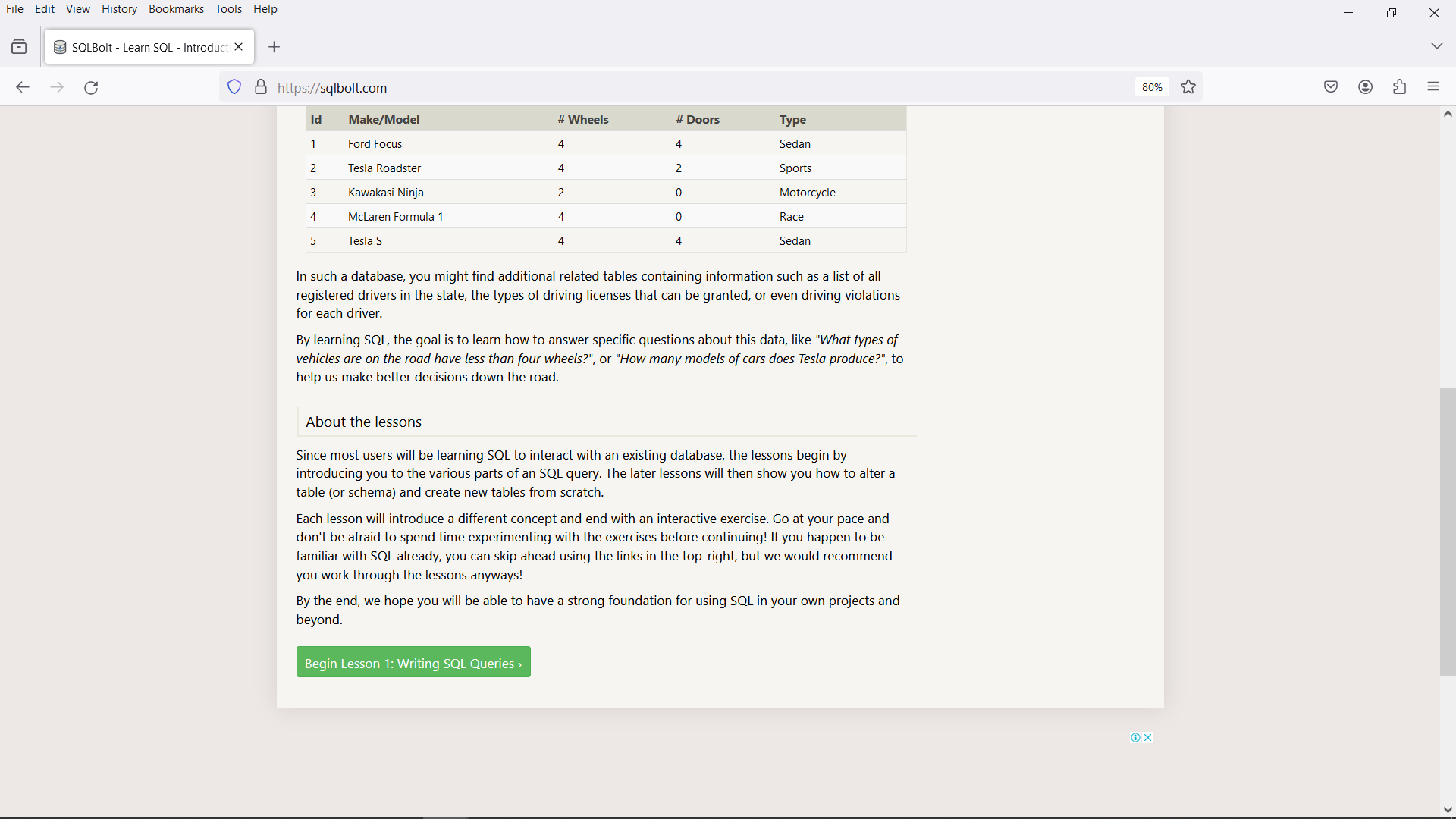
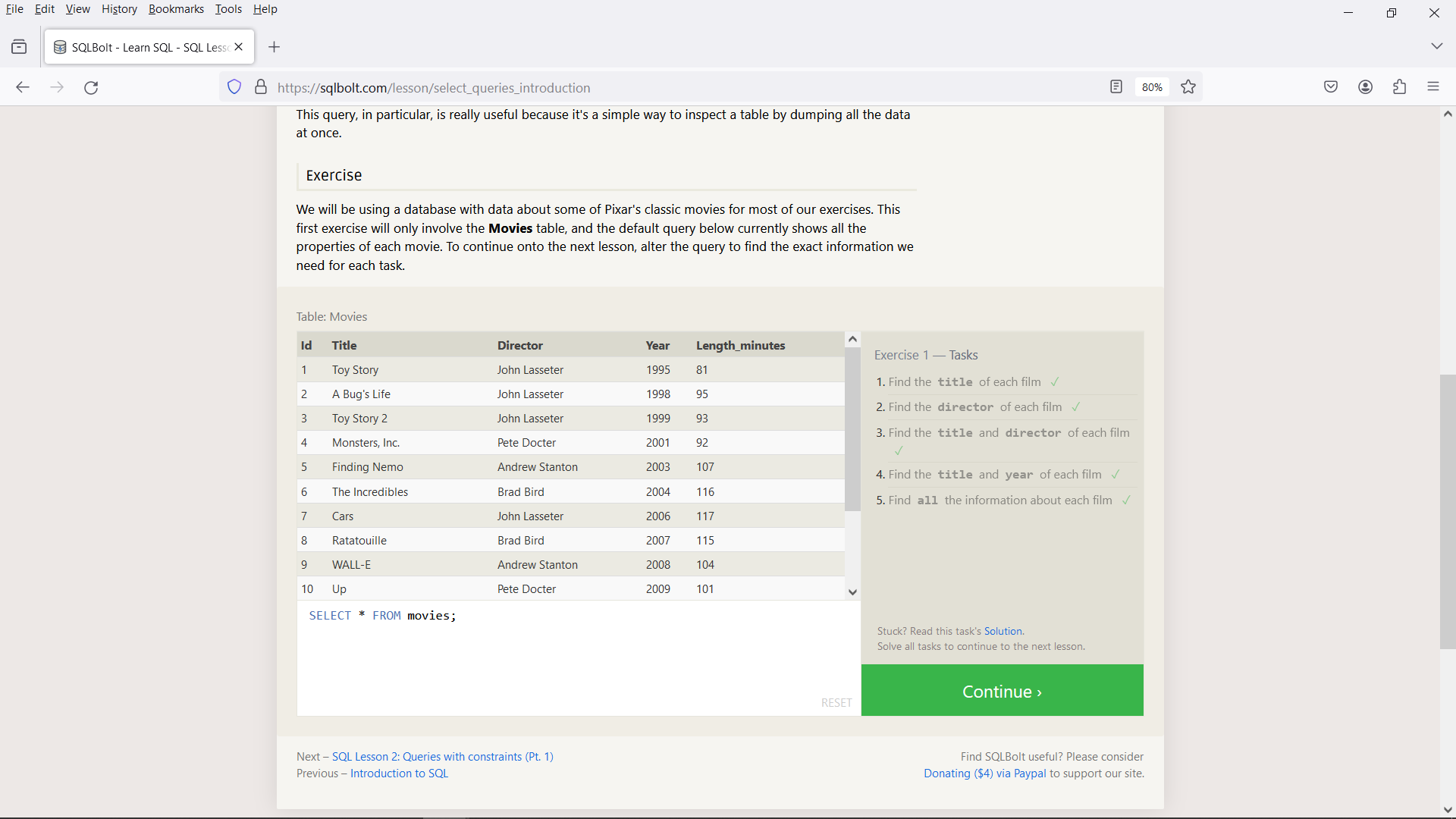
http://sqlbolt.com/ take a screenshot of your executed queries then upload it to GitHub and submit your URL



SQL Queries for the Questions in ScreenShot

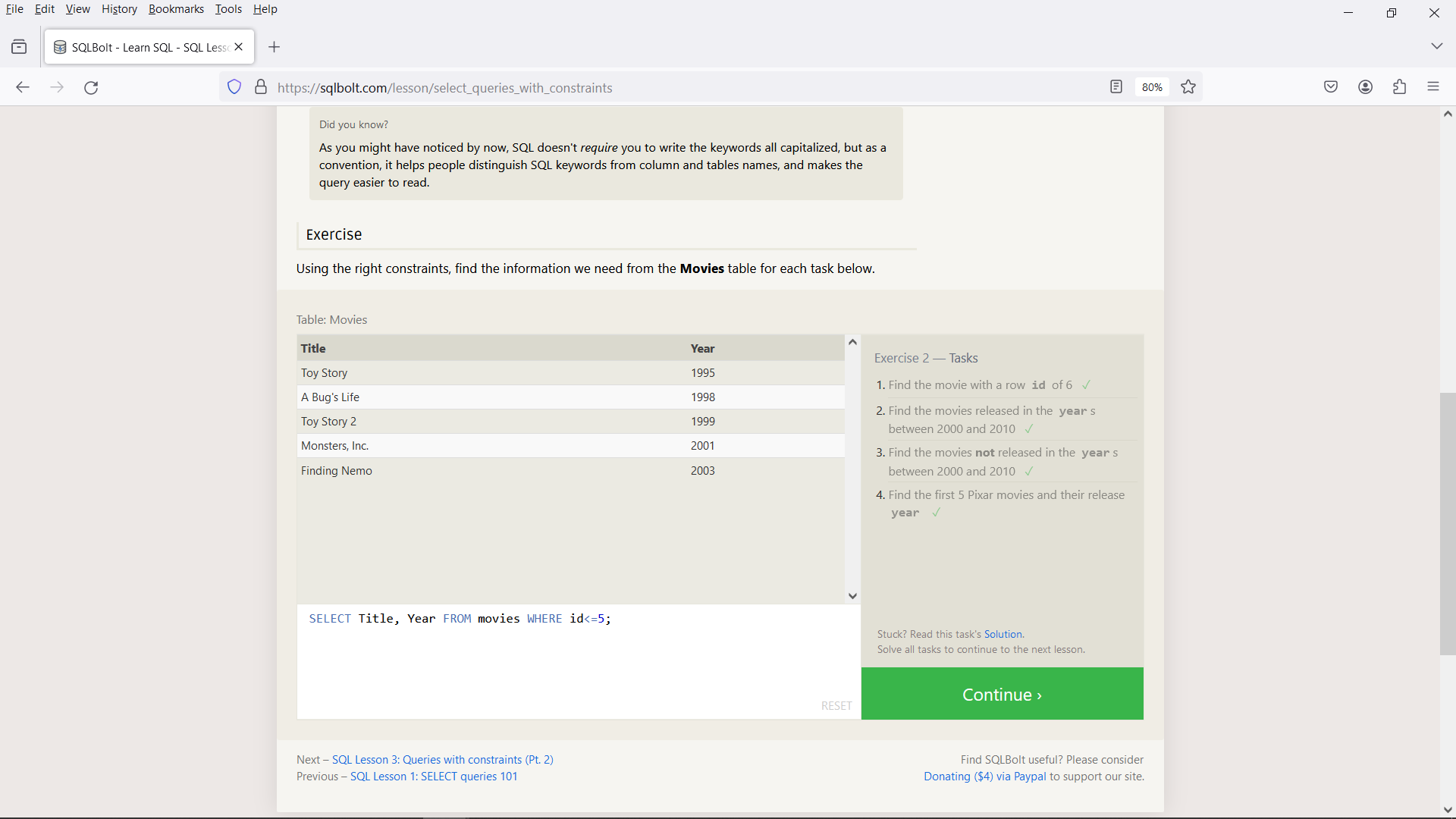
Exercise-1

1. SELECT Title FROM movies;
2. SELECT Director FROM movies;
3. SELECT Title, Director FROM movies;
4. SELECT Title, Year FROM movies;
5. SELECT \* FROM movies;



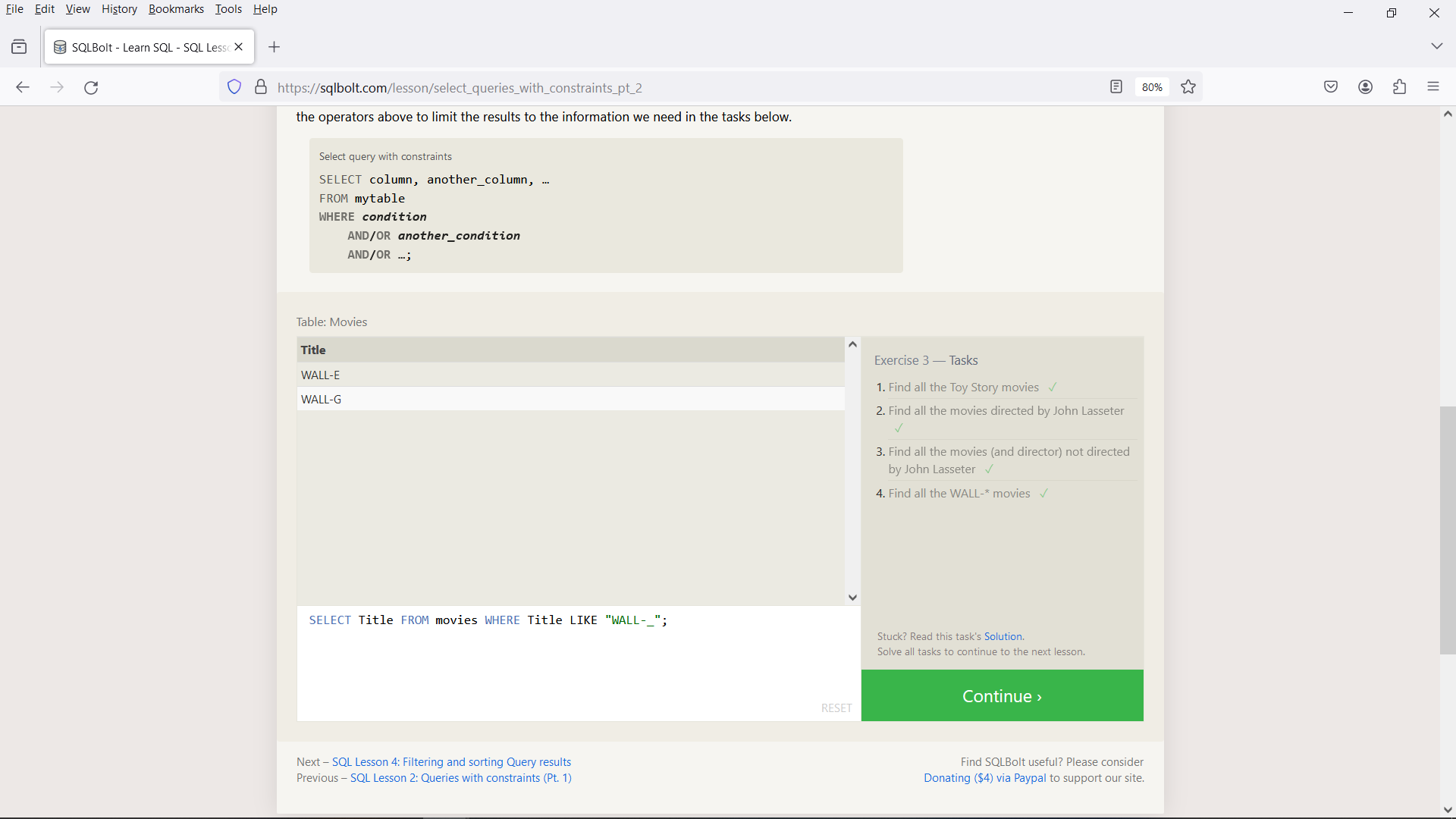
Exercise-2:

1. SELECT Title FROM movies WHERE id=6;
2. SELECT Title, Year FROM movies WHERE Year BETWEEN 2000 AND 2010;
3. SELECT Title, Year FROM movies WHERE Year NOT BETWEEN 2000 AND 2010;
4. SELECT Title, Year FROM movies WHERE id<=5;



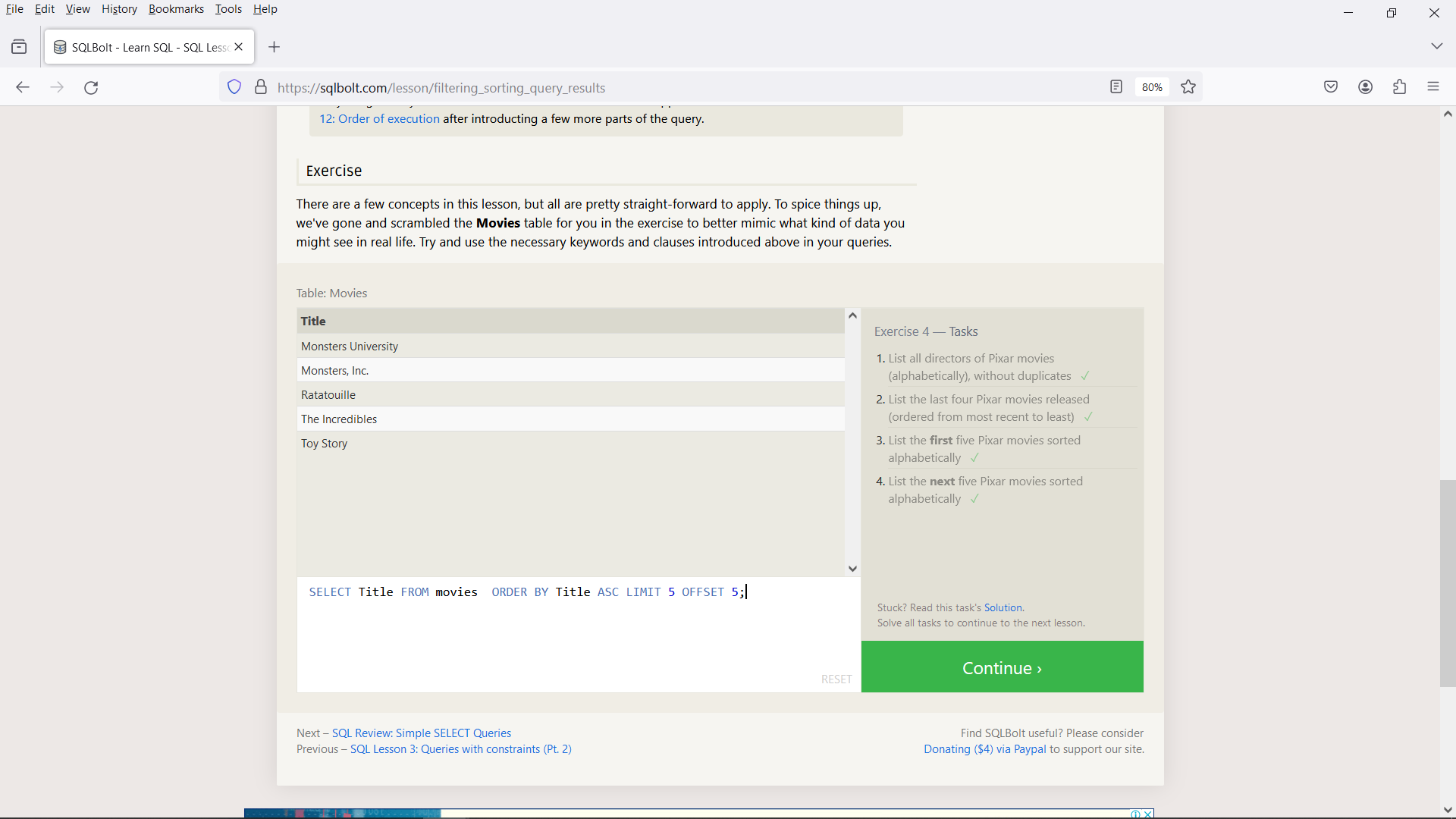
Exercise-3:

1. SELECT Title FROM movies WHERE Title LIKE "Toy Story%";
2. SELECT Title, Director FROM movies WHERE Director LIKE "John Lasseter";
3. SELECT Title, Director FROM movies WHERE Director NOT LIKE "John Lasseter";
4. SELECT Title FROM movies WHERE Title LIKE "WALL-\_";



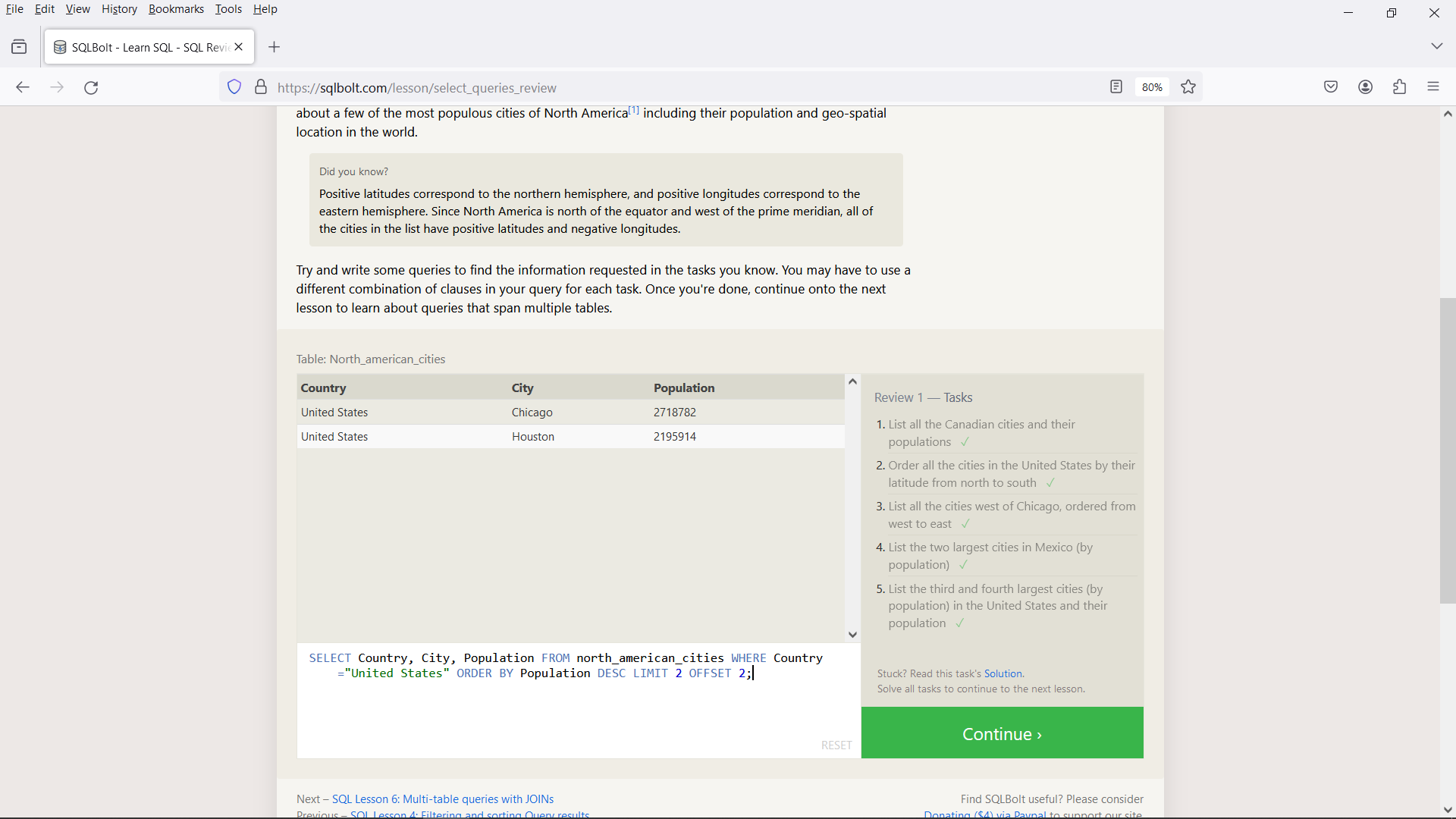
Exercise-4:

1. SELECT DISTINCT Director FROM movies ORDER BY Director ASC;
2. SELECT Title, Year FROM movies ORDER BY Year DESC LIMIT 4;
3. SELECT Title FROM movies ORDER BY Title ASC LIMIT 5;
4. SELECT Title FROM movies ORDER BY Title ASC LIMIT 5 OFFSET 5;



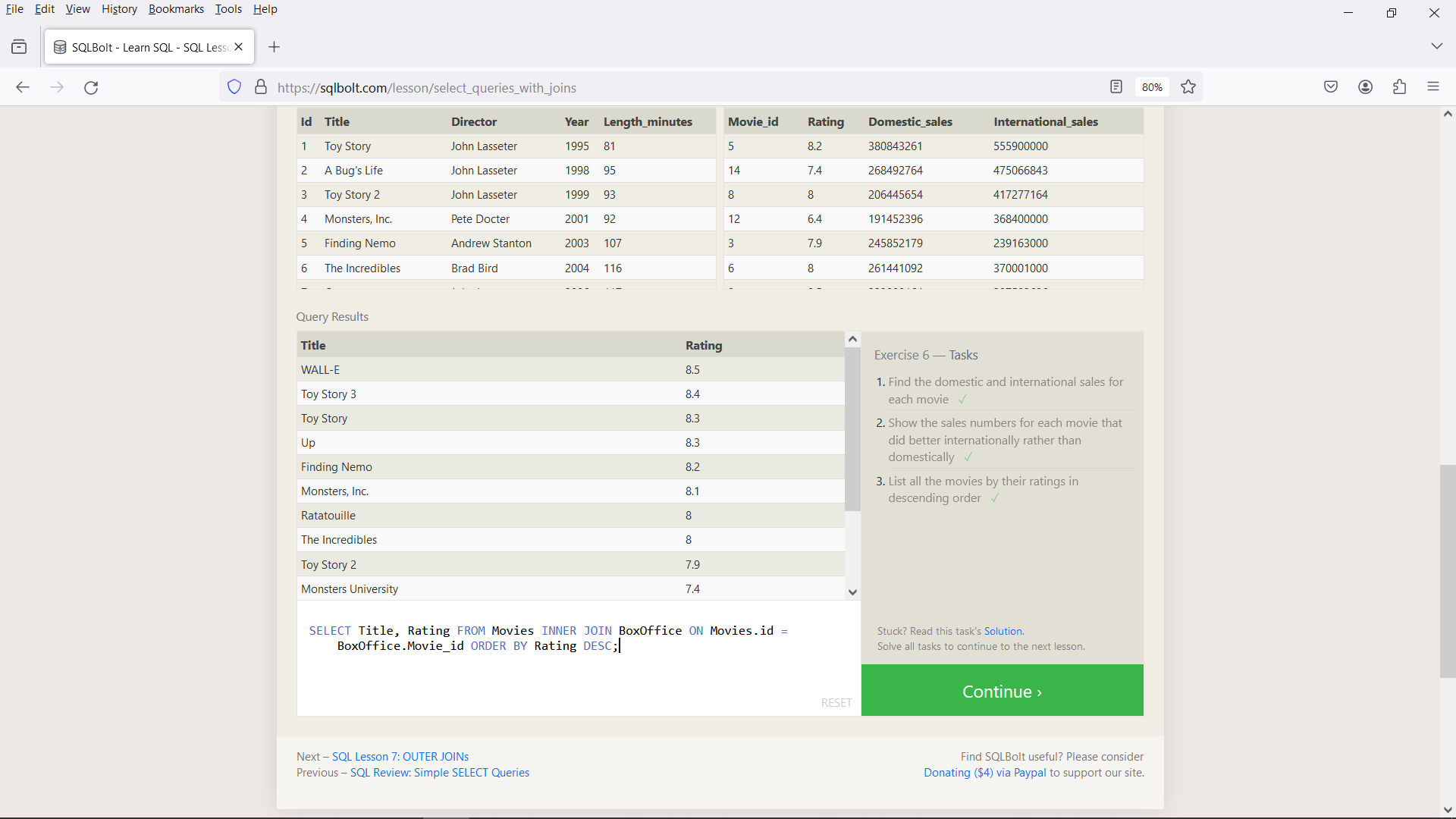
Exercise-5:

1. SELECT Country, City, Population FROM north\_american\_cities WHERE Country="Canada";
2. SELECT Country, City, Latitude FROM north\_american\_cities WHERE Country="United States" ORDER BY Latitude DESC;
3. SELECT City, Longitude FROM north\_american\_cities WHERE Longitude<-87.629798 ORDER BY Longitude ASC;
4. SELECT Country, City, Population FROM north\_american\_cities WHERE Country="Mexico" ORDER BY Population DESC LIMIT 2;
5. SELECT Country, City, Population FROM north\_american\_cities WHERE Country="United States" ORDER BY Population DESC LIMIT 2 OFFSET 2;



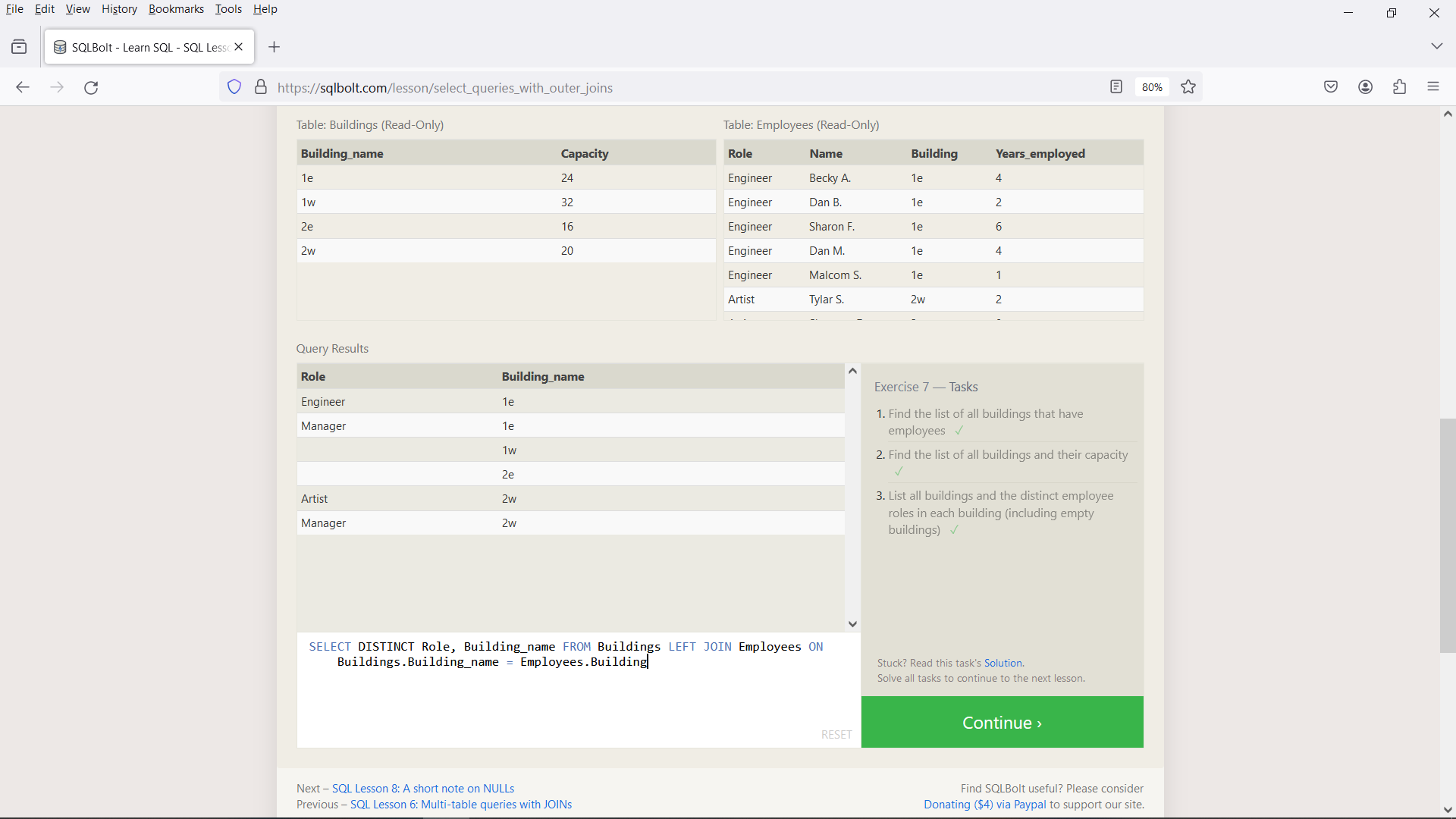
Exercise-6:

1. SELECT Title, domestic\_sales, International\_sales FROM Movies INNER JOIN BoxOffice ON Movies.id = BoxOffice.Movie\_id;
2. SELECT Title, domestic\_sales, International\_sales FROM Movies INNER JOIN BoxOffice ON Movies.id = BoxOffice.Movie\_id WHERE International\_sales>Domestic\_sales;
3. SELECT Title, Rating FROM Movies INNER JOIN BoxOffice ON Movies.id = BoxOffice.Movie\_id ORDER BY Rating DESC;



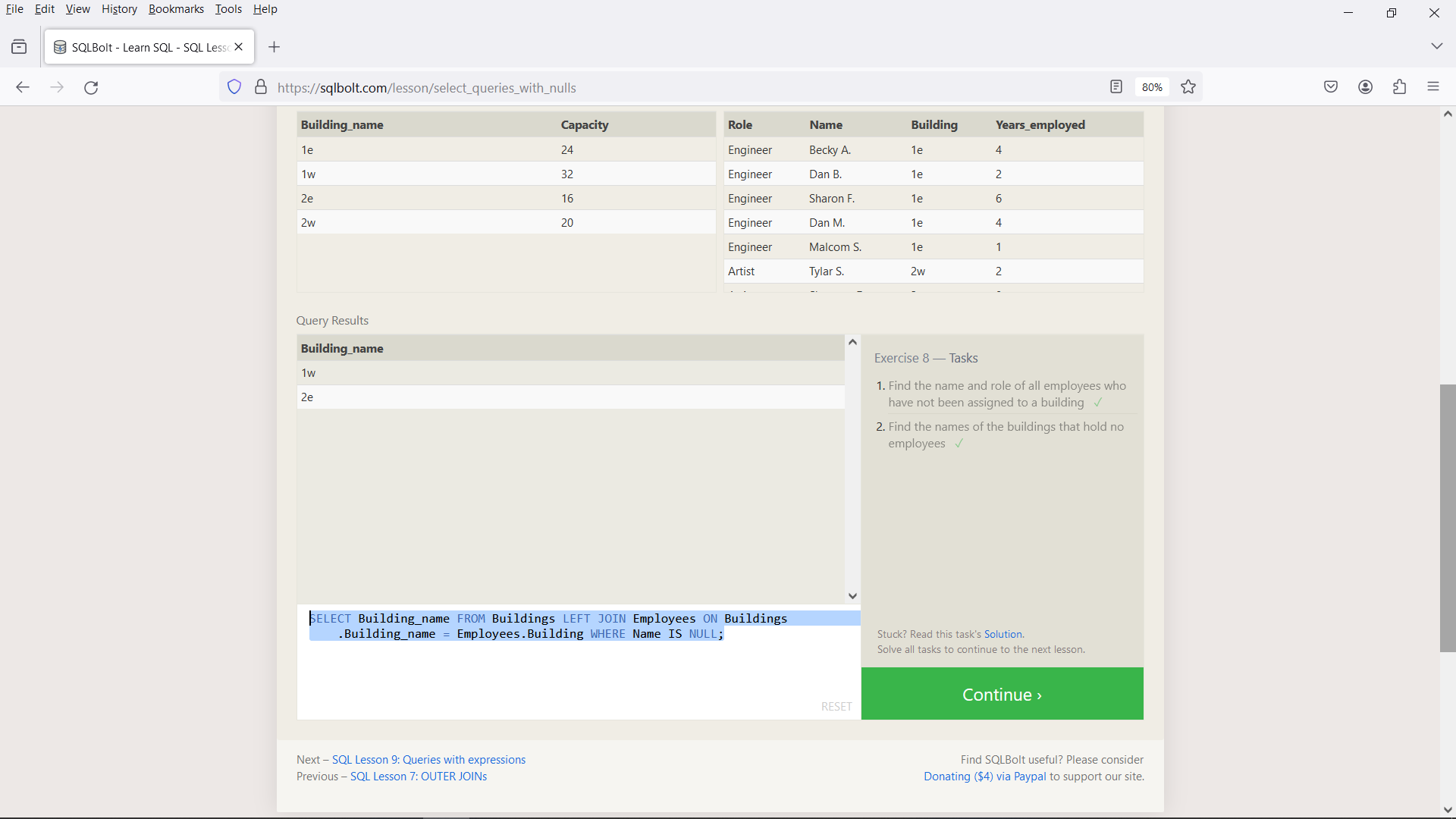
Exercise-7:

1. SELECT DISTINCT Building\_name FROM employees LEFT JOIN Buildings ON Employees.Building = Buildings.Building\_name;
2. SELECT \* FROM Buildings;
3. SELECT DISTINCT Role, Building\_name FROM Buildings LEFT JOIN Employees ON Buildings.Building\_name = Employees.Building



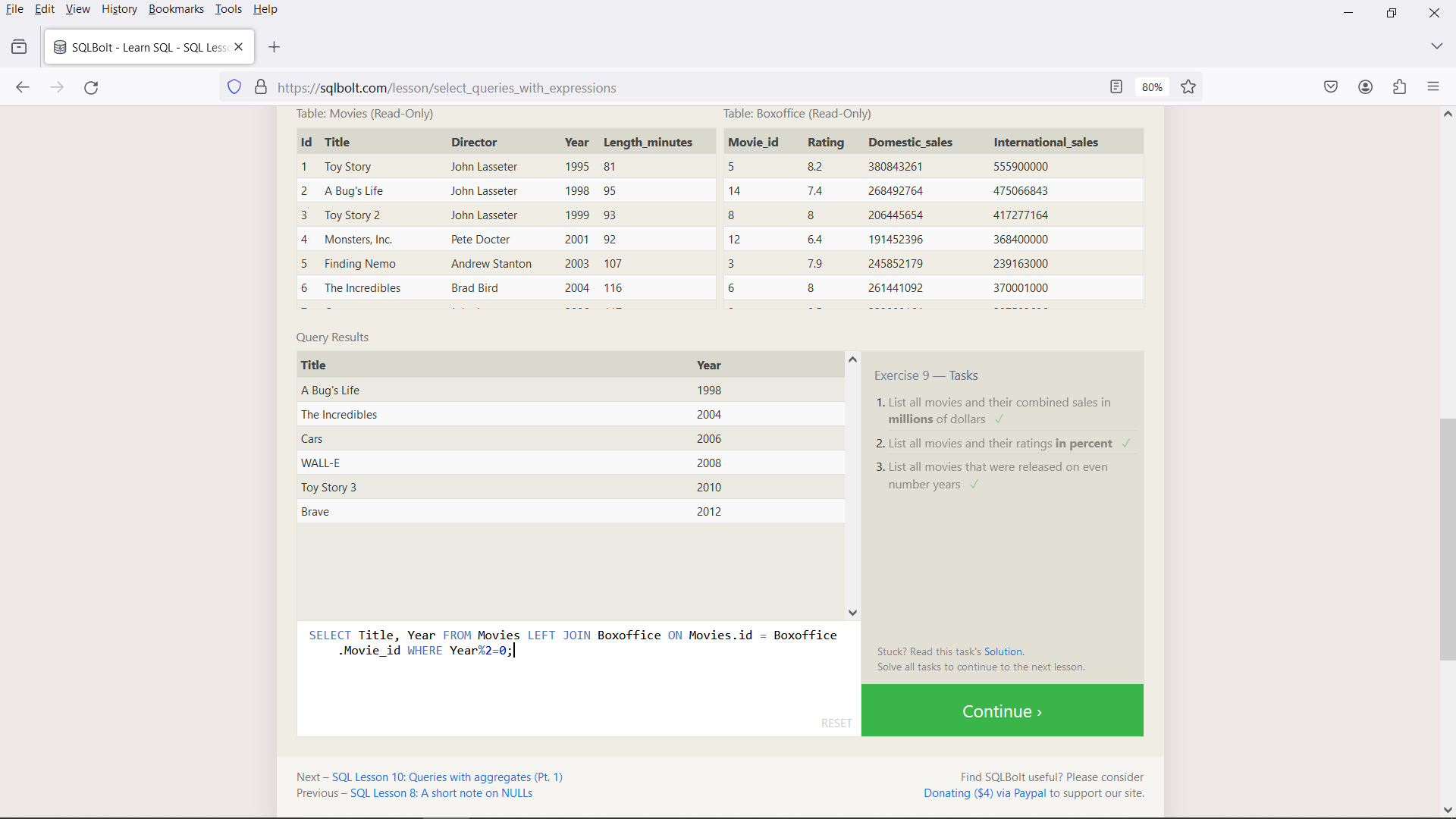
Exercise-8:

1. SELECT Name, Role FROM employees WHERE Building IS NULL;
2. SELECT Building\_name FROM Buildings LEFT JOIN Employees ON Buildings.Building\_name = Employees.Building WHERE Name IS NULL;



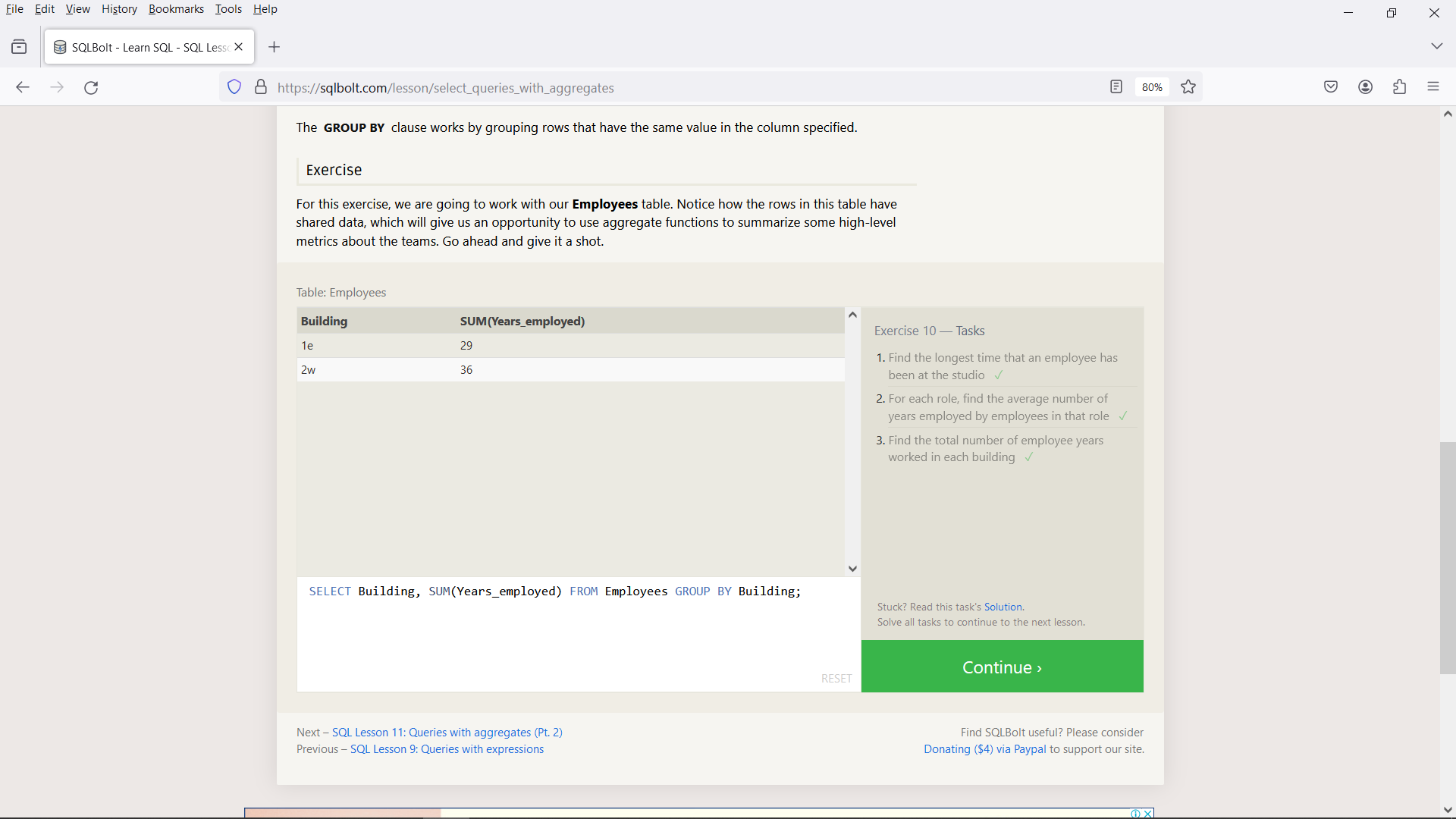
Exercise-9:

1. SELECT Title, ((domestic\_sales+international\_sales)/1000000) AS Sales\_millions FROM Movies LEFT JOIN Boxoffice ON Movies.id = Boxoffice.Movie\_id;
2. SELECT Title, ((Rating\*1000)/100) AS "Rating\_%" FROM Movies LEFT JOIN Boxoffice ON Movies.id = Boxoffice.Movie\_id;
3. SELECT Title, Year FROM Movies LEFT JOIN Boxoffice ON Movies.id = Boxoffice.Movie\_id WHERE Year%2=0;



Exercise-10:

1. SELECT Max(Years\_employed) AS LongestDuration FROM employees;
2. SELECT Role, AVG(Years\_employed) AS AvgYearsWorked FROM Employees GROUP BY Role;
3. SELECT Building, SUM(Years\_employed) FROM Employees GROUP BY Building;

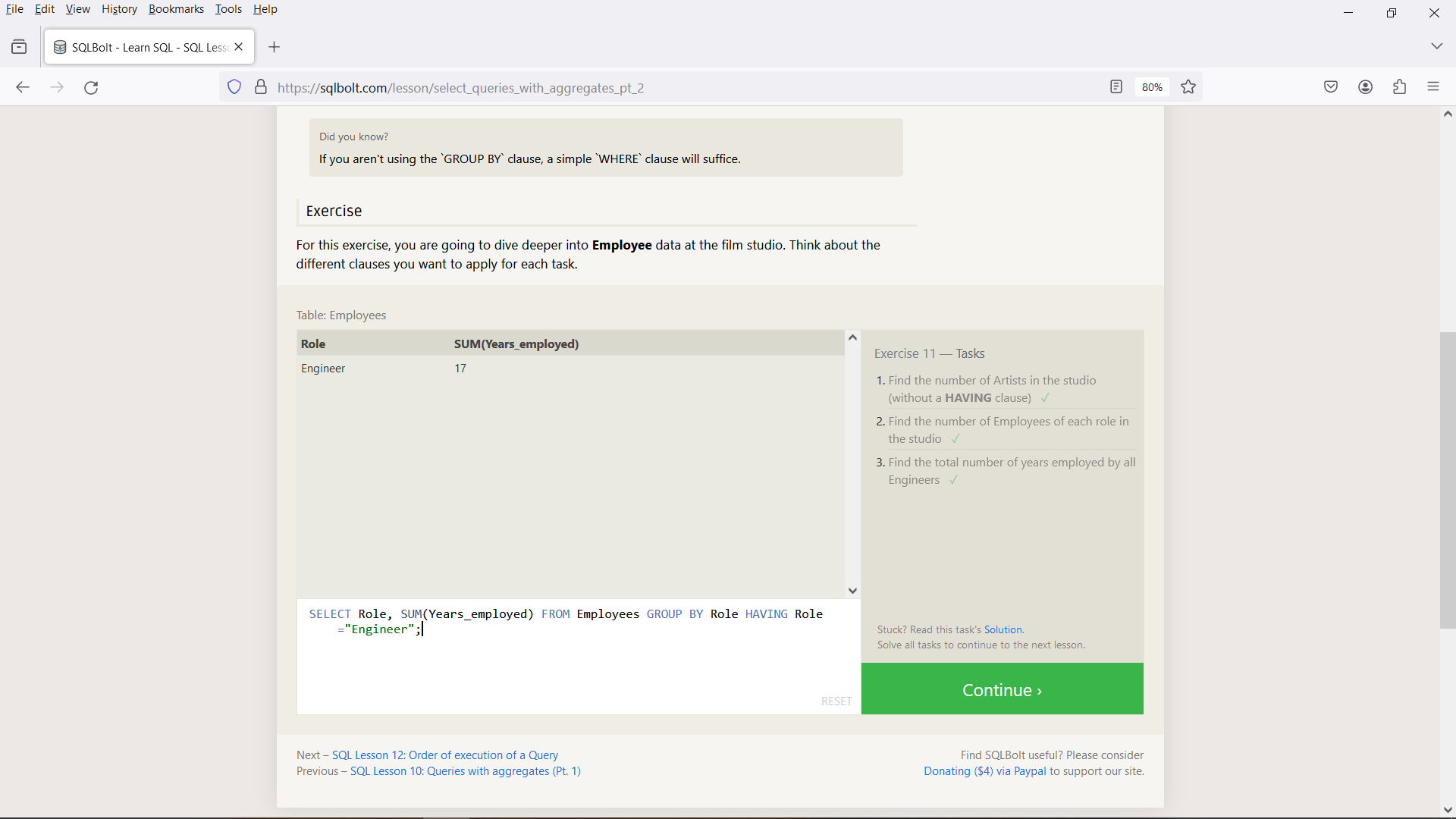


Exercise-11:

SELECT Count(Role) FROM employees WHERE Role="Artist";

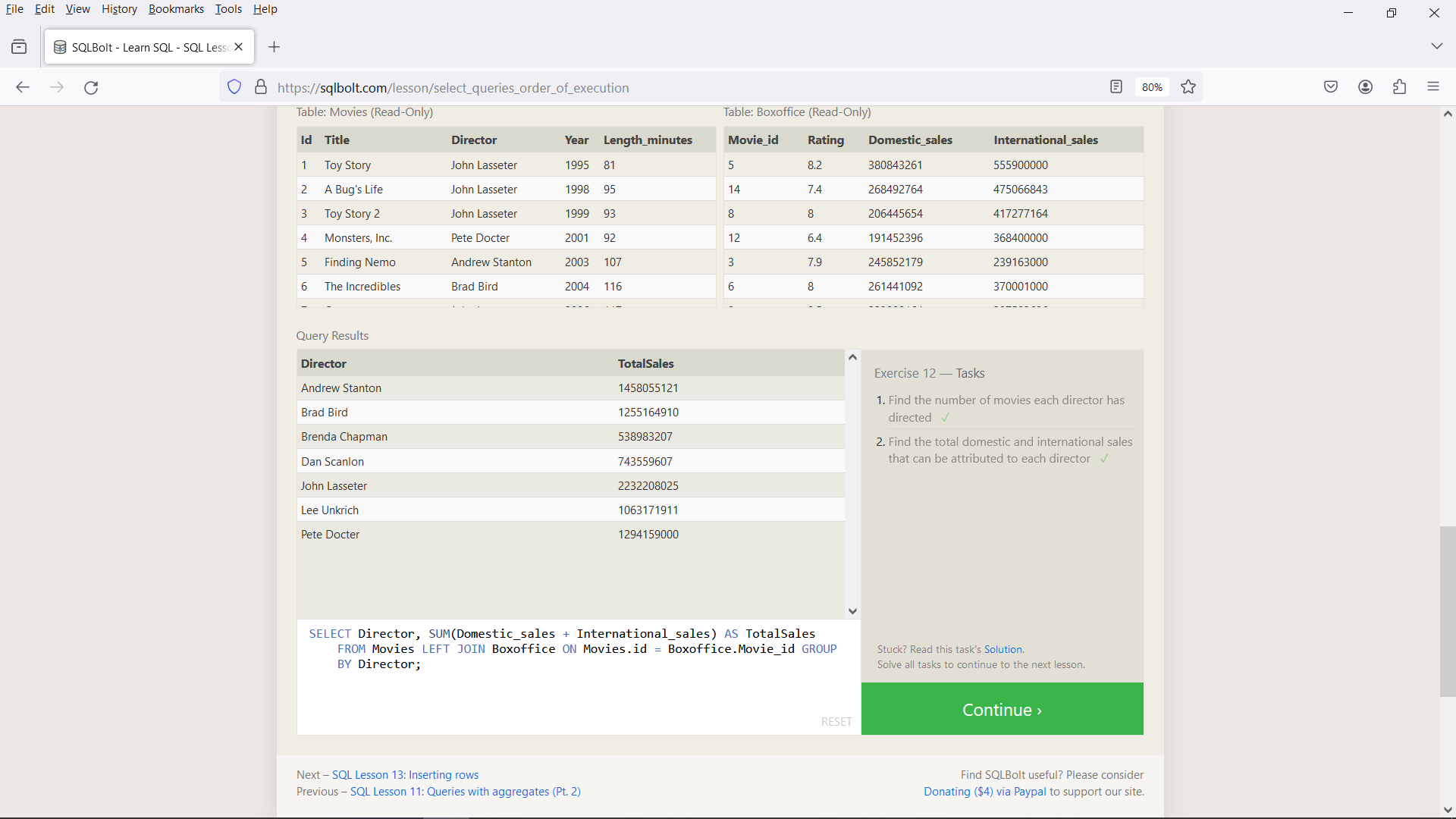
SELECT Role, COUNT(Role) AS TotalEmployees FROM employees GROUP BY Role;

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



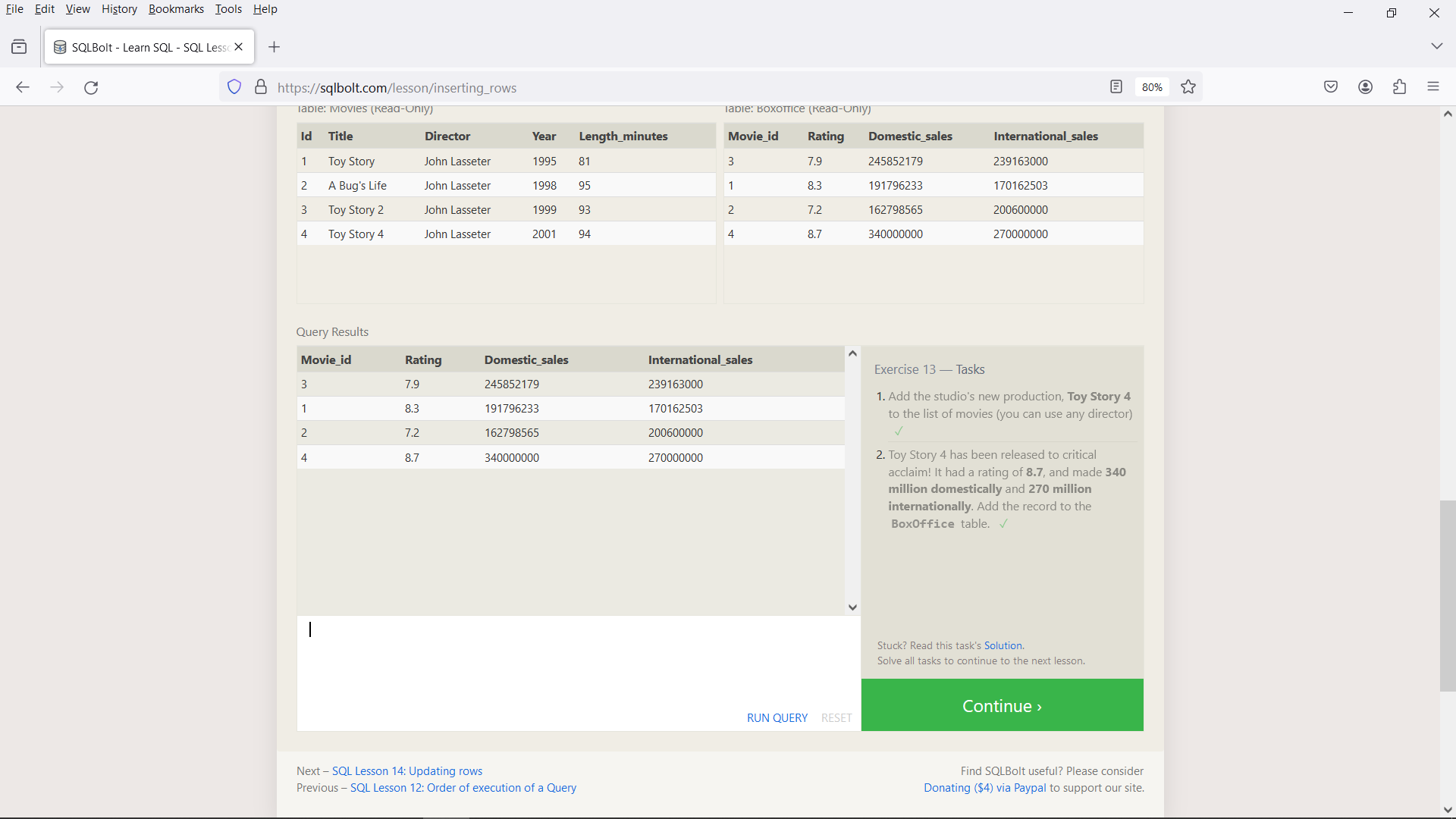
Exercise-12:

1. SELECT Director, COUNT(Title) AS Movies\_Directed FROM Movies GROUP BY Director;
2. SELECT Director, SUM(Domestic\_sales + International\_sales) AS TotalSales FROM Movies LEFT JOIN Boxoffice ON Movies.id = Boxoffice.Movie\_id GROUP BY Director;



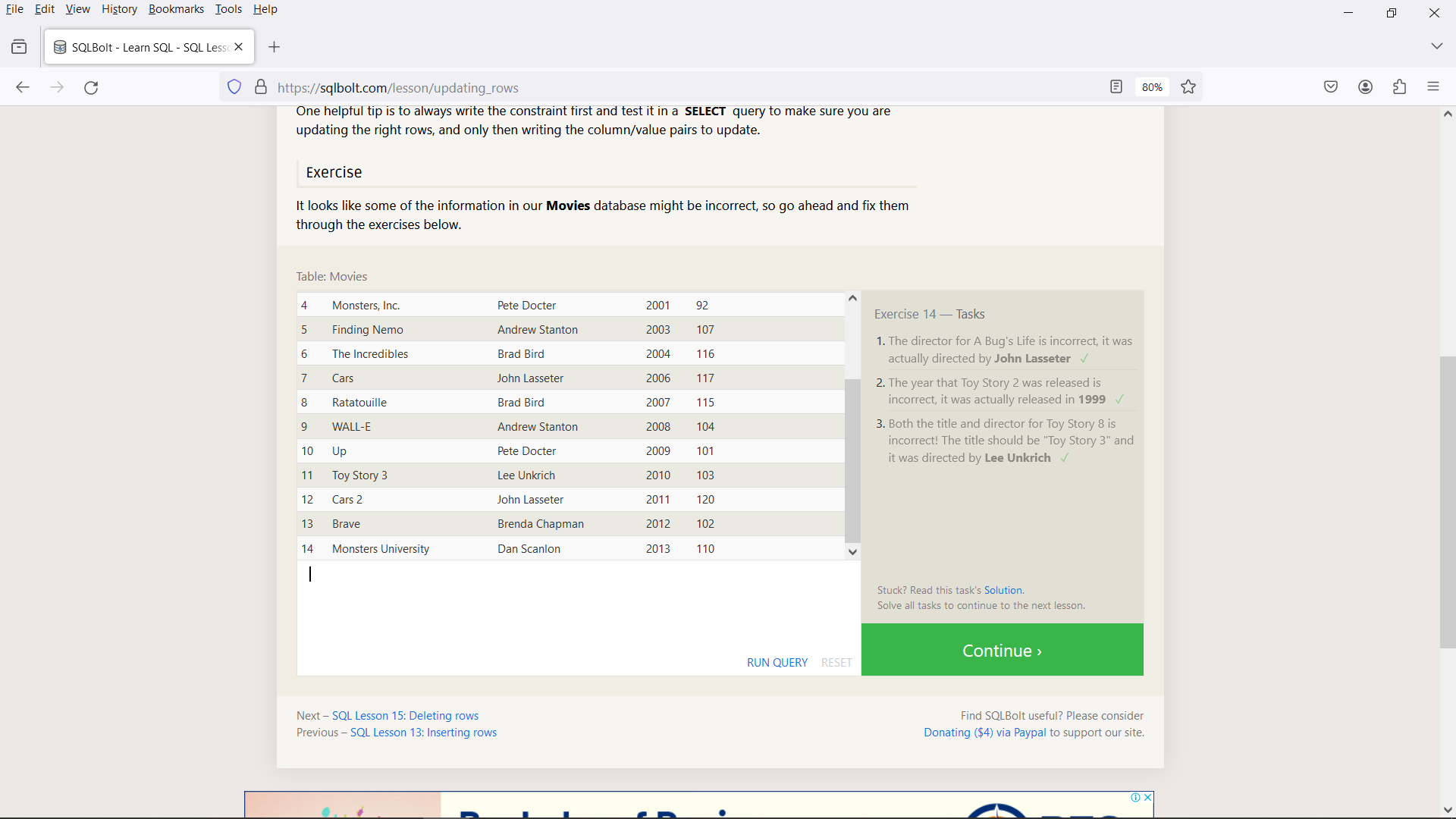
Exercise-13:

1. INSERT INTO Movies VALUES (4, "Toy Story 4", "John Lasseter", 2001, 94);
2. INSERT INTO Boxoffice VALUES(4, 8.7, 340000000, 270000000)



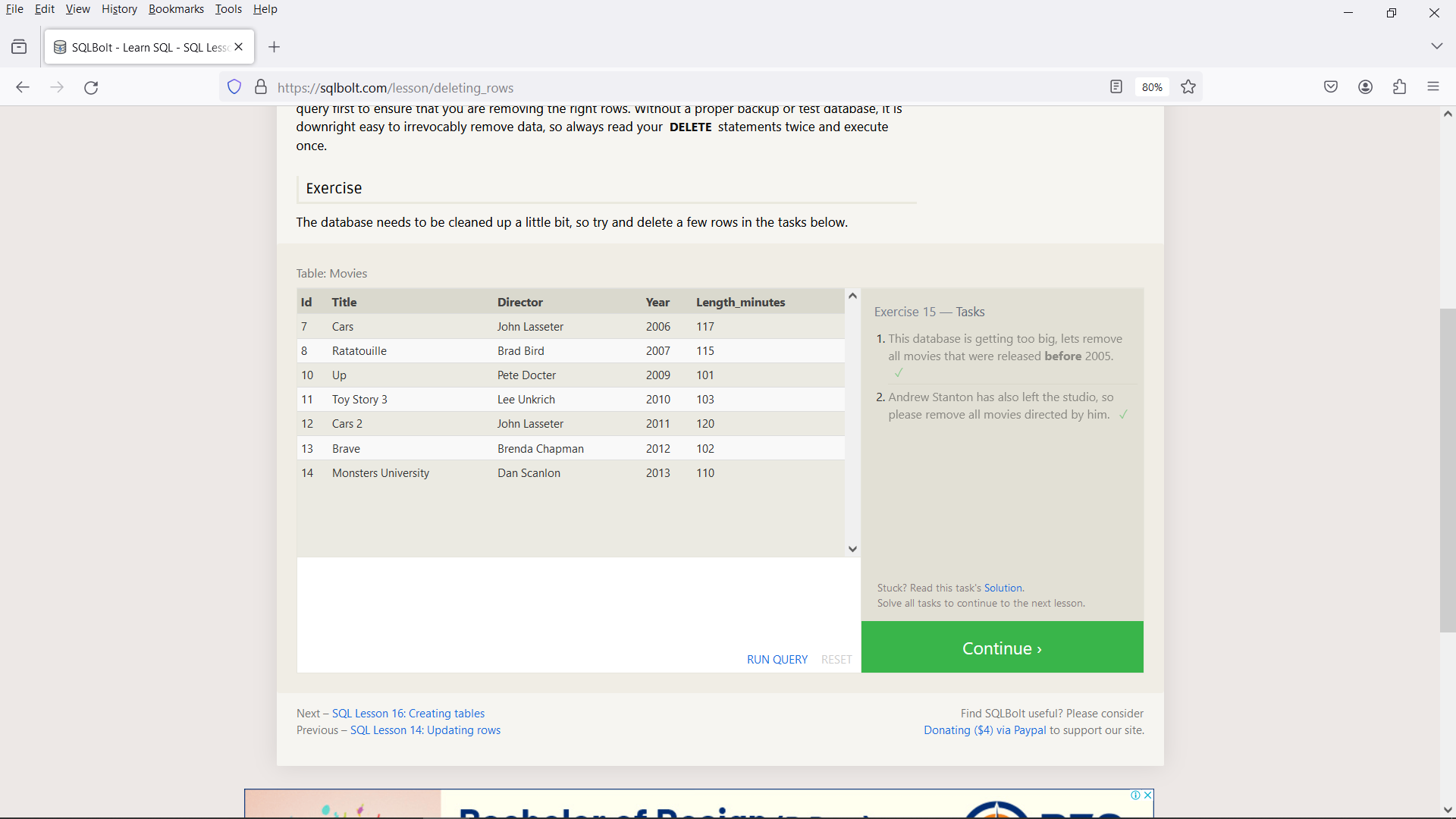
Exercise-14:

1. UPDATE Movies SET Director = "John Lasseter" WHERE Id=2 AND Title="A Bug's Life";
2. UPDATE Movies SET Year=1999 WHERE Id=3 AND Title="Toy Story 2";
3. UPDATE Movies SET Title="Toy Story 3", Director="Lee Unkrich" WHERE Id=11 AND Title="Toy Story 8";



Exercise-15:

1. DELETE FROM Movies WHERE Year<2005;
2. DELETE FROM Movies WHERE Director="Andrew Stanton";



Exercise-16:

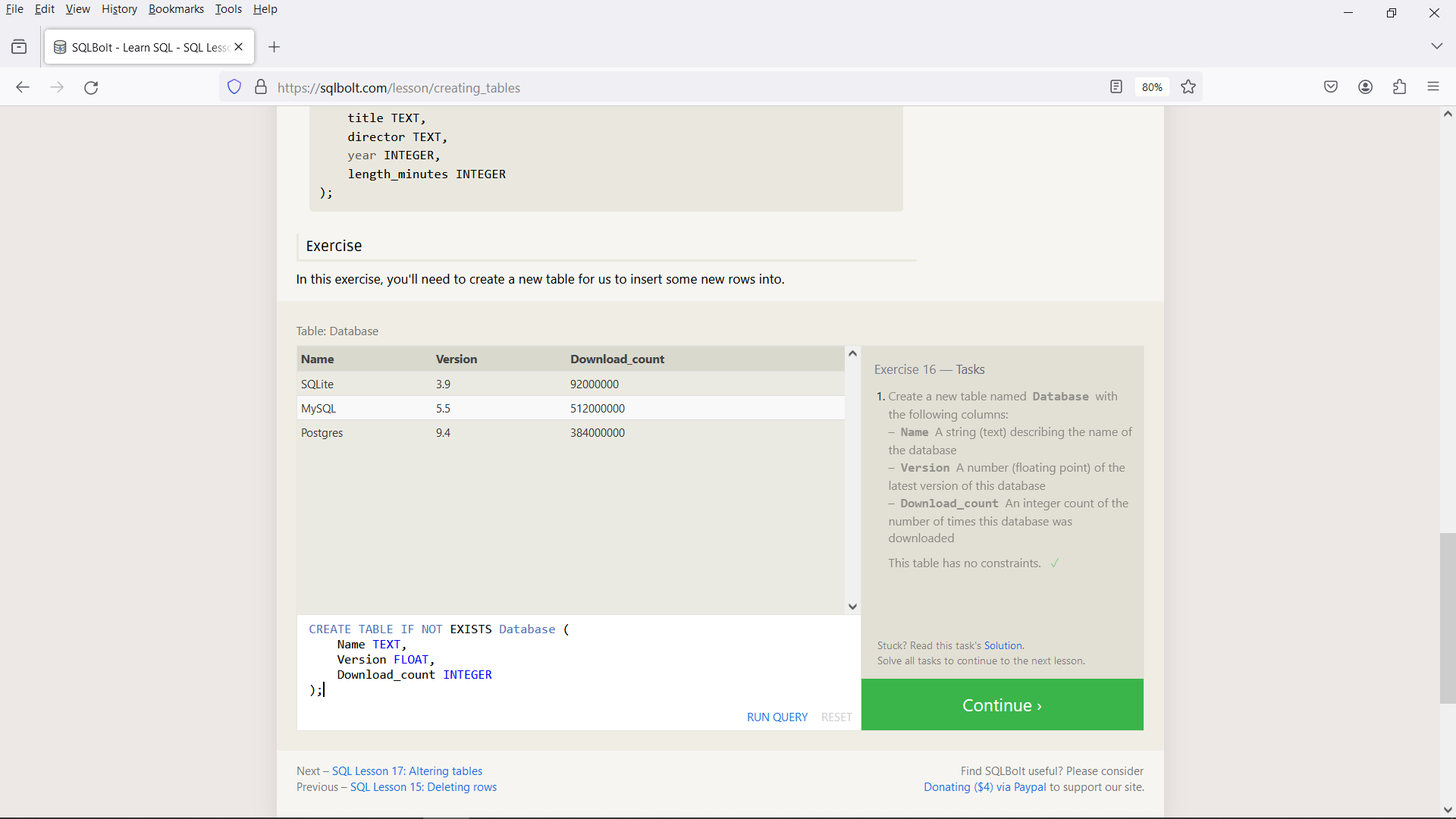
CREATE TABLE IF NOT EXISTS Database (

Name TEXT,

Version FLOAT,

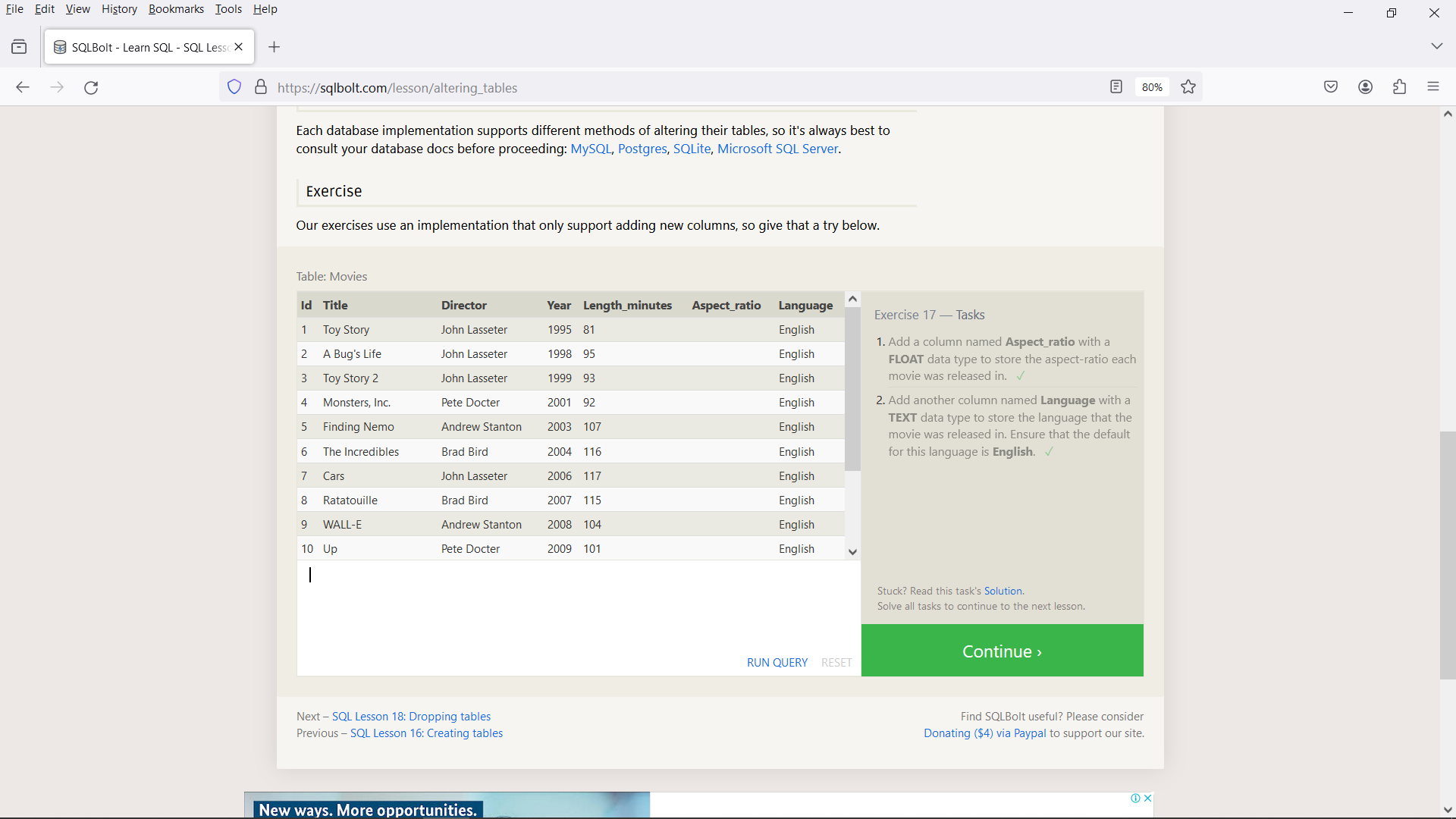
Download\_count INTEGER

);



Exercise-17:

1. ALTER TABLE Movies ADD Aspect\_ratio FLOAT;
2. ALTER TABLE Movies ADD Language TEXT DEFAULT English;



Exercise-18:

1. DROP TABLE Movies;
2. DROP TABLE Boxoffice;

