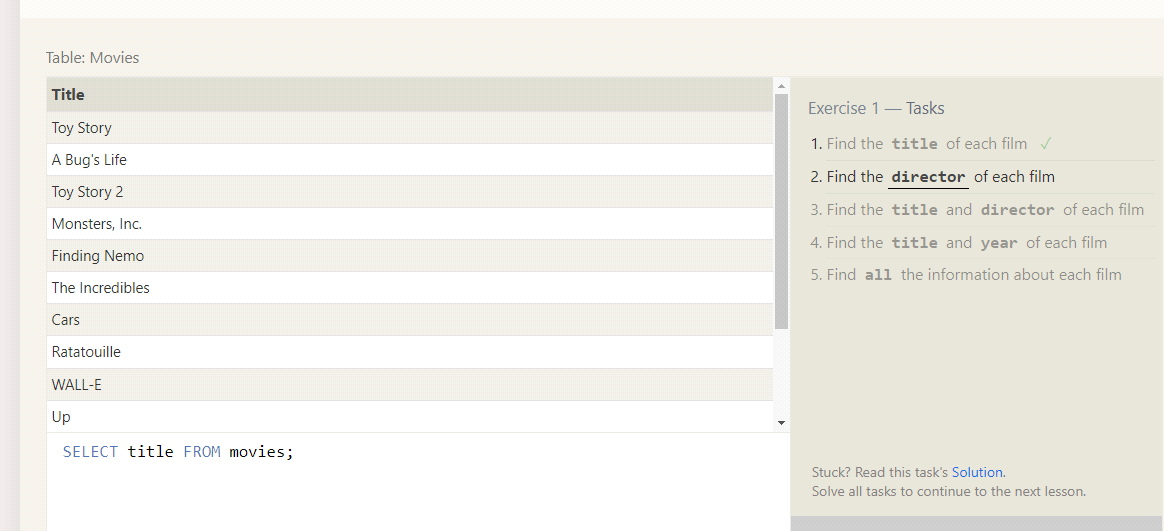
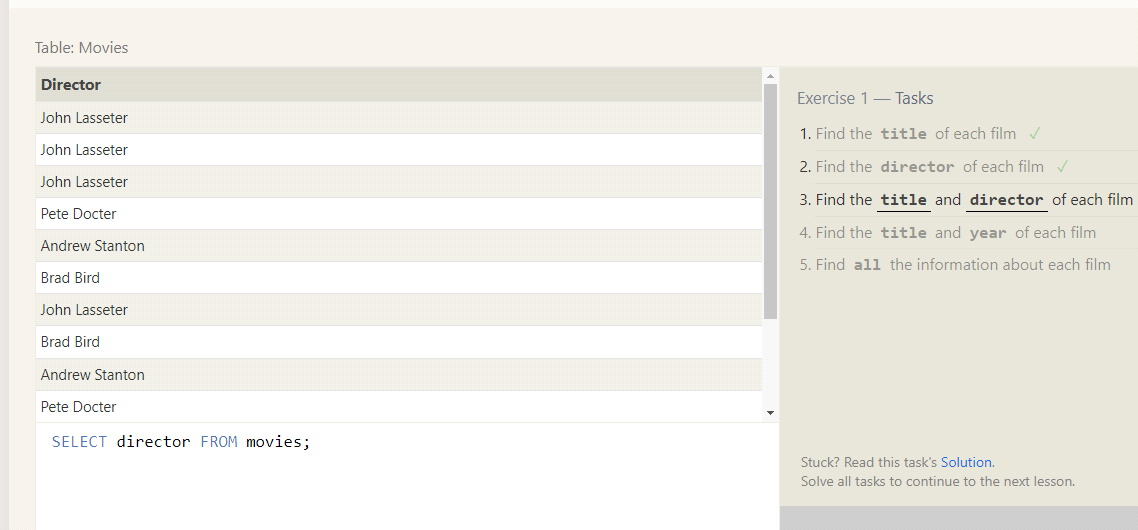
**My SQL Task**

**Task1:Select from queries**

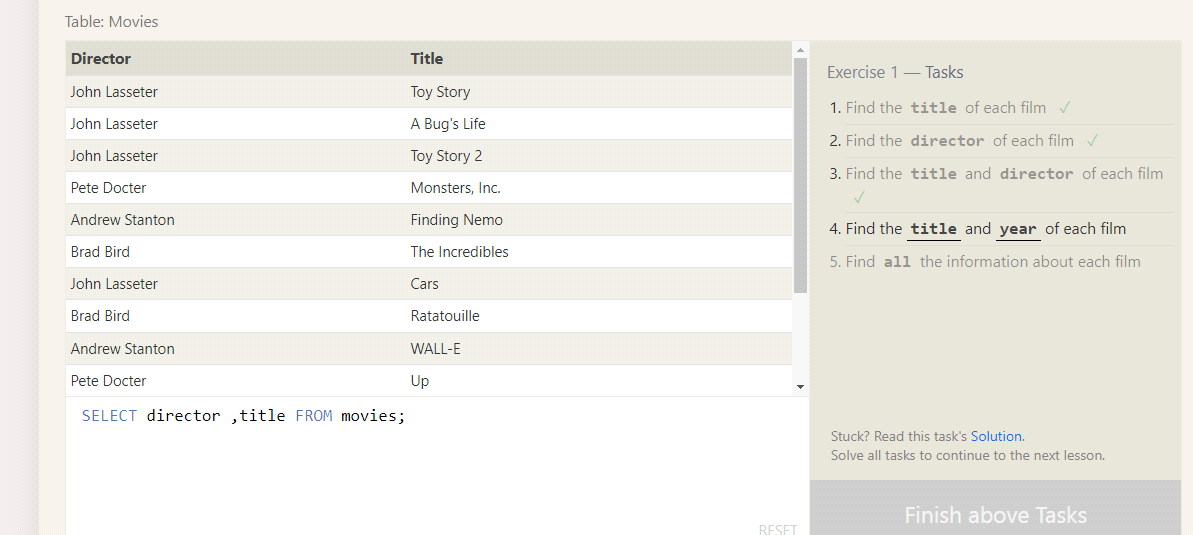
**a.Title from movies**

****

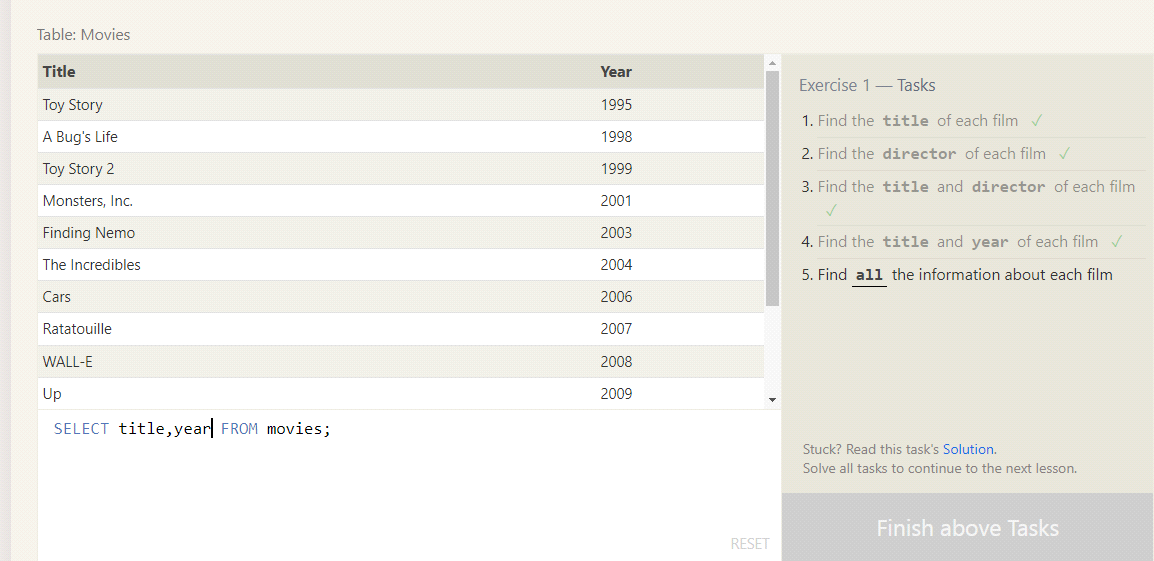
**b.Director of each film**

****

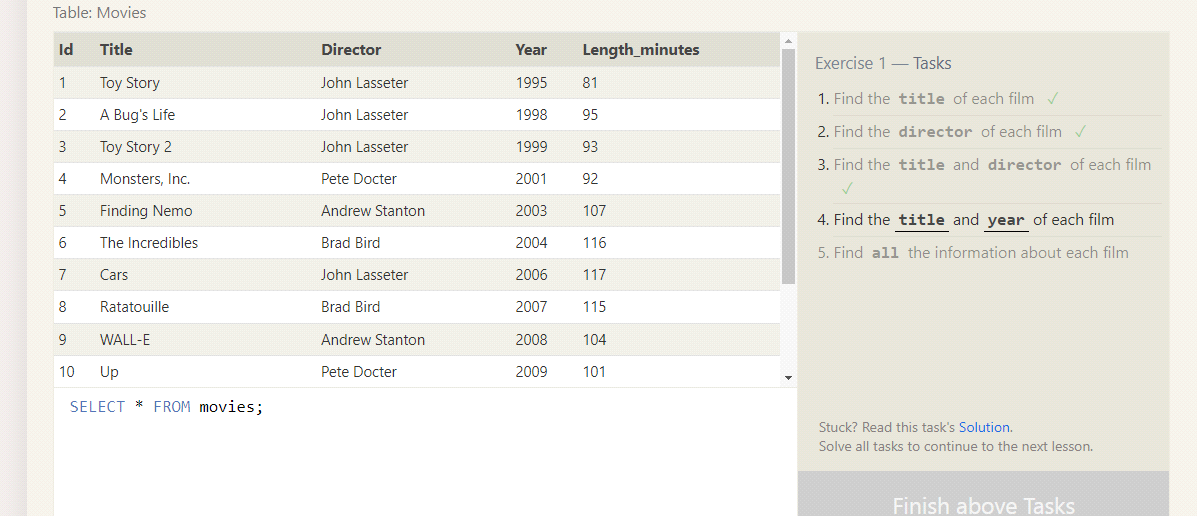
**c.Title and Director**

****

**d.Title and year**

****

**e.Find all info**

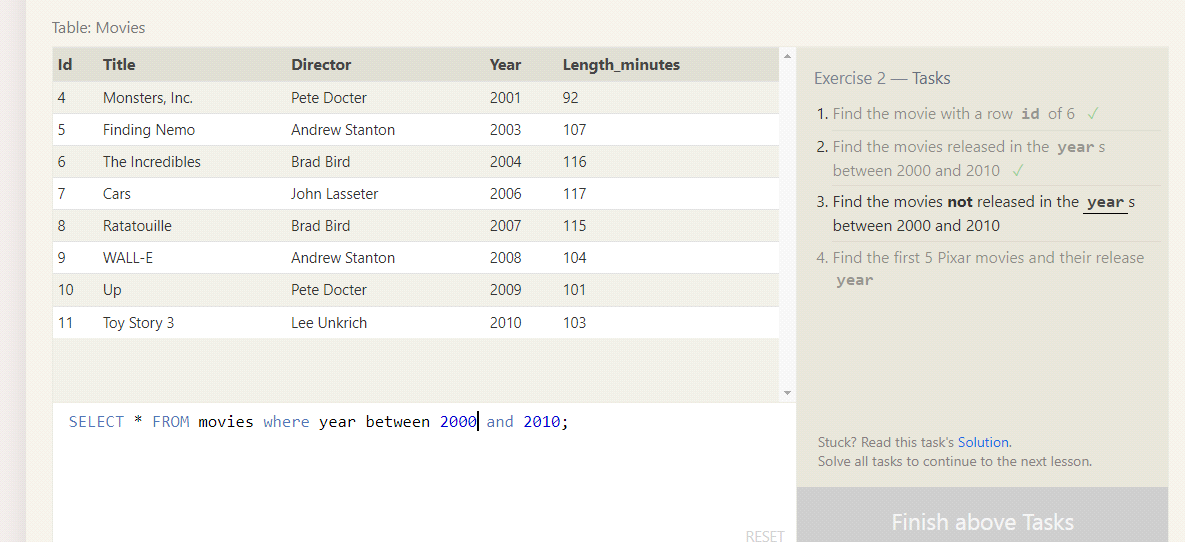
****

**2.Task2:Queries with constraints(Pt-1)**

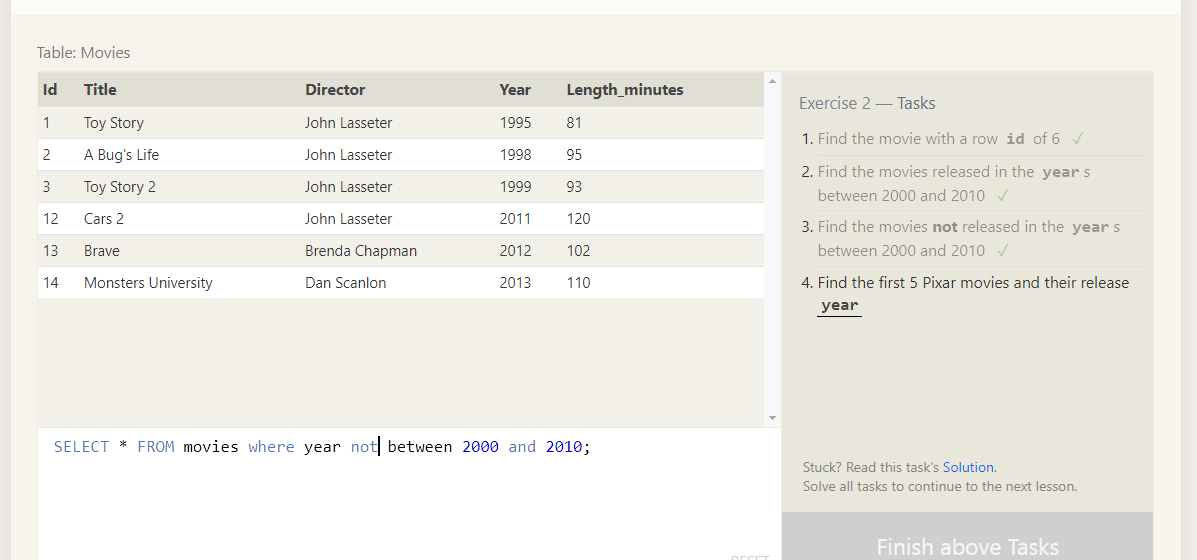
**a.Movie with id=6**

****

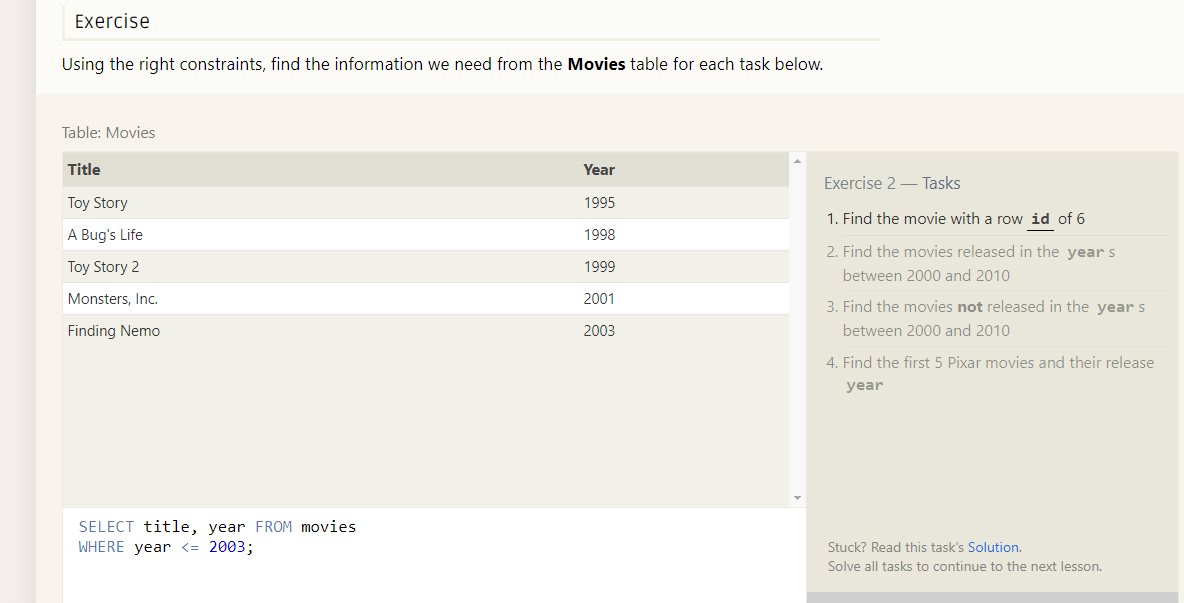
**b.Movies released in year bt 2000 and 2010**

****

**c.Movies not released bt 2000 and 2010**

****

**d.First pixar movies and years**

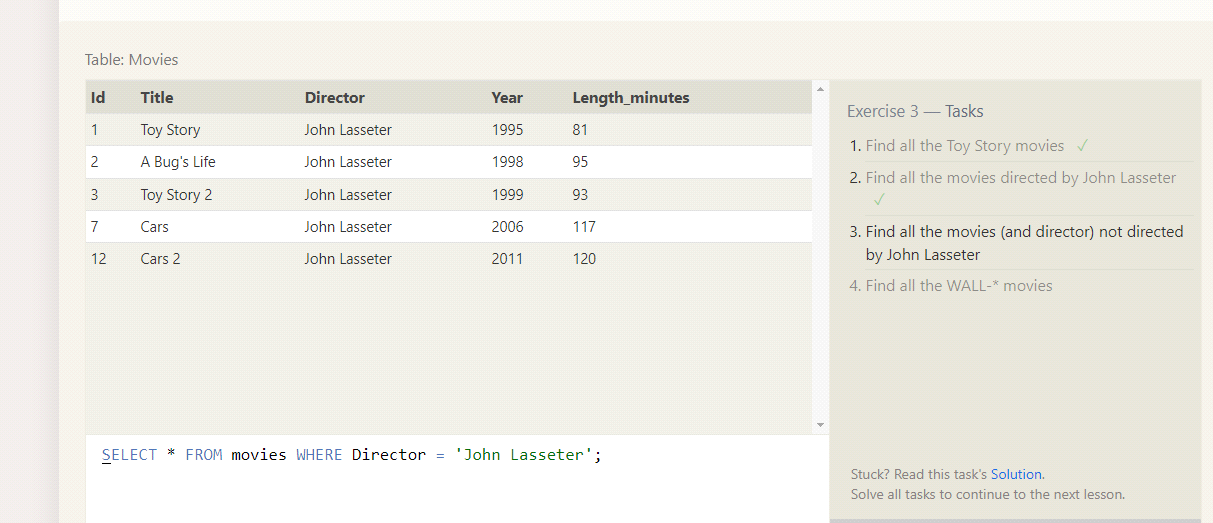
****

**3.Task 3:Queries with constraints(pt-2)**

**a.All Toy story movies**

****

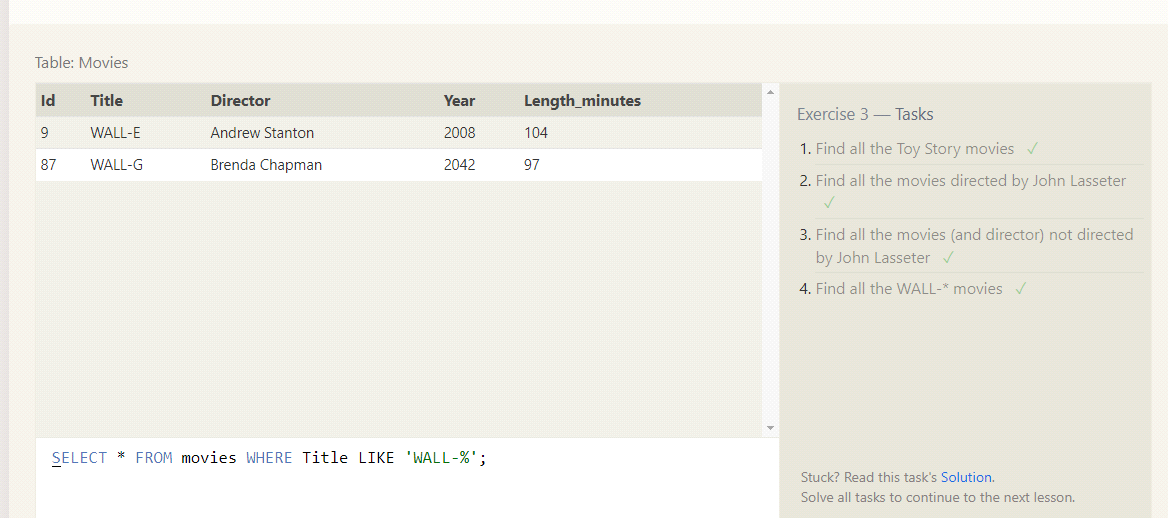
**b.Movies directed by John Lasseter**

****

**c.Movies not directed by John Lasseter**

****

**d.Movies with Wall-\***

****

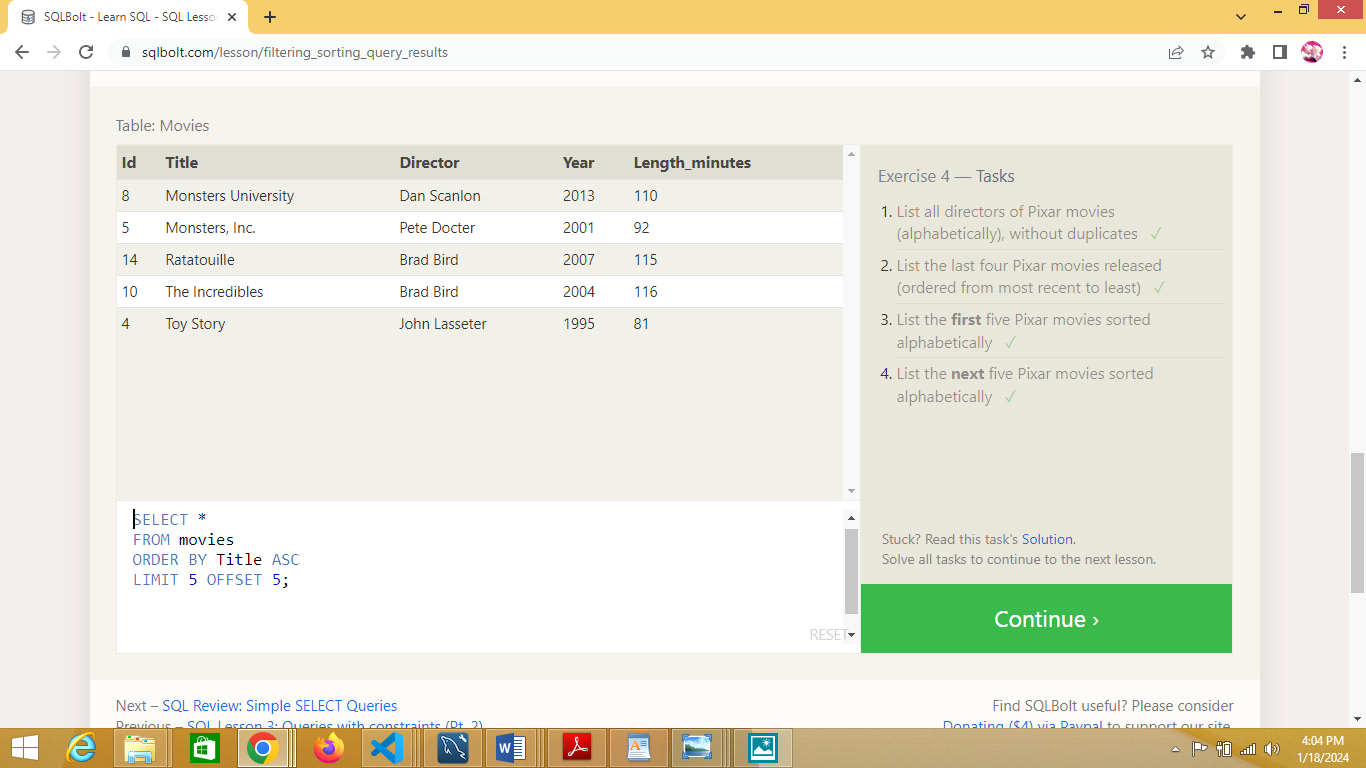
**4.Task 4:Filtering and sorting query results**

**a.Directors of pixar movies without duplicates**

**b.SELECT \*FROM movies ORDER BY Year DESC LIMIT 4;**

**c.SELECT \*FROM movies ORDER BY Title ASC LIMIT 5;**

**d.SELECT \*FROM movies ORDER BY Title ASC LIMIT 5 OFFSET 5;**

****

**5.Task :Simple select queries**

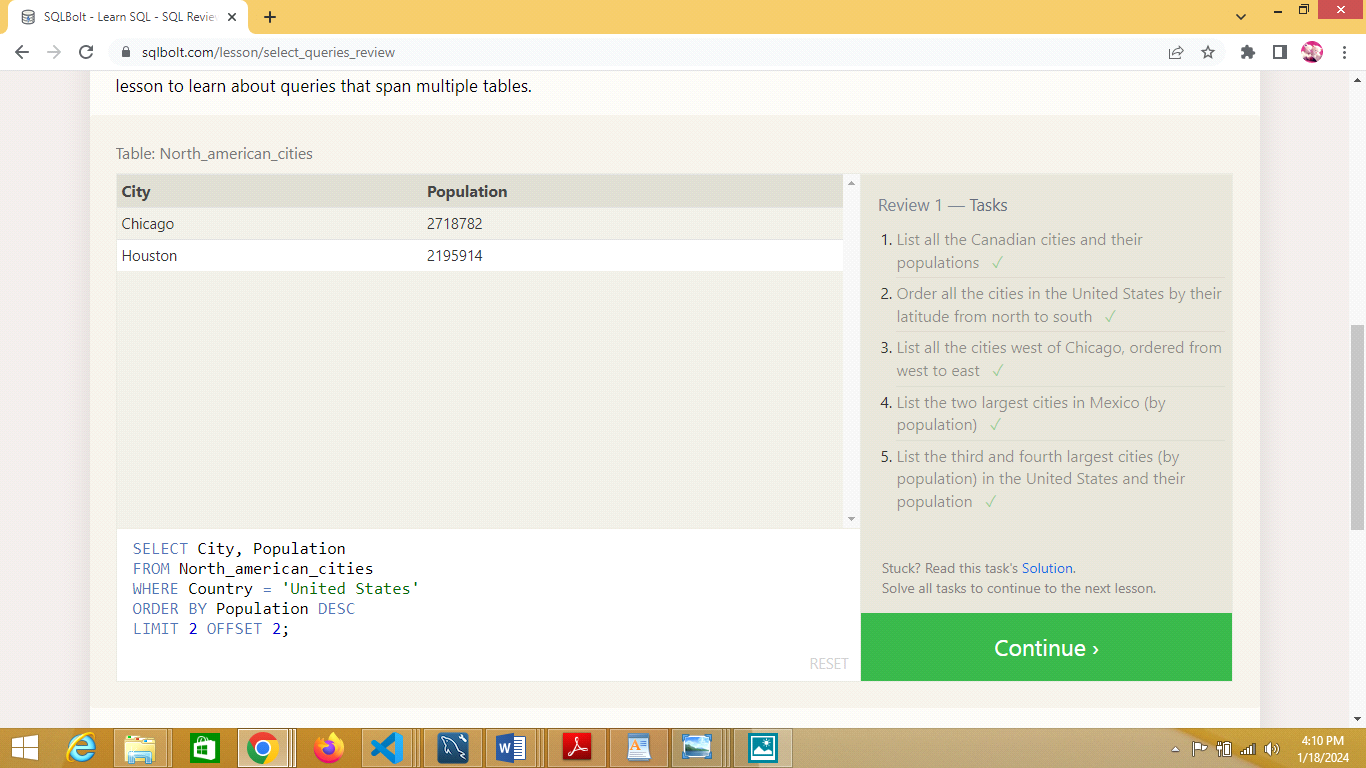
**a.SELECT City, Population FROM North\_american\_cities WHERE Country = 'Canada';**

**b.SELECT City, Latitude FROM North\_american\_cities WHERE Country = 'United States' ORDER BY Latitude DESC;**

**c.SELECT City, Longitude FROM North\_american\_cities WHERE Longitude < (SELECT Longitude FROM North\_american\_cities WHERE City = 'Chicago' ORDER BY Longitude ASC;**

**d.SELECT City, Population FROM North\_american\_cities WHERE Country = 'Mexico' ORDER BY Population DESC LIMIT 2;**

**e.SELECT City, Population FROM North\_american\_cities WHERE Country = 'United States' ORDER BY Population DESC LIMIT 2 OFFSET 2;**

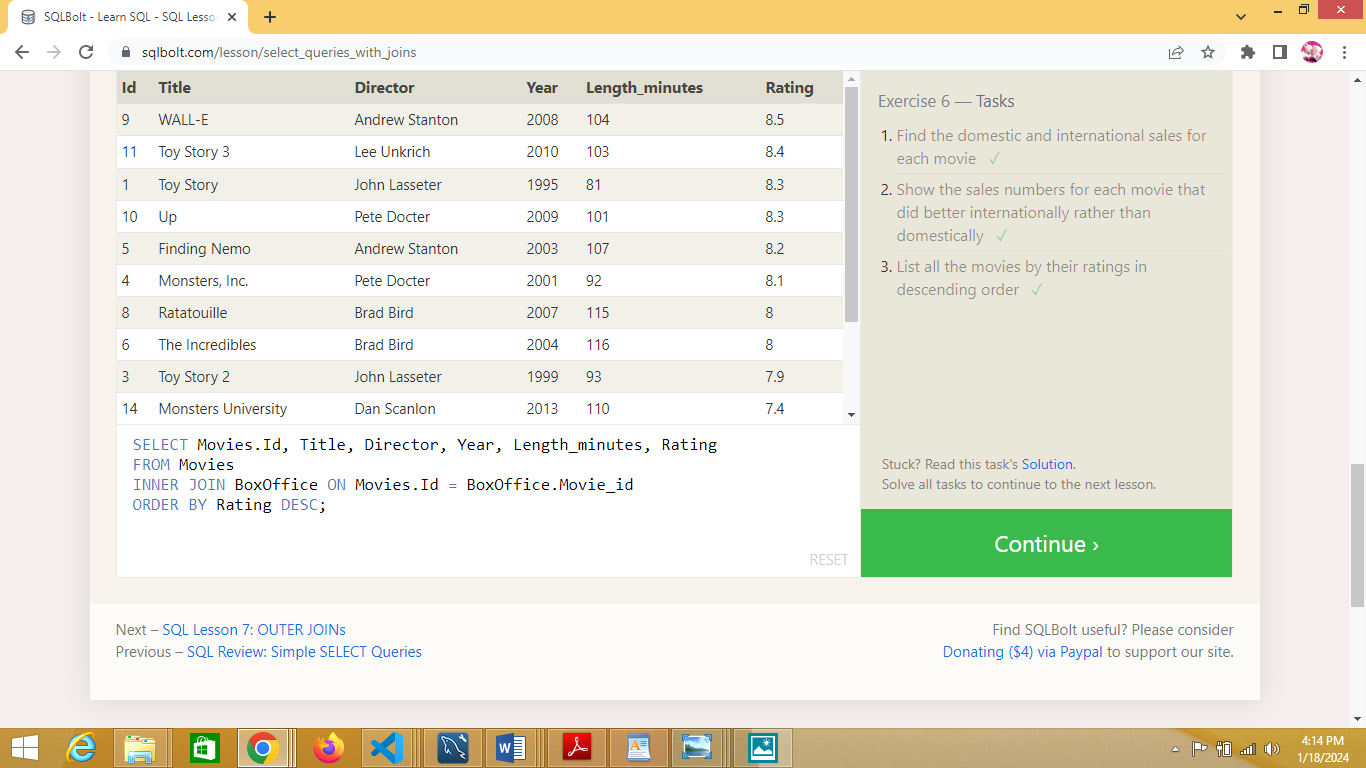
****

**Task 6:Multi-table queries with JOINS**

**a.SELECT Movies.Id, Title, Director, Year, Length\_minutes, Domestic\_sales, International\_sales FROM Movies INNER JOIN BoxOffice ON Movies.Id = BoxOffice.Movie\_id;**

**b.SELECT Movies.Id, Title, Director, Year, Length\_minutes, Domestic\_sales, International\_sales FROM Movies INNER JOIN BoxOffice ON Movies.Id = BoxOffice.Movie\_id WHERE International\_sales > Domestic\_sales;**

**c.SELECT Movies.Id, Title, Director, Year, Length\_minutes, Rating FROM Movies INNER JOIN BoxOffice ON Movies.Id = BoxOffice.Movie\_id ORDER BY Rating DESC;**

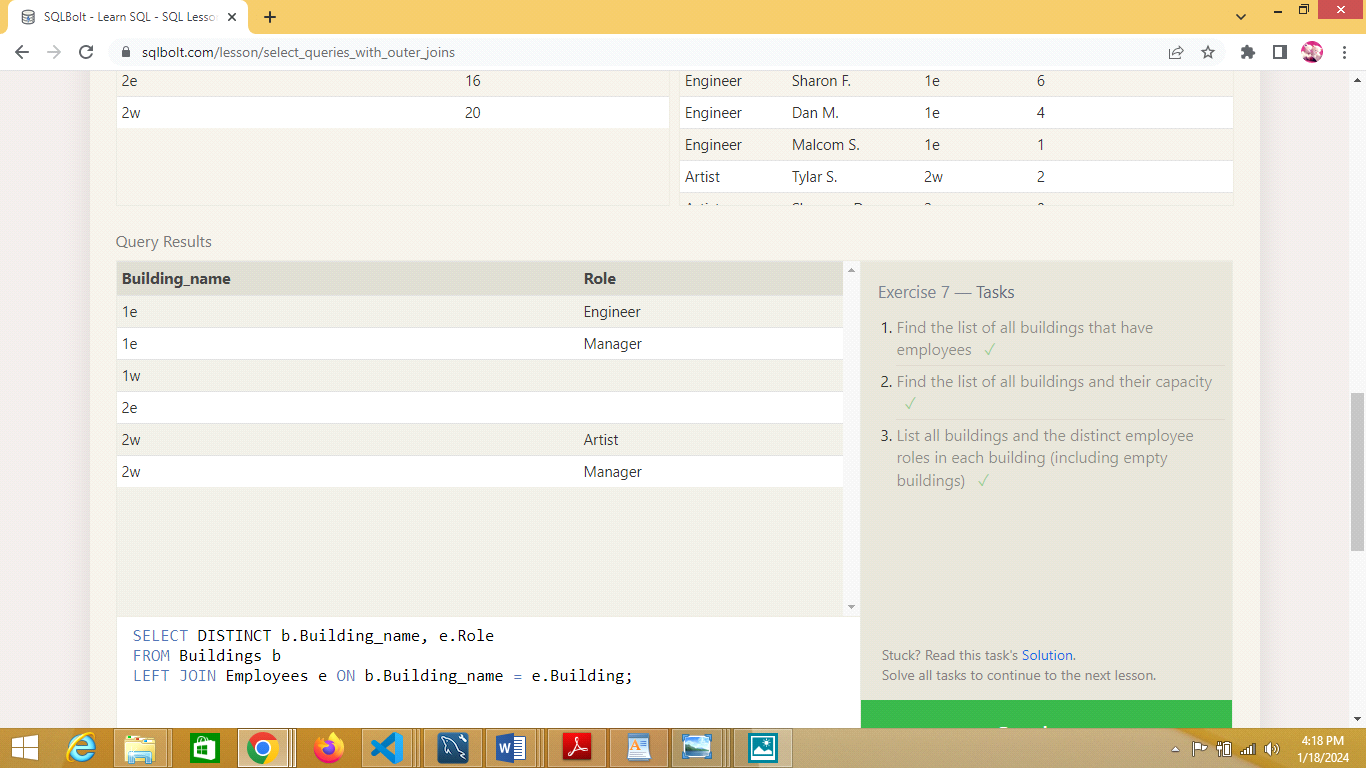
****

**Task 7:Outer JOINS**

**a.SELECT DISTINCT e.Building FROM Employees e LEFT JOIN Buildings b ON e.Building = b.Building\_name WHERE e.Building IS NOT NULL;**

**b.SELECT b.Building\_name, b.Capacity FROM Buildings b LEFT JOIN Employees e ON b.Building\_name = e.Building GROUP BY b.Building\_name, b.Capacity;**

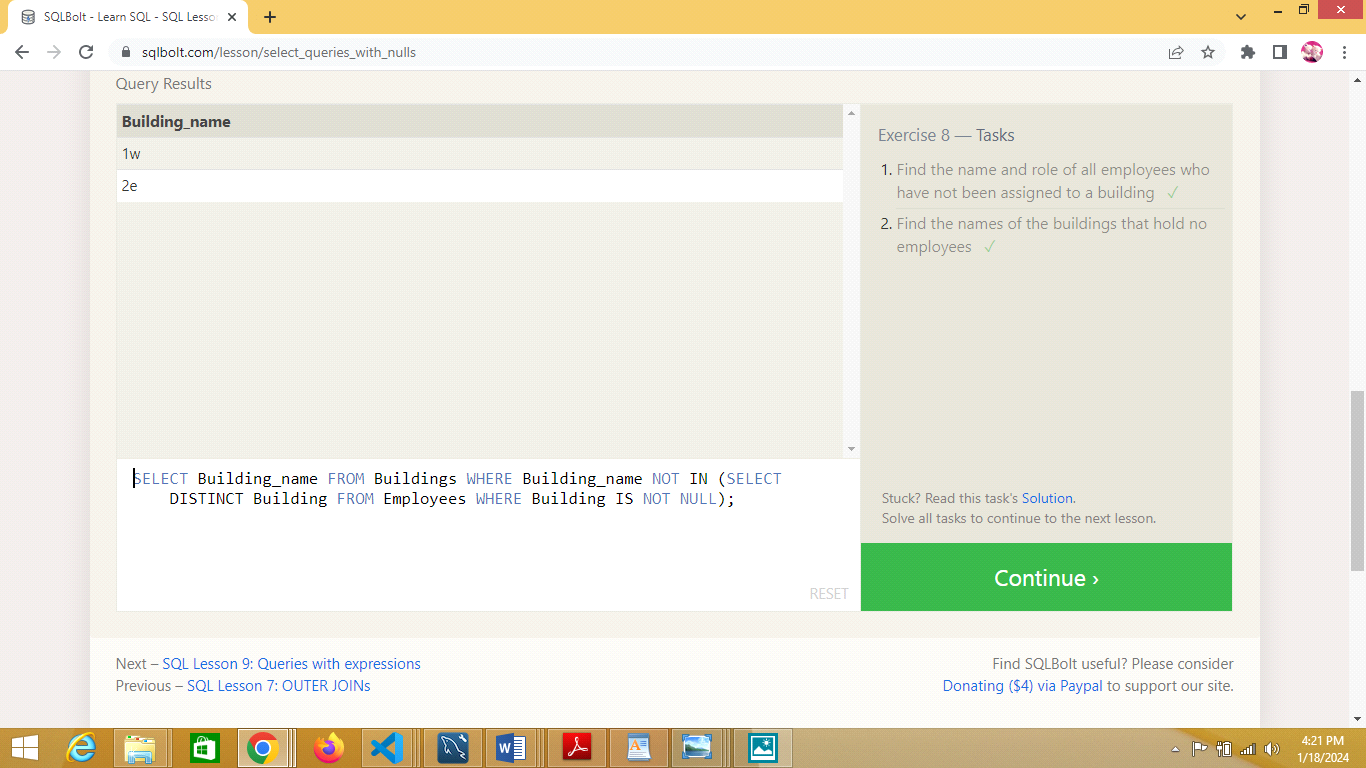
**c.SELECT DISTINCT b.Building\_name, e.Role FROM Buildings LEFT JOIN Employees e ON b.Building\_name = e.Building;**

****

**Task 8:Short note on NULLs**

**a.SELECT Name, Role FROM Employees WHERE Building IS NULL;**

**b.SELECT Building\_name FROM Buildings WHERE Building\_name NOT IN (SELECT DISTINCT Building FROM Employees WHERE Building IS NOT NULL);**

****

**Task 9:Queries with expressions**

**a.SELECT**

**m.Id,**

**m.Title,**

**m.Director,**

**m.Year,**

**m.Length\_minutes,**

**(b.Domestic\_sales + b.International\_sales) / 1000000 AS combined\_sales\_millions FROM Movies m JOINBoxoffice b ON m.Id = b.Movie\_id;**

**b.SELECT**

**m.Id,**

**m.Title,**

**m.Director,**

**m.Year,**

**m.Length\_minutes,**

**b.Rating \* 10 AS rating\_percent FROM Movies m JOIN Boxoffice b ON m.Id = b.Movie\_id;**

**c.SELECT**

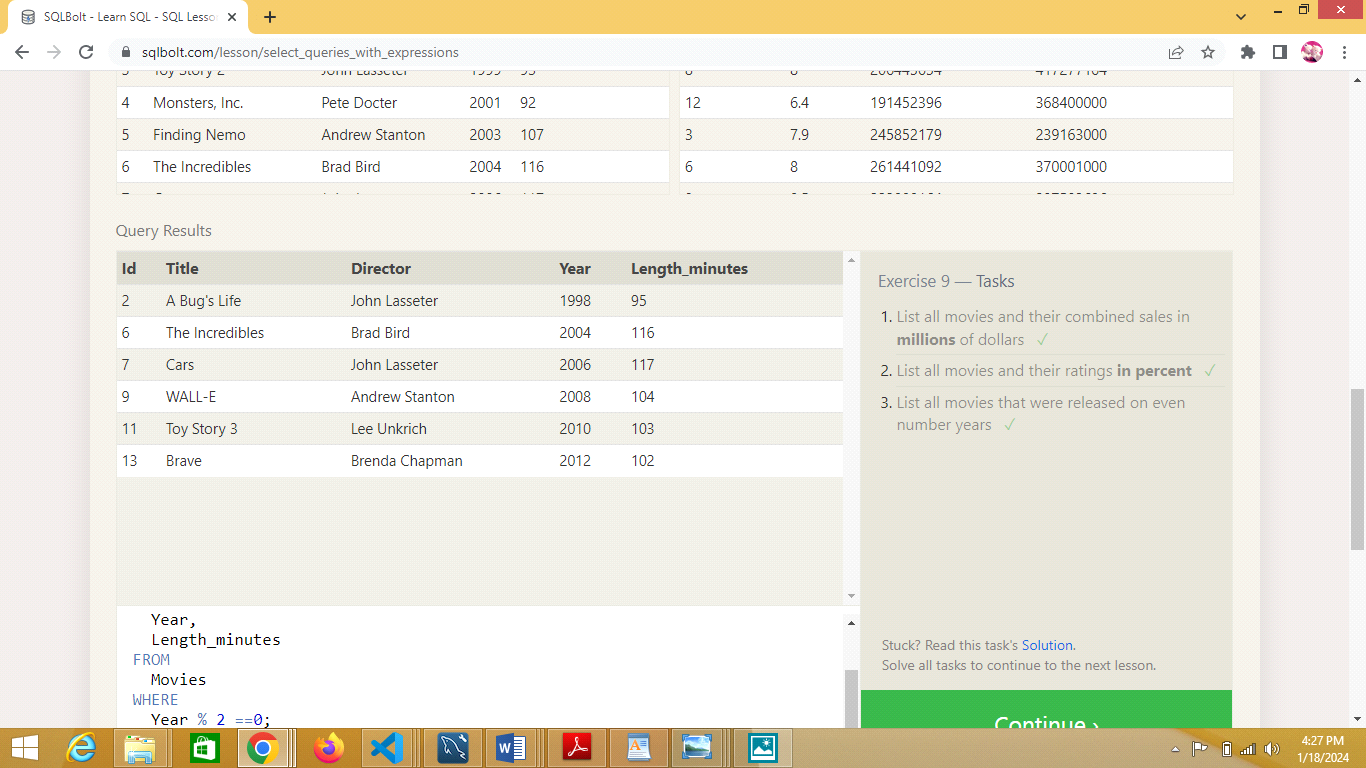
**Id,**

**Title,**

**Director,**

**Year,**

**Length\_minutes FROM Movies WHERE Year % 2 = =0;**

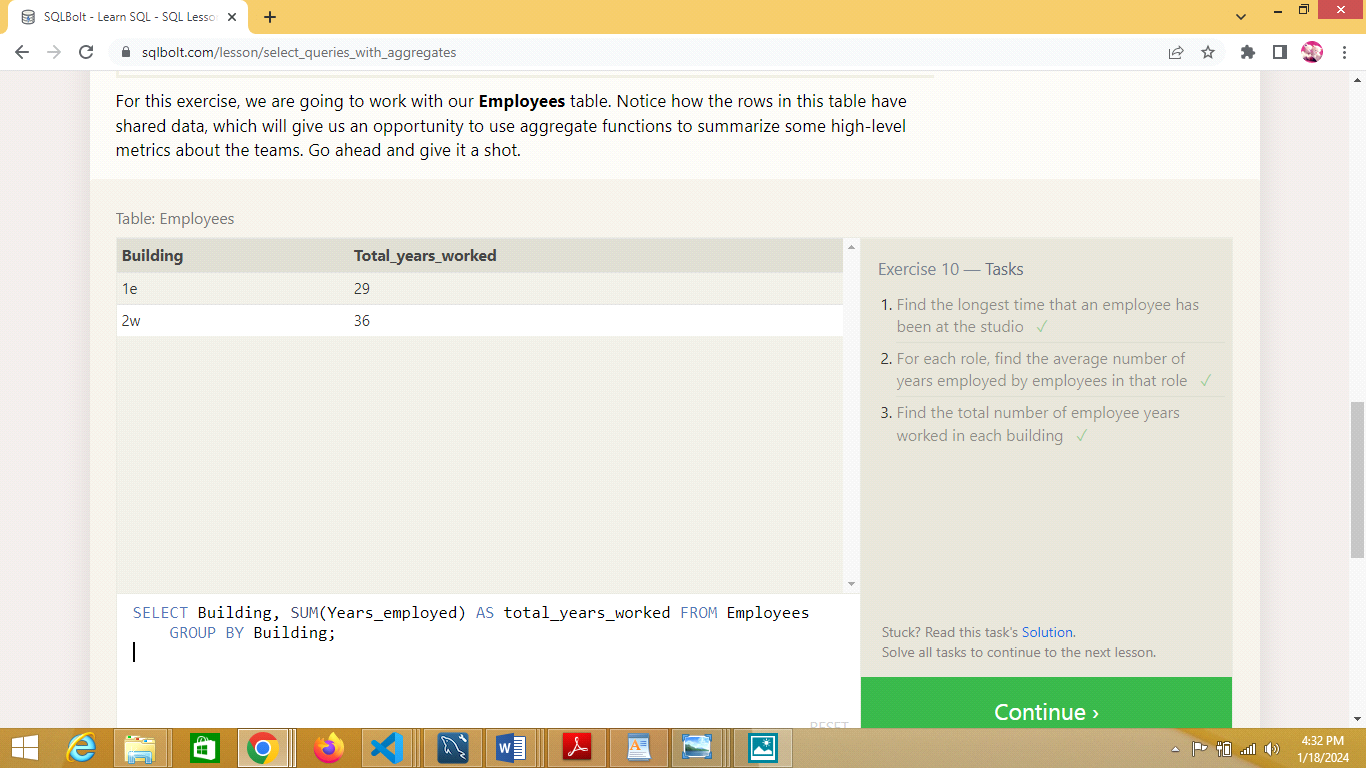
****

**Task 10:Queries with aggregates(pt-1)**

**a.SELECT MAX(Years\_employed) AS longest\_time FROM Employees;**

**b.SELECT Role, AVG(Years\_employed) AS avg\_years\_employed FROM Employees GROUP BY Role;**

**c.SELECT Building, SUM(Years\_employed) AS total\_years\_worked FROM Employees GROUP BY Building;**

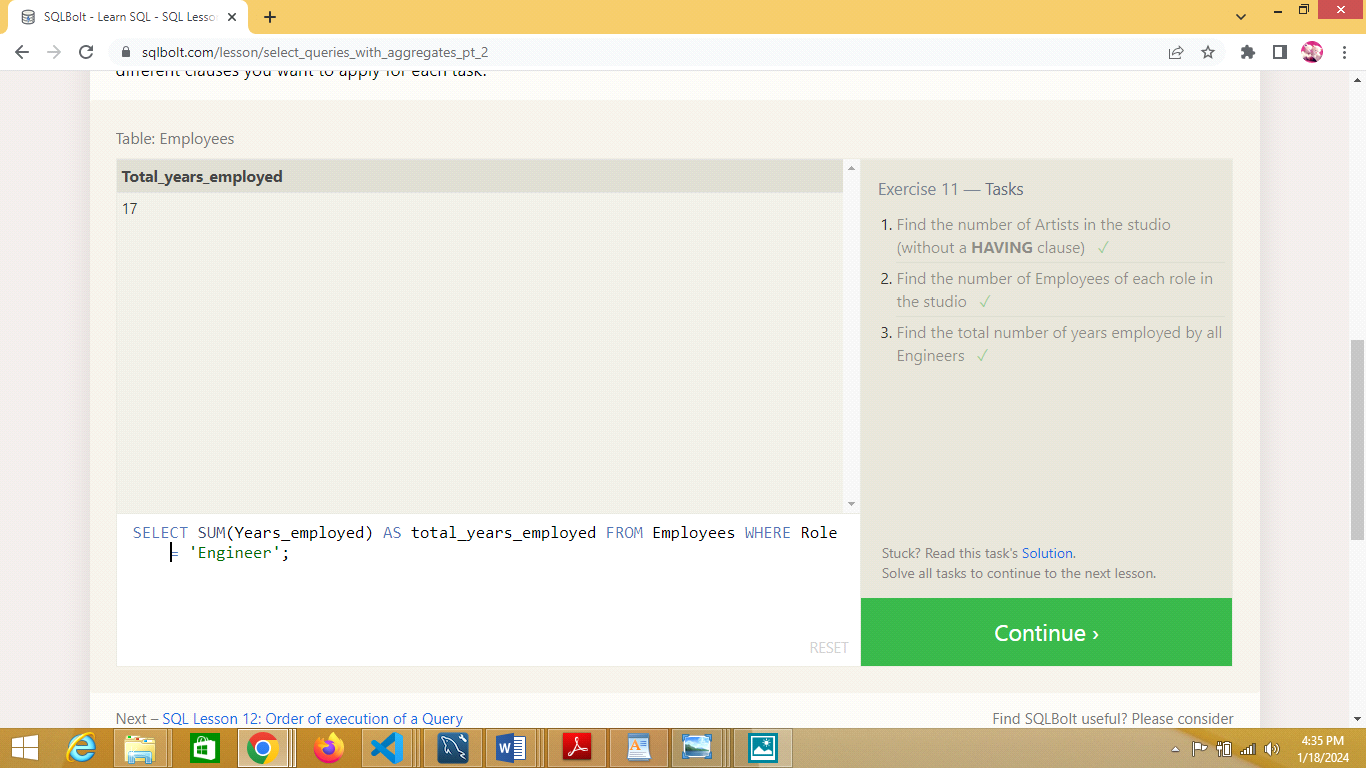
****

**Task 11:Queries with aggregates(pt-2)**

**a.SELECT COUNT(\*) AS num\_artists FROM Employees WHERE Role = 'Artist';**

**b.SELECT Role, COUNT(\*) AS num\_employees FROM Employees GROUP BY Role;**

**c.SELECT SUM(Years\_employed) AS total\_years\_employed FROM Employees WHERE Role = 'Engineer';**

****

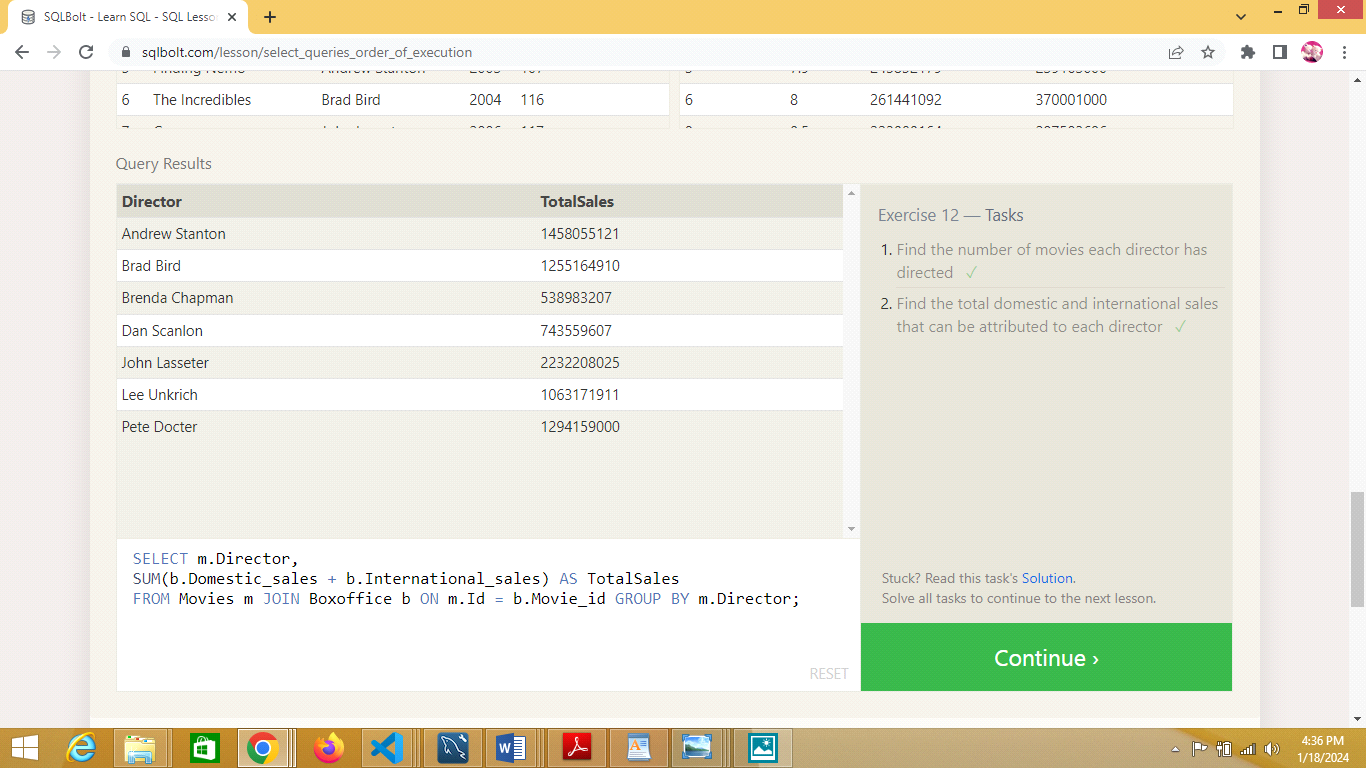
**Task 12:Order of Execution of a Query**

**a.SELECT Director, COUNT(\*) AS NumMoviesDirected FROM Movies GROUP BY Director;**

**b.SELECT m.Director,**

**SUM(b.Domestic\_sales + b.International\_sales) AS TotalSales**

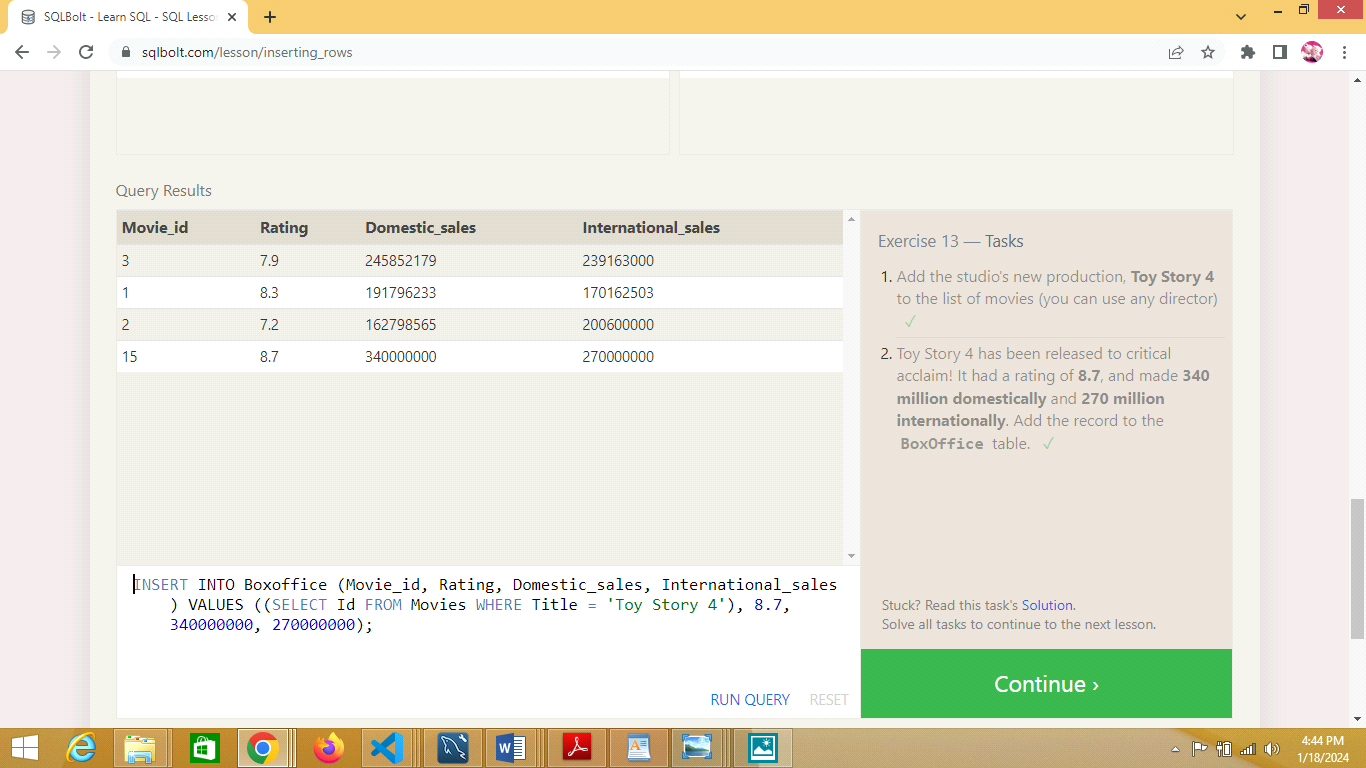
**FROM Movies m JOIN Boxoffice b ON m.Id = b.Movie\_id GROUP BY m.Director;**

****

**Task 13:Inserting rows**

**a.INSERT INTO Movies (Title, Director, Year, Length\_minutes)VALUES ('Toy Story 4', 'Any Director', 2023, 100);**

**b.INSERT INTO Boxoffice (Movie\_id, Rating, Domestic\_sales, International\_sales) VALUES ((SELECT Id FROM Movies WHERE Title = 'Toy Story 4'), 8.7, 340000000, 270000000);**

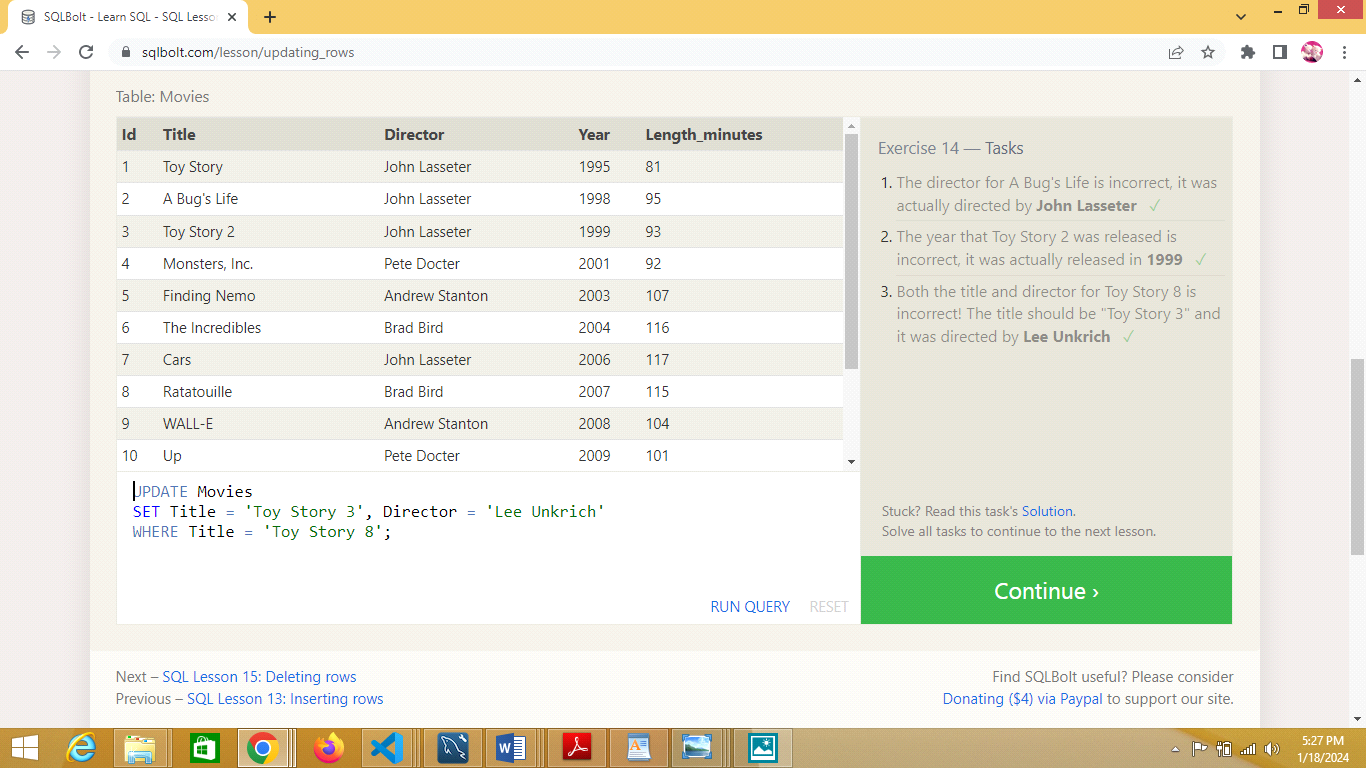
****

**Task 14:Updating rows**

**a.UPDATE MoviesSET Director = 'John Lasseter' WHERE Title = 'A Bug''s Life';**

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

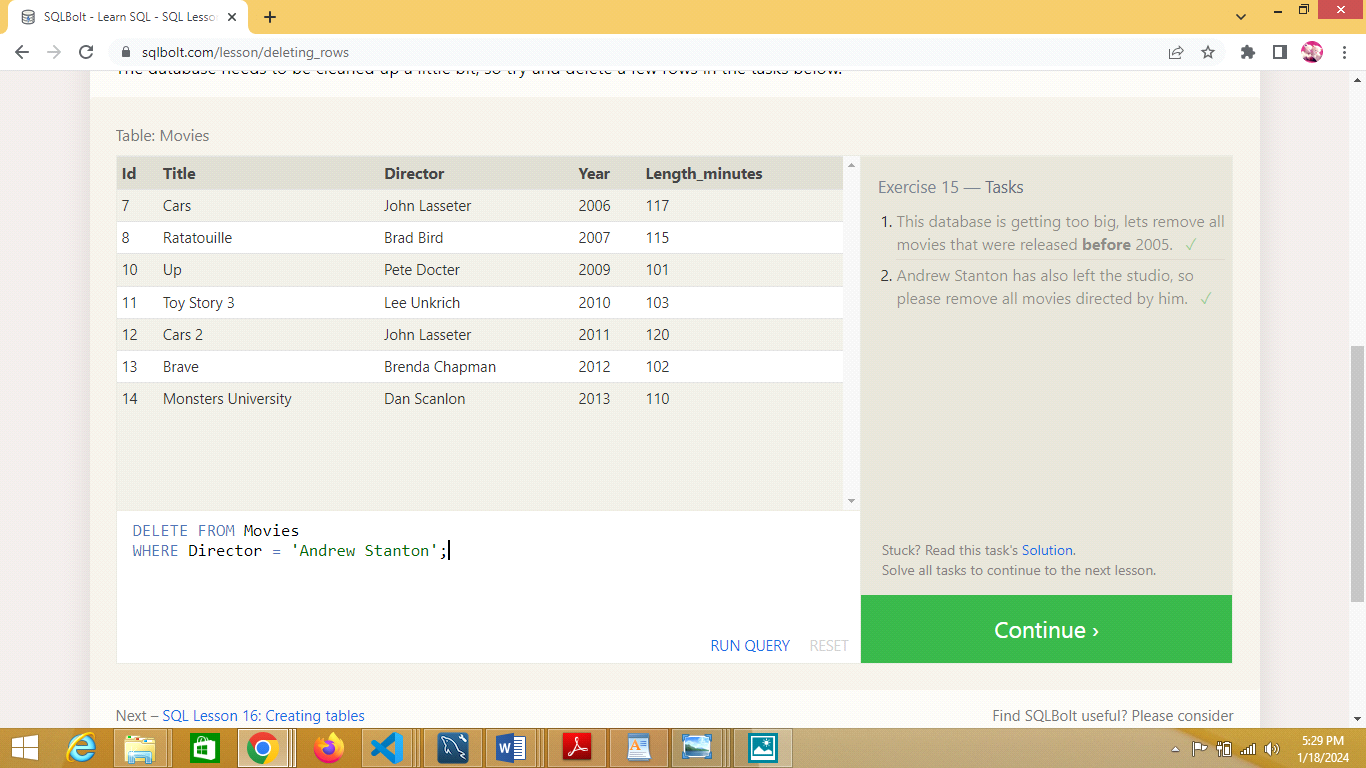
**c.UPDATE Movies SET Title = 'Toy Story 3', Director = 'Lee Unkrich' WHERE Title = 'Toy Story 8';**

****

**Task 15:Deleting rows**

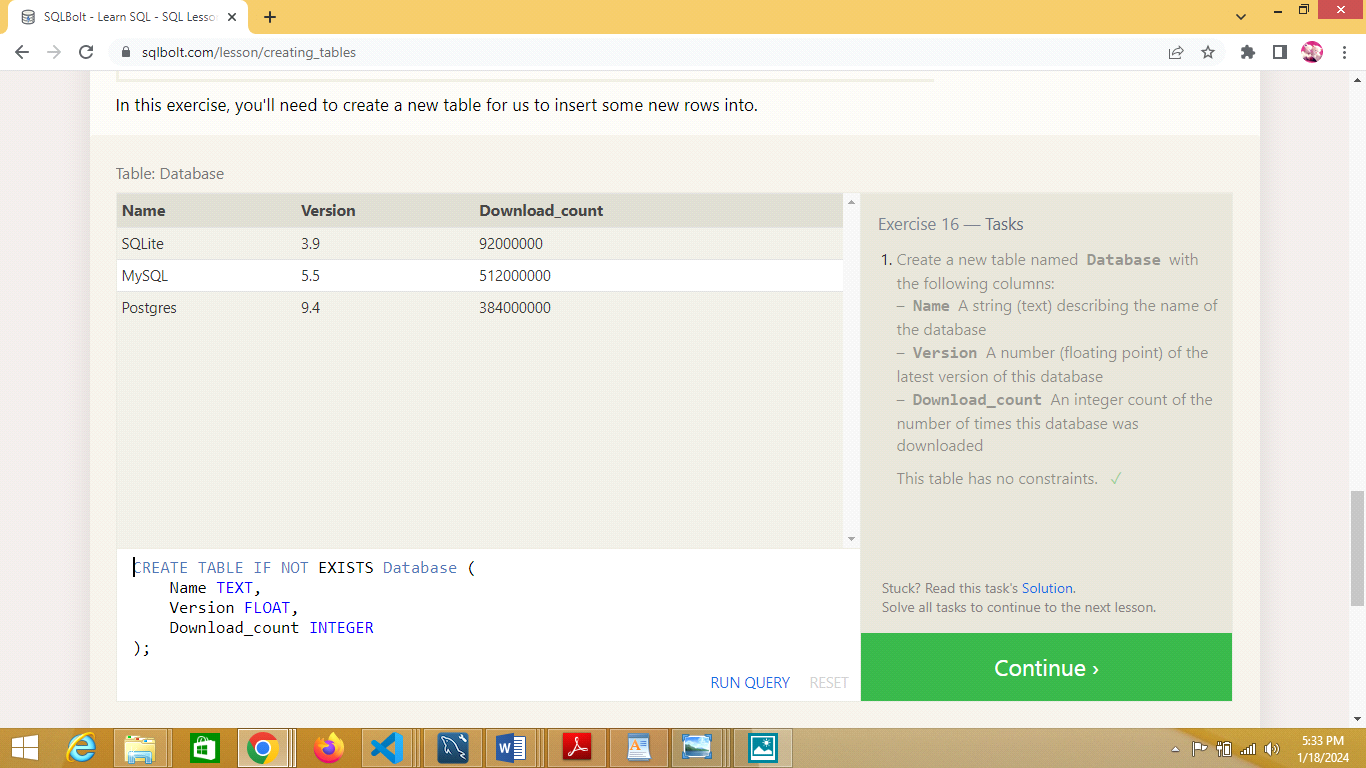
**a.DELETE FROM Movies WHERE Year < 2005;**

**b.DELETE FROM Movies WHERE Director = 'Andrew Stanton';**

****

**Task 16:Creating Tables**

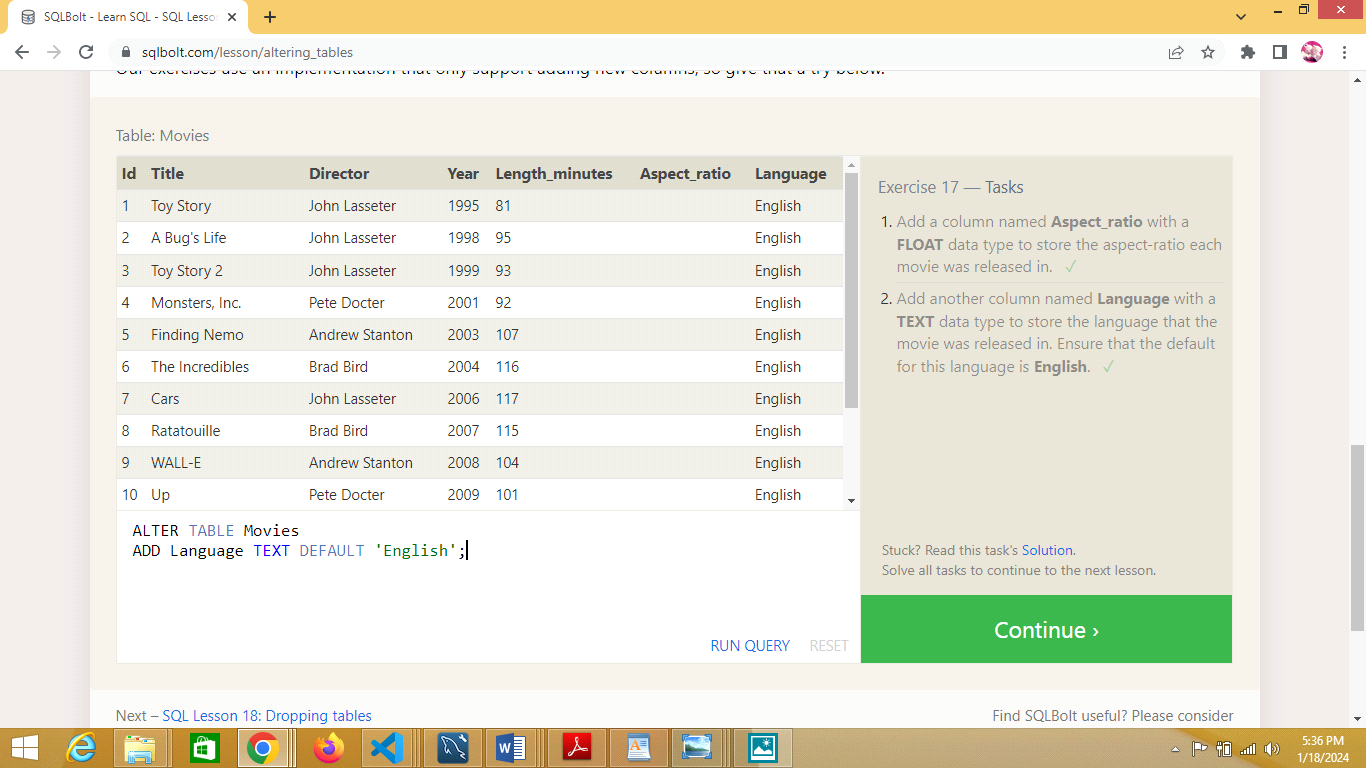
**a.CREATE TABLE IF NOT EXISTS Database ( Name TEXT, Version FLOAT, Download\_count INTEGER );**

****

**Task 17:Altering Tables**

**a.ALTER TABLE Movies ADD Aspect\_ratio FLOAT;**

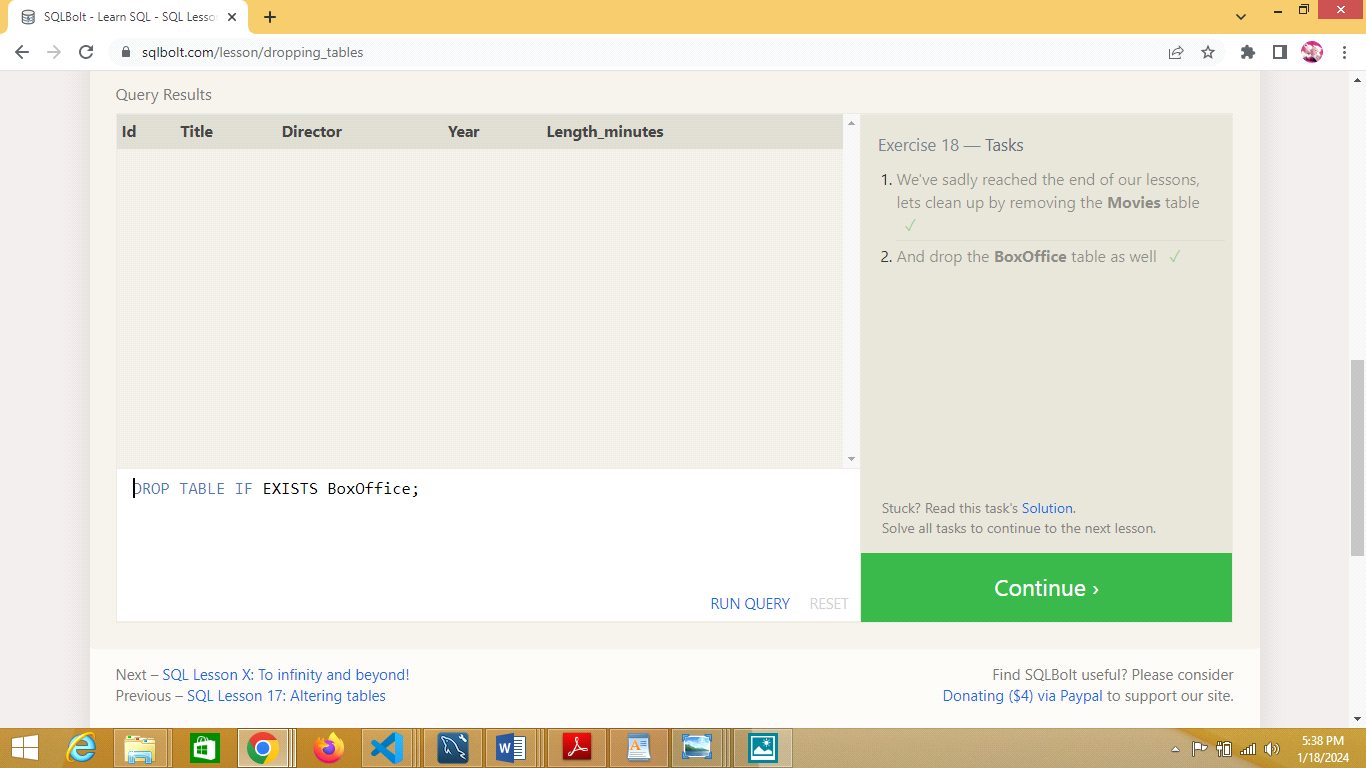
**b.ALTER TABLE Movies ADD Language TEXT DEFAULT 'English';**

****

**Task 18:Dropping Tables**

**a.DROP TABLE IF EXISTS Movies;**

**b.DROP TABLE IF EXISTS BoxOffice;**

****

****