SQL bolt Task

**1.SQL Lesson 1: SELECT queries 101:**

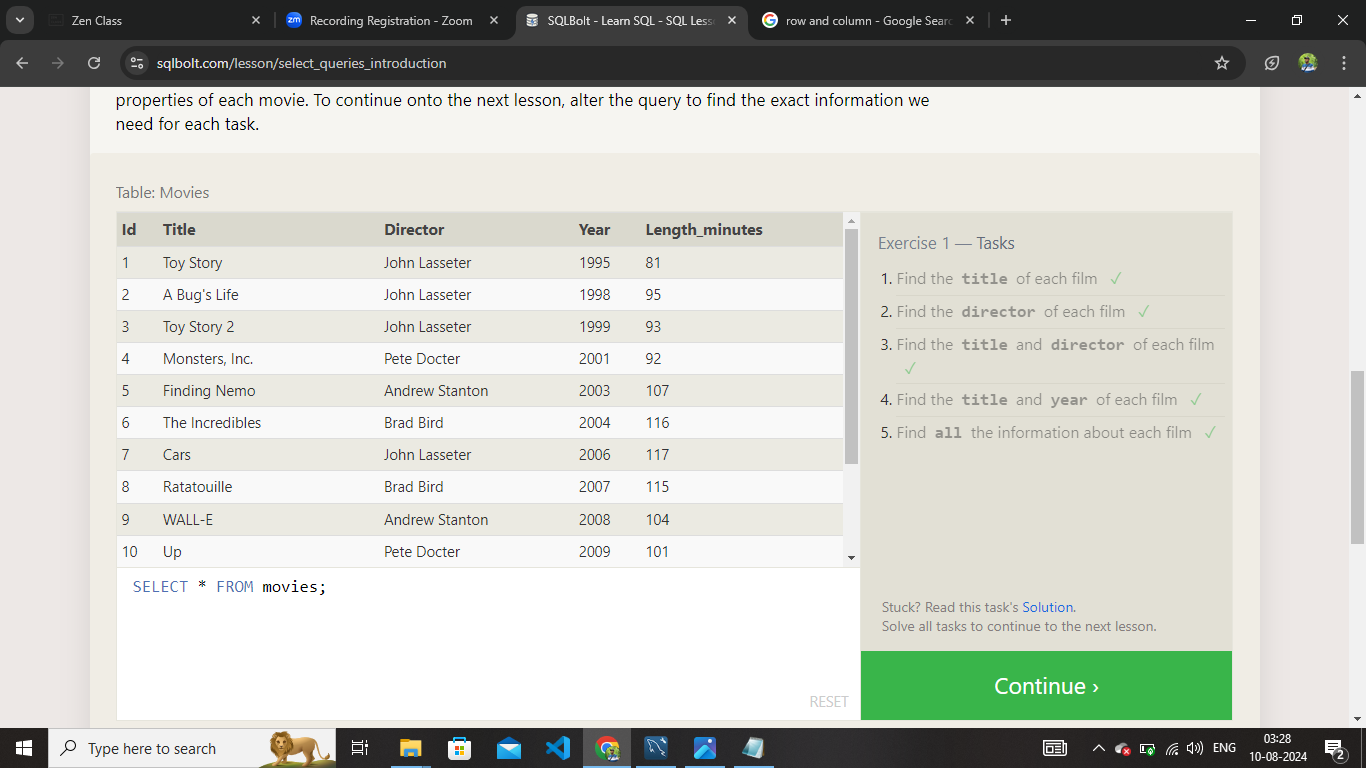
1.SELECT title FROM movies;

2.SELECT director FROM movies;

3.SELECT title, director FROM movies;

4.SELECT title, year FROM movies;

5.SELECT \* FROM movies;

****

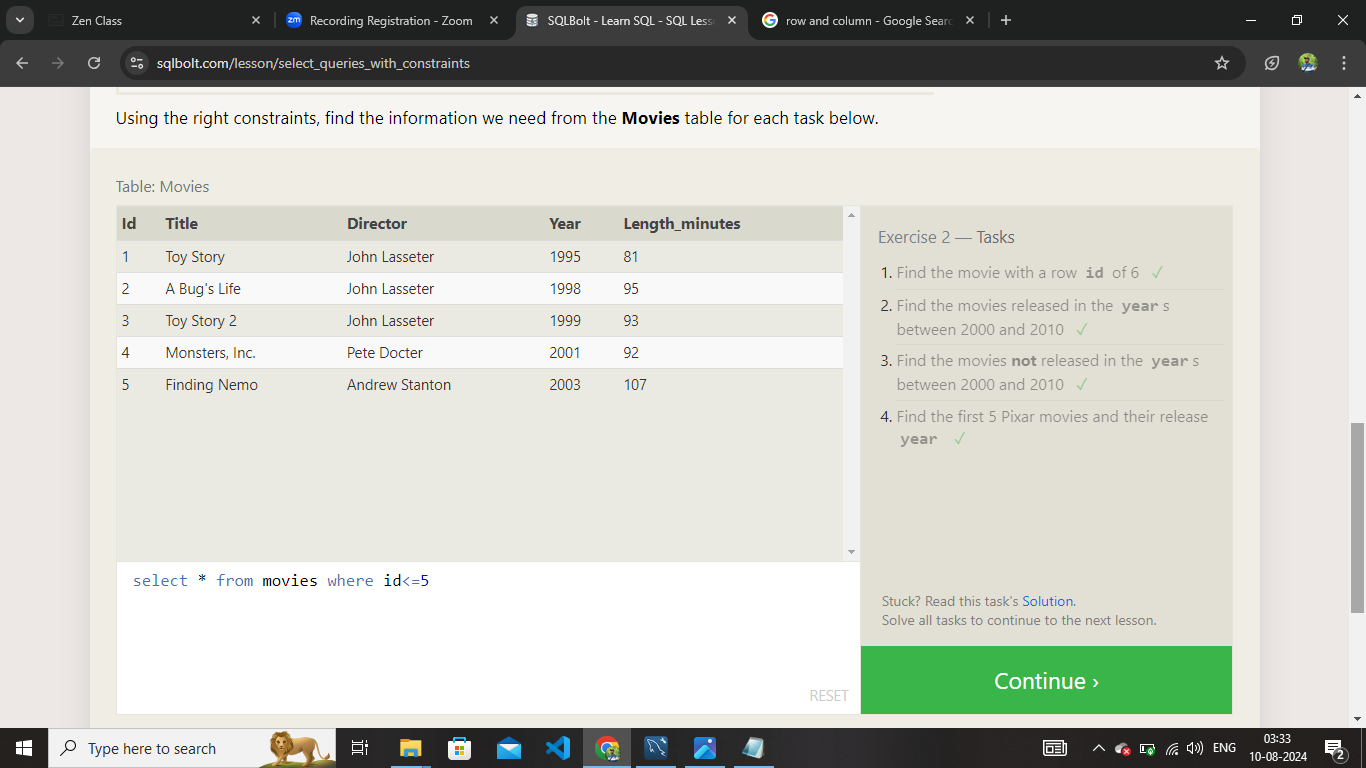
**2.SQL Lesson 2: Queries with constraints (Pt. 1)**

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<=5;



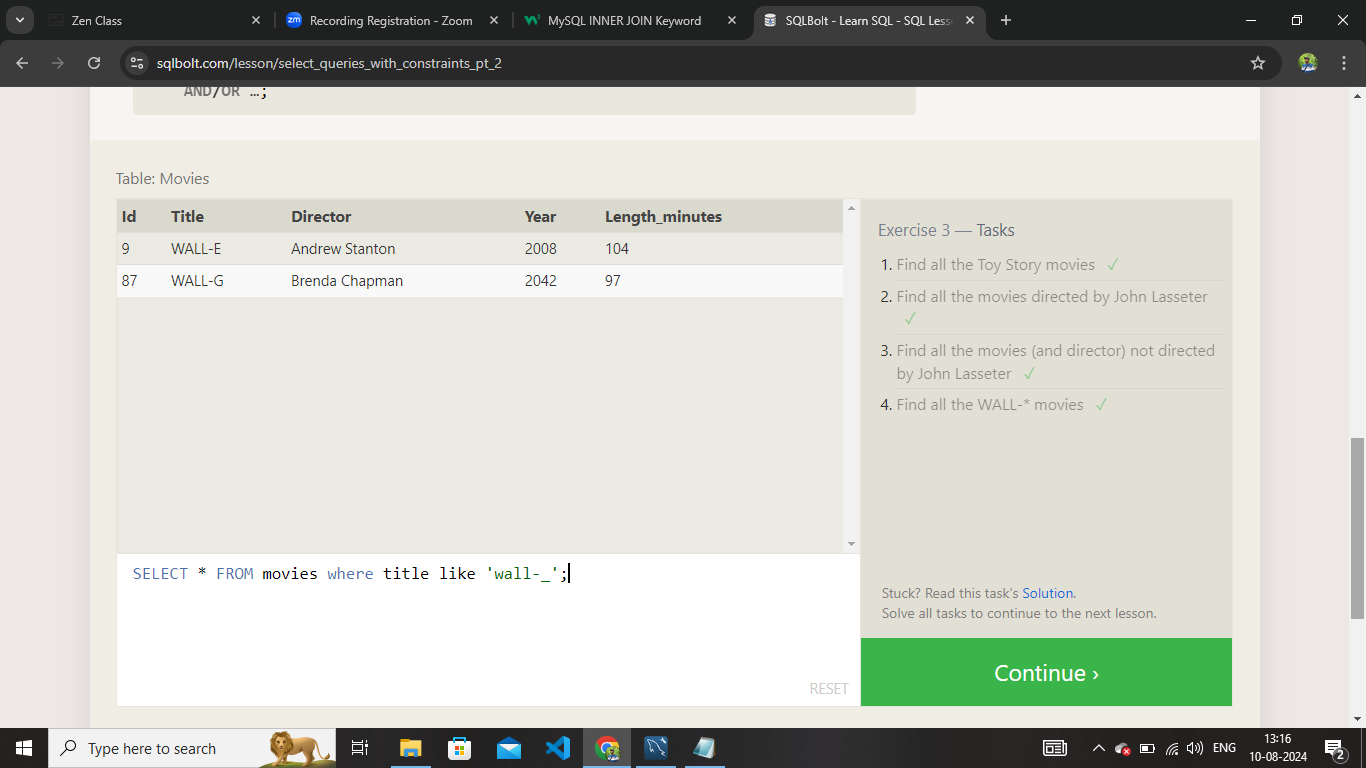
**3.SQL Lesson 3: Queries with constraints (Pt. 2)**

1.SELECT \* FROM movies where title like 'Toy Story%';

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-\_';



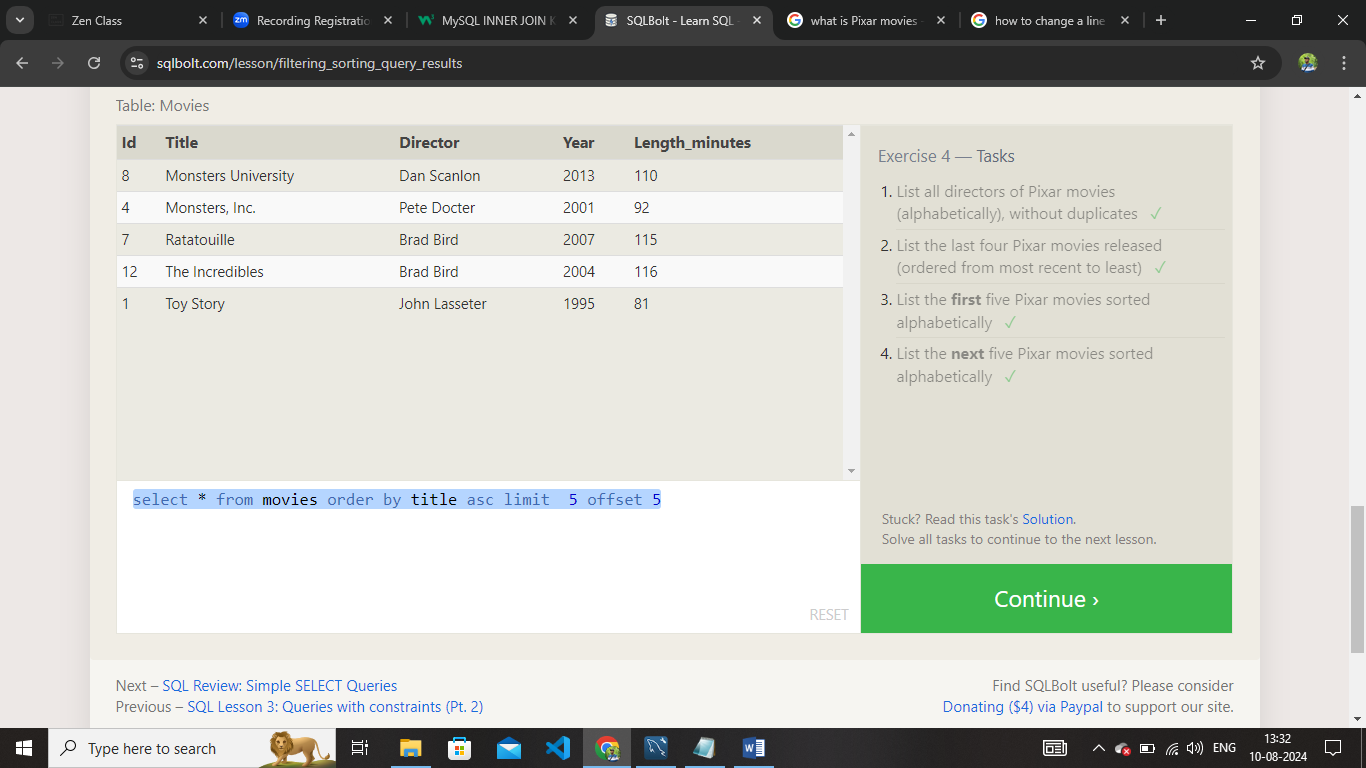
**4.SQL Lesson 4: Filtering and sorting Query results**

1.SELECT distinct director from movies order by director;

2.select \* from movies order by year desc limit 4;

3.select \* from movies order by title asc limit 5;

4.select \* from movies order by title asc limit 5 offset 5;



**5.SQL Review: Simple SELECT Queries**

1.SELECT \* FROM north\_american\_cities where country like 'Canada'

2.SELECT city, latitude

FROM north\_american\_cities

WHERE country = 'United States'

ORDER BY latitude DESC;

3.SELECT city, longitude

FROM north\_american\_cities

WHERE longitude < -87.6298

ORDER BY longitude ASC;

4.SELECT city, population

FROM north\_american\_cities

WHERE country = 'Mexico'

ORDER BY population DESC

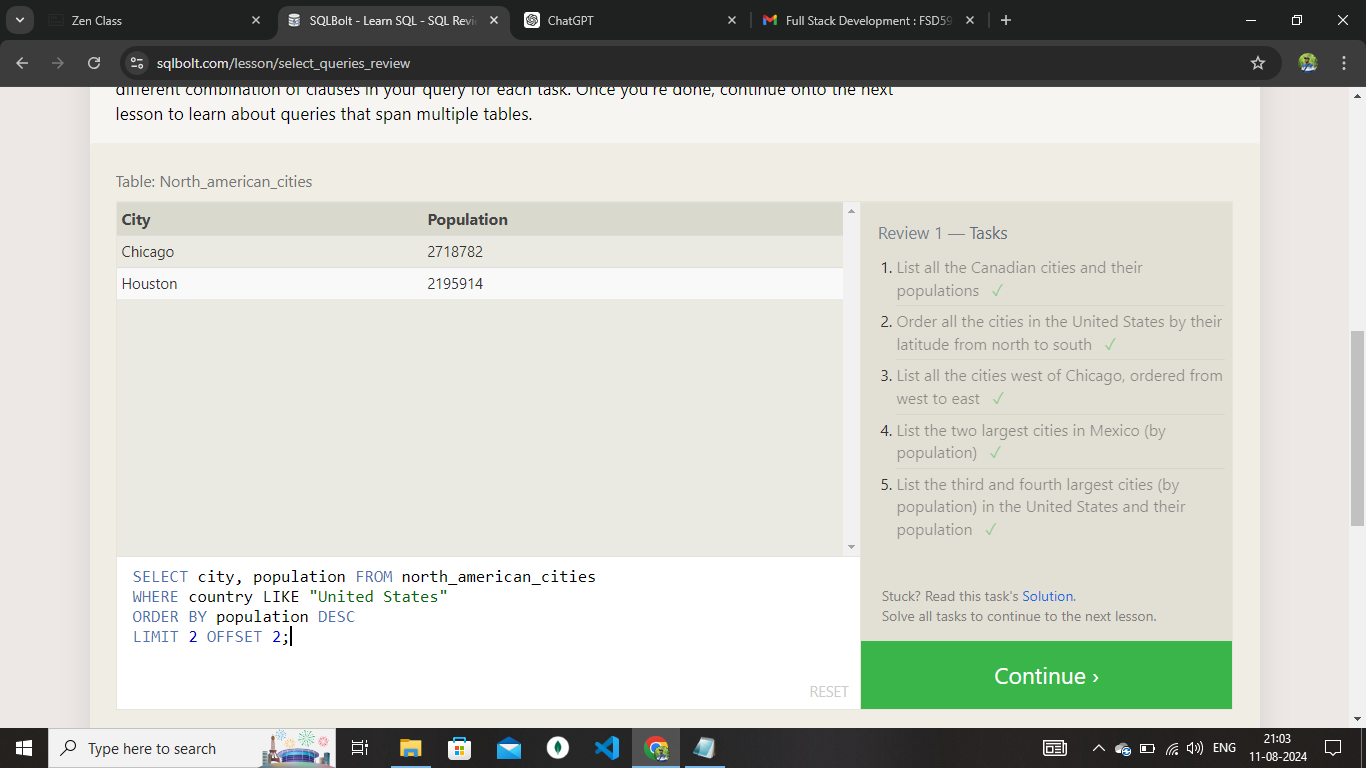
LIMIT 2;

5.SELECT city, population FROM north\_american\_cities

WHERE country LIKE "United States"

ORDER BY population DESC

LIMIT 2 OFFSET 2;



**6.SQL Lesson 6: Multi-table queries with JOINs**

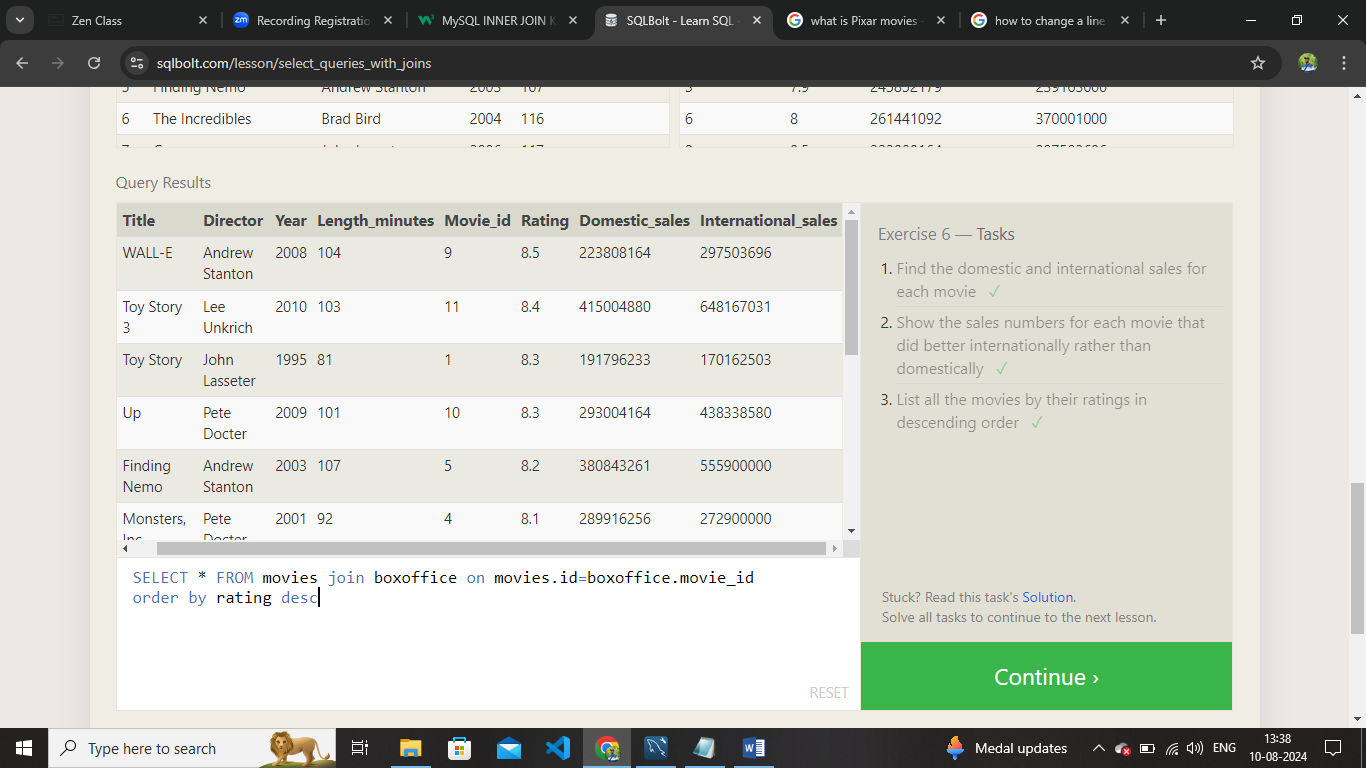
1.SELECT \* FROM movies join boxoffice on movies.id=boxoffice.movie\_id ;

2.SELECT \* FROM movies join boxoffice on movies.id=boxoffice.movie\_id

where international\_sales>domestic\_sales;

3.SELECT \* FROM movies join boxoffice on movies.id=boxoffice.movie\_id

order by rating desc;

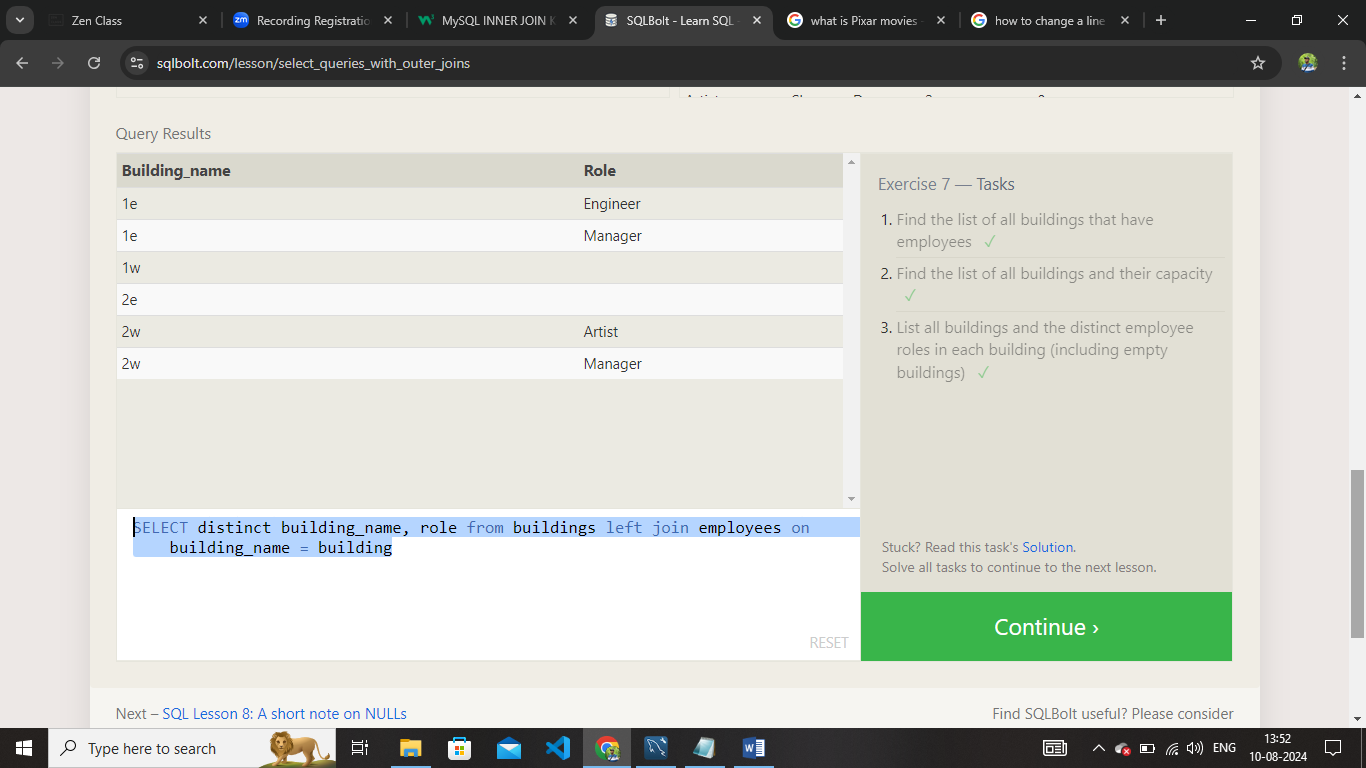


**7.SQL Lesson 7: OUTER JOINs**

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;



**8.SQL Lesson 8: A short note on NULLs**

1.SELECT name, role FROM employees

WHERE building IS NULL;

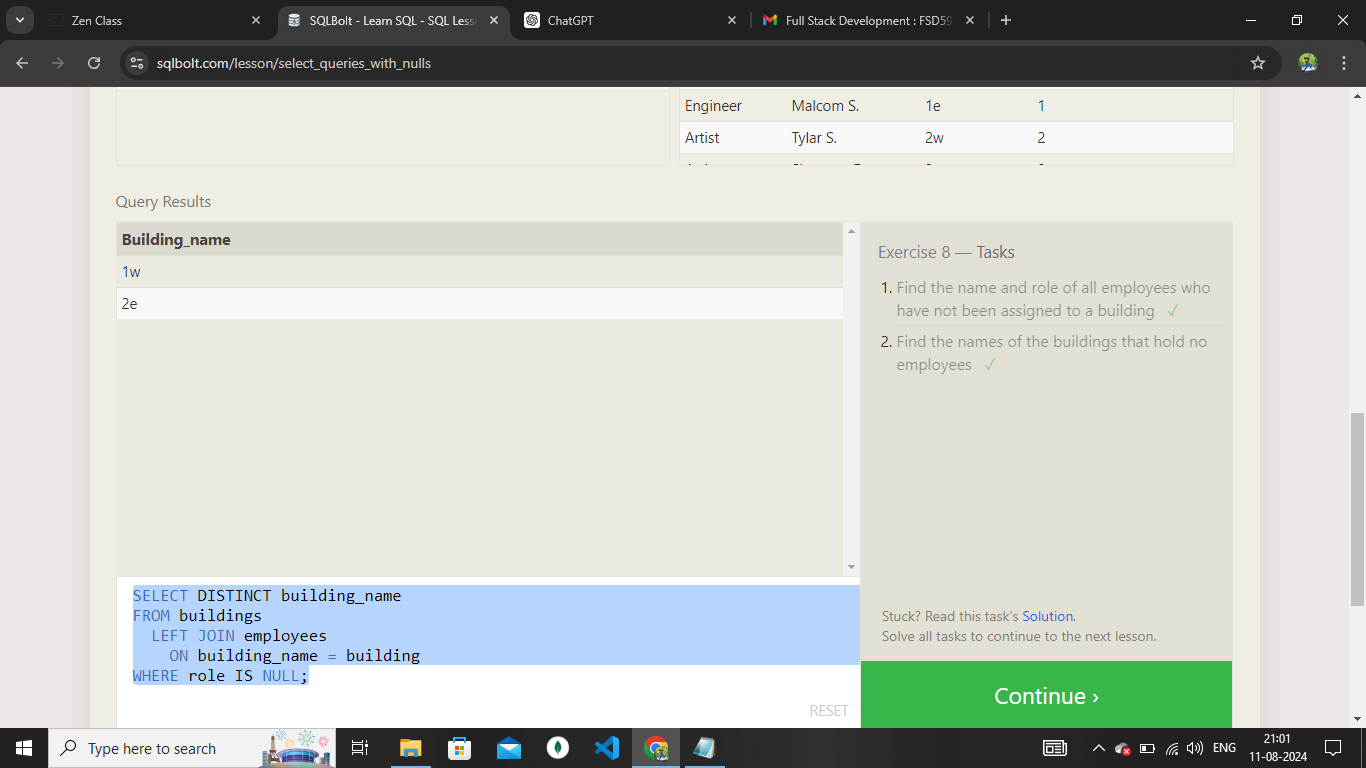
2.SELECT DISTINCT building\_name

FROM buildings

LEFT JOIN employees

ON building\_name = building

WHERE role IS NULL;



**9.SQL Lesson 9: Queries with expressions**

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

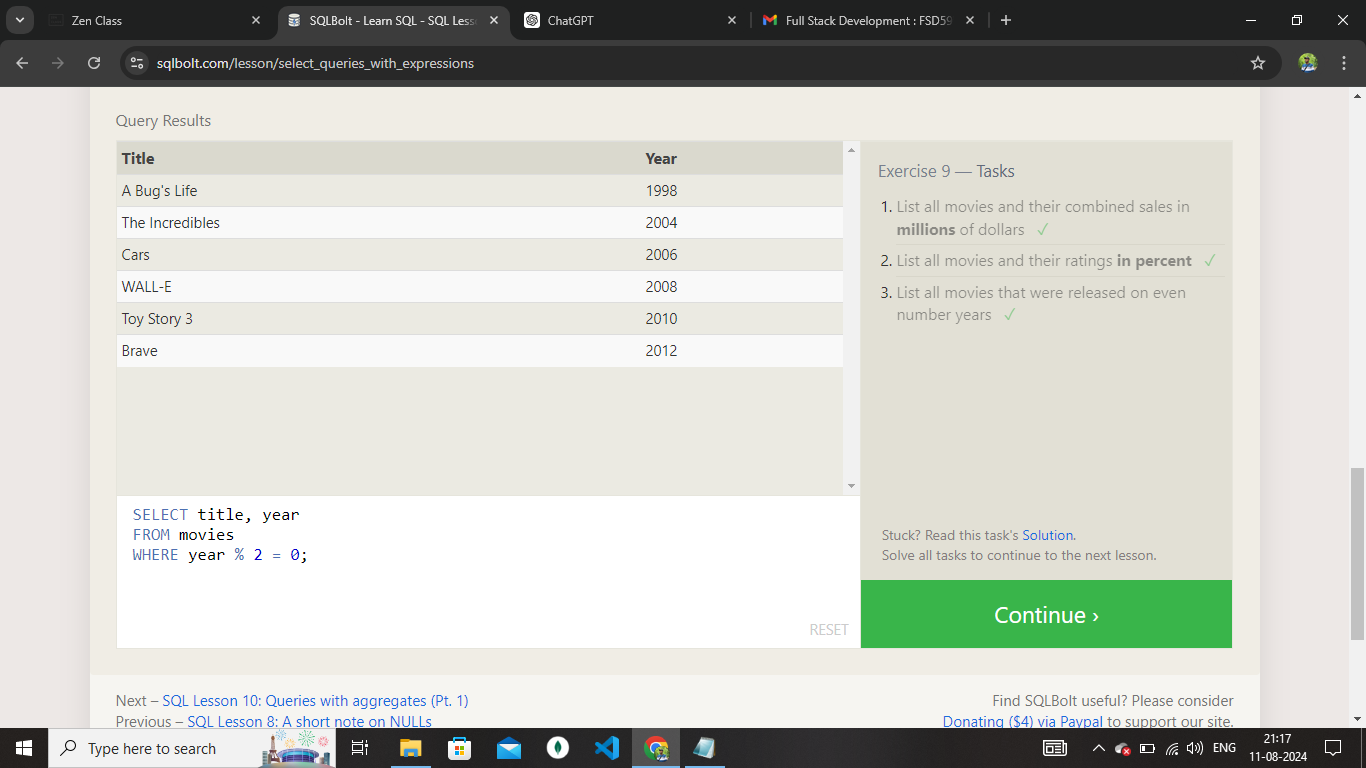
JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

3.SELECT title, year

FROM movies

WHERE year % 2 = 0;



**10.SQL Lesson 10: Queries with aggregates (Pt. 1)**

1.SELECT MAX(years\_employed) as Max\_years\_employed

FROM employees;

2.SELECT role, AVG(years\_employed) as Average\_years\_employed

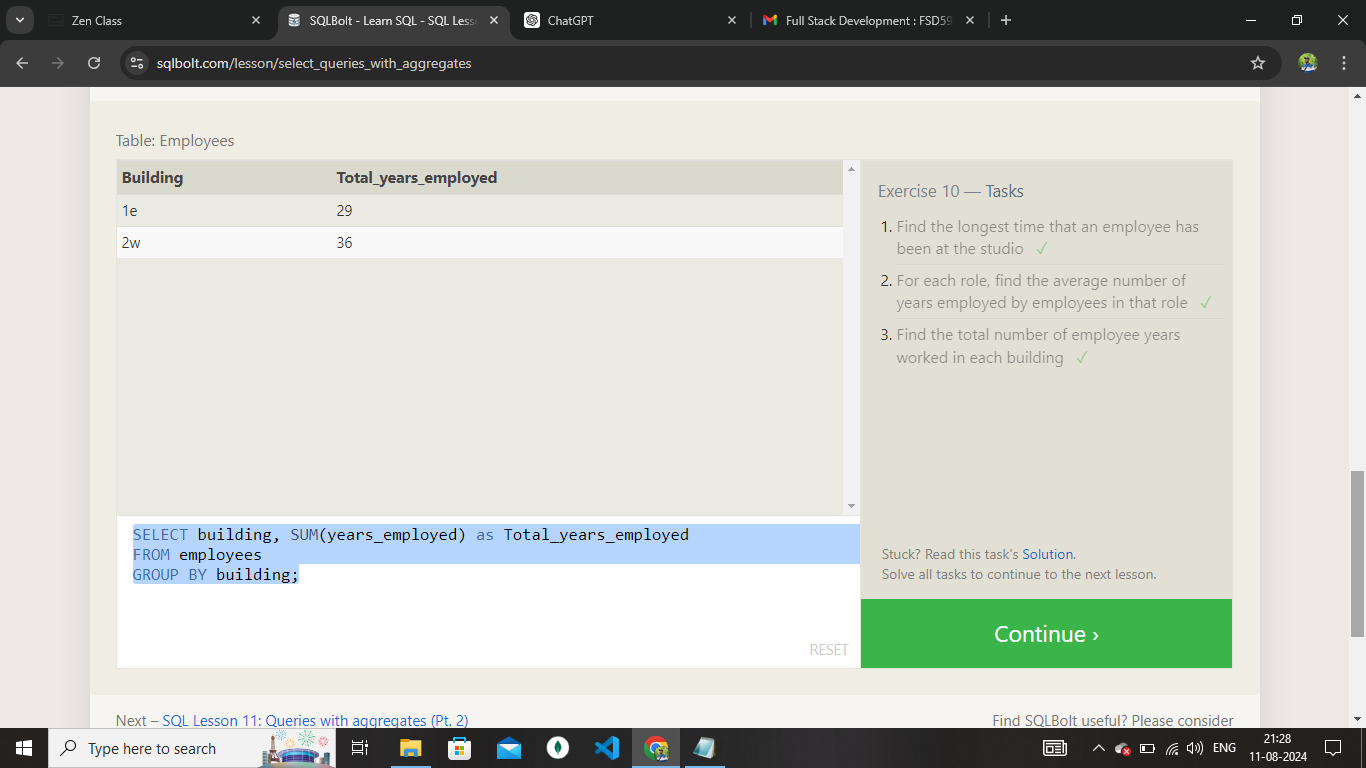
FROM employees

GROUP BY role;

3.SELECT building, SUM(years\_employed) as Total\_years\_employed

FROM employees

GROUP BY building;



**11.SQL Lesson 11: Queries with aggregates (Pt. 2)**

1.SELECT role, COUNT(\*) as Number\_of\_artists

FROM employees

WHERE role = "Artist";

2.SELECT role, COUNT(\*)

FROM employees

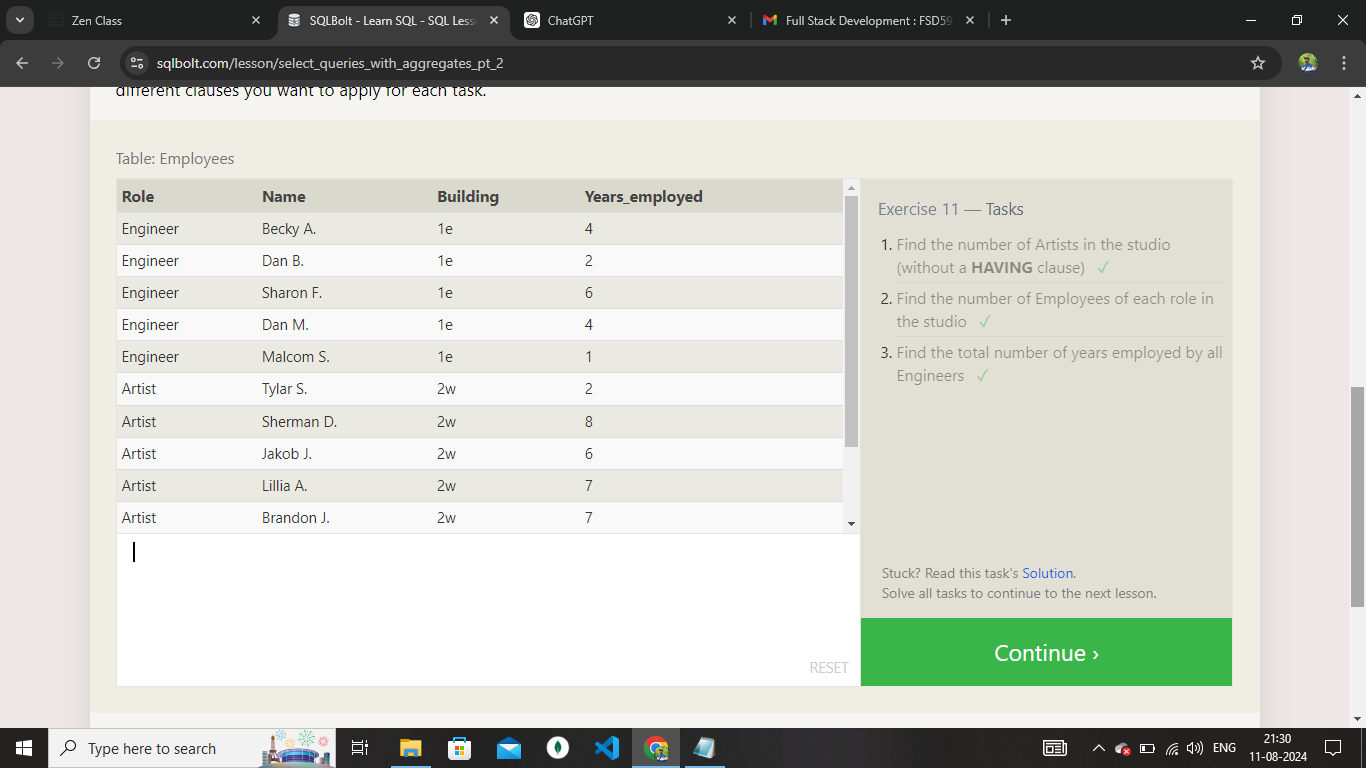
GROUP BY role;

3.SELECT role, SUM(years\_employed)

FROM employees

GROUP BY role

HAVING role = "Engineer";



**12.SQL Lesson 12: Order of execution of a Query**

1.SELECT director, COUNT(id) as Num\_movies\_directed

FROM movies

GROUP BY director;

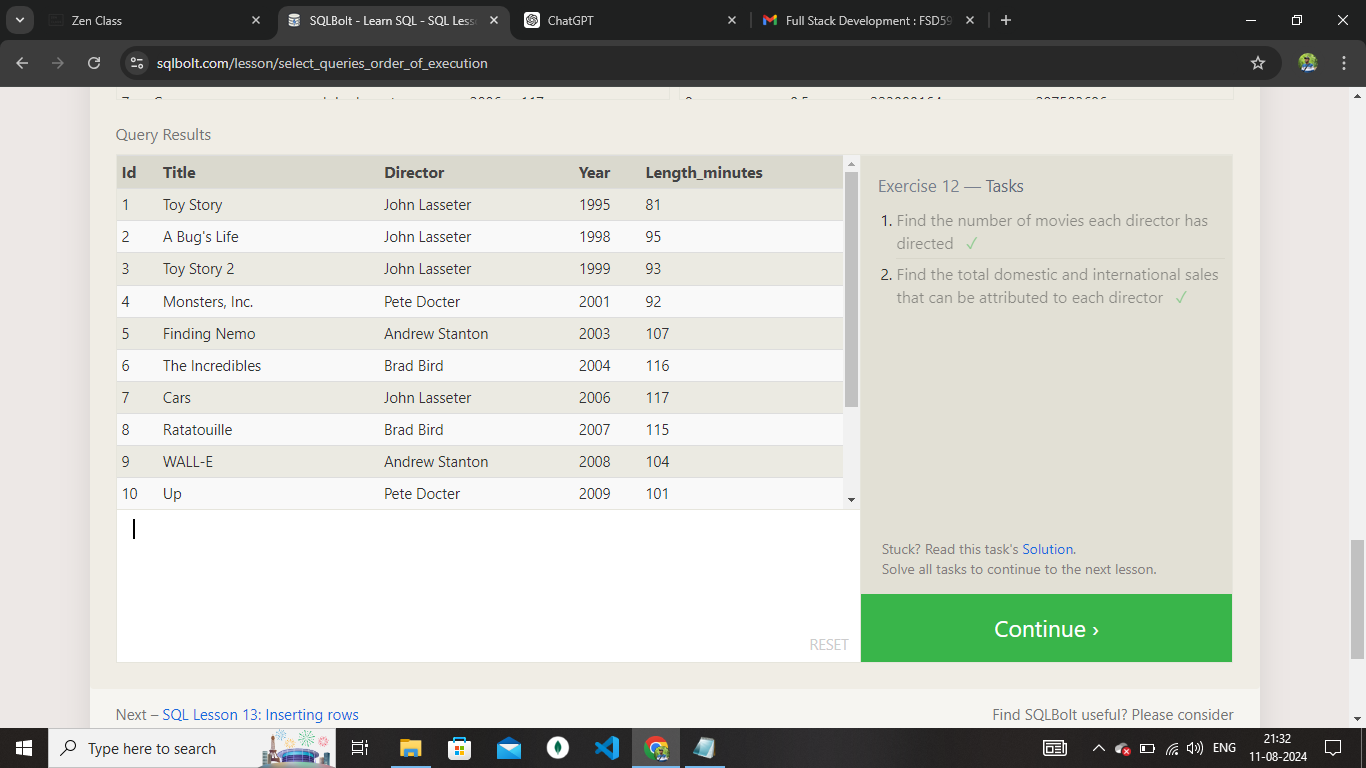
2.SELECT director, SUM(domestic\_sales + international\_sales) as Cumulative\_sales\_from\_all\_movies

FROM movies

INNER JOIN boxoffice

ON movies.id = boxoffice.movie\_id

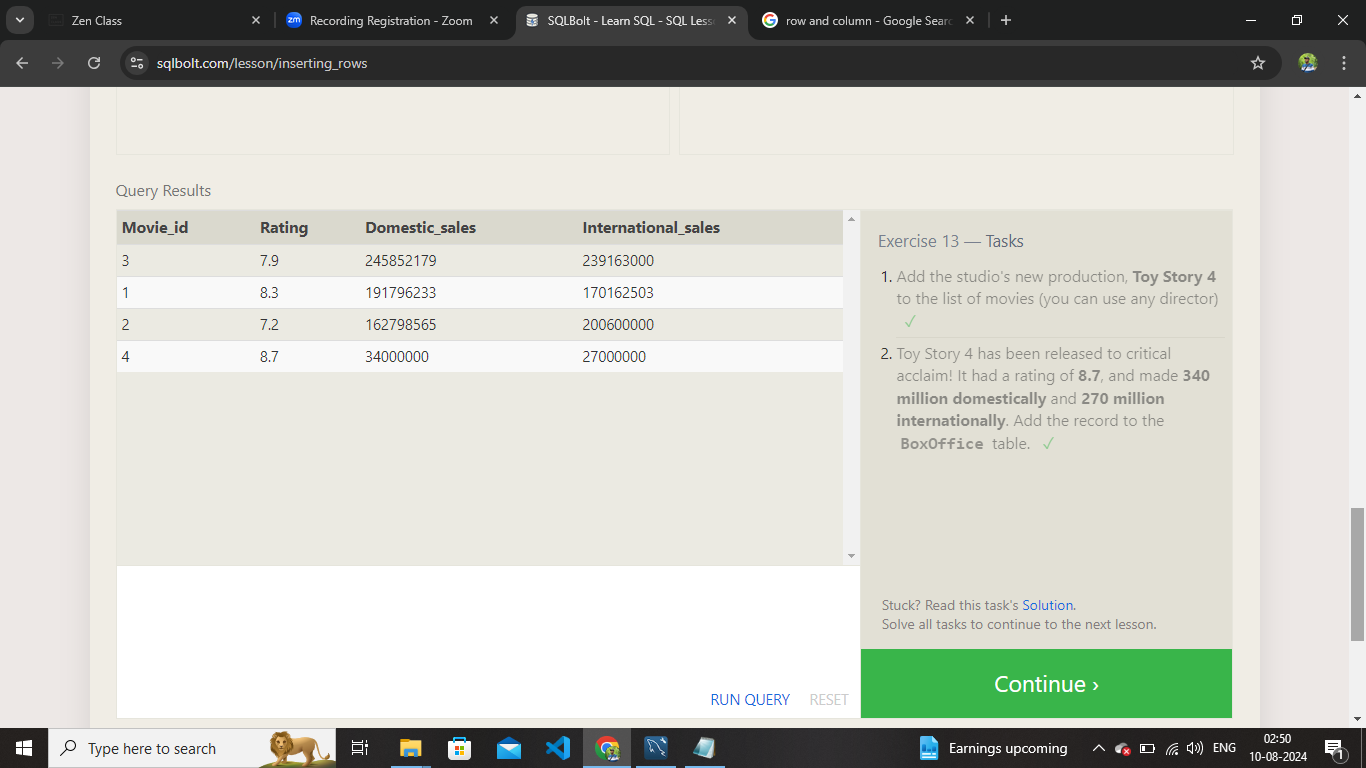
GROUP BY director;



**13.SQL Lesson 13: Inserting rows**

1.insert into movies values(4,'Toy Story 4','John Lasseter',2024,90);

2.insert into Boxoffice values (4,8.7,34000000,27000000);



**14.SQL Lesson 14: Updating rows**

1.update movies

set director = 'John Lasseter'

where id = 2;

2.update movies

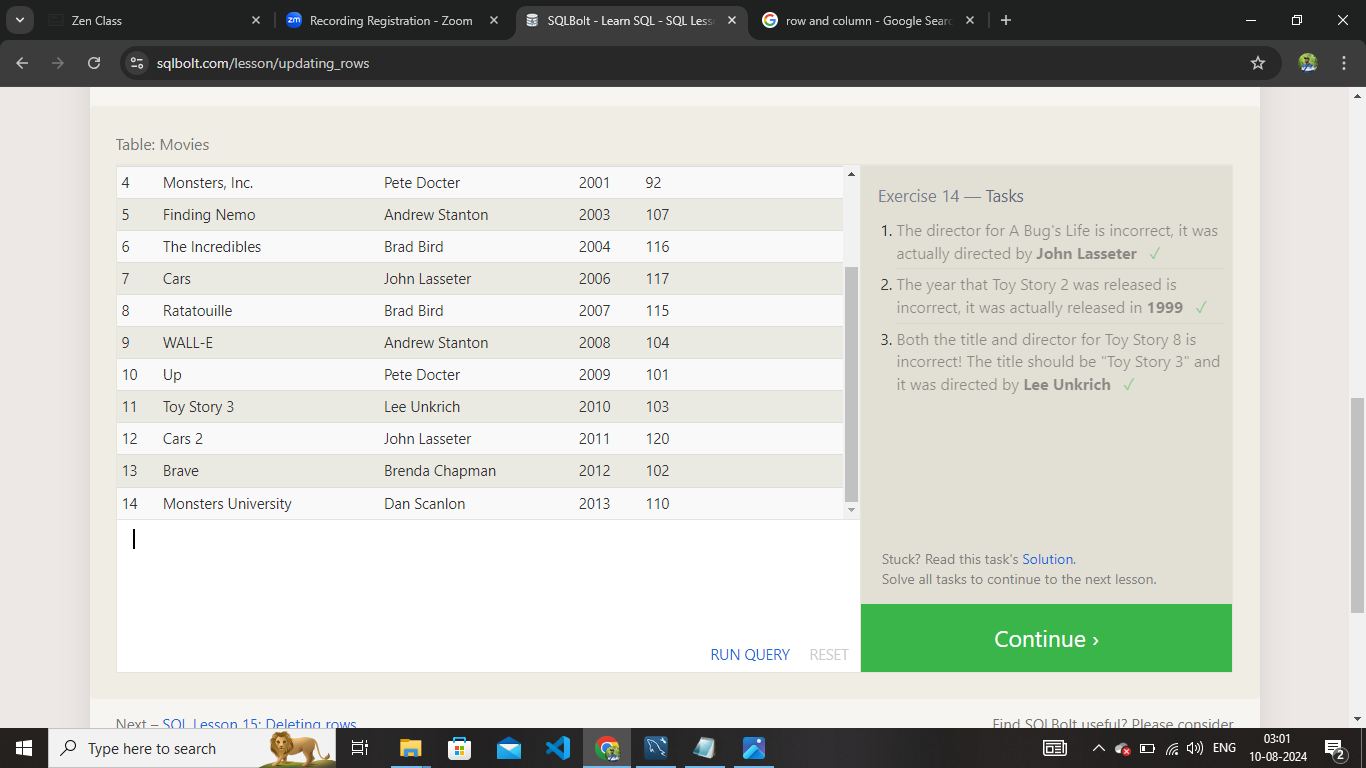
set year = 1999

where id = 3;

3.update Movies

set Director = Lee Unkrich,Title = 'Toy Story'

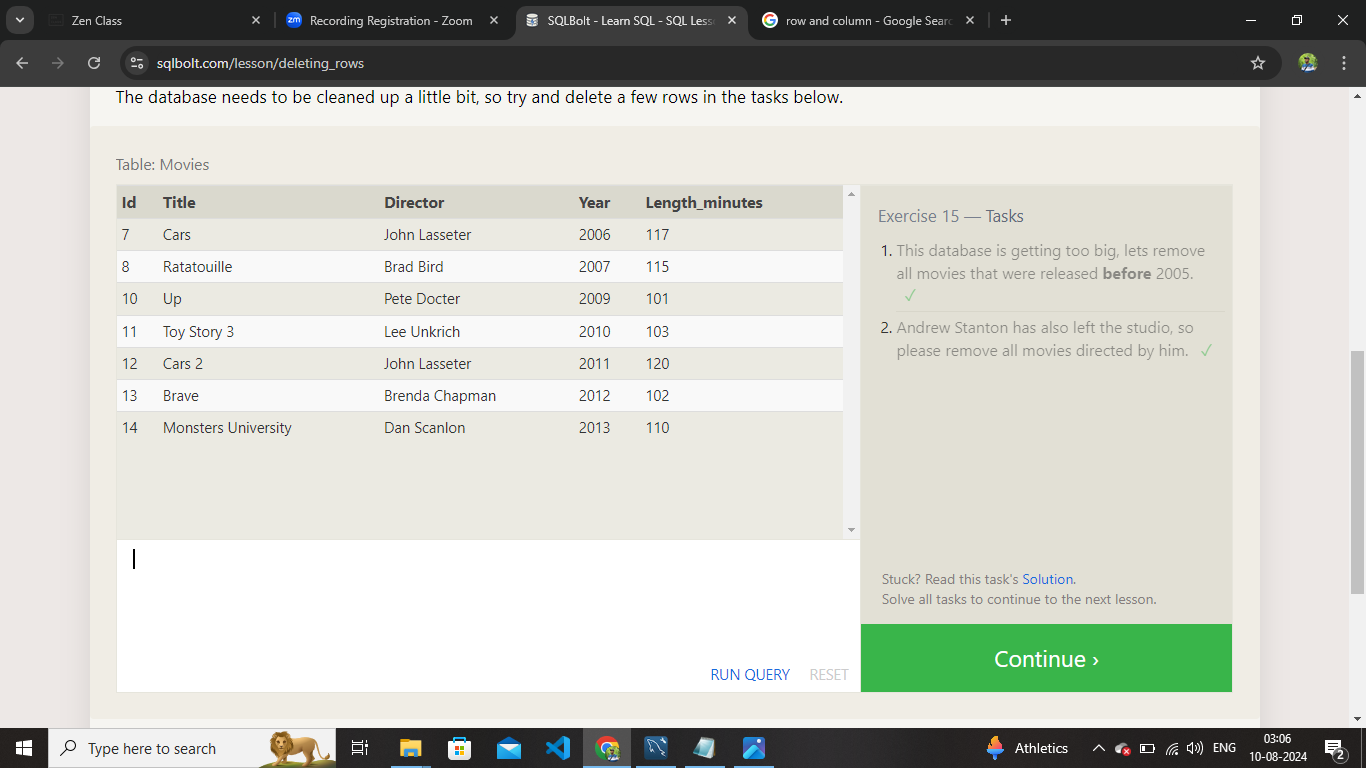
where id = 11;



**15.SQL Lesson 15: Deleting rows**

1.delete from movies where year<2005;

2.delete from movies where director = 'Andrew Stanton';



**16.SQL Lesson 16: Creating tables**

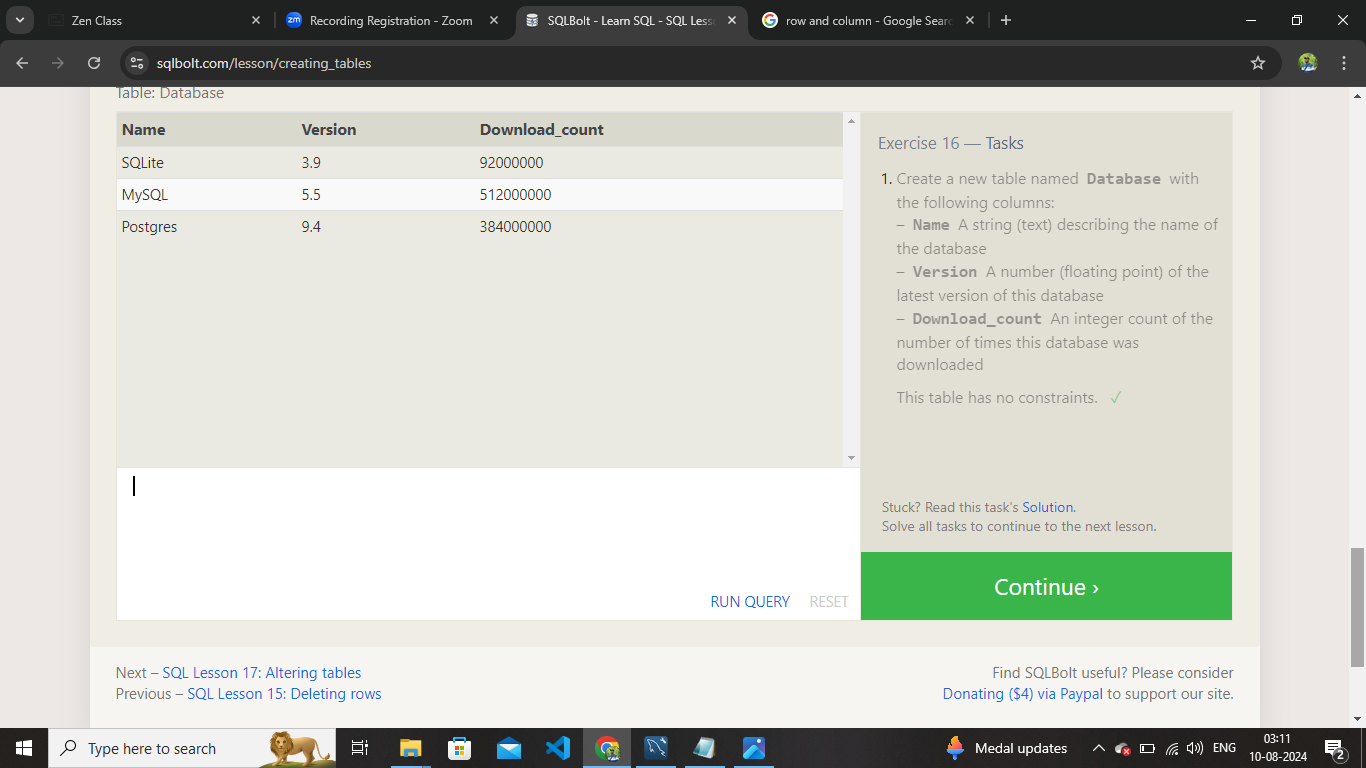
1.create table Database (

Name varchar(255),

Version decimal,

Download\_Count int

);

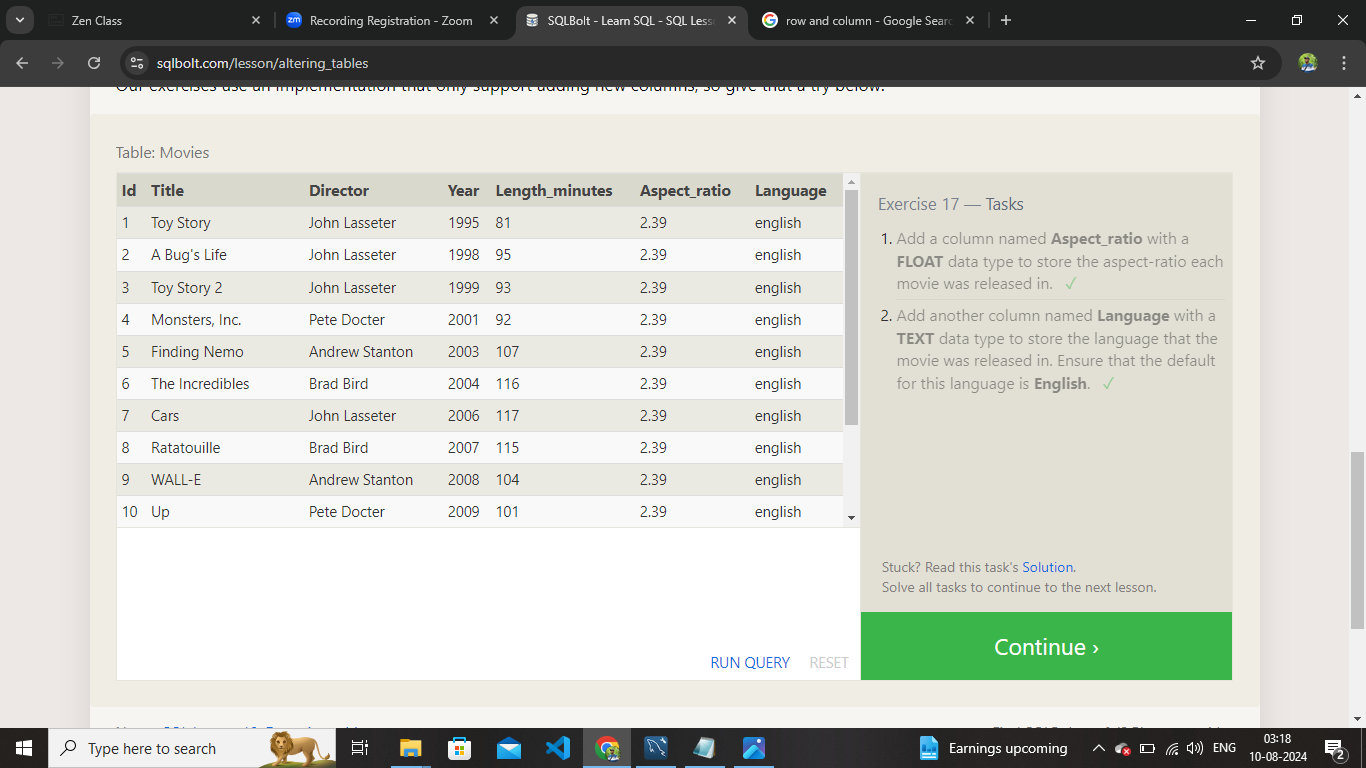


**17.SQL Lesson 17: Altering tables**

1.alter table movies

add column Aspect\_ratio float default 2.39;

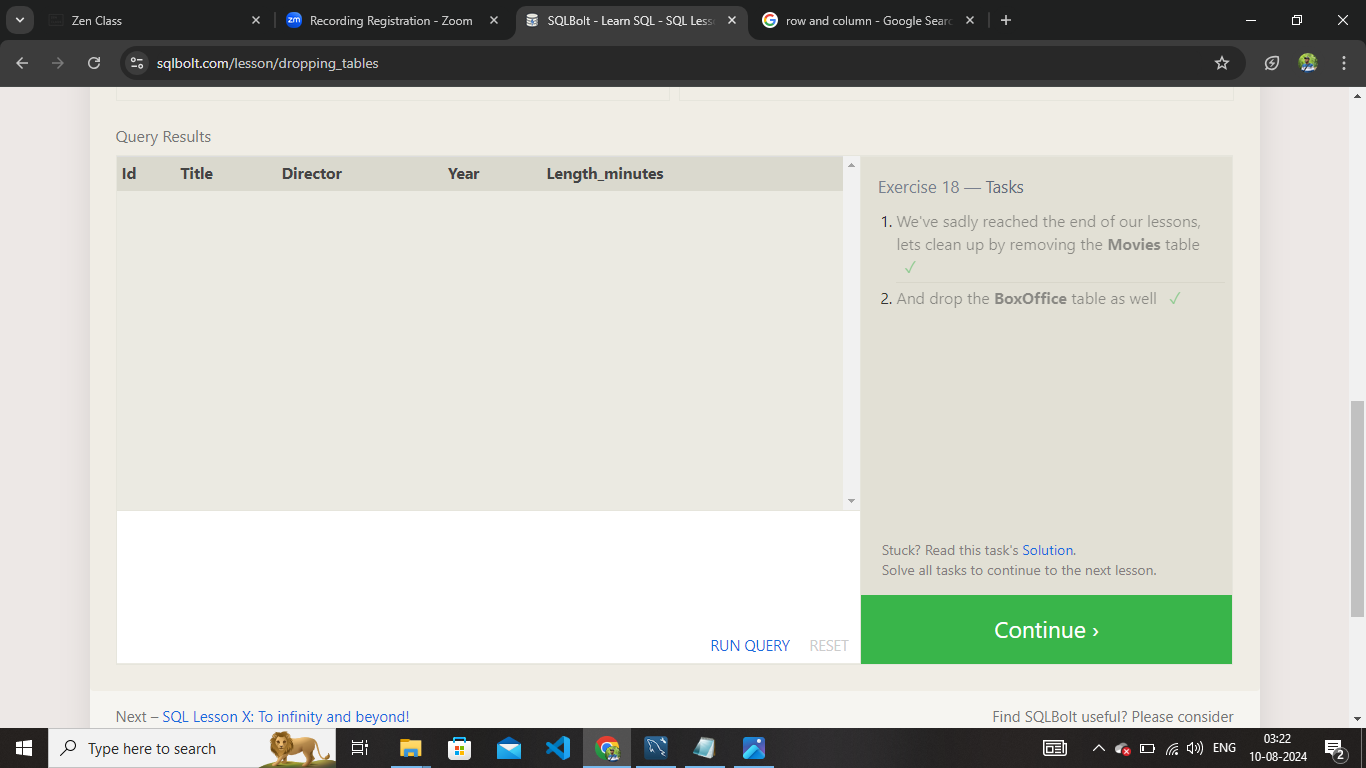
2.alter table movies add language varchar (255) default 'english';



**18.SQL Lesson 18: Dropping tables**

1.drop table movies;

2.drop table boxoffice;



Thank You