TABLE : MOVIES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Title** | **Director** | **Year** | **Length\_minutes** |
| 1 | Toy Story | John Lasseter | 1995 | 81 |
| 2 | A Bug's Life | John Lasseter | 1998 | 95 |
| 3 | Toy Story 2 | John Lasseter | 1999 | 93 |
| 4 | Monsters, Inc. | Pete Docter | 2001 | 92 |
| 5 | Finding Nemo | Andrew Stanton | 2003 | 107 |
| 6 | The Incredibles | Brad Bird | 2004 | 116 |
| 7 | Cars | John Lasseter | 2006 | 117 |
| 8 | Ratatouille | Brad Bird | 2007 | 115 |
| 9 | WALL-E | Andrew Stanton | 2008 | 104 |
| 10 | Up | Pete Docter | 2009 | 101 |
| 11 | Toy Story 3 | Lee Unkrich | 2010 | 103 |
| 12 | Cars 2 | John Lasseter | 2011 | 120 |
| 13 | Brave | Brenda Chapman | 2012 | 102 |
| 14 | Monsters University | Dan Scanlon | 2013 | 110 |

**Exercise - 1 Tasks**

1. Find the **“title”** of each film

SELECT title FROM movies;

1. Find the **“director”** of each film

SELECT director FROM movies;

1. Find the **“title”** and **“director”** of each film

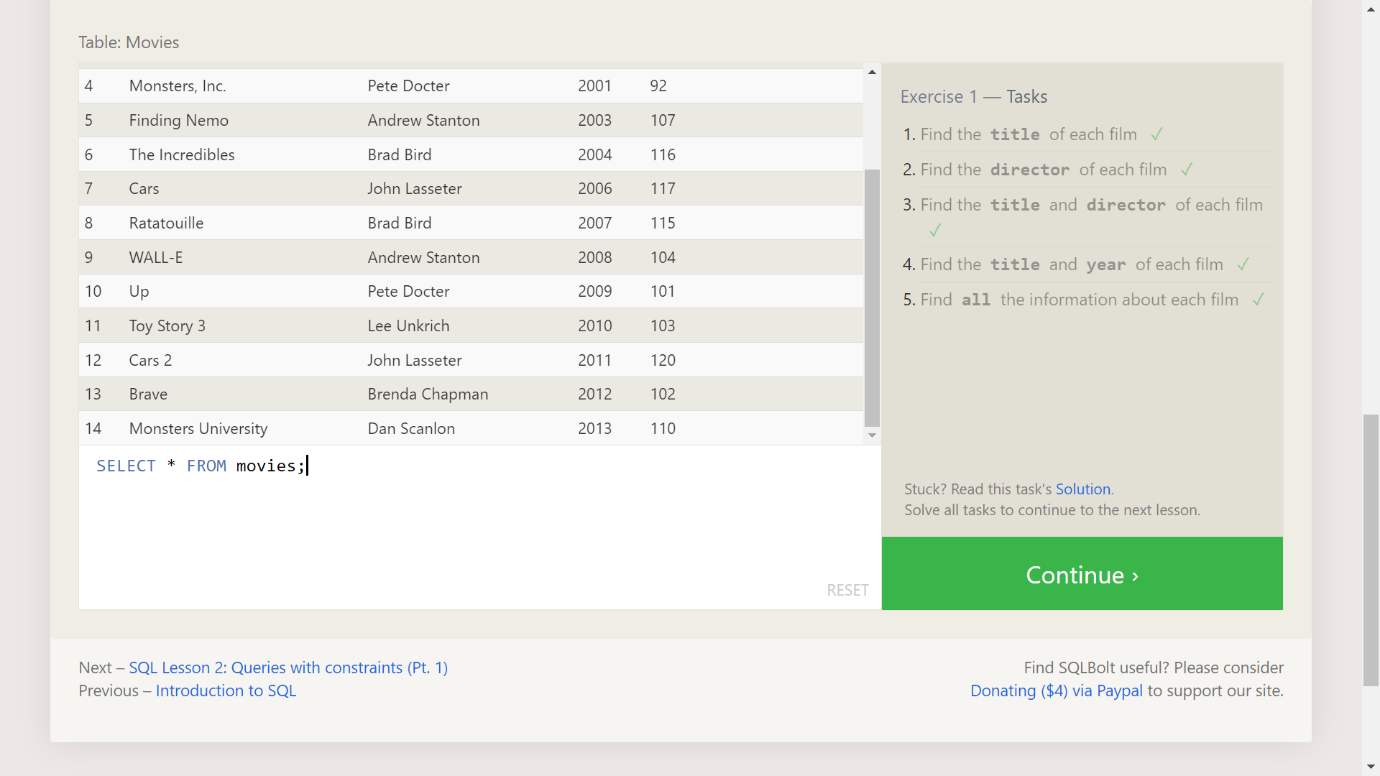
SELECT title, director FROM movies;

1. Find the **“title”** and **“year”** of each film

SELECT title, year FROM movies;

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

SELECT \* FROM movies;



**Exercise - 2 Tasks**

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

SELECT \* FROM movies WHERE id = 6;

1. Find the movies realeased in the **year** between 2000 and 2010

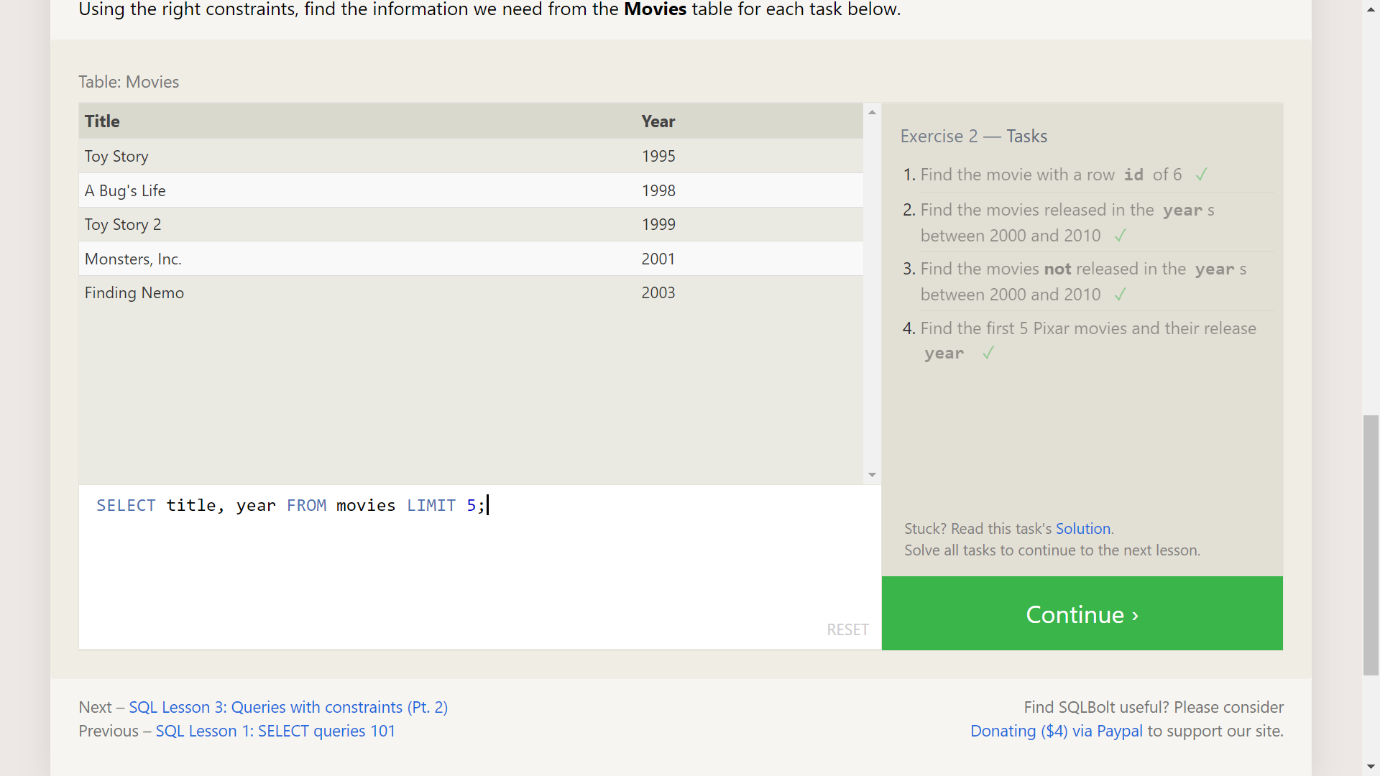
SELECT \* FROM movies WHERE year BETWEEN 2000 AND 2010;

1. Find the movies **not** realeased in the **year** between 2000 and 2010

SELECT \* FROM movies WHERE year NOT BETWEEN 2000 AND 2010;

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

SELECT title, year FROM movies LIMIT 5;



**Exercise - 3 Tasks**

1. Find all the Toy Story movies

SELECT \* FROM movies WHERE title LIKE '%Toy%';

1. Find all the movies directed by Jhon Lasseter

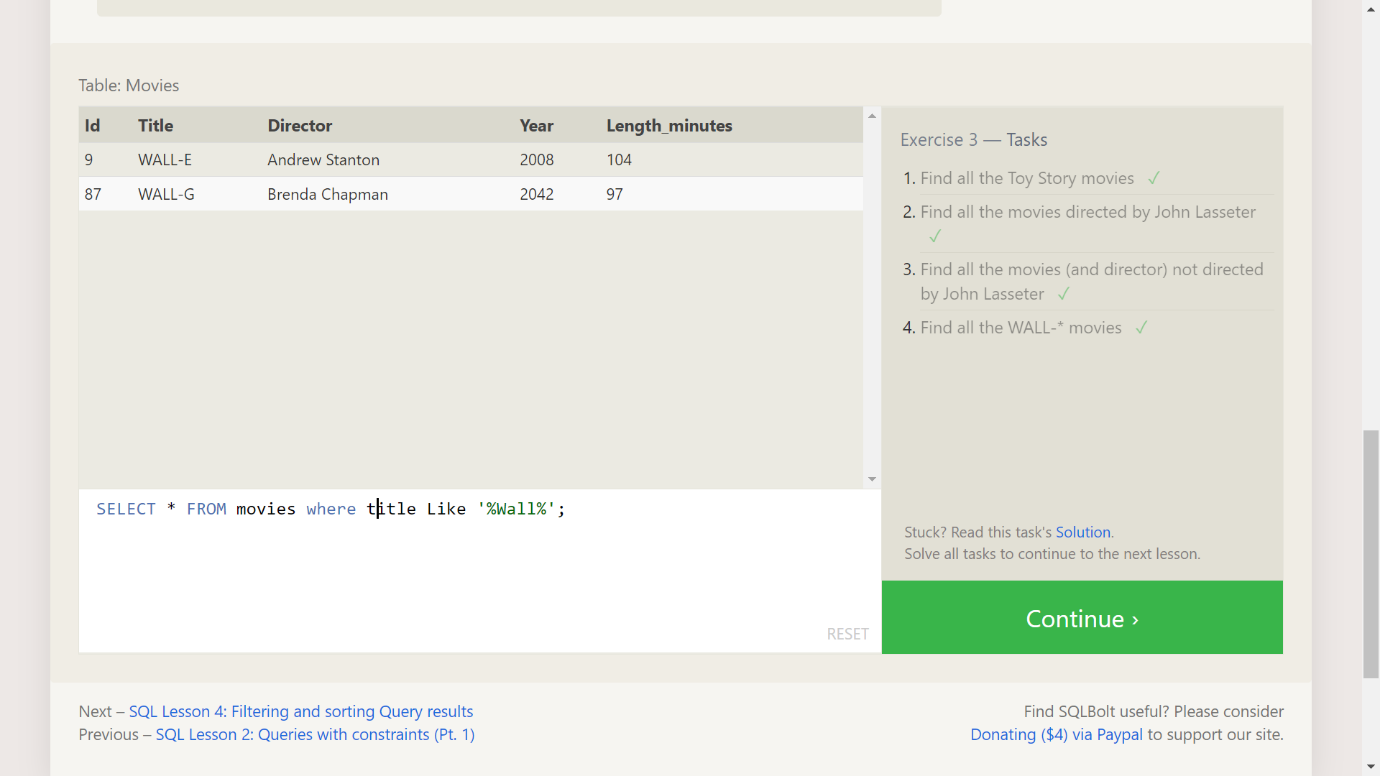
SELECT \* FROM movies WHERE director LIKE '%John Lasseter%';

1. Find all the movies (and director) not directed by Jhon Lasseter

SELECT \* FROM movies WHERE director NOT LIKE '%John Lasseter%';

1. Find all the WALL-\* movies

SELECT \* FROM movies where title Like '%Wall%';



**Exercise - 4 Tasks**

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

SELECT DISTINCT director FROM movies ORDER BY director;

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

SELECT title,year FROM movies ORDER BY year DESC LIMIT 4;

1. List the first five Pixar movies sorted alphabetically

SELECT title FROM movies ORDER BY title LIMIT 5;

1. List the next five Pixar movies sorted alphabetically

SELECT title FROM movies ORDER BY title LIMIT 5 OFFSET 5;



Table: North\_american\_cities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **City** | **Country** | **Population** | **Latitude** | **Longitude** |
| Guadalajara | Mexico | 1500800 | 20.659699 | -103.349609 |
| Toronto | Canada | 2795060 | 43.653226 | -79.383184 |
| Houston | United States | 2195914 | 29.760427 | -95.369803 |
| New York | United States | 8405837 | 40.712784 | -74.005941 |
| Philadelphia | United States | 1553165 | 39.952584 | -75.165222 |
| Havana | Cuba | 2106146 | 23.05407 | -82.345189 |
| Mexico City | Mexico | 8555500 | 19.432608 | -99.133208 |
| Phoenix | United States | 1513367 | 33.448377 | -112.074037 |
| Los Angeles | United States | 3884307 | 34.052234 | -118.243685 |
| Ecatepec de Morelos | Mexico | 1742000 | 19.601841 | -99.050674 |
| Montreal | Canada | 1717767 | 45.501689 | -73.567256 |
| Chicago | United States | 2718782 | 41.878114 | -87.629798 |

**Review - 1 Tasks**

1. List all the Canadian cities and their populations

SELECT city, population FROM North\_american\_cities WHERE country = "Canada";

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

SELECT city, 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

SELECT city,longitude FROM North\_american\_cities WHERE longitude < -87.629798 ORDER BY longitude;

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

SELECT city,population FROM North\_american\_cities WHERE country LIKE "Mexico" ORDER BY Population DESC LIMIT 2;

1. 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 LIKE "United States" ORDER BY population DESC LIMIT 2 OFFSET 2;

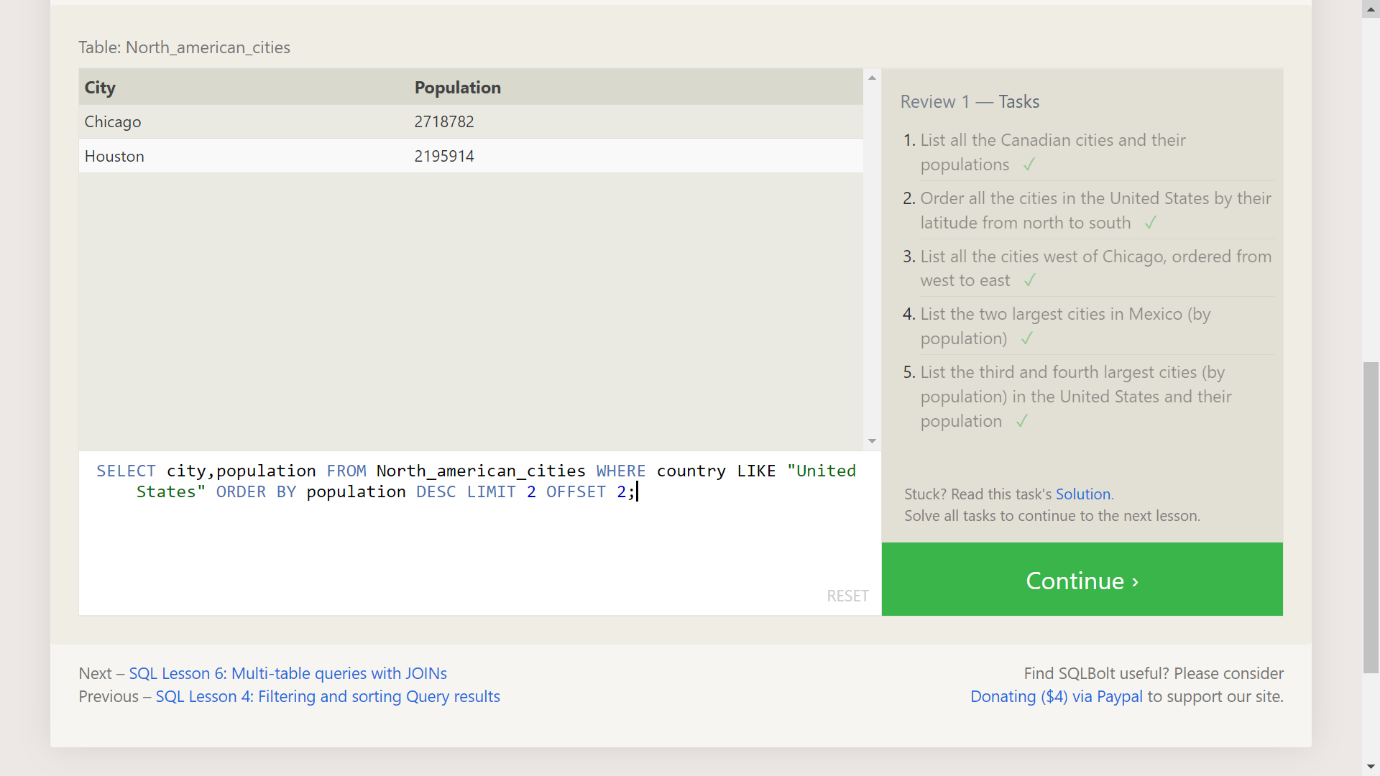


Table: Movies (Read-Only)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Title** | **Director** | **Year** | **Length\_minutes** |
| 1 | Toy Story | John Lasseter | 1995 | 81 |
| 2 | A Bug's Life | John Lasseter | 1998 | 95 |
| 3 | Toy Story 2 | John Lasseter | 1999 | 93 |
| 4 | Monsters, Inc. | Pete Docter | 2001 | 92 |
| 5 | Finding Nemo | Andrew Stanton | 2003 | 107 |
| 6 | The Incredibles | Brad Bird | 2004 | 116 |
| 7 | Cars | John Lasseter | 2006 | 117 |
| 8 | Ratatouille | Brad Bird | 2007 | 115 |
| 9 | WALL-E | Andrew Stanton | 2008 | 104 |
| 10 | Up | Pete Docter | 2009 | 101 |
| 11 | Toy Story 3 | Lee Unkrich | 2010 | 103 |
| 12 | Cars 2 | John Lasseter | 2011 | 120 |
| 13 | Brave | Brenda Chapman | 2012 | 102 |
| 14 | Monsters University | Dan Scanlon | 2013 | 110 |

Table: Boxoffice (Read-Only)

|  |  |  |  |
| --- | --- | --- | --- |
| **Movie\_id** | **Rating** | **Domestic\_sales** | **International\_sales** |
| 5 | 8.2 | 380843261 | 555900000 |
| 14 | 7.4 | 268492764 | 475066843 |
| 8 | 8 | 206445654 | 417277164 |
| 12 | 6.4 | 191452396 | 368400000 |
| 3 | 7.9 | 245852179 | 239163000 |
| 6 | 8 | 261441092 | 370001000 |
| 9 | 8.5 | 223808164 | 297503696 |
| 11 | 8.4 | 415004880 | 648167031 |
| 1 | 8.3 | 191796233 | 170162503 |
| 7 | 7.2 | 244082982 | 217900167 |
| 10 | 8.3 | 293004164 | 438338580 |
| 4 | 8.1 | 289916256 | 272900000 |
| 2 | 7.2 | 162798565 | 200600000 |
| 13 | 7.2 | 237283207 | 301700000 |

**Excercise - 6 Tasks**

1. Find the domestic and international sales for each movie

SELECT Title, Domestic\_sales, International\_sales FROM Movies JOIN Boxoffice ON Movies.id = Boxoffice.movie\_id;

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

SELECT Title, Domestic\_sales, International\_sales FROM Movies JOIN Boxoffice ON Movies.id = Boxoffice.movie\_id WHERE International\_sales > Domestic\_sales;

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

SELECT Title, Rating FROM Movies JOIN Boxoffice ON Movies.id = Boxoffice.Movie\_id ORDER BY Rating DESC;

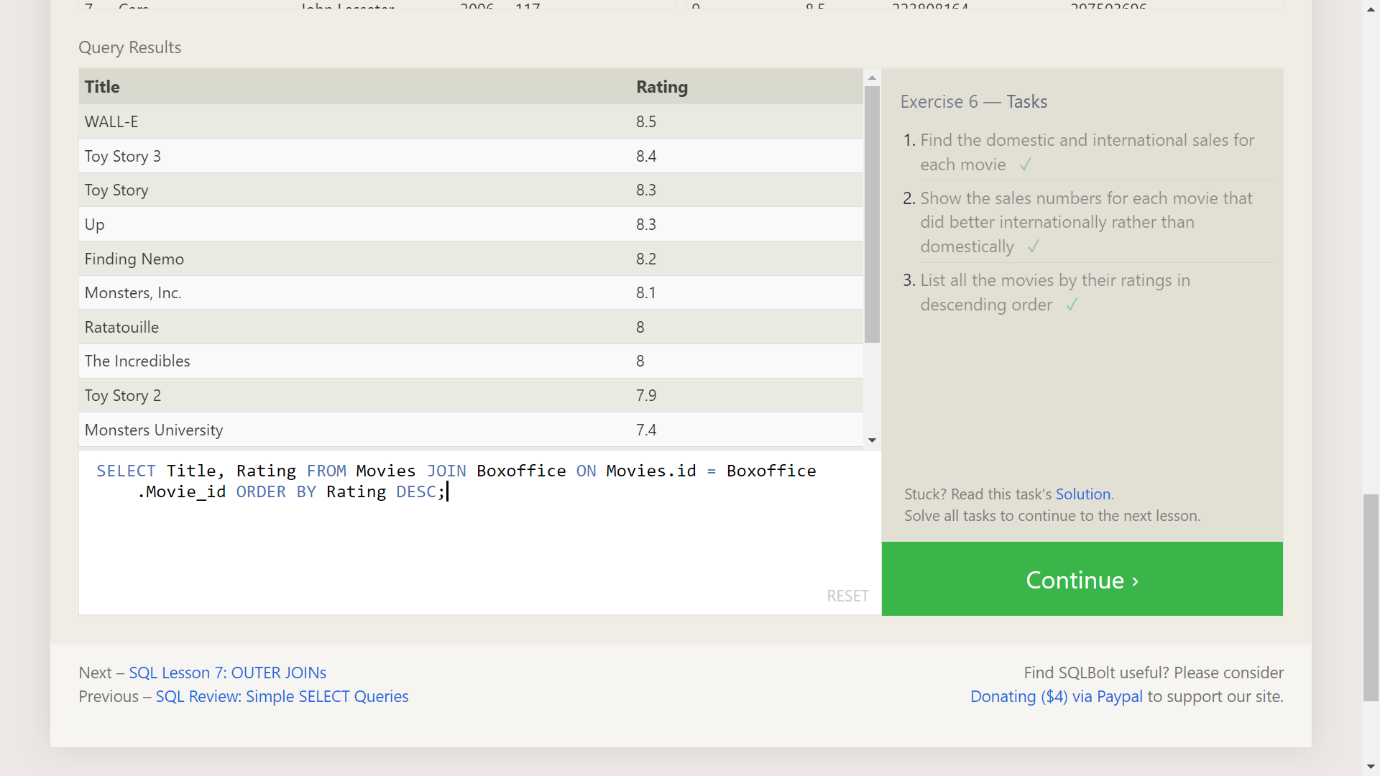


Table: Buildings (Read-Only)

|  |  |
| --- | --- |
| **Building\_name** | **Capacity** |
| 1e | 24 |
| 1w | 32 |
| 2e | 16 |
| 2w | 20 |

Table: Employees (Read-Only)

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Building** | **Years\_employed** |
| Engineer | Becky A. | 1e | 4 |
| Engineer | Dan B. | 1e | 2 |
| Engineer | Sharon F. | 1e | 6 |
| Engineer | Dan M. | 1e | 4 |
| Engineer | Malcom S. | 1e | 1 |
| Artist | Tylar S. | 2w | 2 |
| Artist | Sherman D. | 2w | 8 |
| Artist | Jakob J. | 2w | 6 |
| Artist | Lillia A. | 2w | 7 |
| Artist | Brandon J. | 2w | 7 |
| Manager | Scott K. | 1e | 9 |
| Manager | Shirlee M. | 1e | 3 |
| Manager | Daria O. | 2w | 6 |

**Excercise - 7 Tasks**

1. Find the list of all buildings that have employees

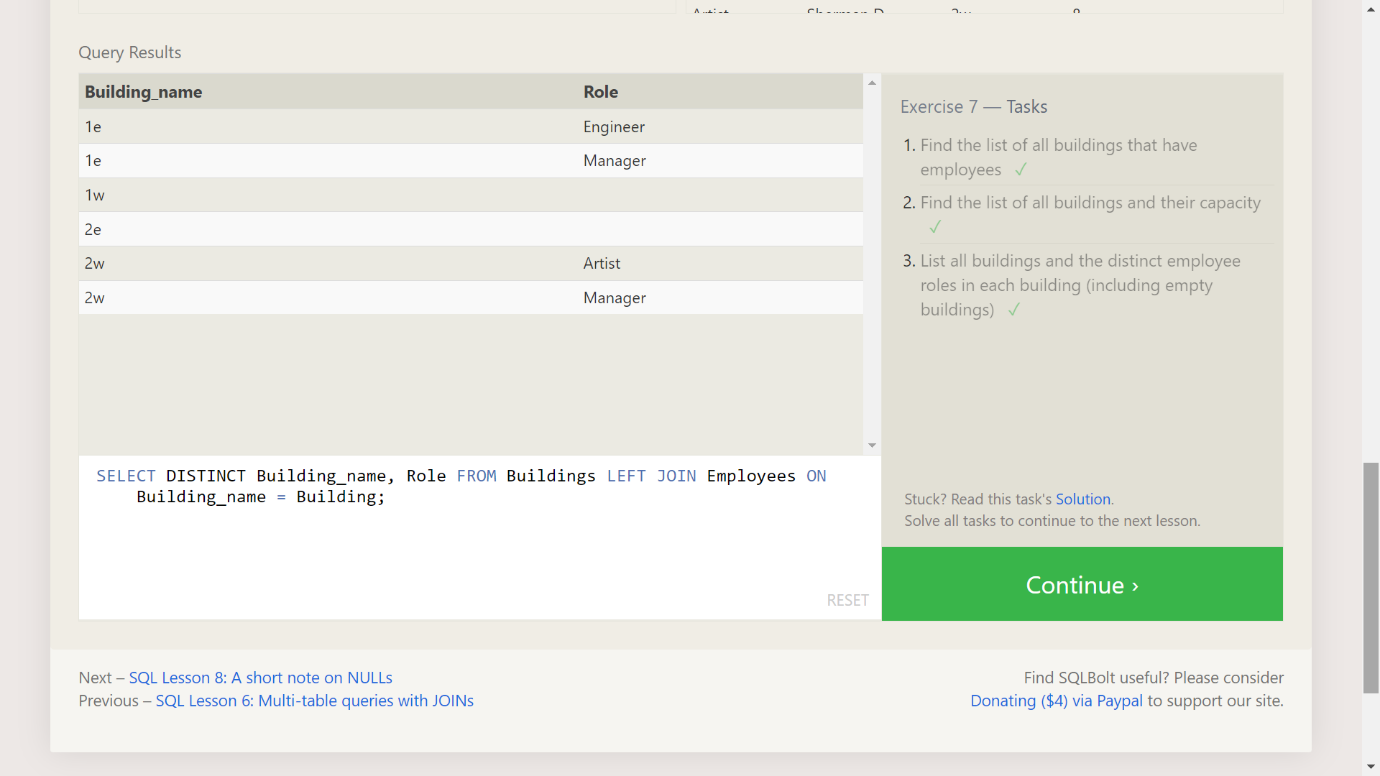
SELECT DISTINCT Building FROM Employees;

1. Find the list of all buildings and their capacity

SELECT \* FROM Buildings;

1. 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 Building\_name = Building;



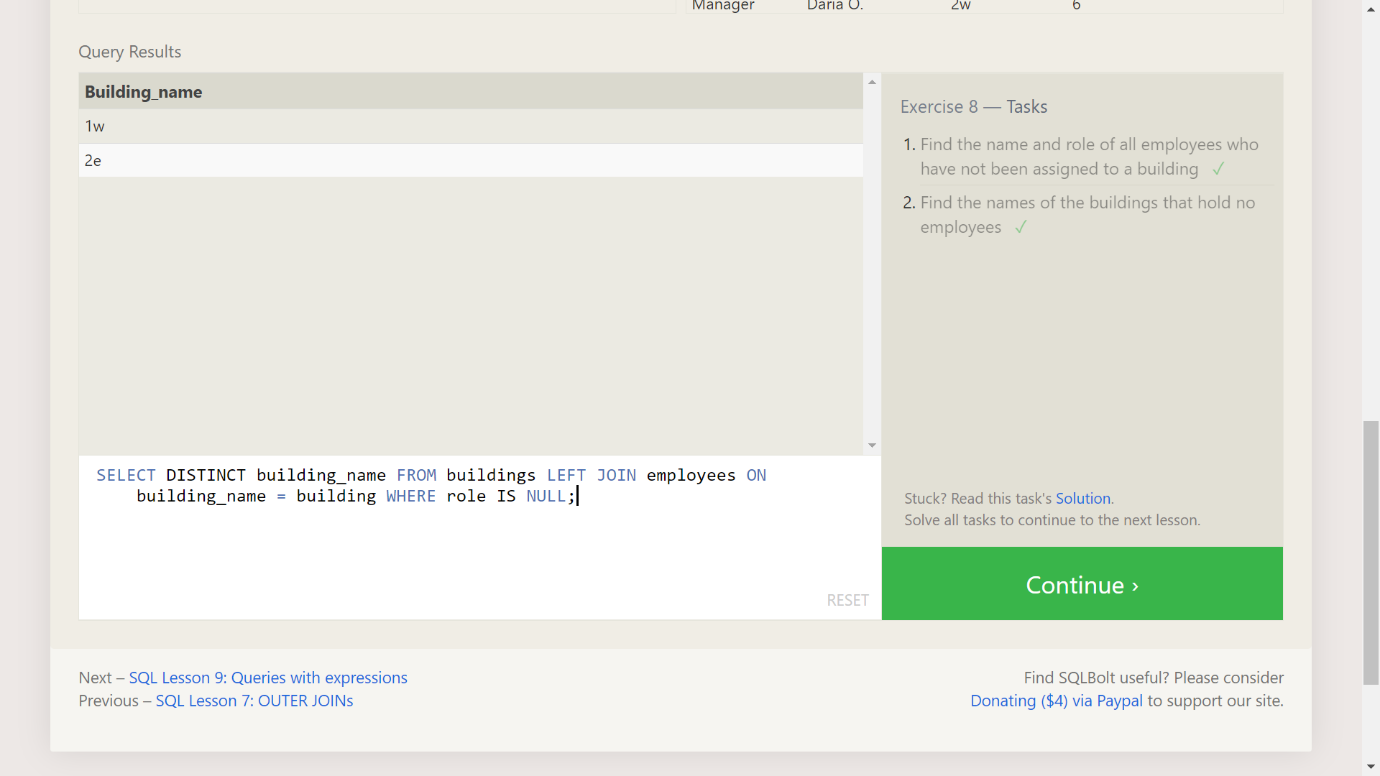
**Excercise - 8 Tasks**

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

SELECT Name, Role FROM Employees WHERE Building IS NULL;

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

SELECT DISTINCT building\_name FROM buildings LEFT JOIN employees ON building\_name = building WHERE role IS NULL;



**Excercise - 9 Tasks**

1. Find the list of all buildings that have employees

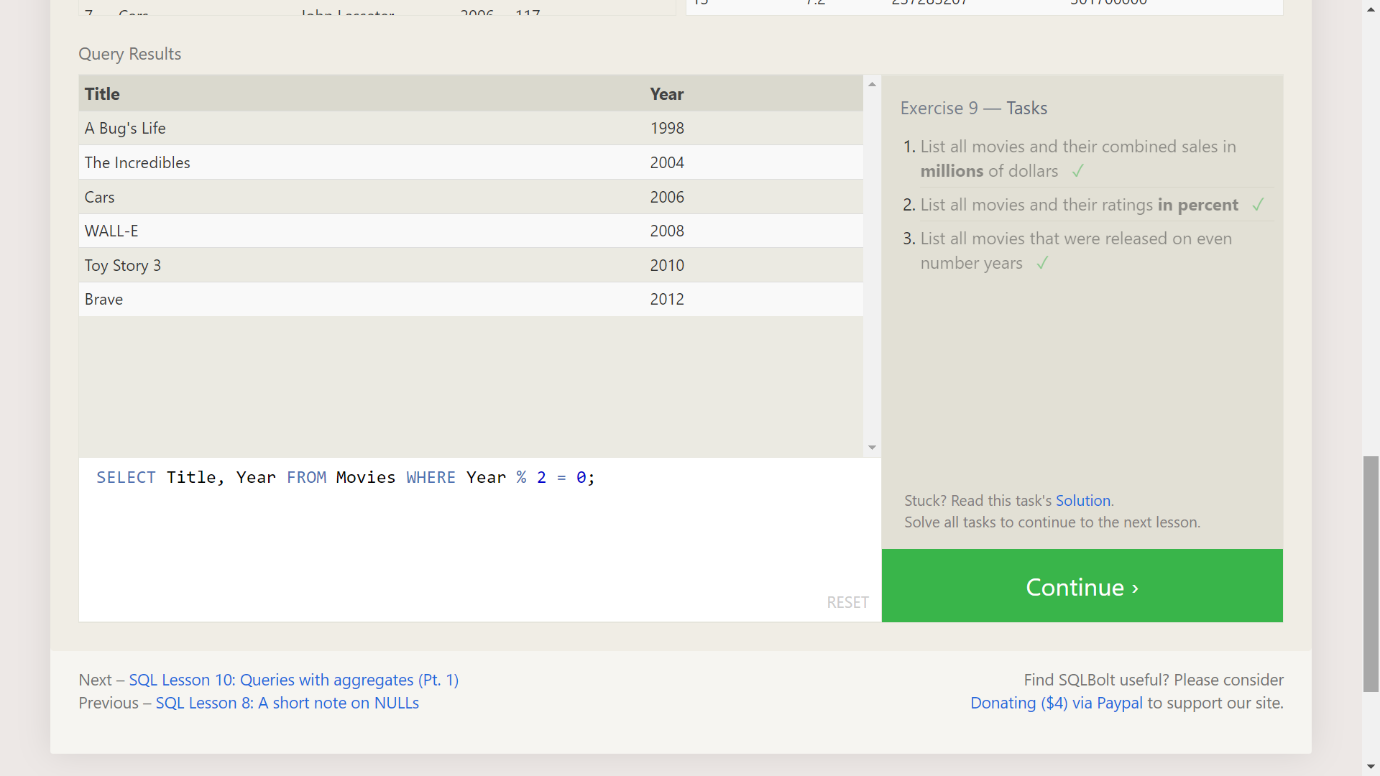
SELECT title, (domestic\_sales + international\_sales) / 1000000 AS Gross\_sales\_millions FROM Movies JOIN Boxoffice ON movies.id = Boxoffice.Movie\_id;

1. Find the list of all buildings and their capacity

SELECT Title, Rating \* 10 AS rating\_percent FROM Movies JOIN Boxoffice ON Movies.id = Boxoffice.Movie\_id;

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

SELECT Title, Year FROM Movies WHERE Year % 2 = 0;



**Excercise - 10 Tasks**

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

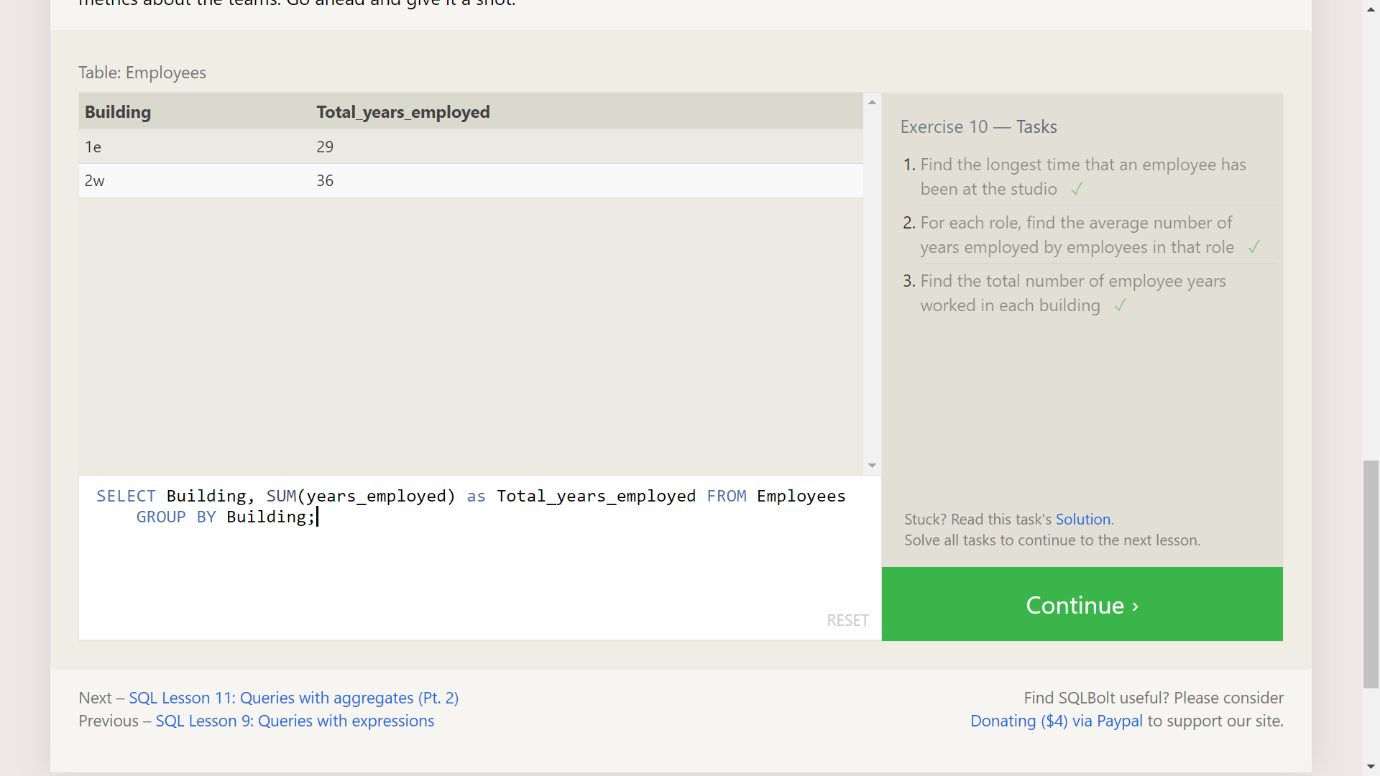
SELECT MAX(years\_employed) as Max\_years\_employed FROM employees;

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;

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

SELECT Building, SUM(years\_employed) as Total\_years\_employed FROM Employees GROUP BY Building;



**Excercise - 11 Tasks**

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

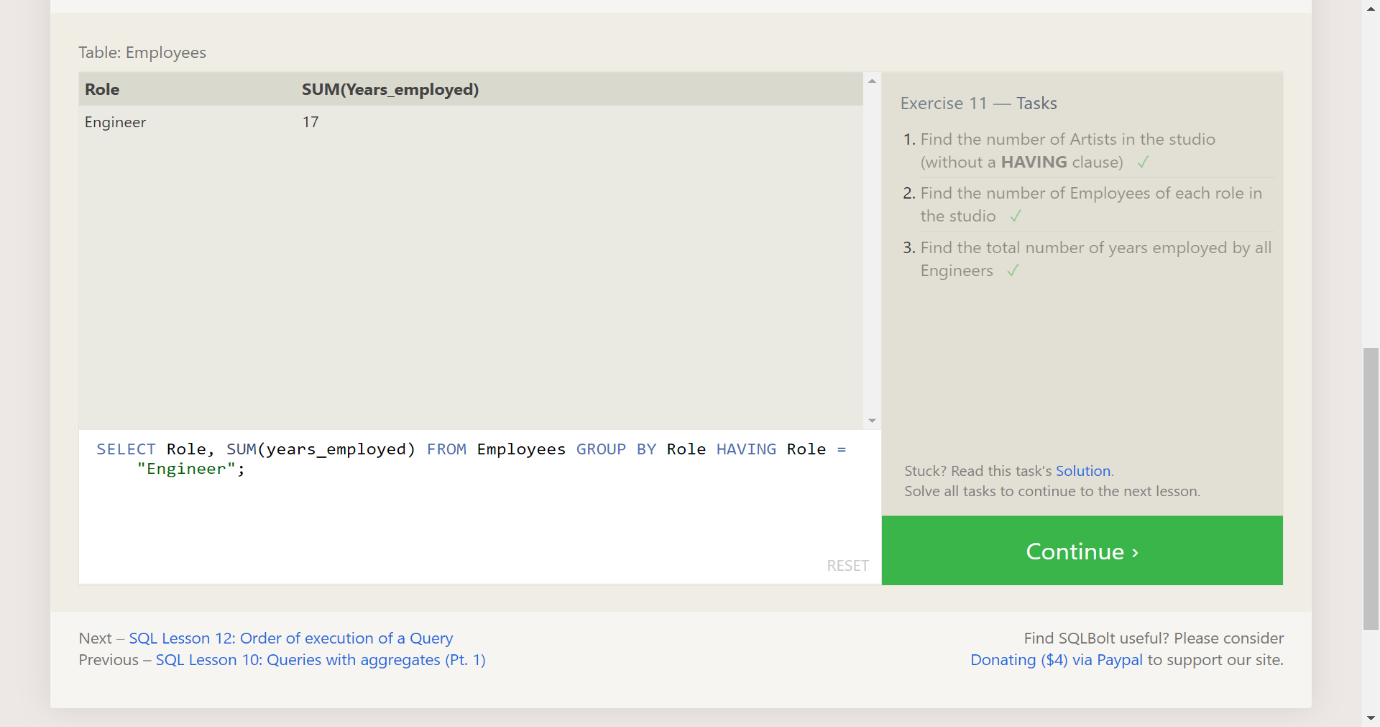
SELECT Role, COUNT(\*) as Number\_of\_artists FROM Employees WHERE Role = "Artist";

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

SELECT Role, COUNT(\*)FROM Employees GROUP BY Role;

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

SELECT Role, SUM(years\_employed) FROM Employees GROUP BY Role HAVING Role = "Engineer";



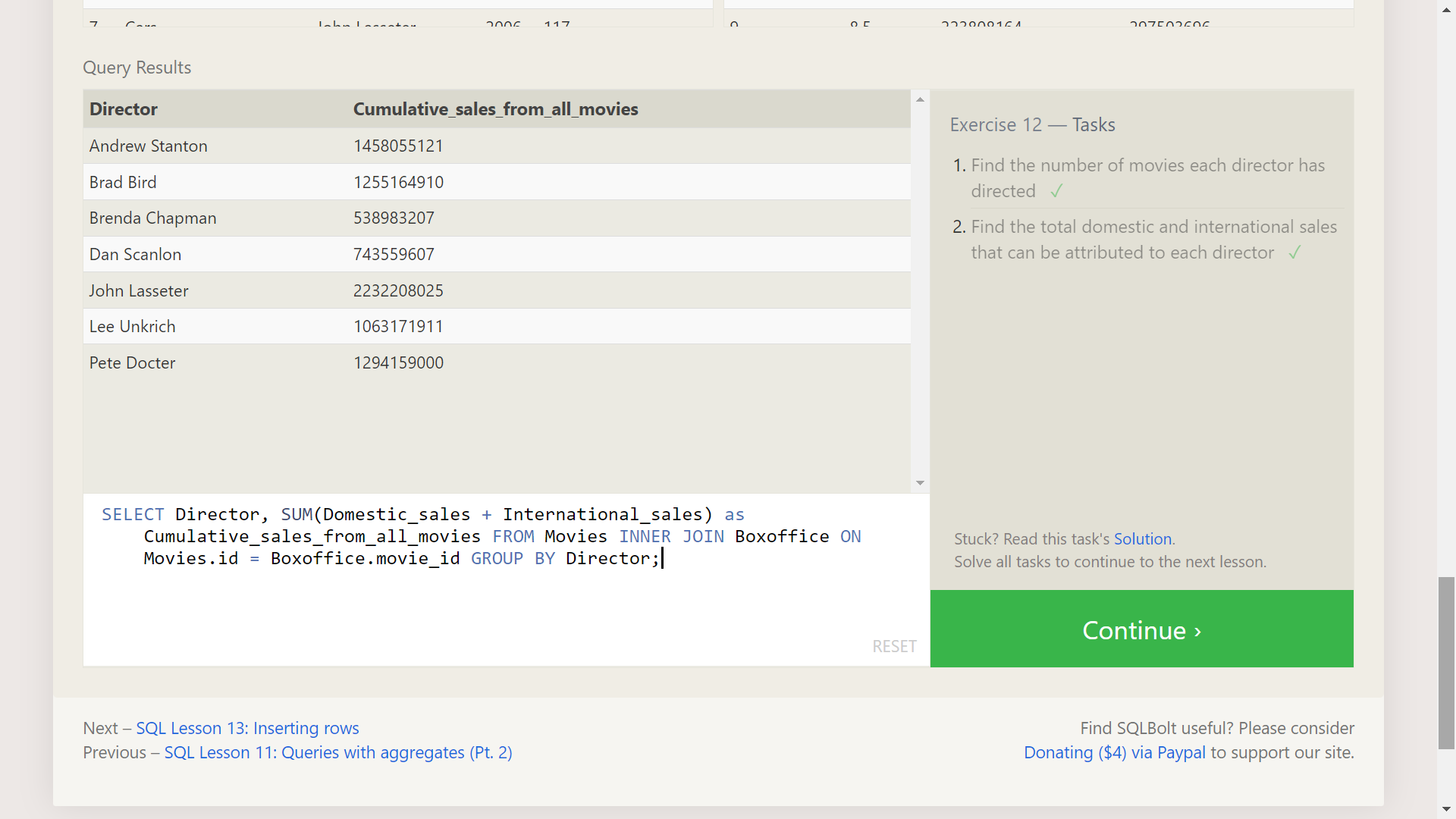
**Excercise - 12 Tasks**

1. Find the number of movies each director has directed

SELECT Director, COUNT(id) as Num\_movies\_directed FROM Movies GROUP BY Director;

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

SELECT Director, SUM(Domestic\_sales + International\_sales) as Cumulative\_sales\_from\_all\_movies FROM Movies INNER JOIN Boxoffice ON Movies.id = Boxoffice.movie\_id GROUP BY Director;



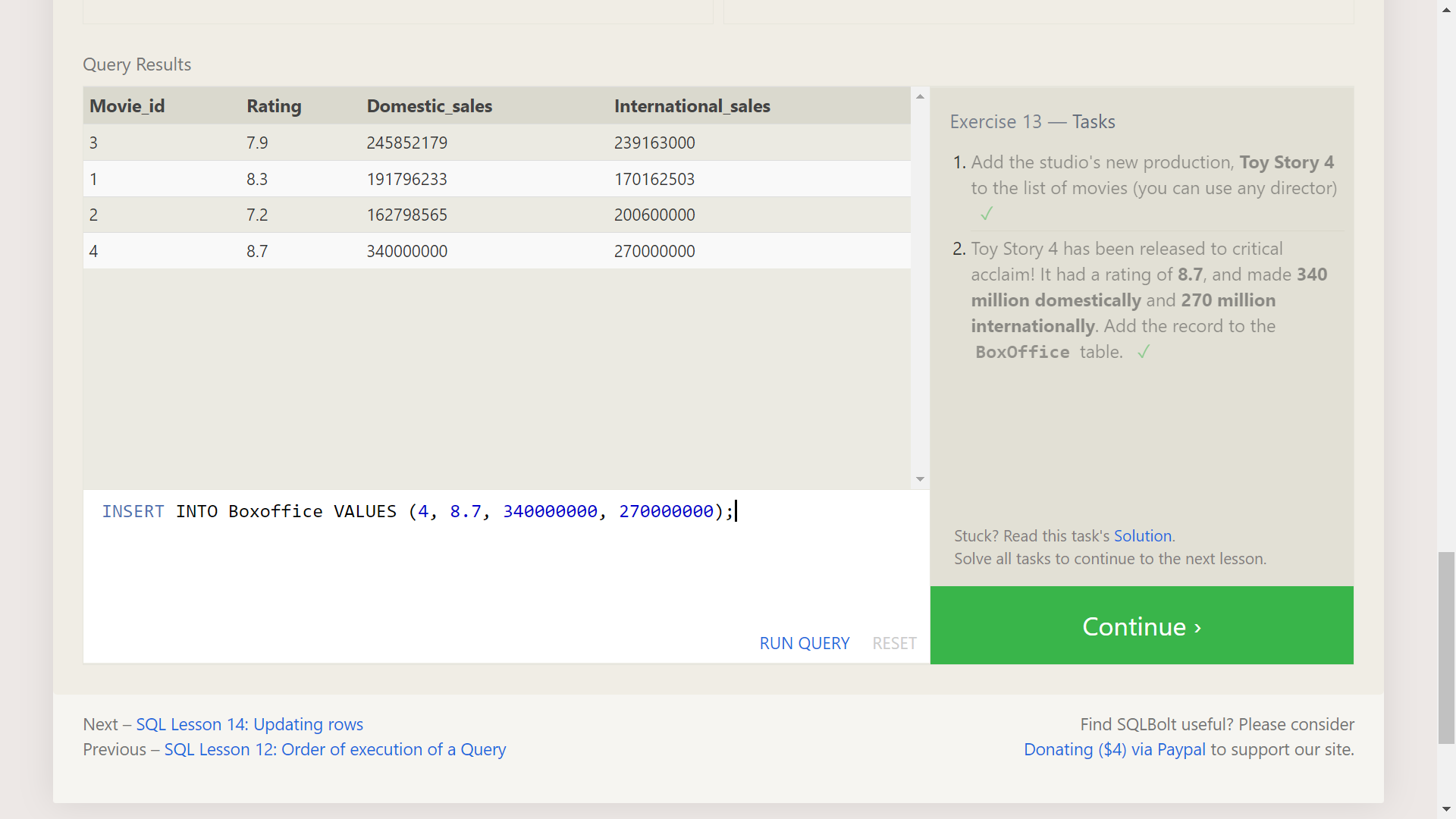
**Excercise - 13 Tasks**

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

SELECT Director, COUNT(id) as Num\_movies\_directed FROM Movies GROUP BY Director;

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.

INSERT INTO Boxoffice VALUES (4, 8.7, 340000000, 270000000);



**Excercise - 14 Tasks**

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

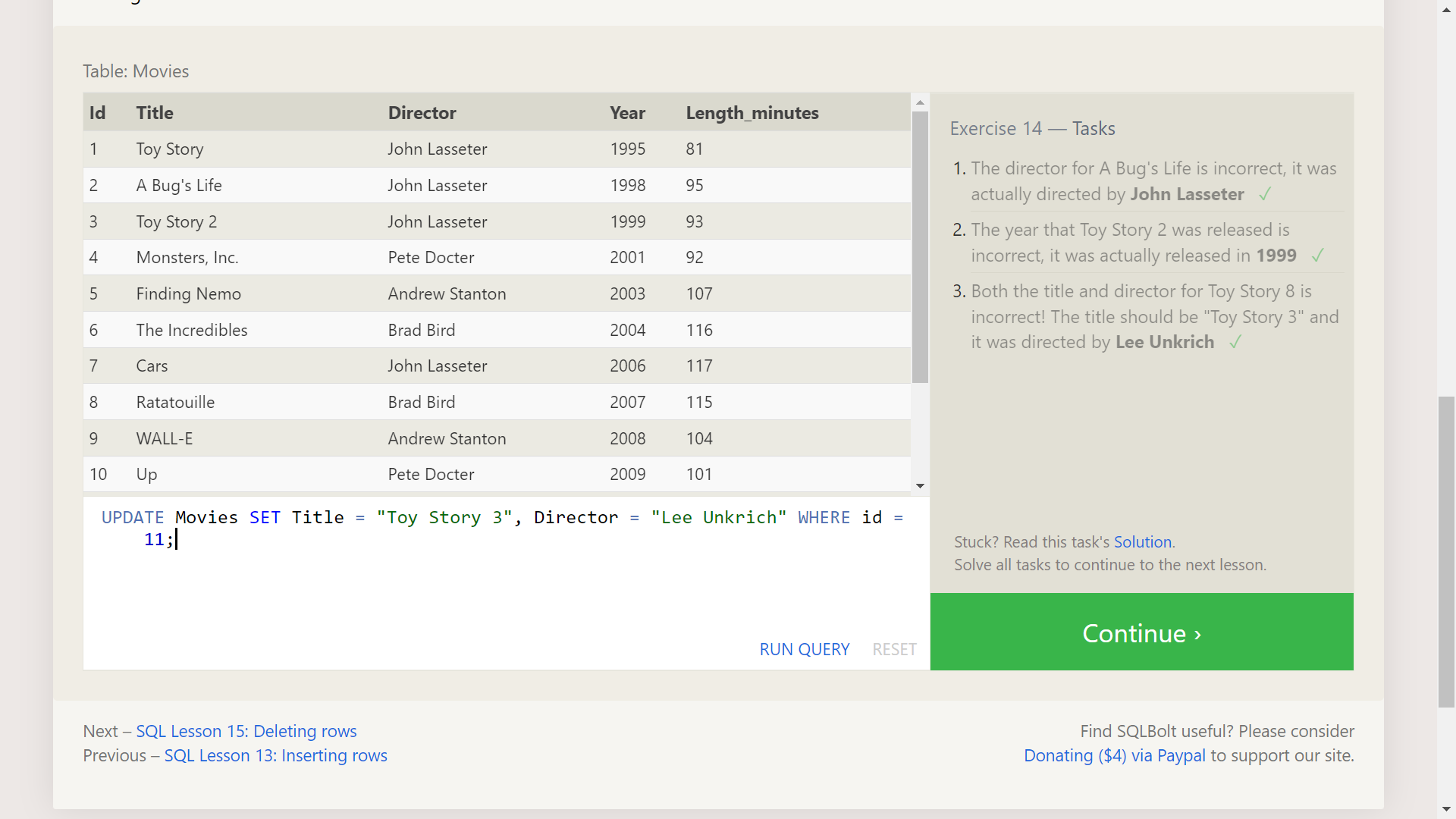
UPDATE Movies SET Director = "John Lasseter" WHERE id = 2;

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

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

UPDATE Movies SET Title = "Toy Story 3", Director = "Lee Unkrich" WHERE id = 11;



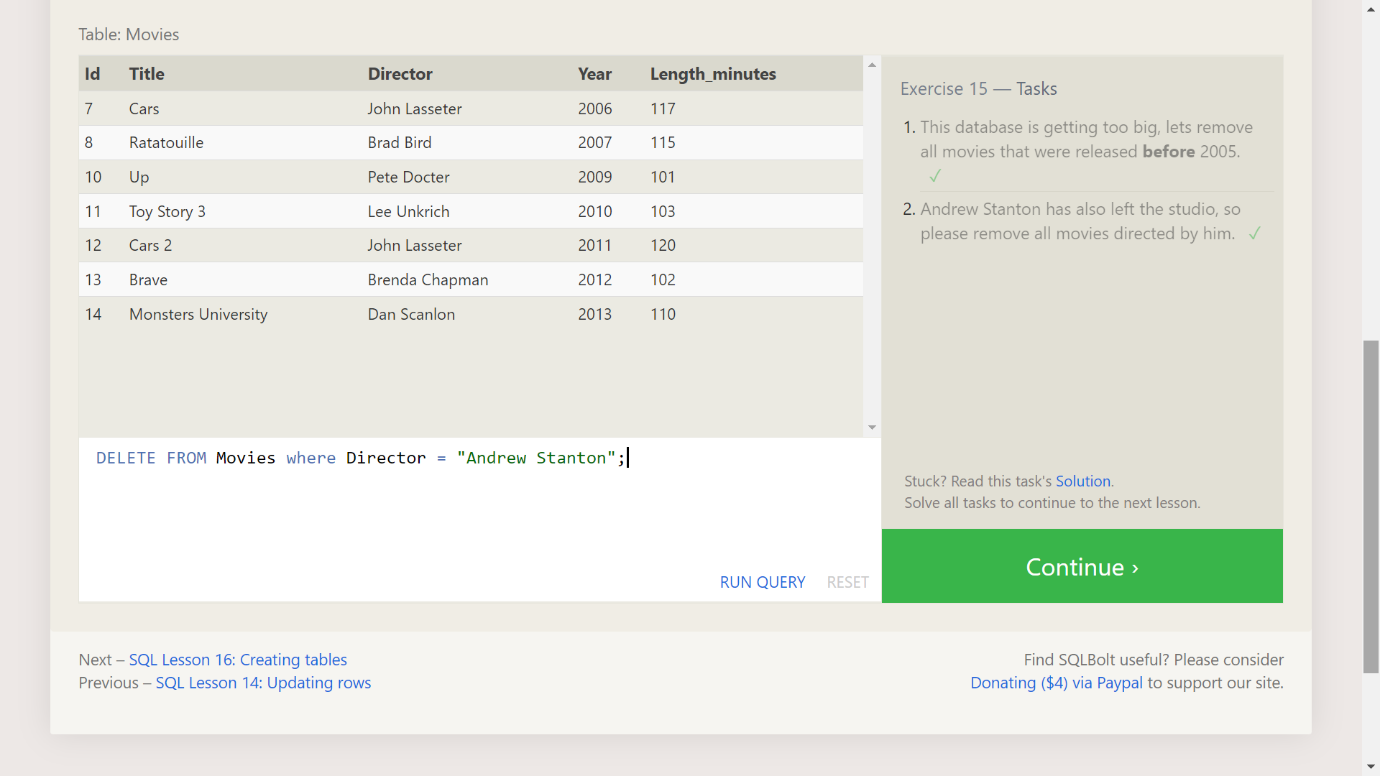
**Excercise - 15 Tasks**

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

DELETE FROM Movies where Year < 2005;

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

DELETE FROM Movies where Director = "Andrew Stanton";



**Excercise - 16 Tasks**

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

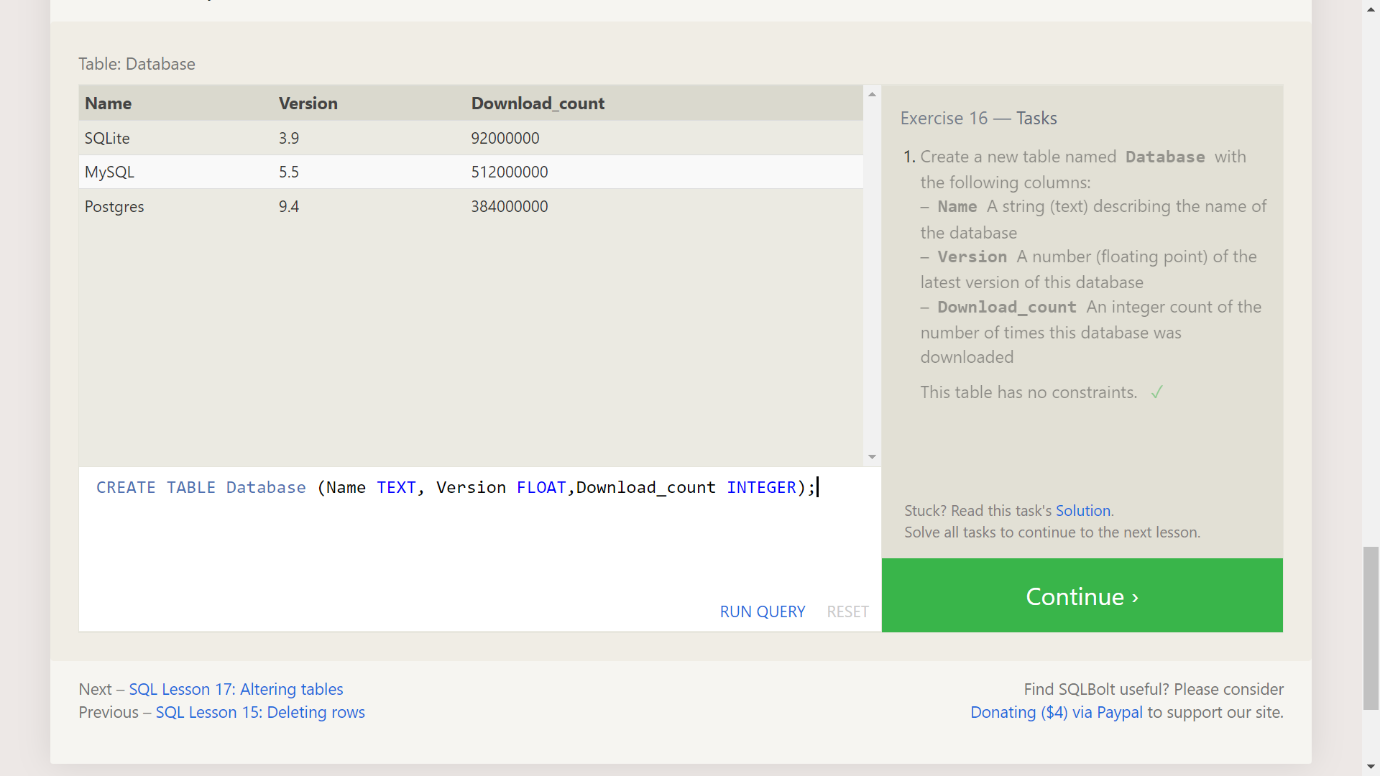
– Name A string (text) describing the name of the database

– 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 TEXT, Version FLOAT,Download\_count INTEGER);



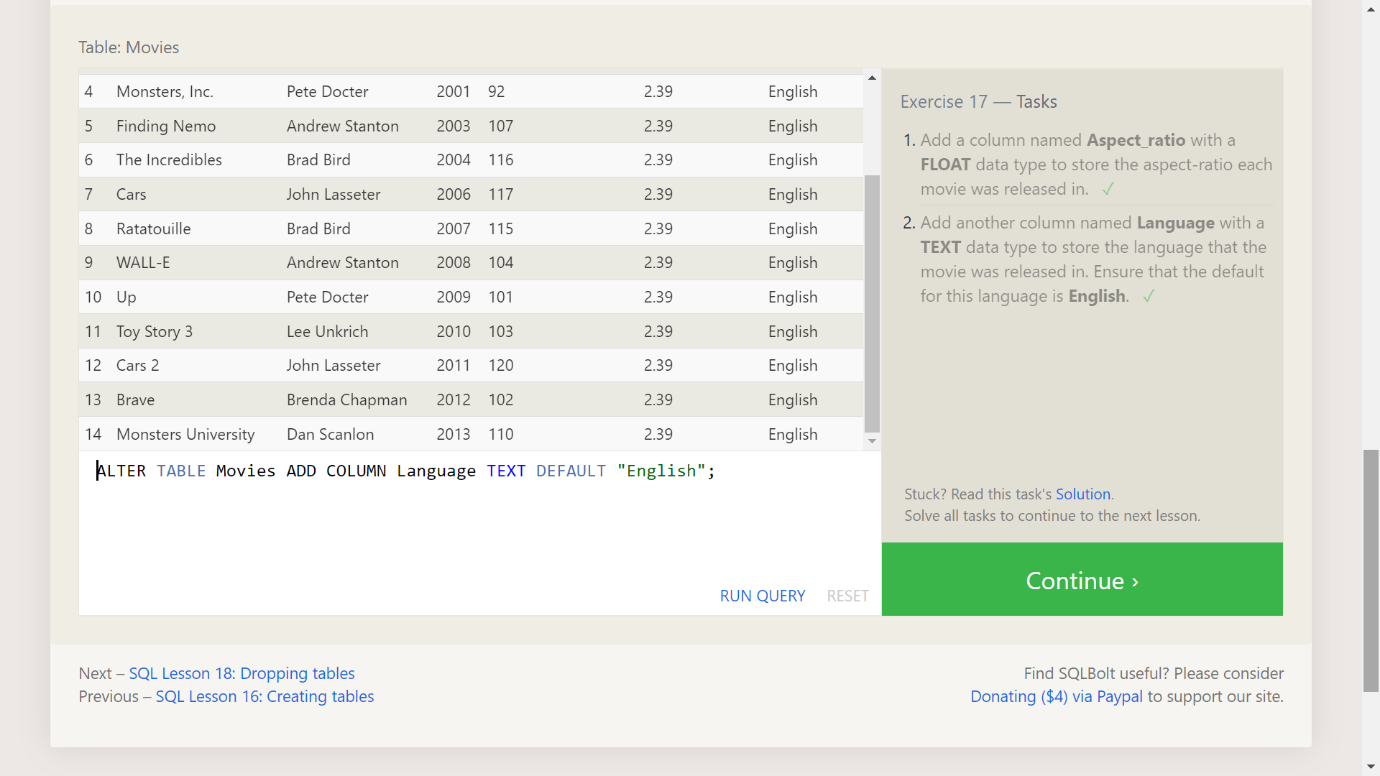
**Excercise - 17 Tasks**

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 COLUMN Aspect\_ratio FLOAT DEFAULT 2.39;

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.

ALTER TABLE Movies ADD COLUMN Language TEXT DEFAULT "English";



**Excercise - 18 Tasks**

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

DROP TABLE Movies;

1. And drop the BoxOffice table as well.

DROP TABLE BoxOffice;

