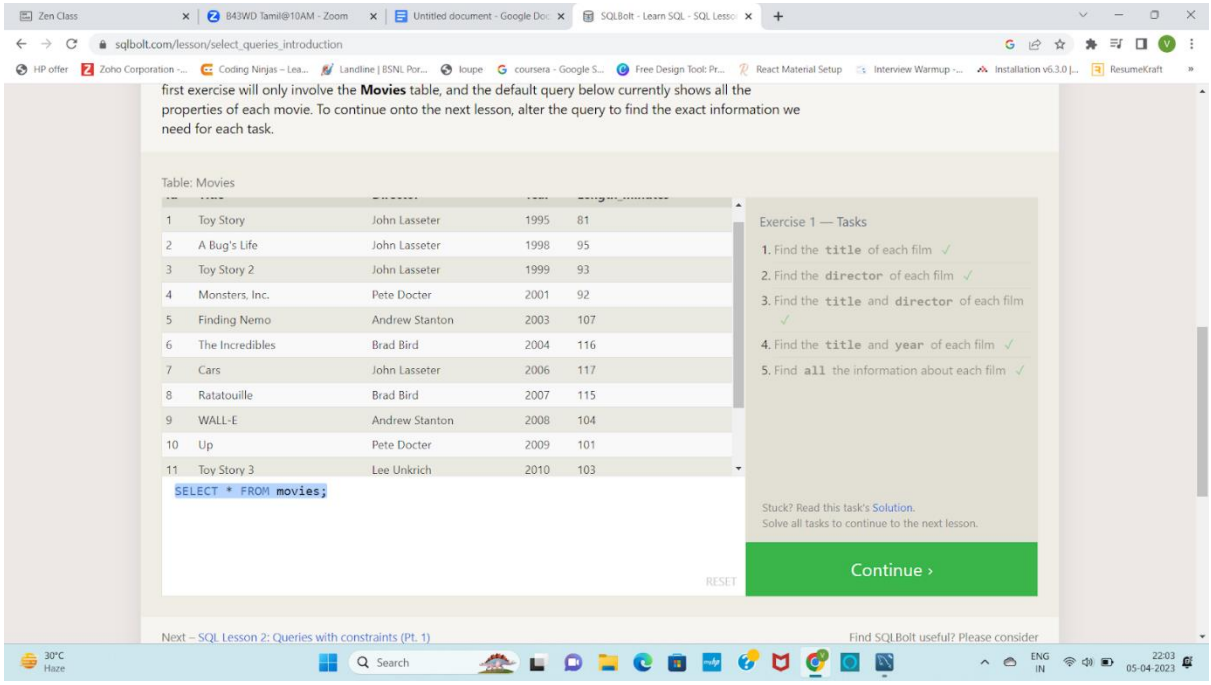
Lesson 1



1.SELECT title FROM movies;

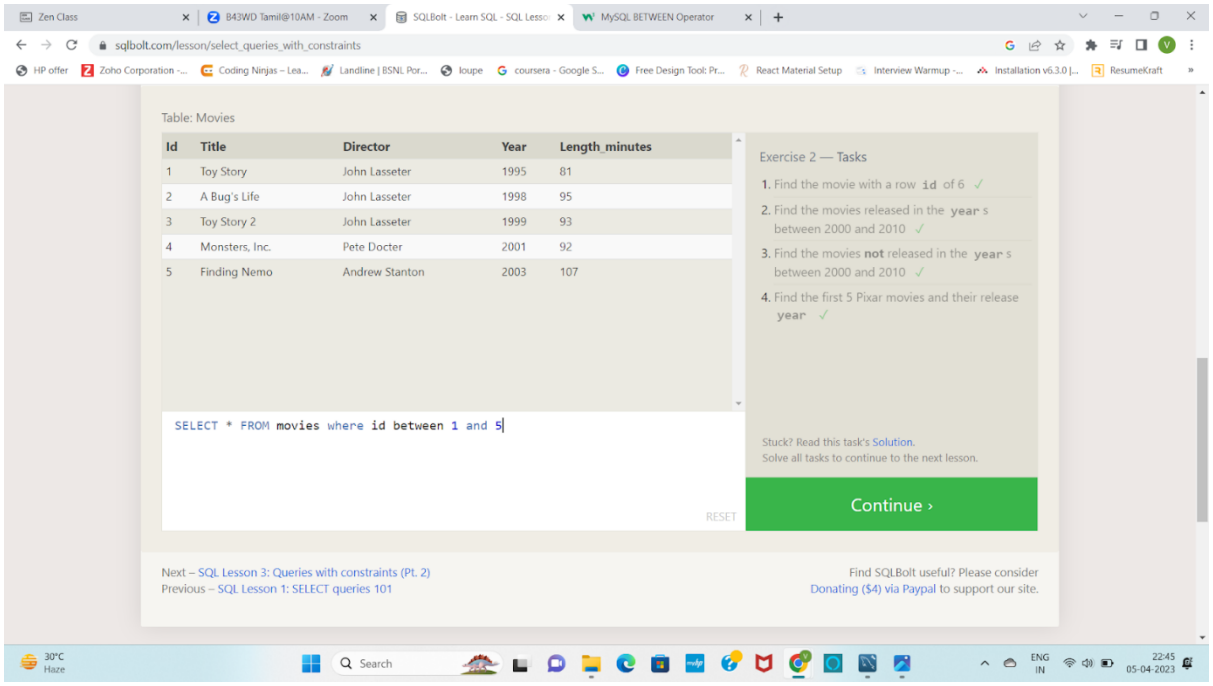
2.SELECT director FROM movies;

3.SELECT title,director FROM movies;

4.SELECT title,year FROM movies;

5.SELECT * FROM movies;

Lesson 2



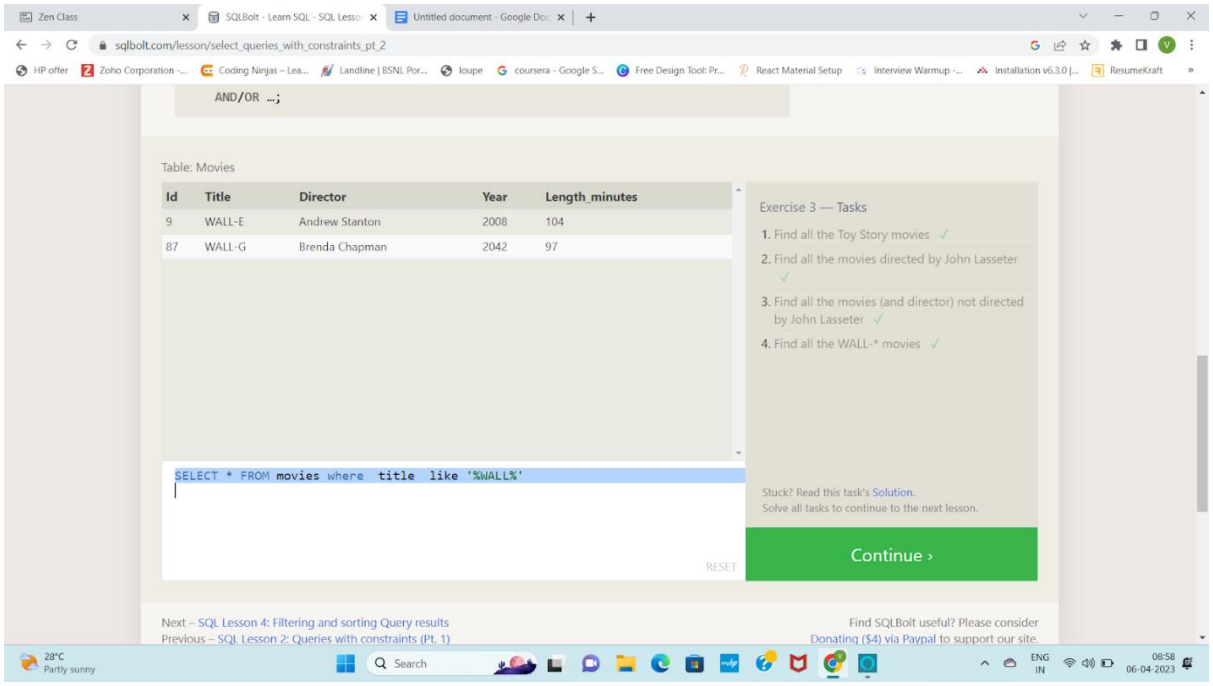
1.SELECT * FROM movies where id=6;

2.SELECT * FROM movies where year between 2000 and 2010

3.SELECT * FROM movies where year not between 2000 and 2010

4.SELECT * FROM movies where id between 1 and 5

Lesson 3



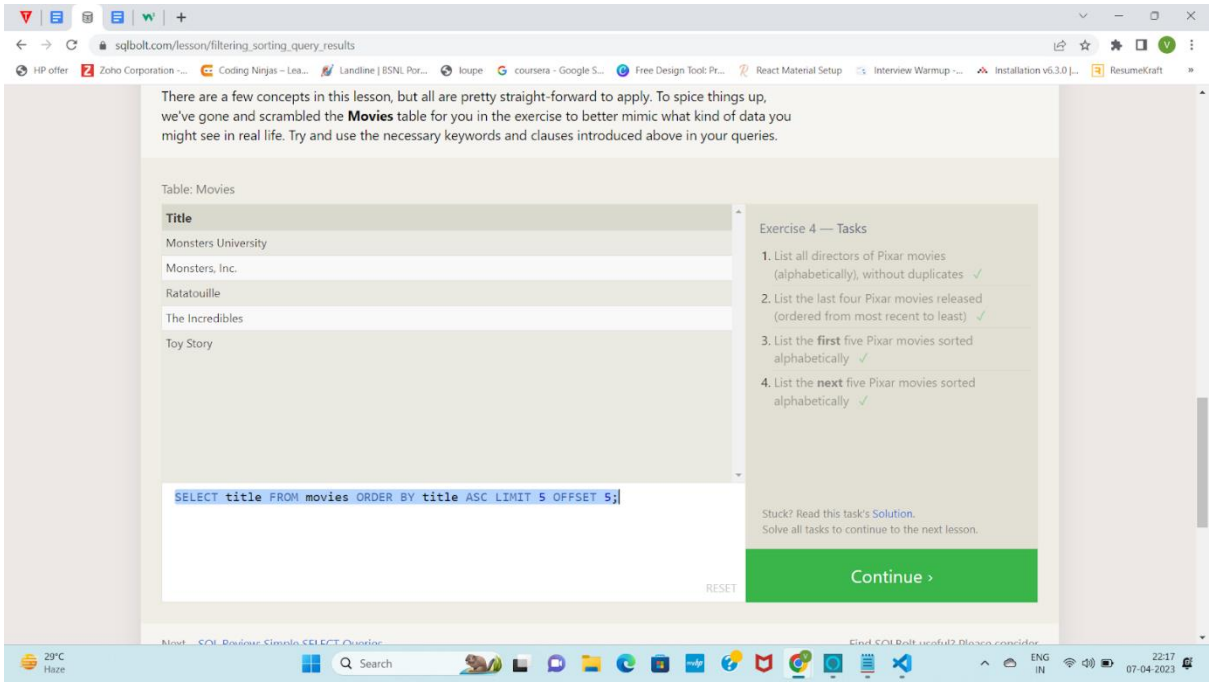
1.SELECT * FROM movies where title like '%toy%'

2.SELECT * FROM movies where director like '%John Lasseter%'

3.SELECT * FROM movies where director not like '%John Lasseter%'

4.SELECT * FROM movies where title like '%WALL%'

Lesson 4



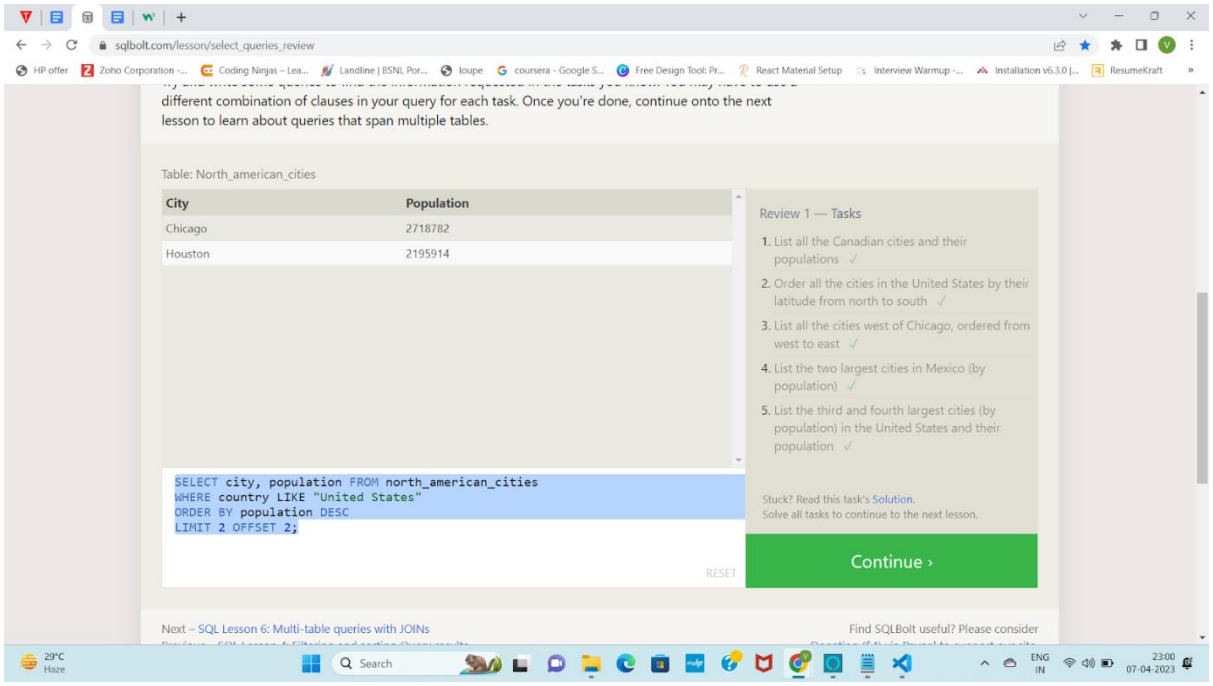
1.SELECT distinct director FROM movies ORDER BY director ASC;

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;

Lesson 5



1.SELECT city, population FROM north_american_cities WHERE country = "Canada";

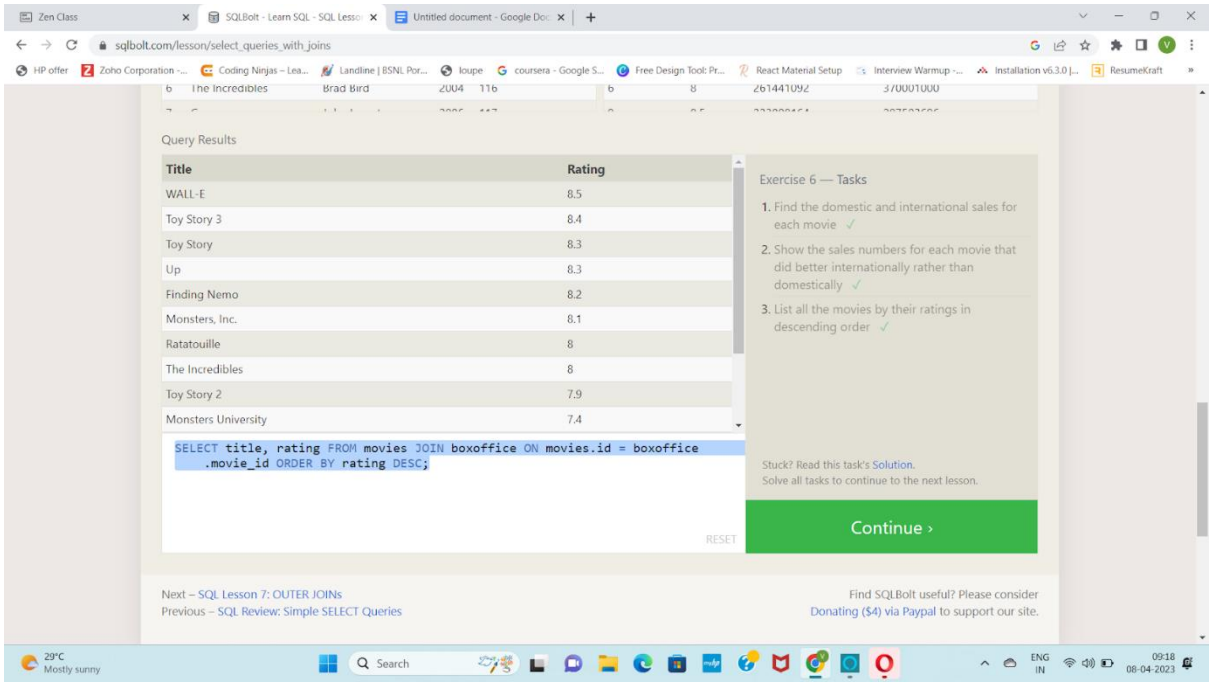
2.SELECT city, latitude FROM north_american_cities WHERE country = "United States"
ORDER BY latitude DESC;

3.SELECT city, latitude FROM north_american_cities WHERE country = "United States"
ORDER BY latitude DESC;

4.SELECT city, population FROM north_american_cities WHERE country LIKE "Mexico"
ORDER BY population DESC LIMIT 2;

5.SELECT city, population FROM north_american_cities WHERE country LIKE "United States"
ORDER BY population DESC LIMIT 2 OFFSET 2;

Lessons 6

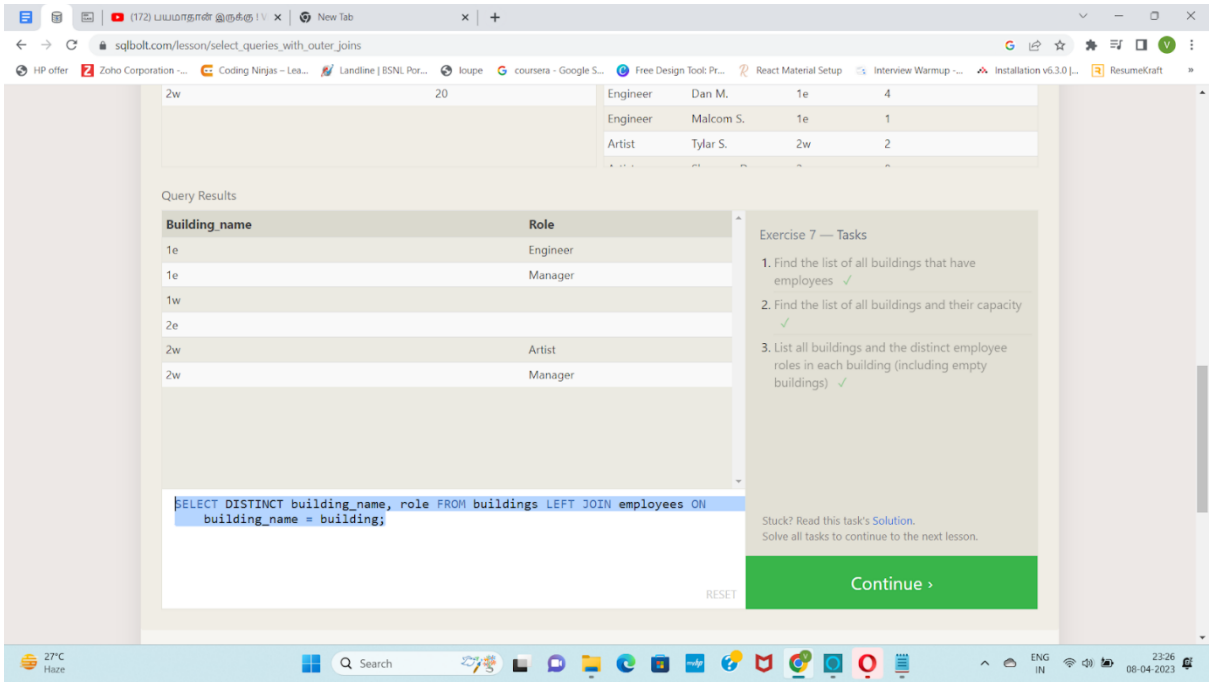


1.SELECT title, domestic_sales, international_sales FROM movies JOIN boxoffice
ON movies.id = boxoffice.movie_id;

2.SELECT title, domestic_sales, international_sale FROM movies JOIN boxoffice
ON movies.id = boxoffice.movie_id WHERE international_sales > domestic_sale;

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

Lesson 7

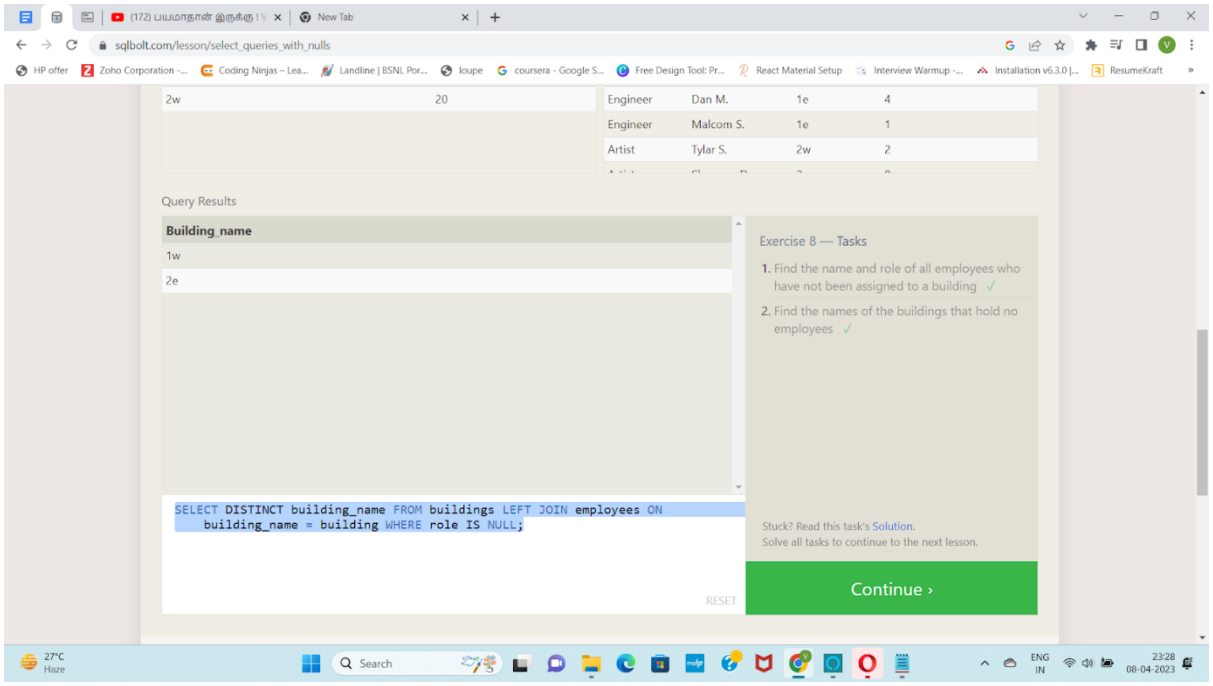


1.SELECT DISTINCT building FROM employees;

2.SELECT * FROM buildings;

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

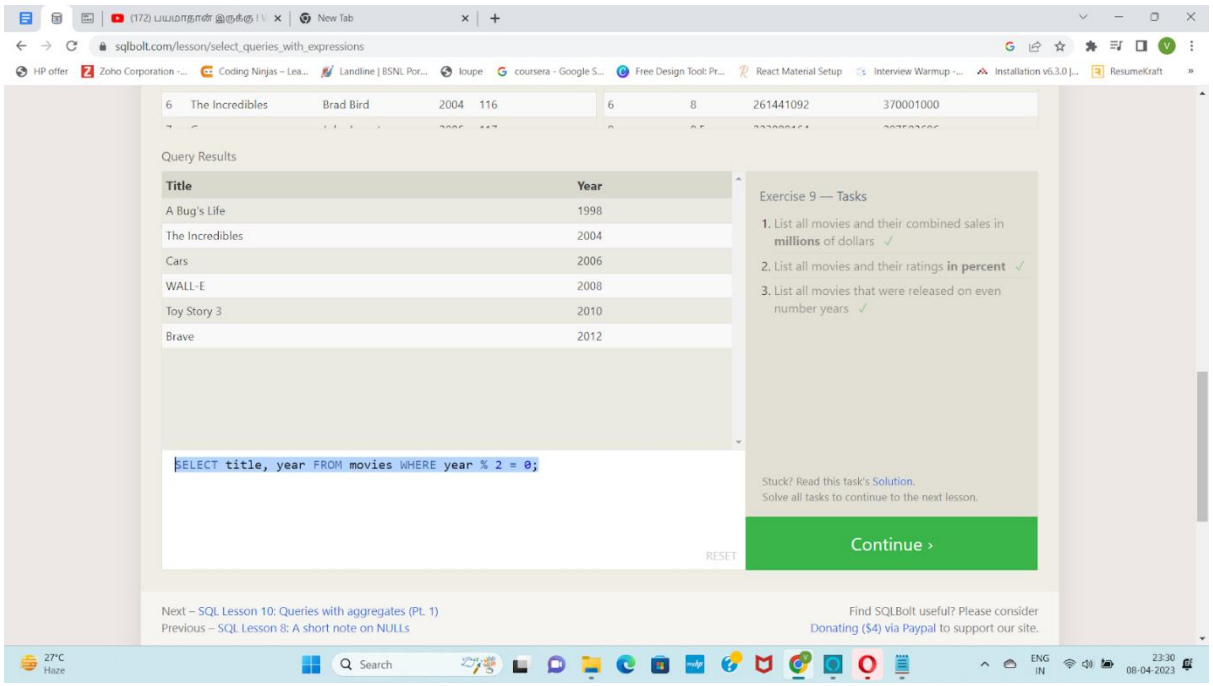
Lesson 8



1.SELECT name, role FROM employees WHERE building IS NULL;

2.SELECT DISTINCT building_name FROM buildings LEFT JOIN employees ON building_name = building WHERE role IS NULL;

Lesson 9

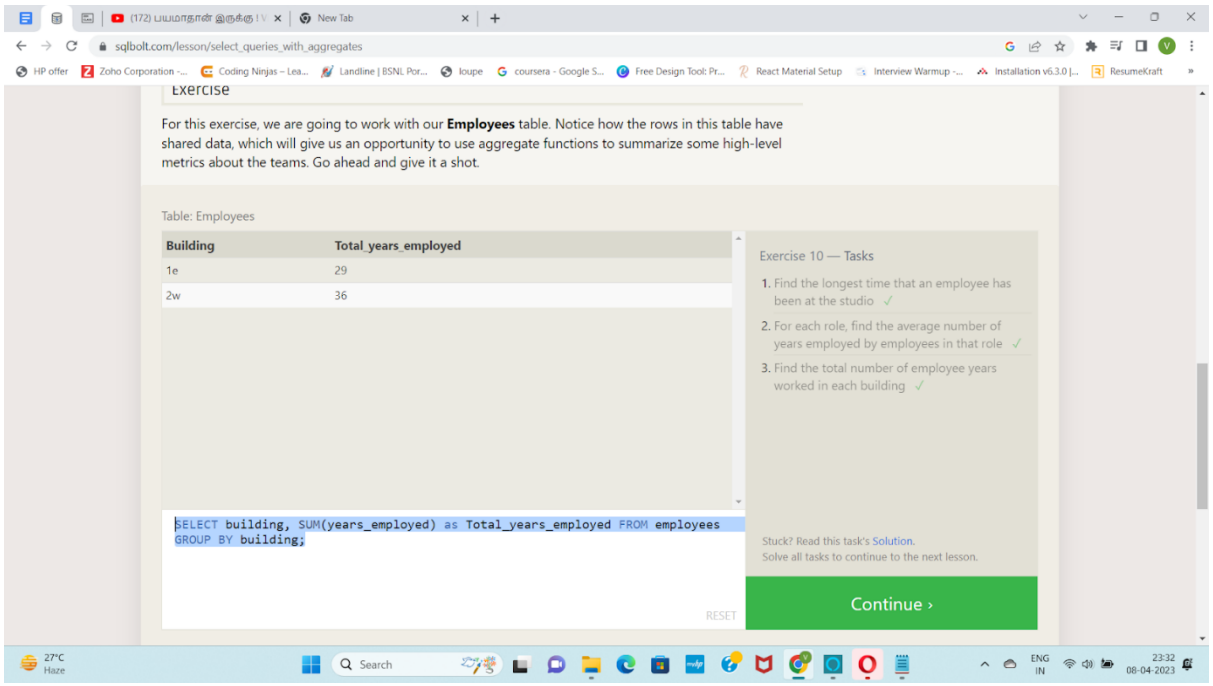


1.SELECT title, (domestic_sales + international_sales) / 1000000 AS gross_sales_millions FROM movies JOIN boxoffice ON movies.id = boxoffice.movie_id;

2.SELECT title, rating * 10 AS rating_percent FROM movies JOIN boxoffice
ON movies.id = boxoffice.movie_id;

3.SELECT title, year FROM movies WHERE year % 2 = 0;

Lesson 10

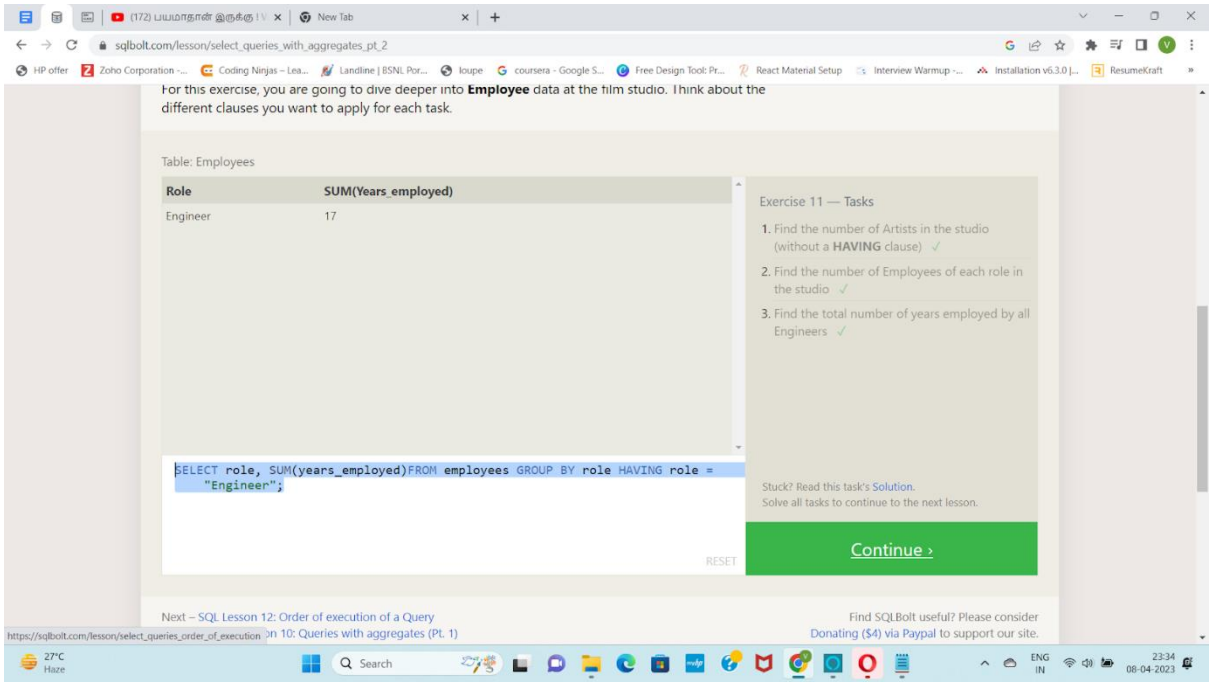


1.SELECT MAX(years_employed) as Max_years_employed FROM employees;

2.SELECT role, AVG(years_employed) as Average_years_employed FROM employees
GROUP BY role;

3.SELECT building, SUM(years_employed) as Total_years_employed FROM employees
GROUP BY building;

Lesson 11

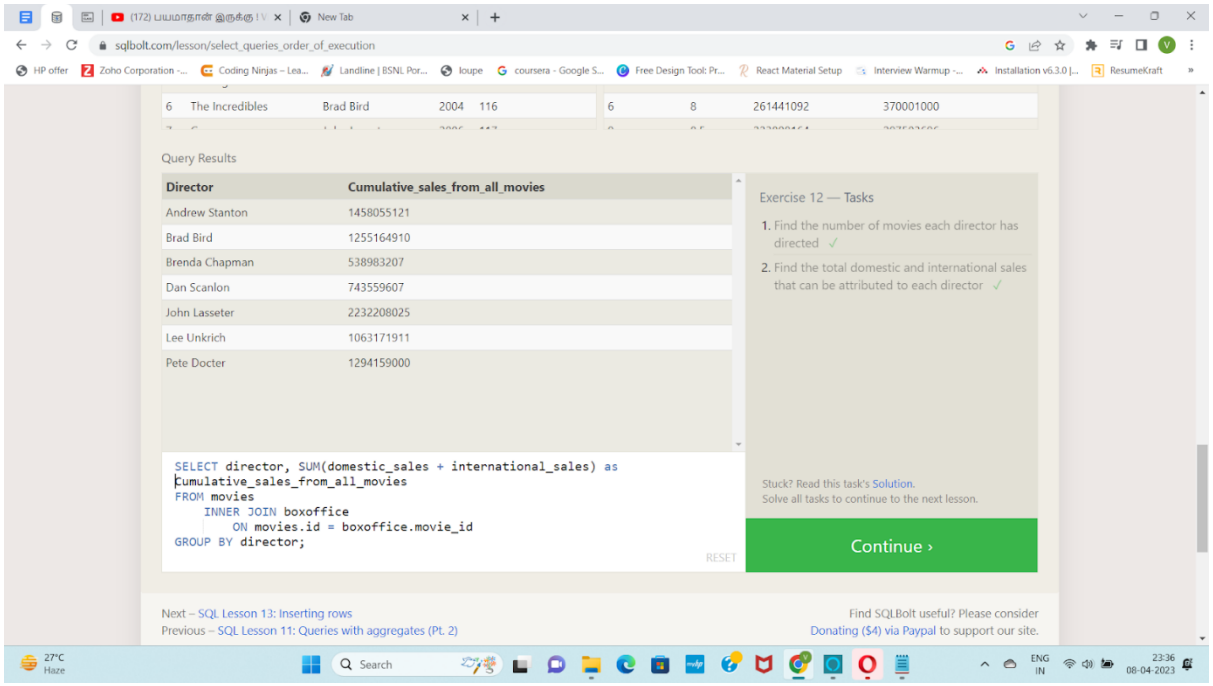


1.SELECT role, COUNT(*) as Number_of_artists FROM employees WHERE role = "Artist";

2.SELECT role, COUNT(*) FROM employees GROUP BY role;

3.SELECT role, SUM(years_employed)FROM employees GROUP BY role HAVING role = "Engineer";

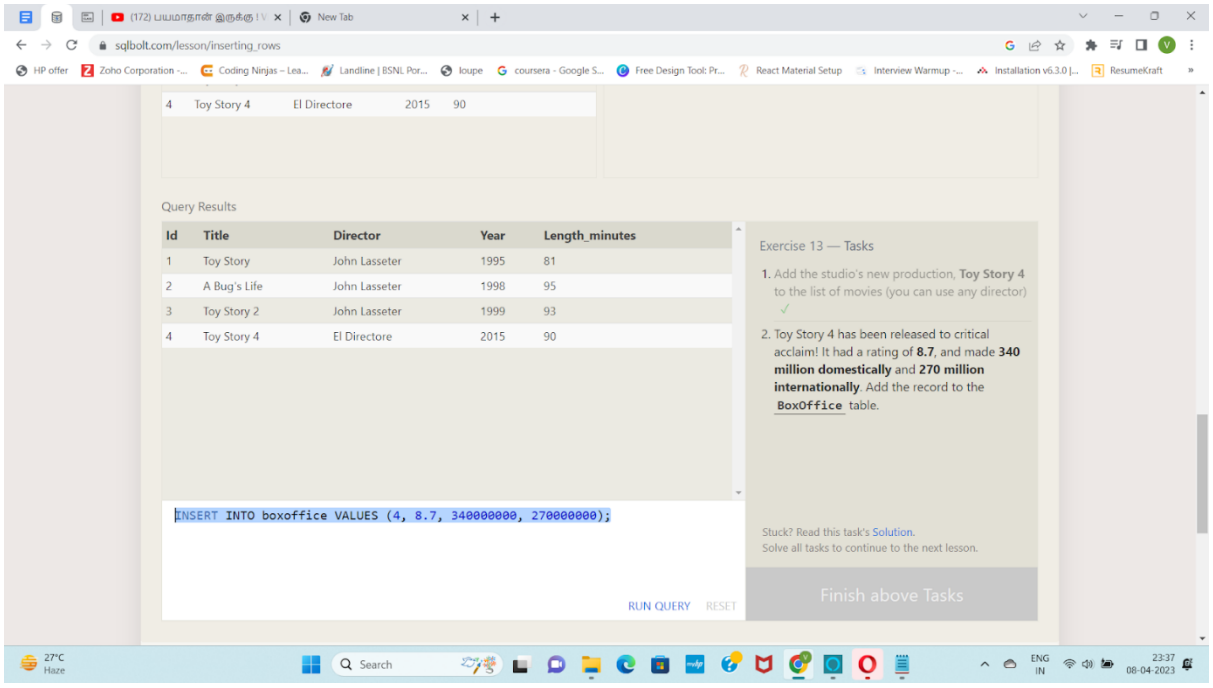
Lesson 12



1.SELECT director, COUNT(id) as Num_movies_directed FROM movies GROUP BY director;

2.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;

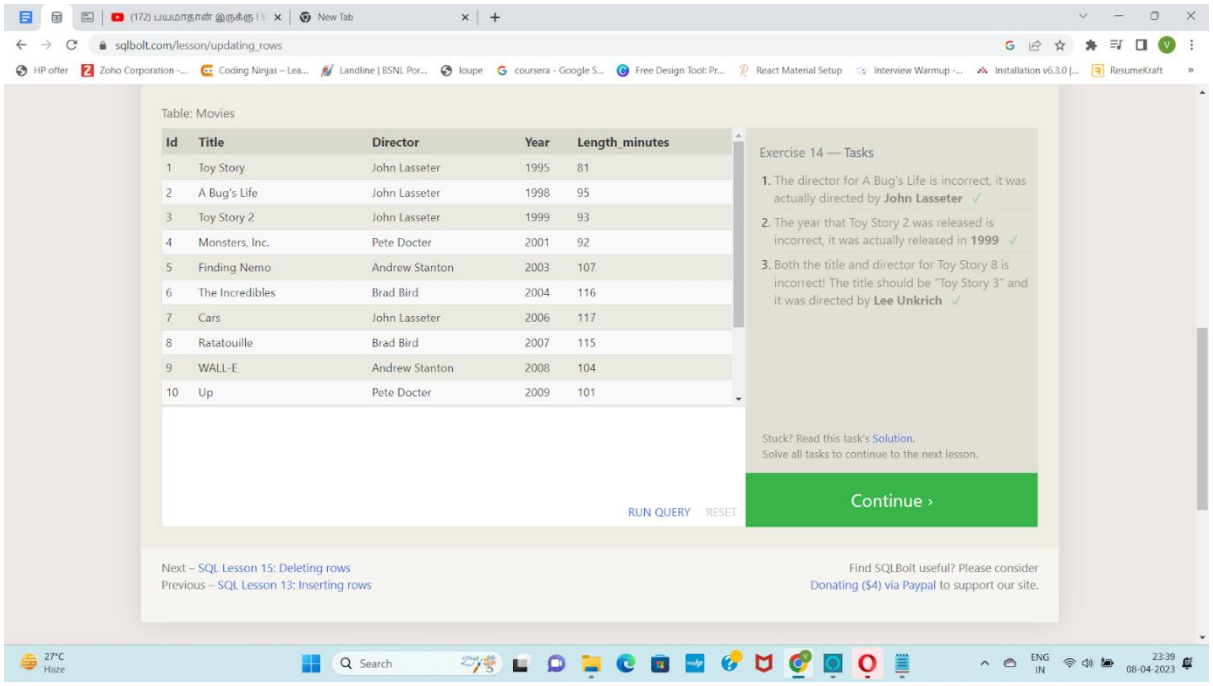
Lesson 13



1.INSERT INTO movies VALUES (4, "Toy Story 4", "El Directore", 2015, 90);

2.INSERT INTO boxoffice VALUES (4, 8.7, 340000000, 270000000);

Lesson 14

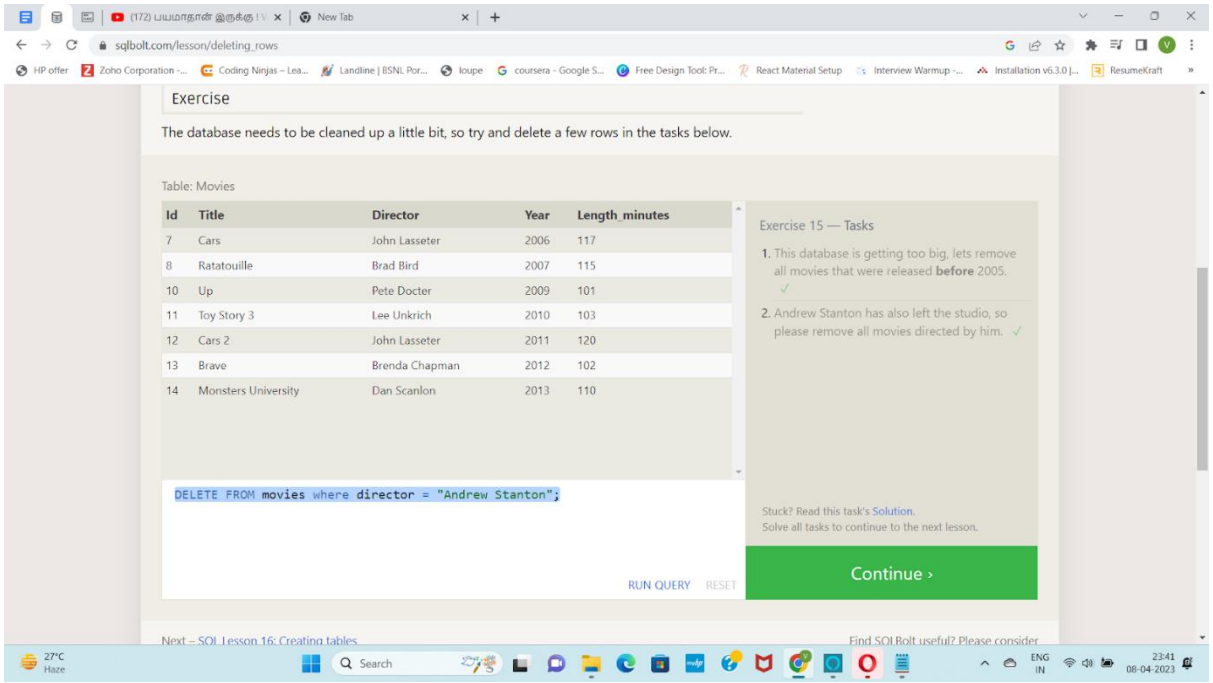


1.UPDATE movies SET director = "John Lasseter" WHERE id = 2;

2.UPDATE movies SET year = 1999 WHERE id = 3;

3.UPDATE movies SET title = "Toy Story 3", director = "Lee Unkrich"
WHERE id = 11;

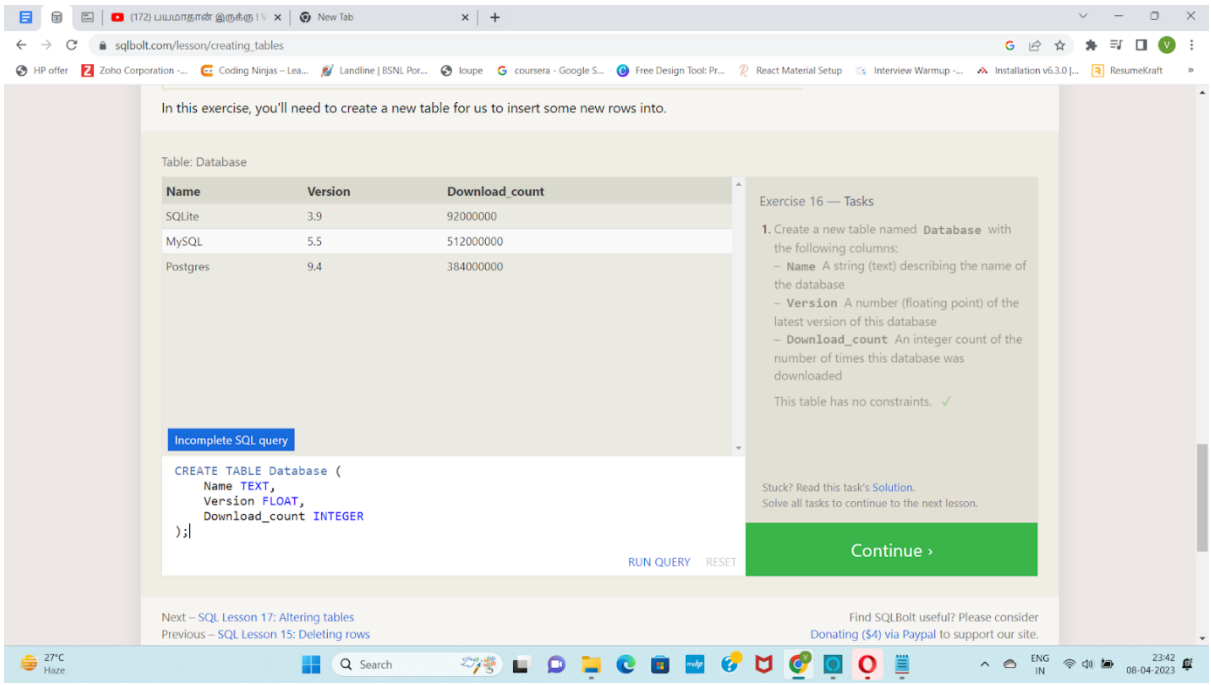
Lesson 15



1.DELETE FROM movies where year < 2005;

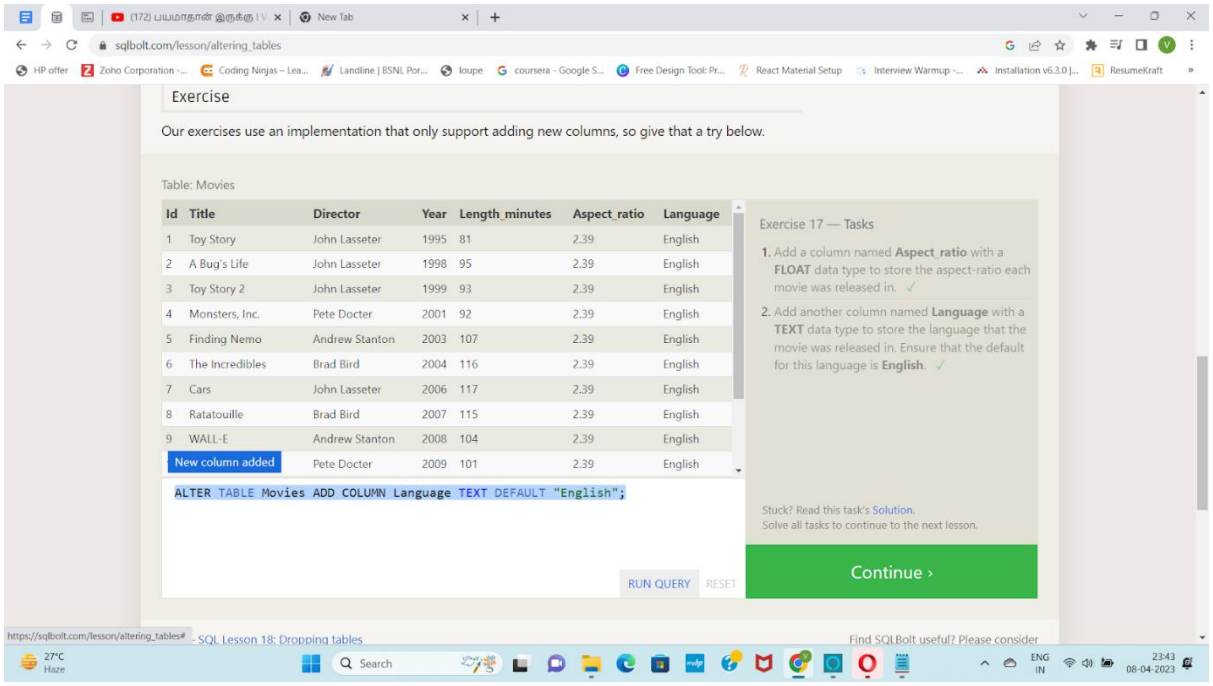
2.DELETE FROM movies where director = "Andrew Stanton";

Lesson 16



1.CREATE TABLE Database (
    Name TEXT,
    Version FLOAT,
    Download_count INTEGER
);

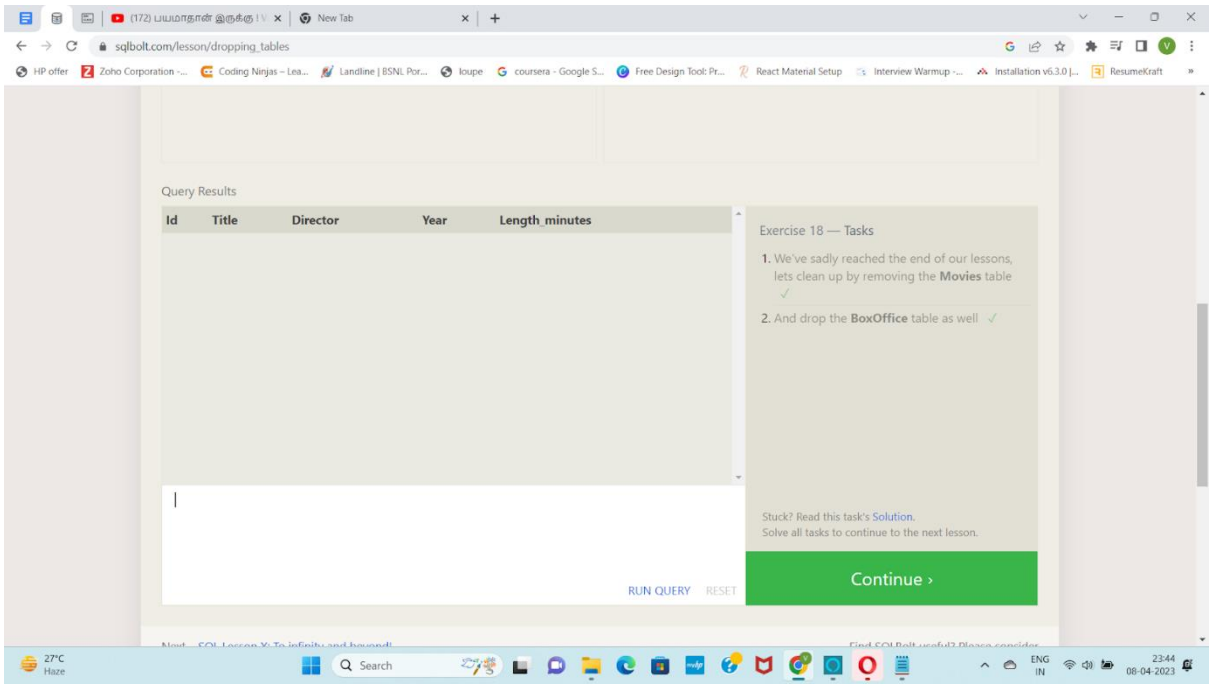
Lesson 17



1.ALTER TABLE Movies ADD COLUMN Aspect_ratio FLOAT DEFAULT 2.39;

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

Lesson 18



1.DROP TABLE Movies;

2.DROP TABLE BoxOffice;