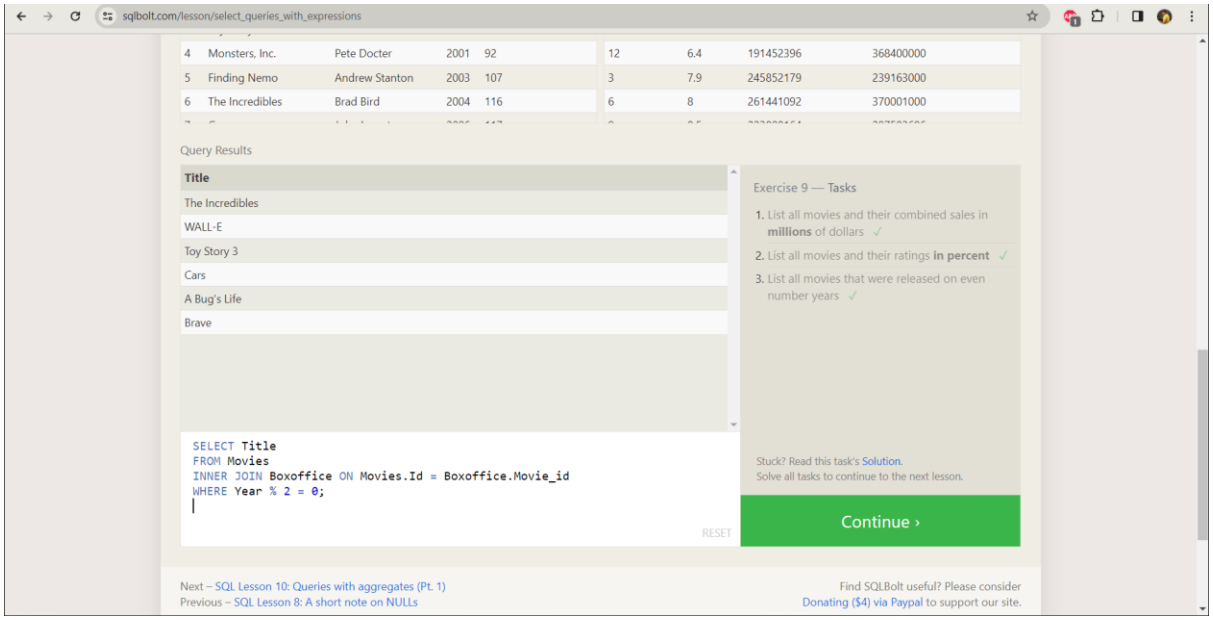
SELECT Title, (Domestic_sales + International_sales) / 1000000 AS "Combined Sales" FROM Movies INNER JOIN Boxoffice ON Movies.Id = Boxoffice.Movie_id;

List all movies and their ratings in percent:

SELECT Title, ROUND((Rating / 10), 2) * 100 AS Ratings FROM Movies INNER JOIN Boxoffice ON Movies.Id = Boxoffice.Movie_id ORDER BY Ratings DESC;

List all movies that were released on even-numbered years:

SELECT Title FROM Movies INNER JOIN Boxoffice ON Movies.Id = Boxoffice.Movie_id WHERE Year % 2 = 0;



EmailID:Vishnuravichandran.28@gmail.com

Exercise-10

Find the longest time that an employee has been at the studio:

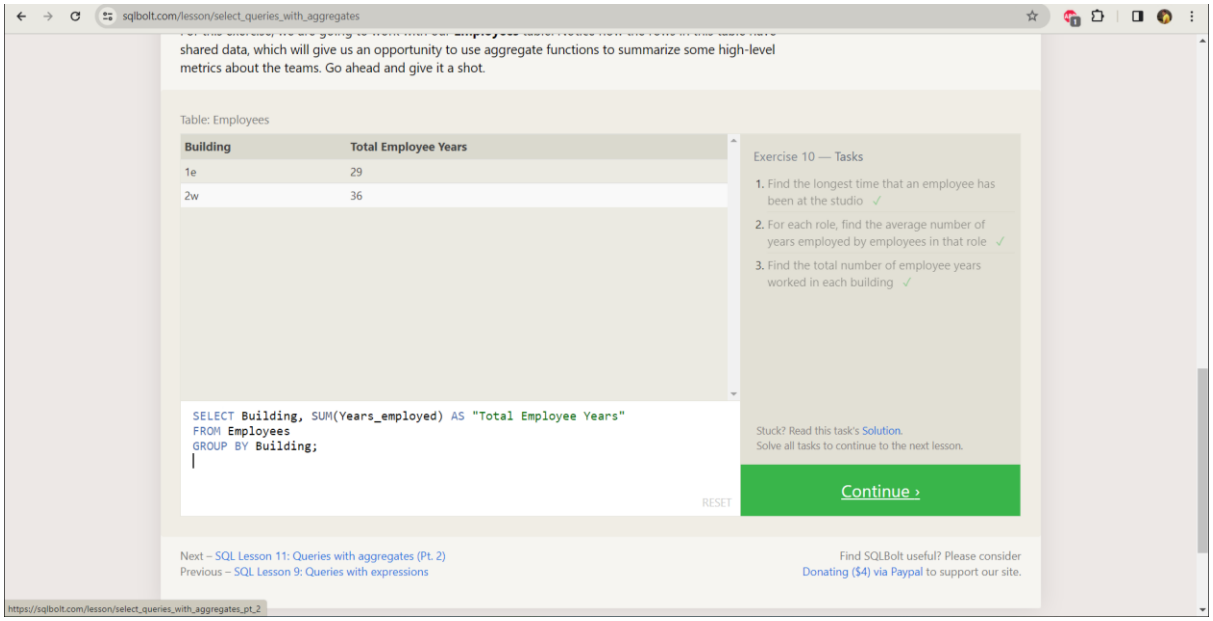
SELECT Name, SUM(Years_employed) AS "Longest Time" FROM Employees GROUP BY Name ORDER BY "Longest Time" DESC LIMIT 1;

For each role, find the average number of years employed by employees in that role:

SELECT Role, AVG(Years_employed) AS "Average Years Employed" FROM Employees GROUP BY Role;

Find the total number of employee years worked in each building:

SELECT Building, SUM(Years_employed) AS "Total Employee Years" FROM Employees GROUP BY Building;



EmailID:Vishnuravichandran.28@gmail.com

Exercise-11

Find the number of Artists in the studio (without a HAVING clause):

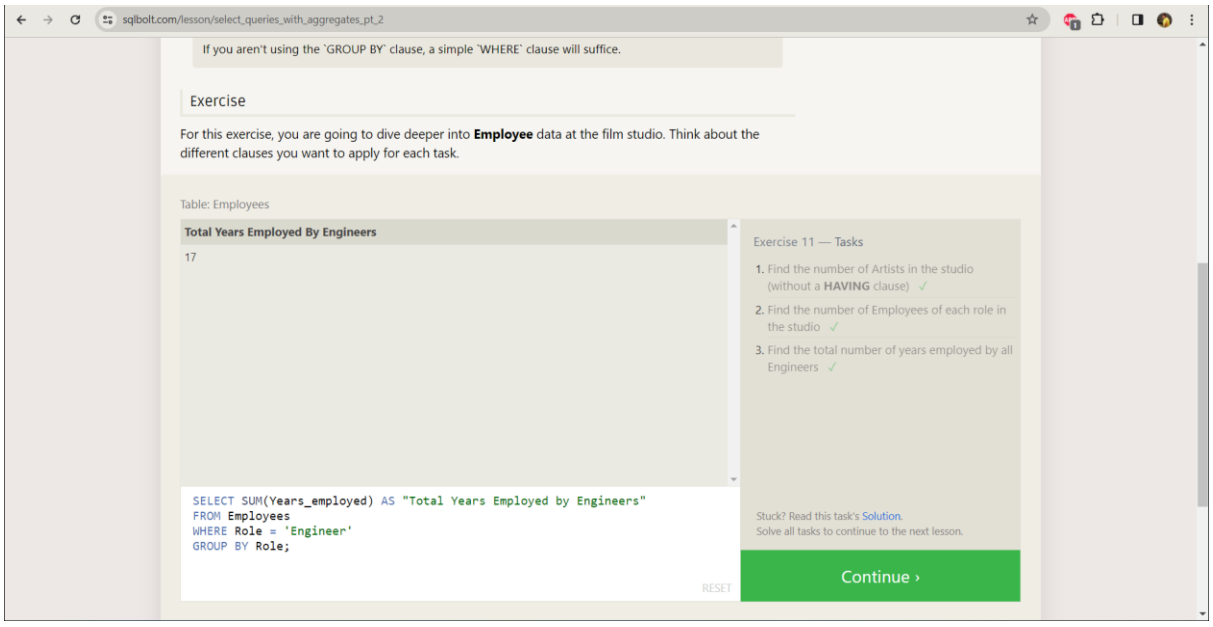
SELECT COUNT(Role) AS "Number of Artists" FROM Employees WHERE Role = 'Artist';

Find the number of Employees of each role in the studio:

SELECT Role, COUNT(Role) AS "Number of Employees" FROM Employees GROUP BY Role;

Find the total number of years employed by all Engineers:

SELECT SUM(Years_employed) AS "Total Years Employed by Engineers" FROM Employees WHERE Role = 'Engineer' GROUP BY Role;



EmailID:Vishnuravichandran.28@gmail.com

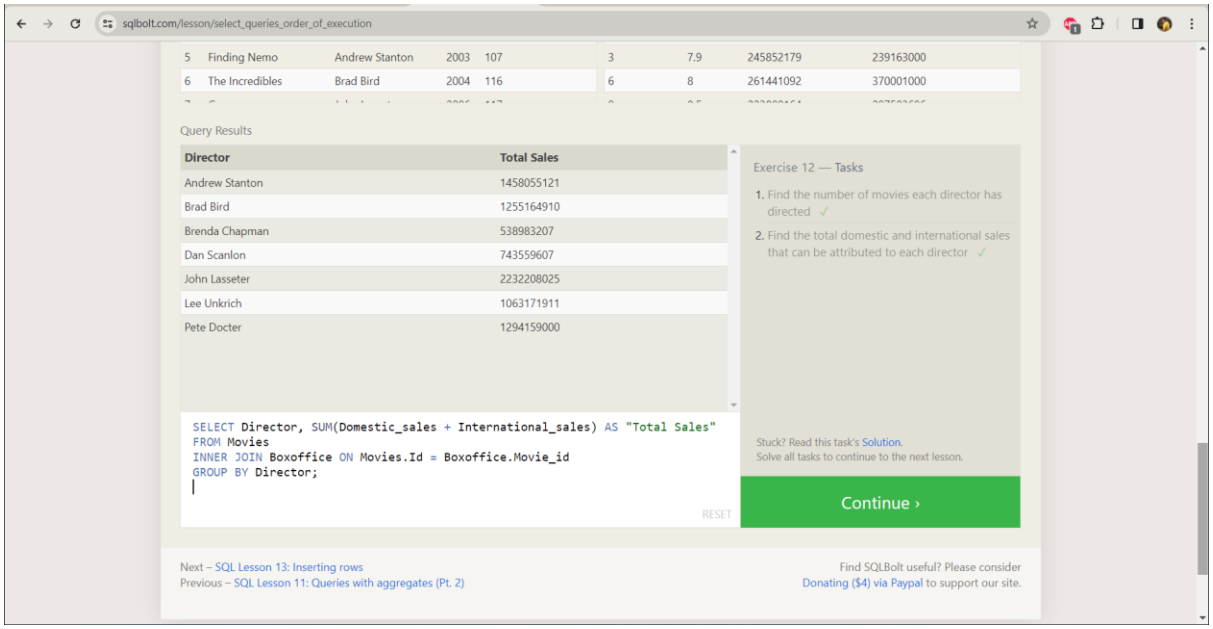
Exercise-12

Find the number of movies each director has directed:

SELECT Director, COUNT(Title) AS "Number of Movies" FROM Movies GROUP BY Director;

Find the total domestic and international sales that can be attributed to each director:

SELECT Director, SUM(Domestic_sales + International_sales) AS "Total Sales" FROM Movies INNER JOIN Boxoffice ON Movies.Id = Boxoffice.Movie_id GROUP BY Director;



EmailID:Vishnuravichandran.28@gmail.com

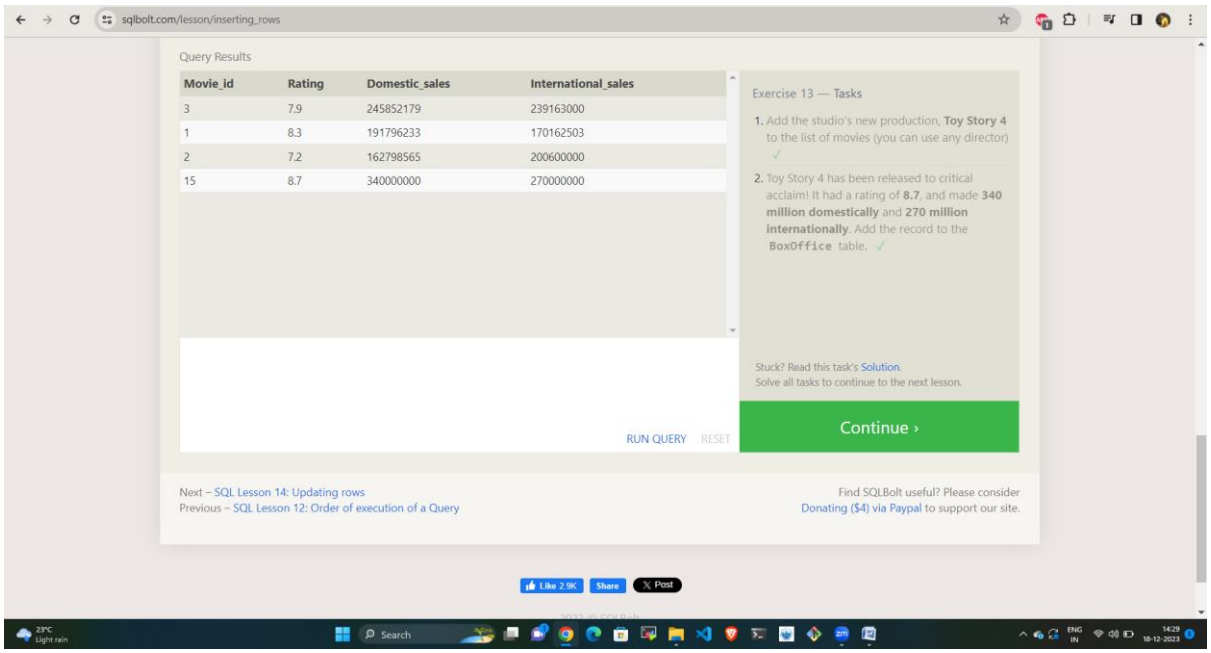
Exercise-13

Add the studio's new production, Toy Story 4 to the list of movies:

INSERT INTO Movies (Title, Director) VALUES ('Toy Story 4', 'Vishnu');

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:

INSERT INTO Boxoffice (Movie_id, Rating, Domestic_sales, International_sales) VALUES (15, 8.7, 340000000, 270000000);



EmailID:Vishnuravichandran.28@gmail.com

Exercise-14

Update the director for "A Bug's Life" to be John Lasseter:

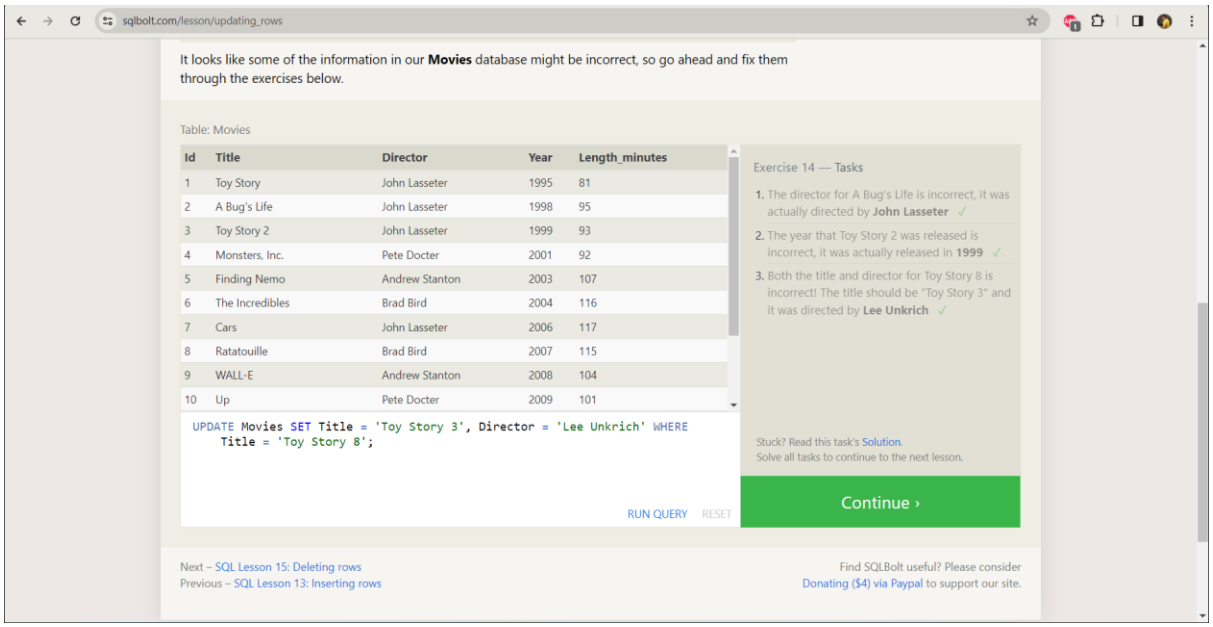
UPDATE Movies SET Director = 'John Lasseter' WHERE Title = 'A Bug's Life';

Update the release year for "Toy Story 2" to be 1999:

UPDATE Movies SET Year = 1999 WHERE Title = 'Toy Story 2';

Update the title and director for "Toy Story 8" to be "Toy Story 3" directed by Lee Unkrich:

UPDATE Movies SET Title = 'Toy Story 3', Director = 'Lee Unkrich' WHERE Title = 'Toy Story 8';



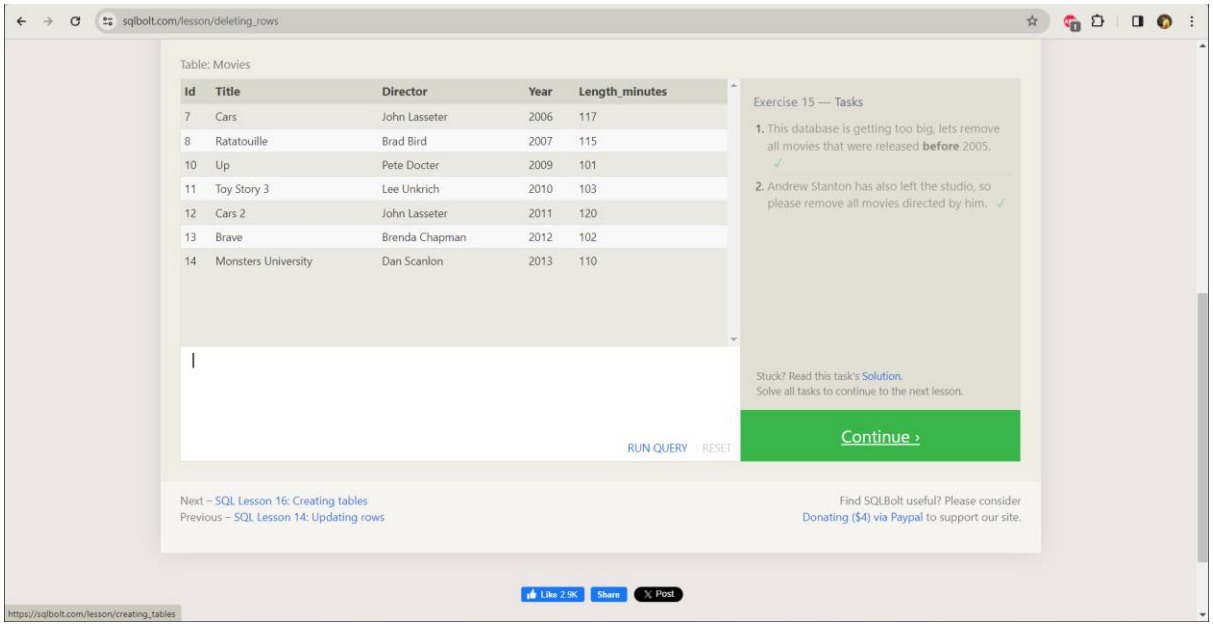
Exercise-15

Remove all movies that were released before 2005:

DELETE FROM Movies WHERE Year < 2005;

Remove all movies directed by Andrew Stanton:

DELETE FROM Movies WHERE Director = 'Andrew Stanton';

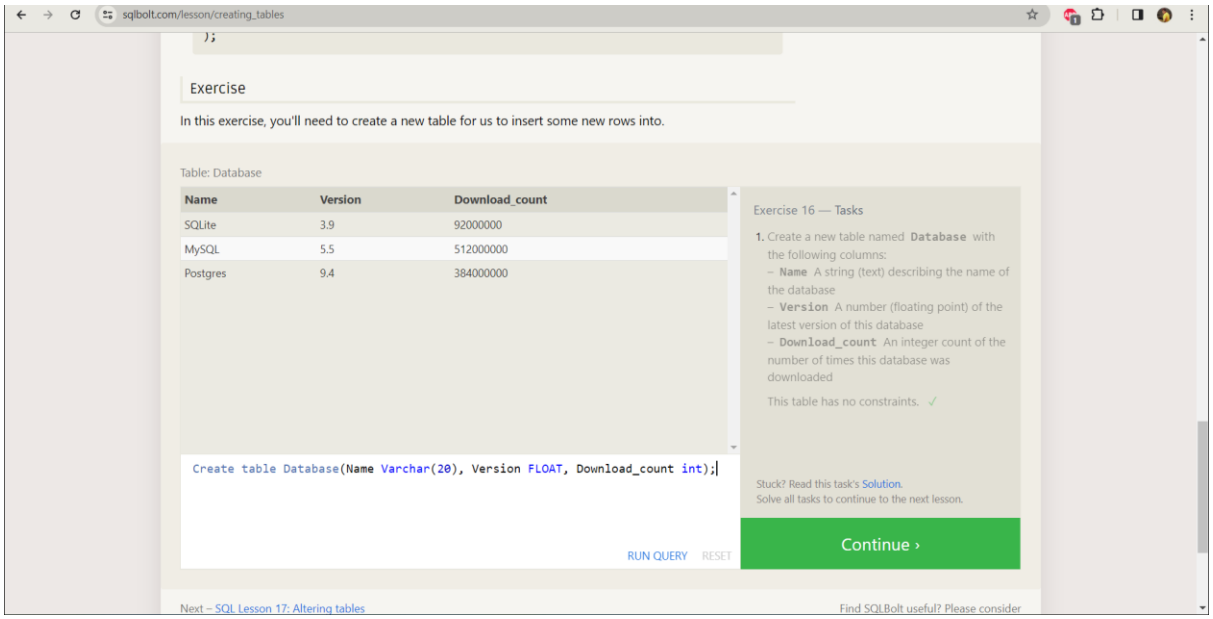


EmailID:Vishnuravichandran.28@gmail.com

Exercise-16

- **Problem 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.

Create table Database(Name Varchar(20), Version FLOAT, Download_count int);



EmailID:Vishnuravichandran.28@gmail.com

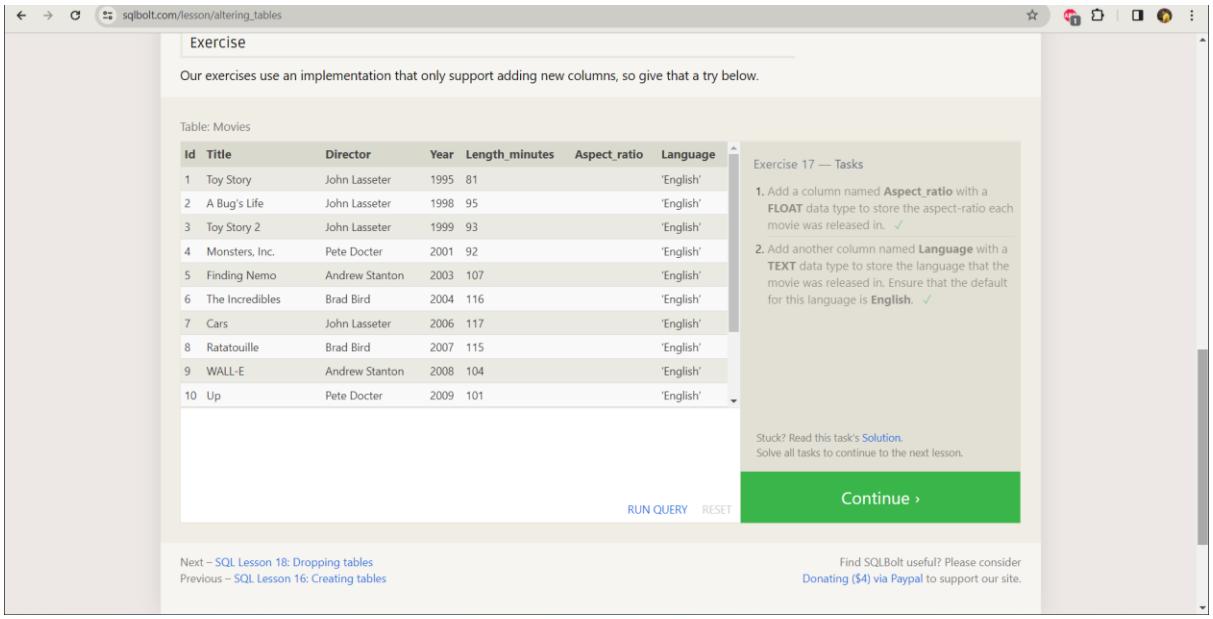
Exercise-17

Problem 1: Add a column named Aspect_ratio with a FLOAT data type to store the aspect-ratio each movie was released in.

ALTER TABLE Movies ADD Aspect_ratio FLOAT;

Problem 2: 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.

ALTER TABLE Movies ADD Language VARCHAR(20) DEFAULT 'English';



EmailID:Vishnuravichandran.28@gmail.com

Exercise-18

Drop the Movies table:

DROP TABLE Movies;

Drop the BoxOffice table:

DROP TABLE BoxOffice;