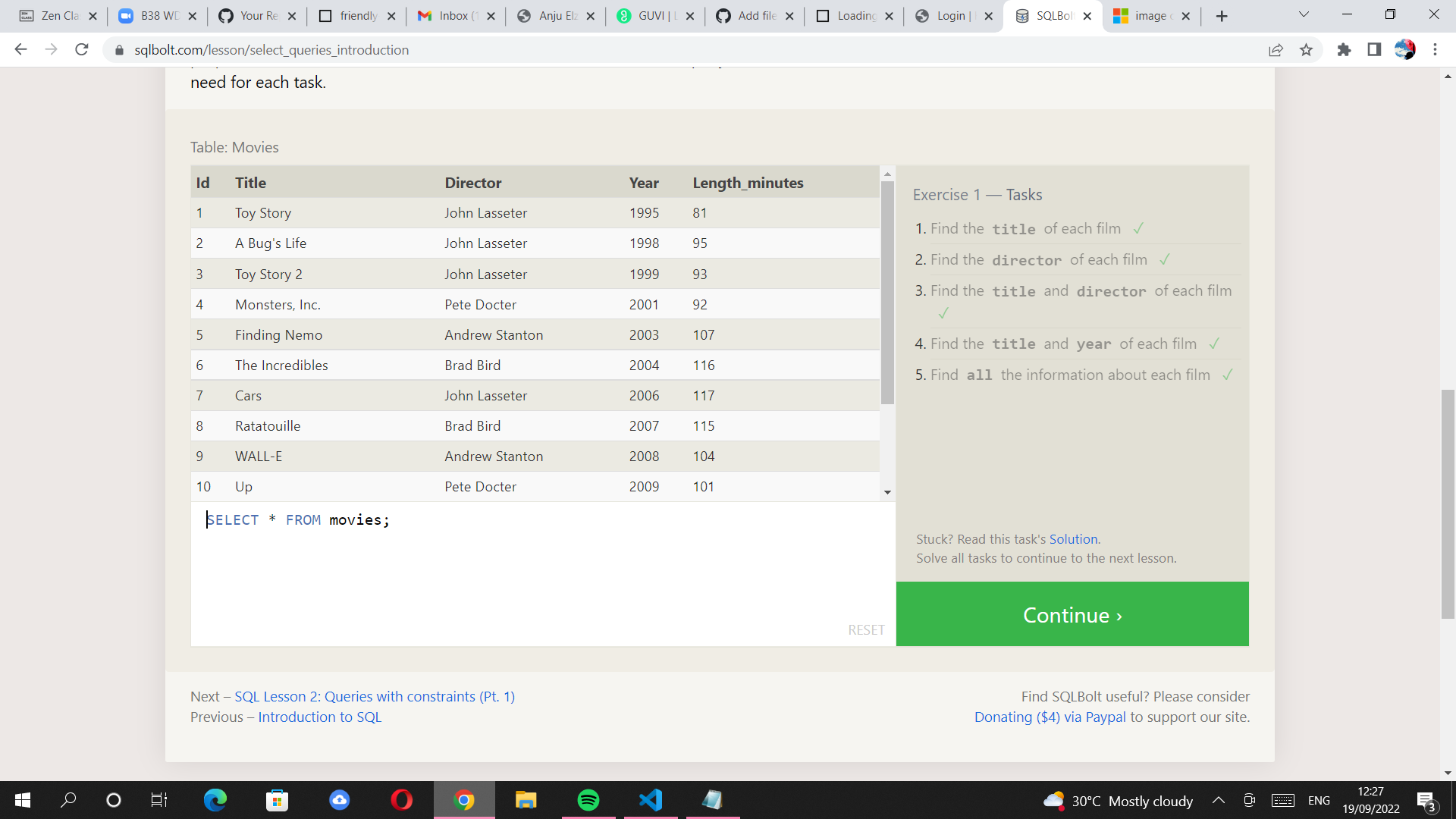
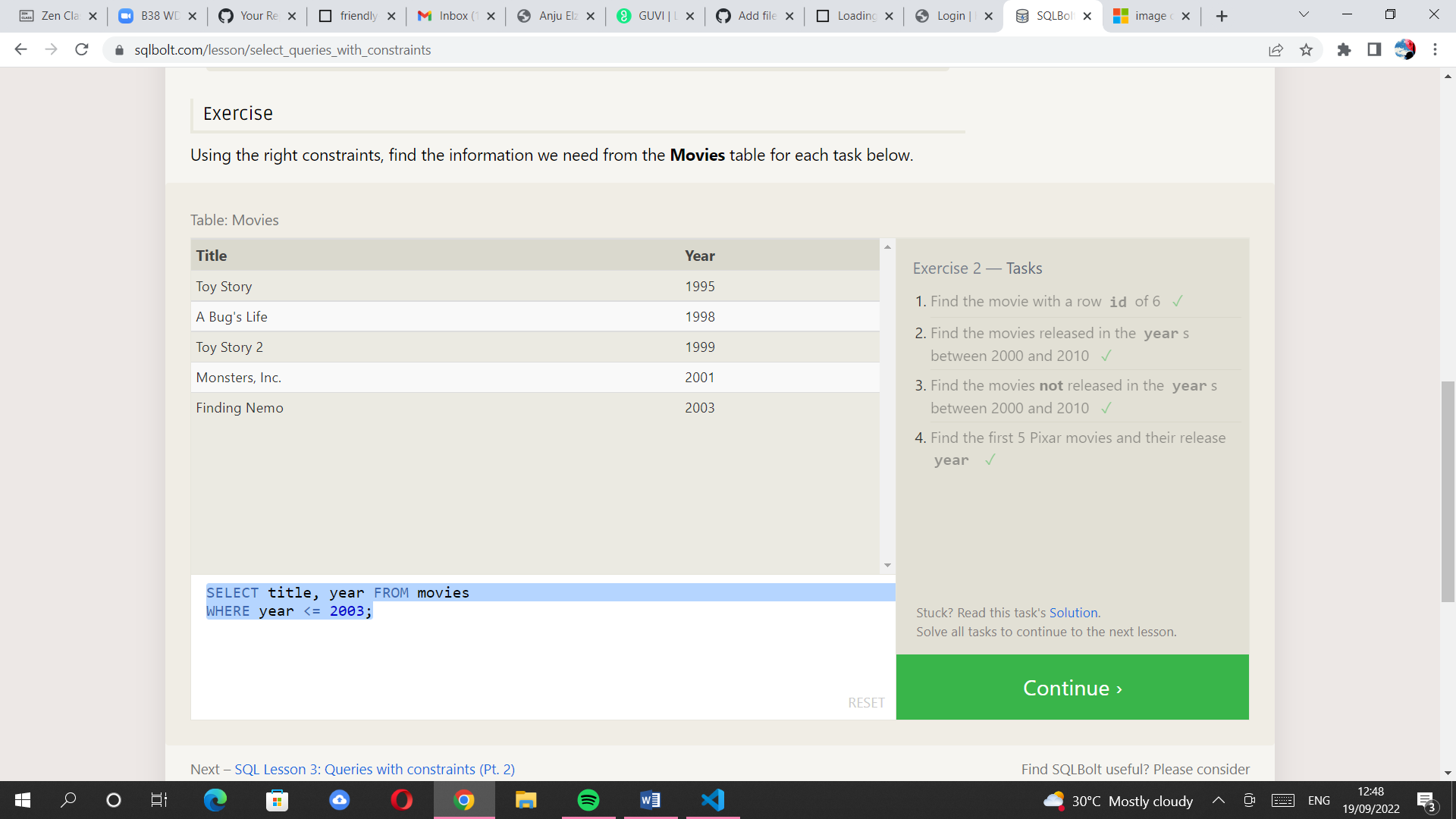
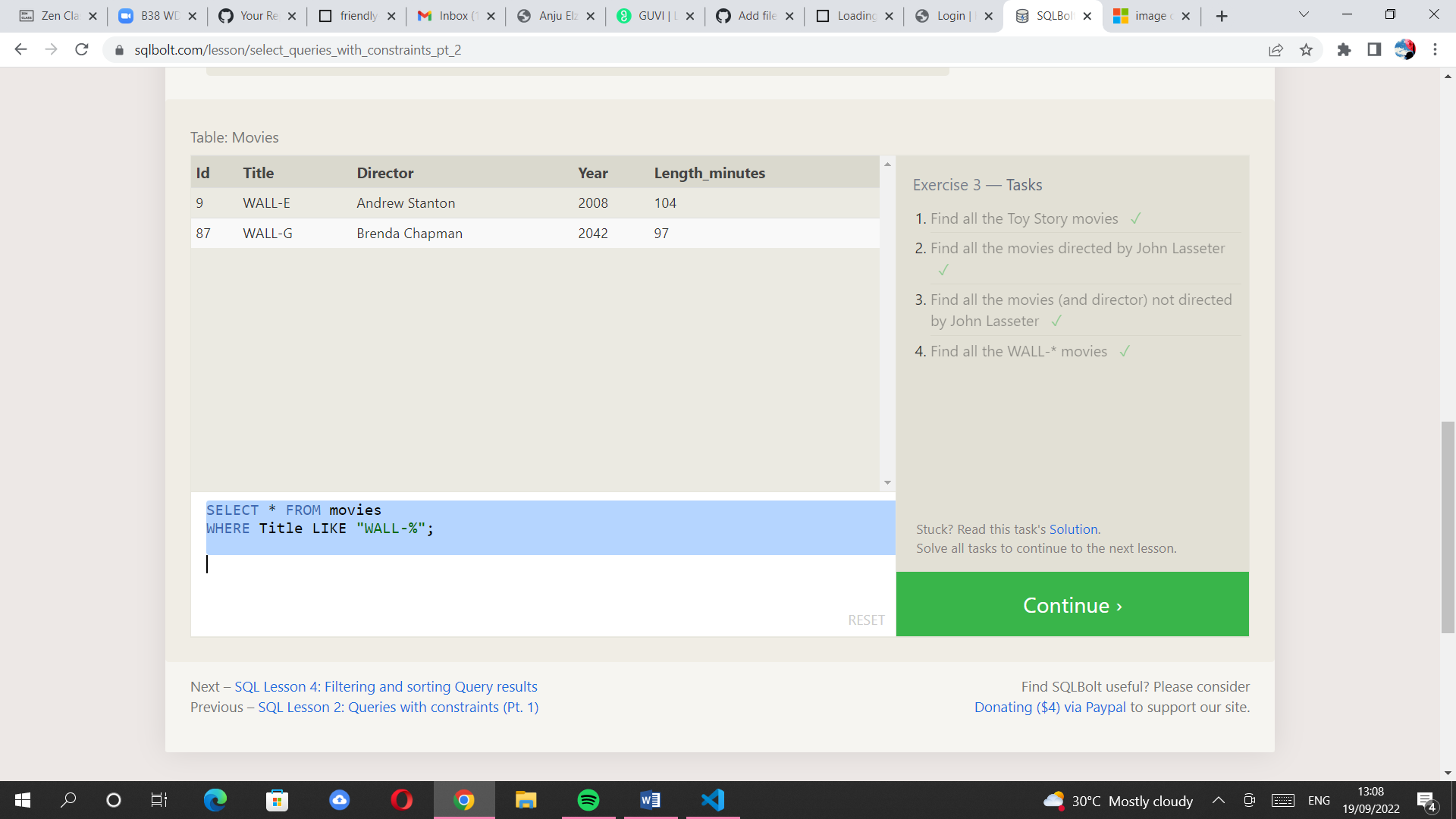
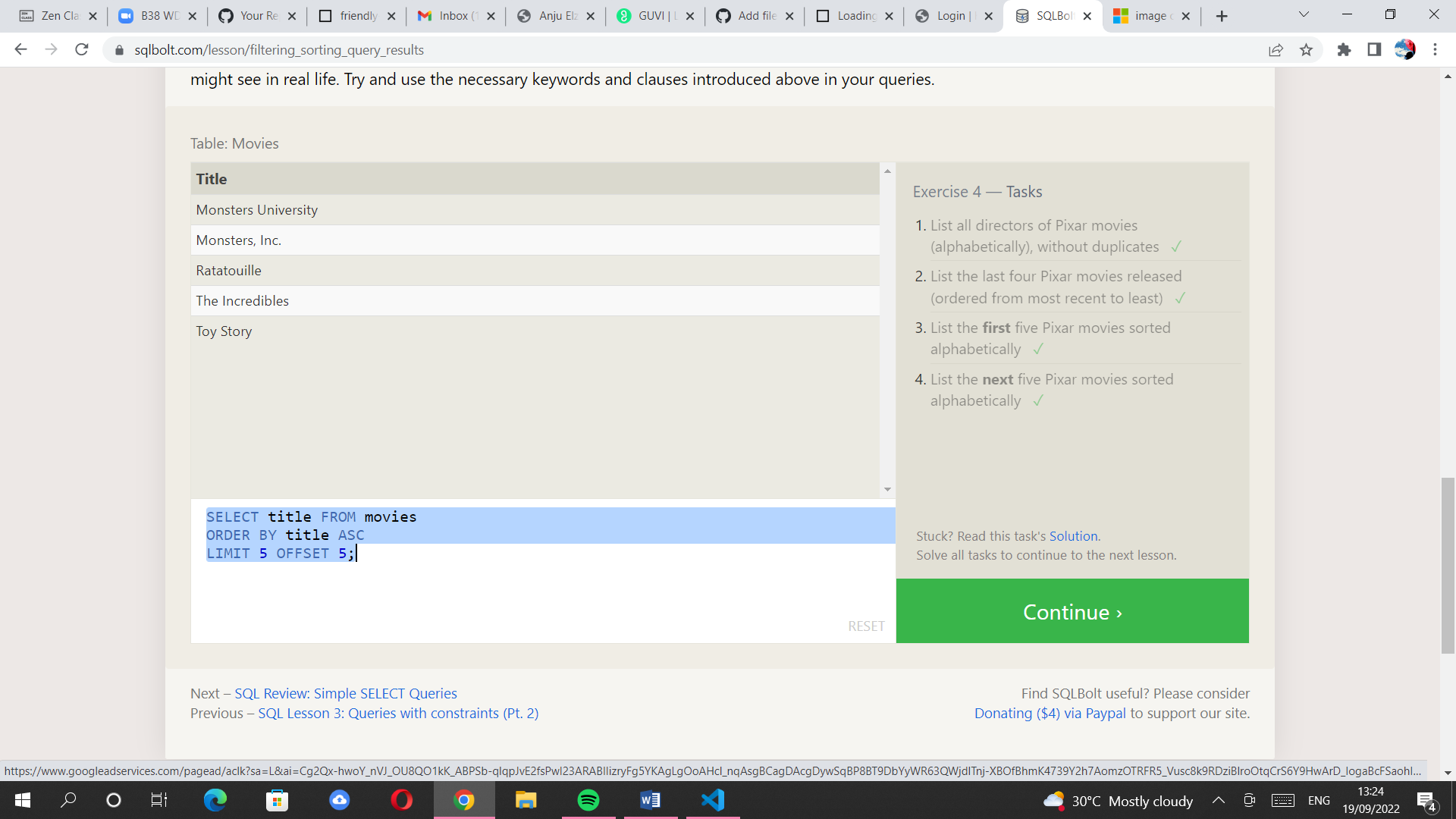
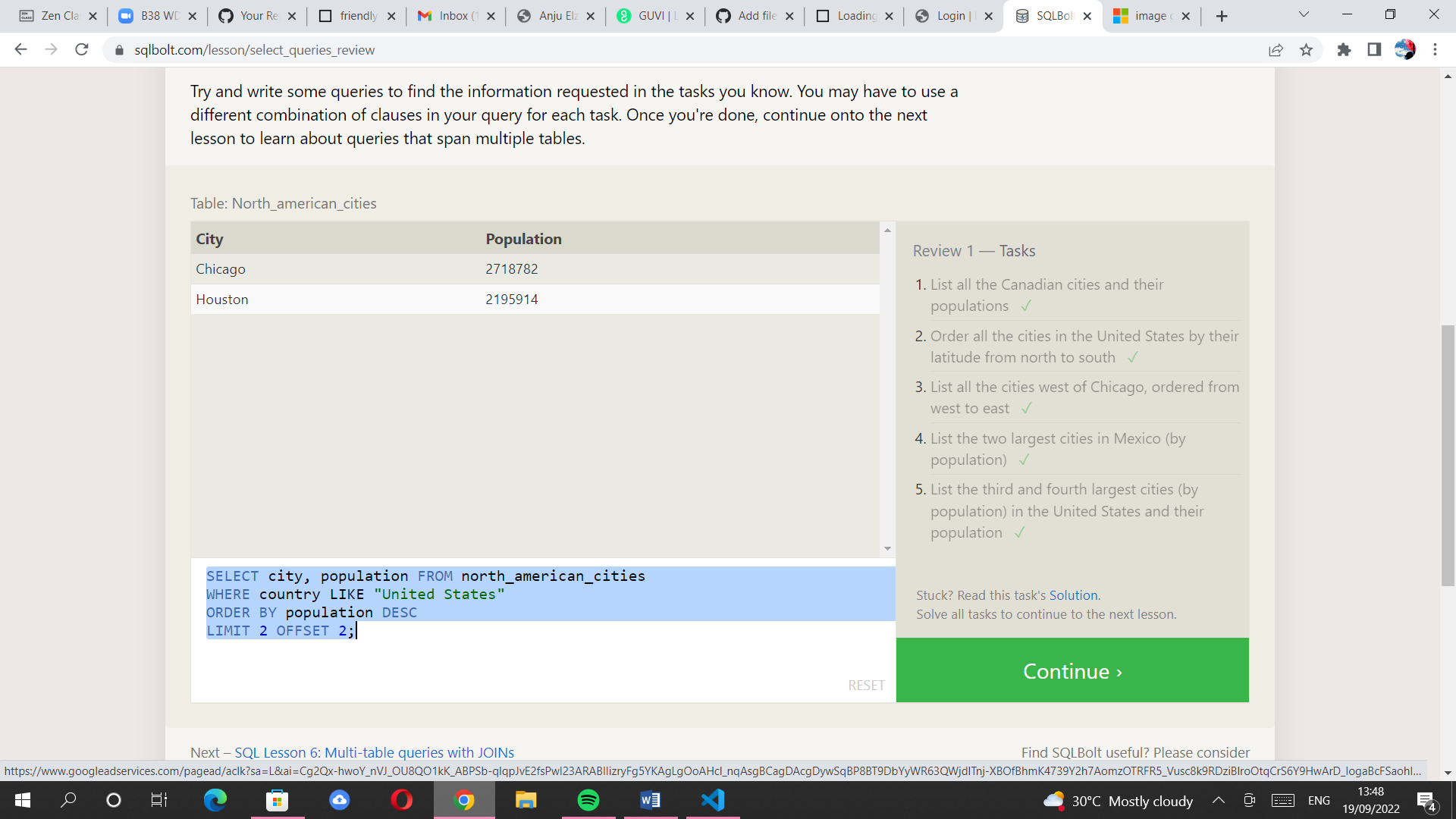
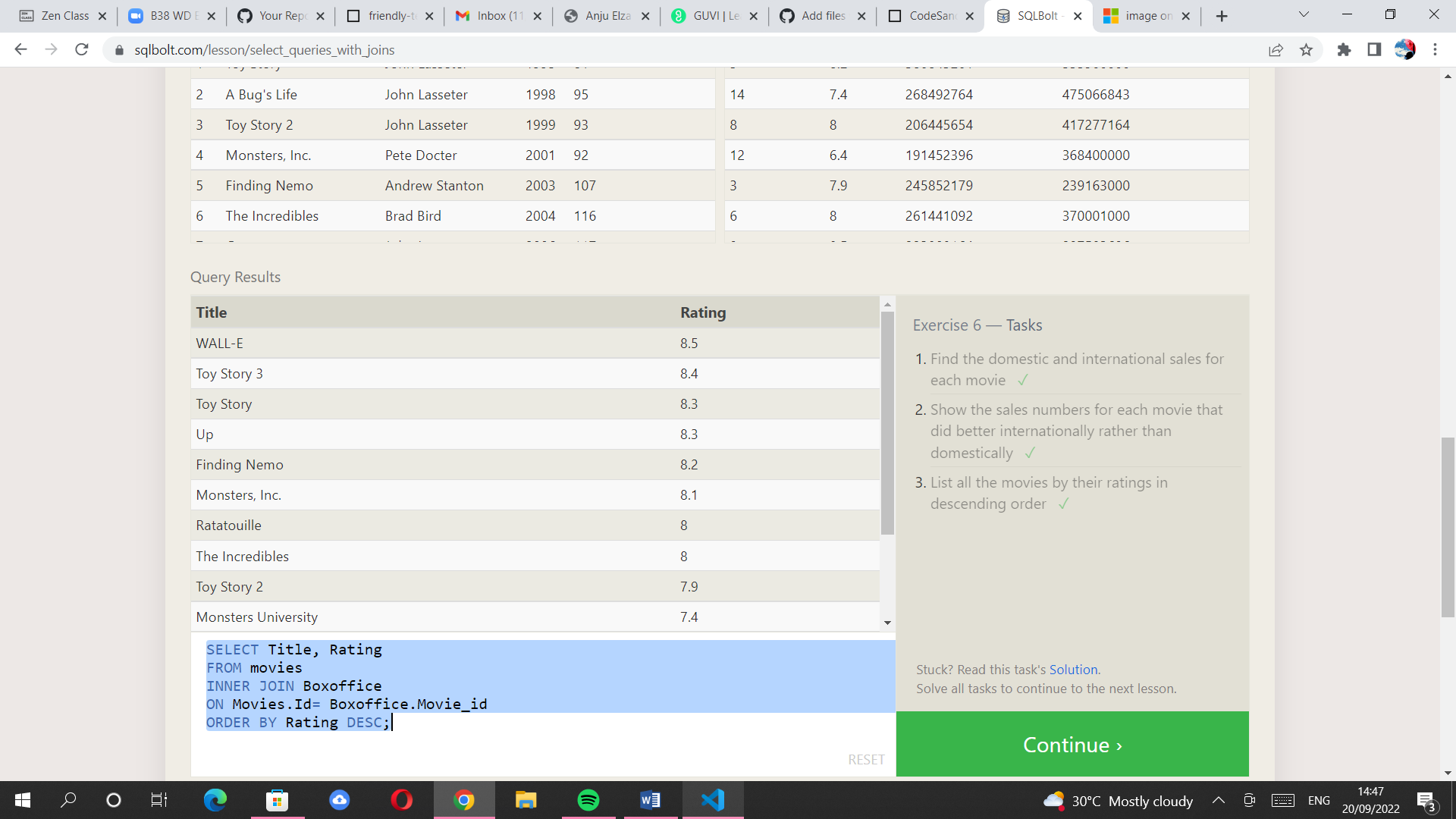
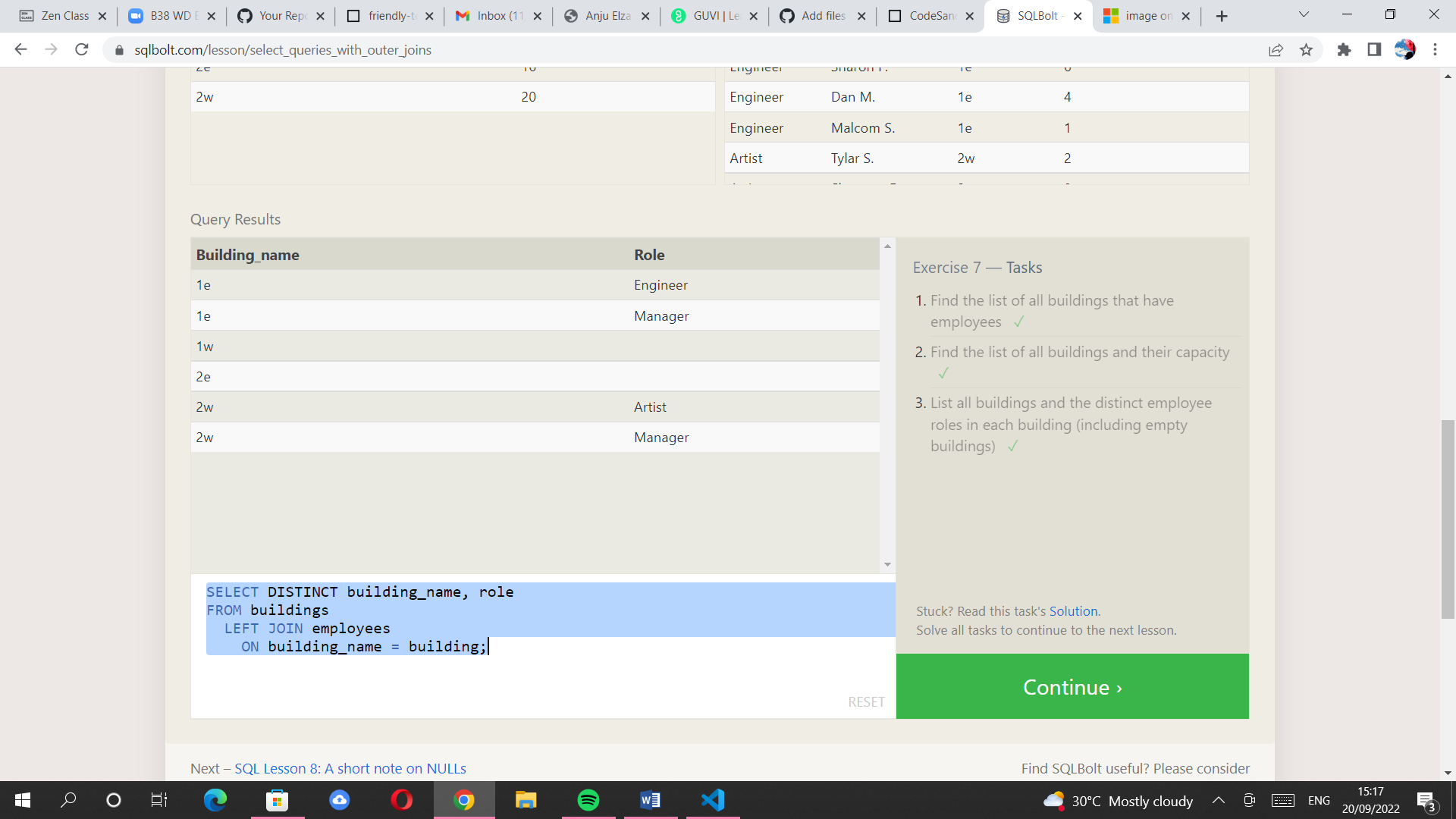
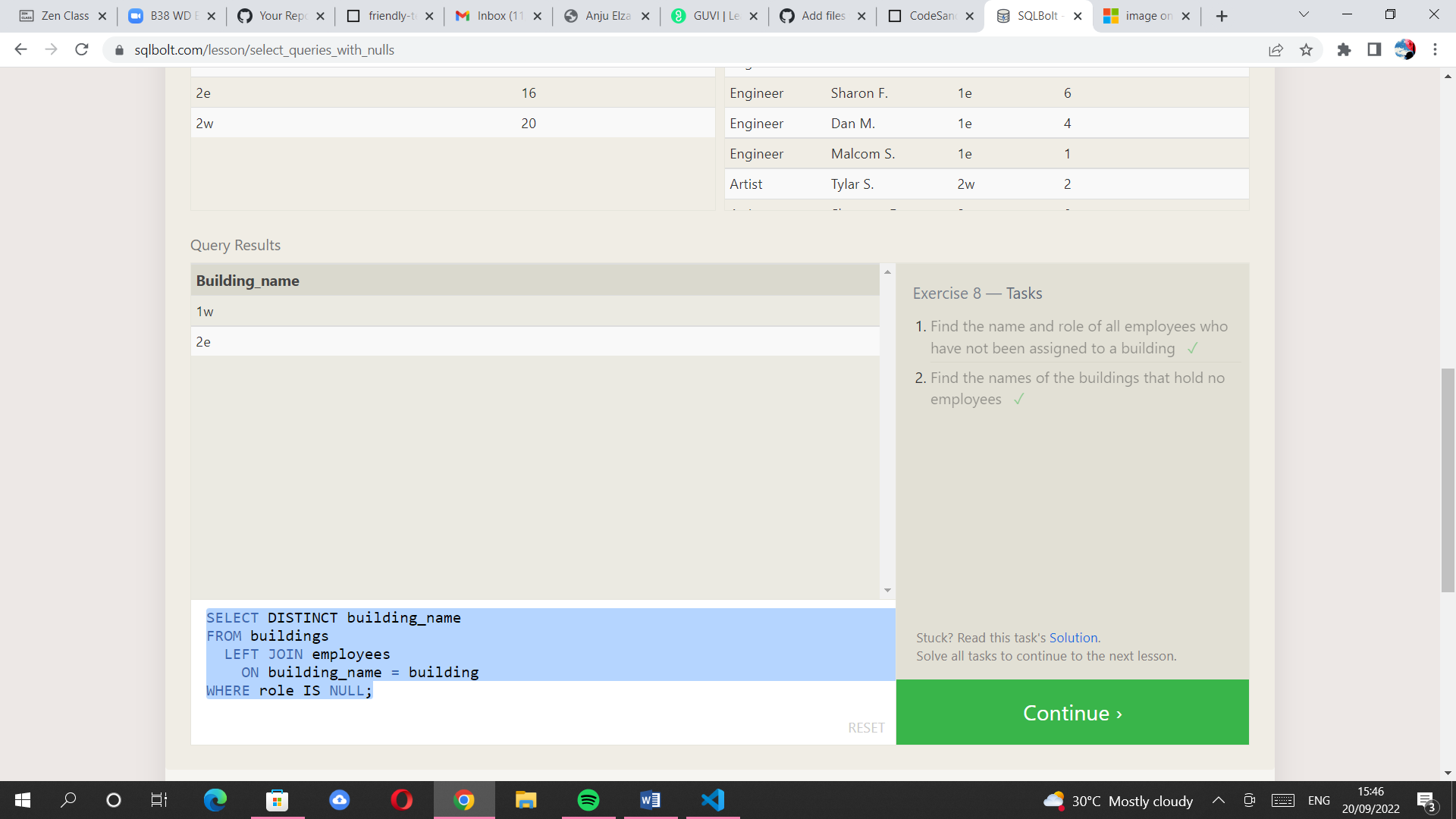
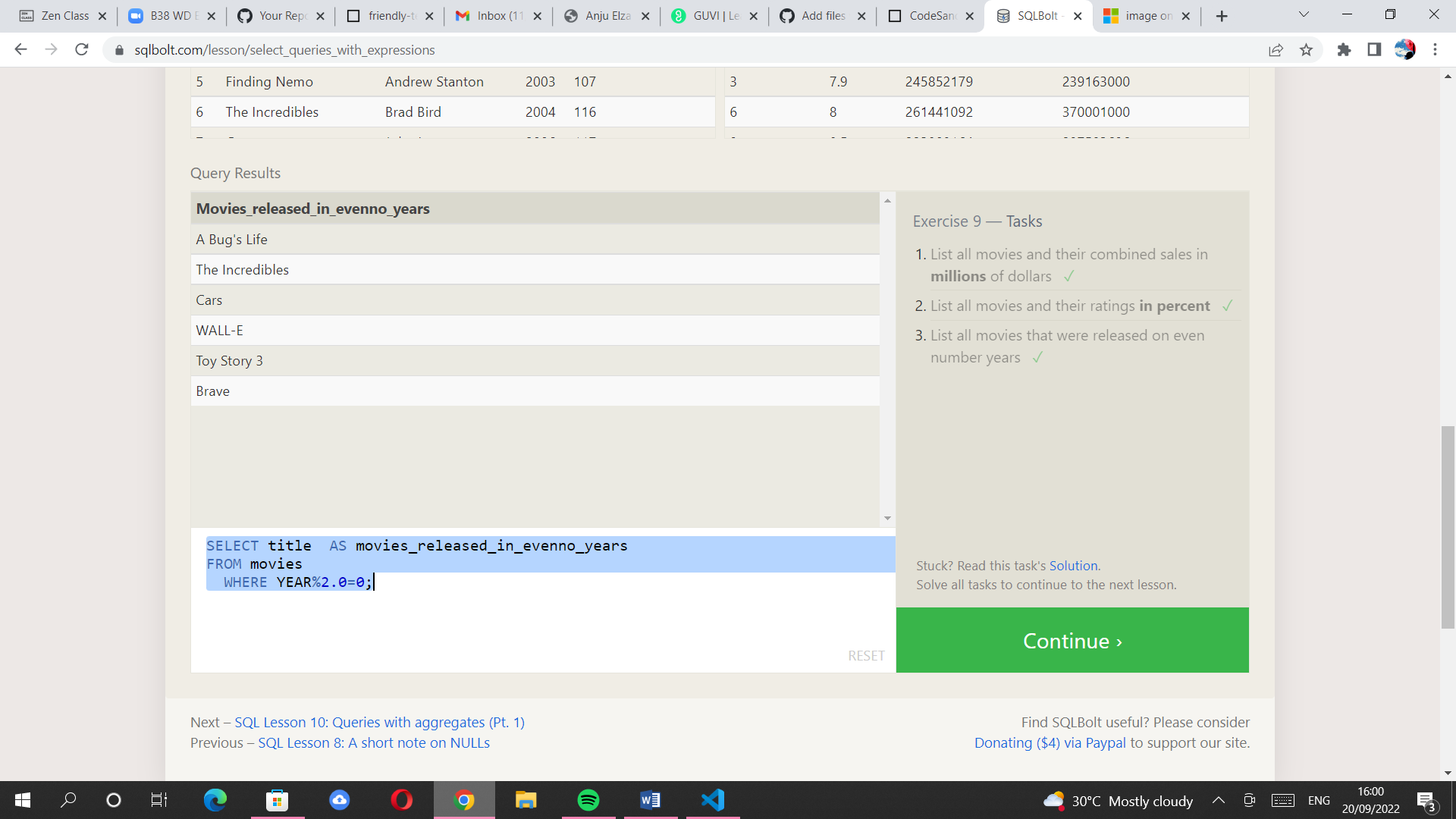
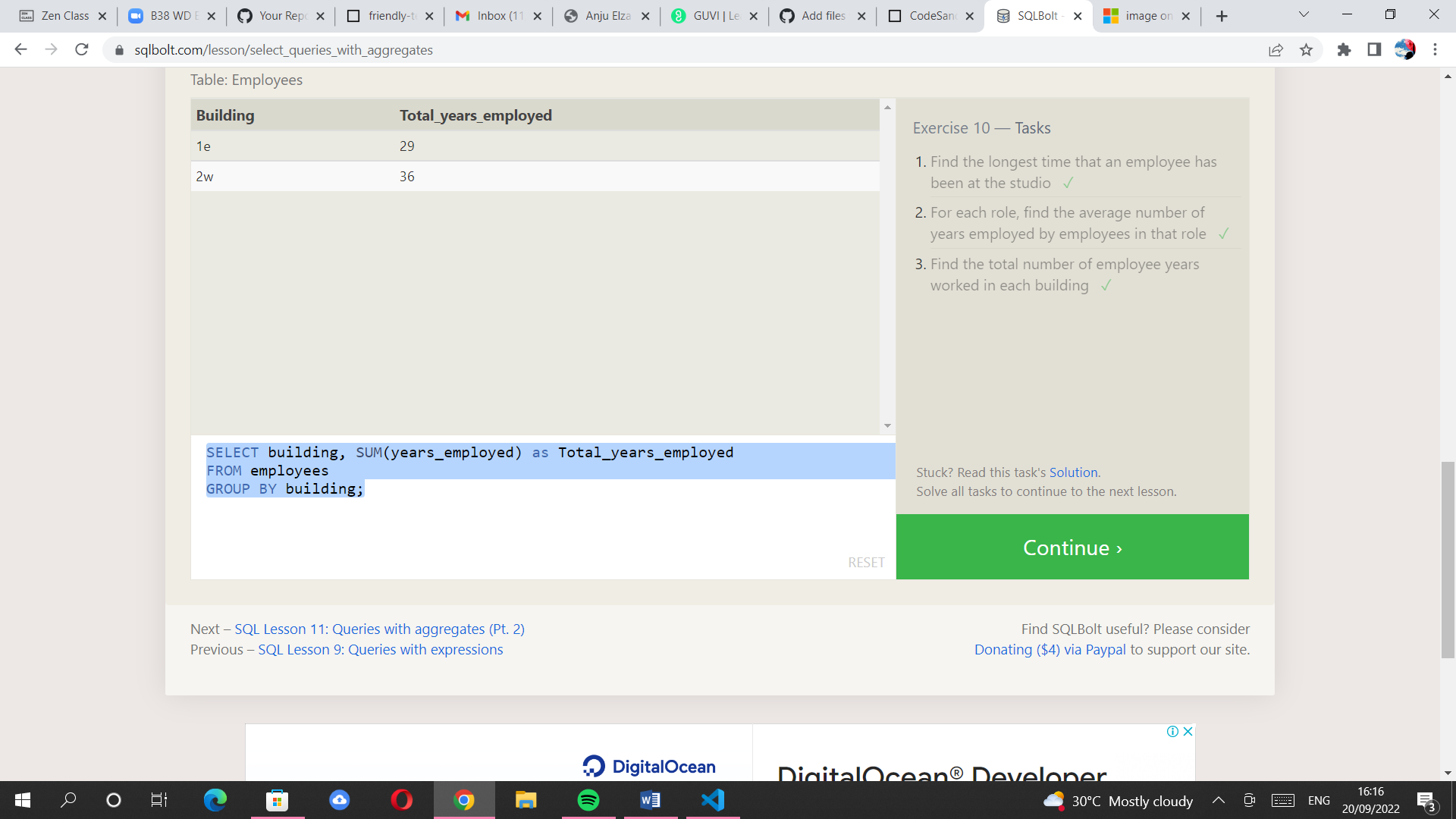
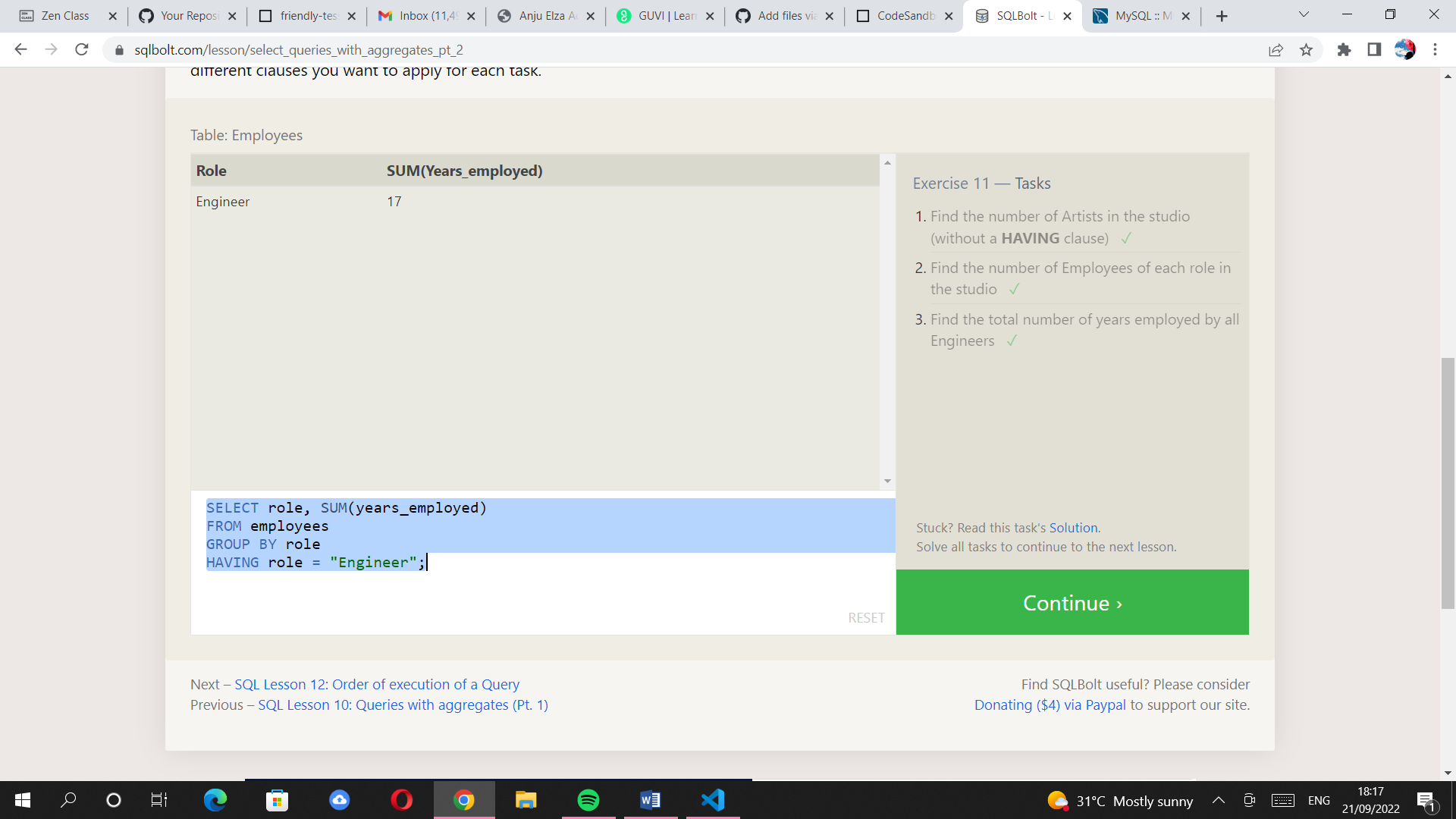
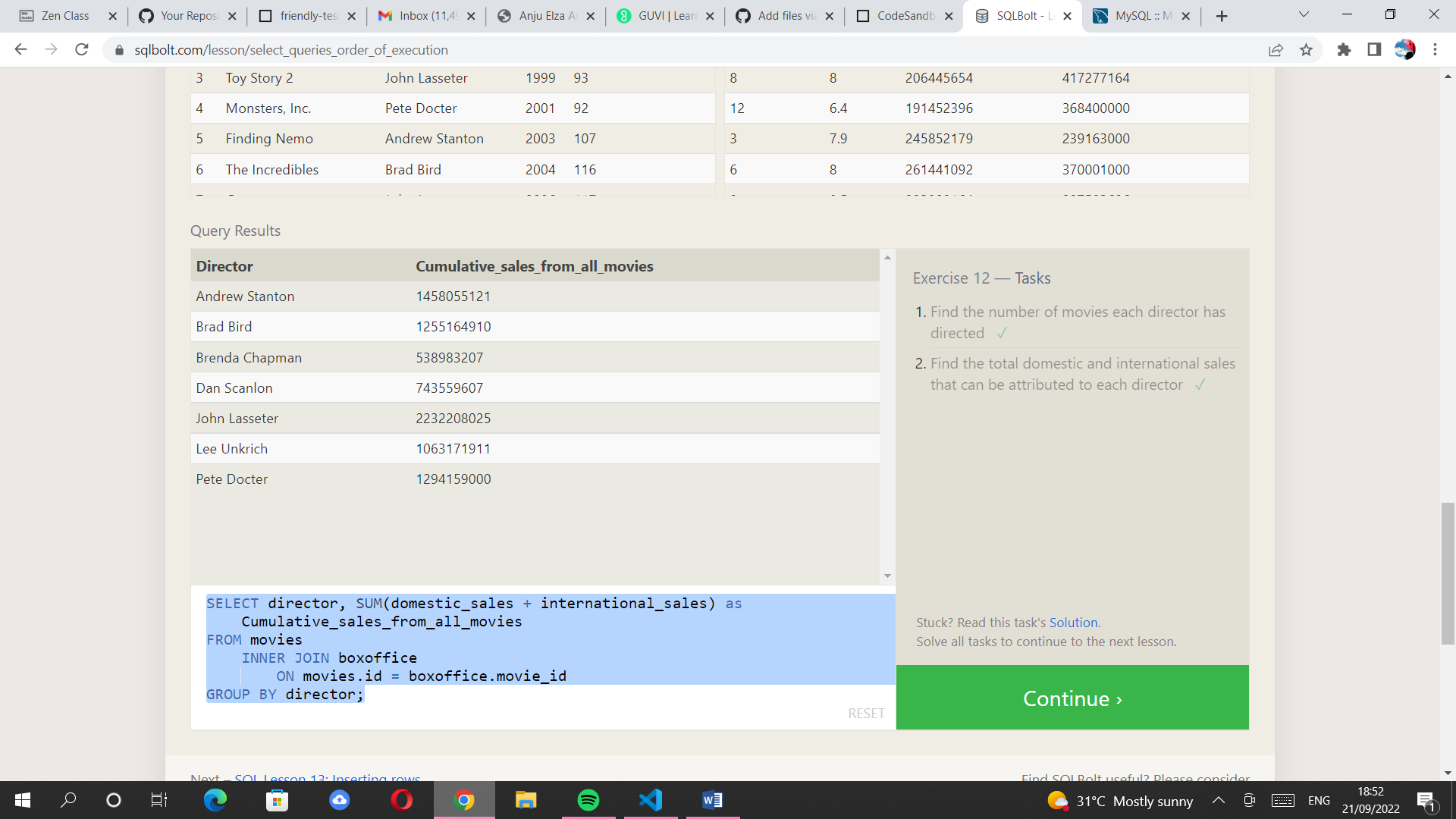
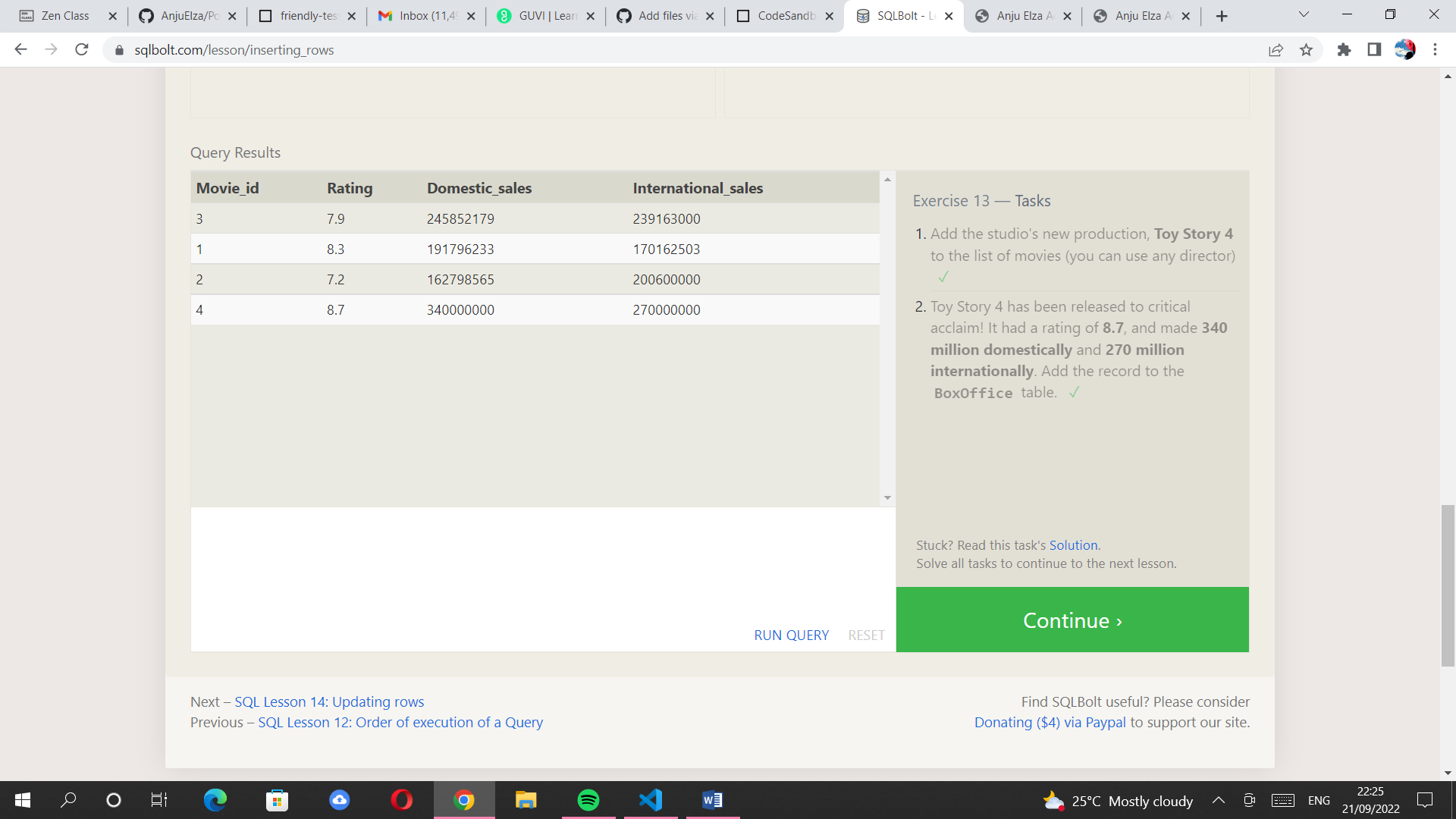
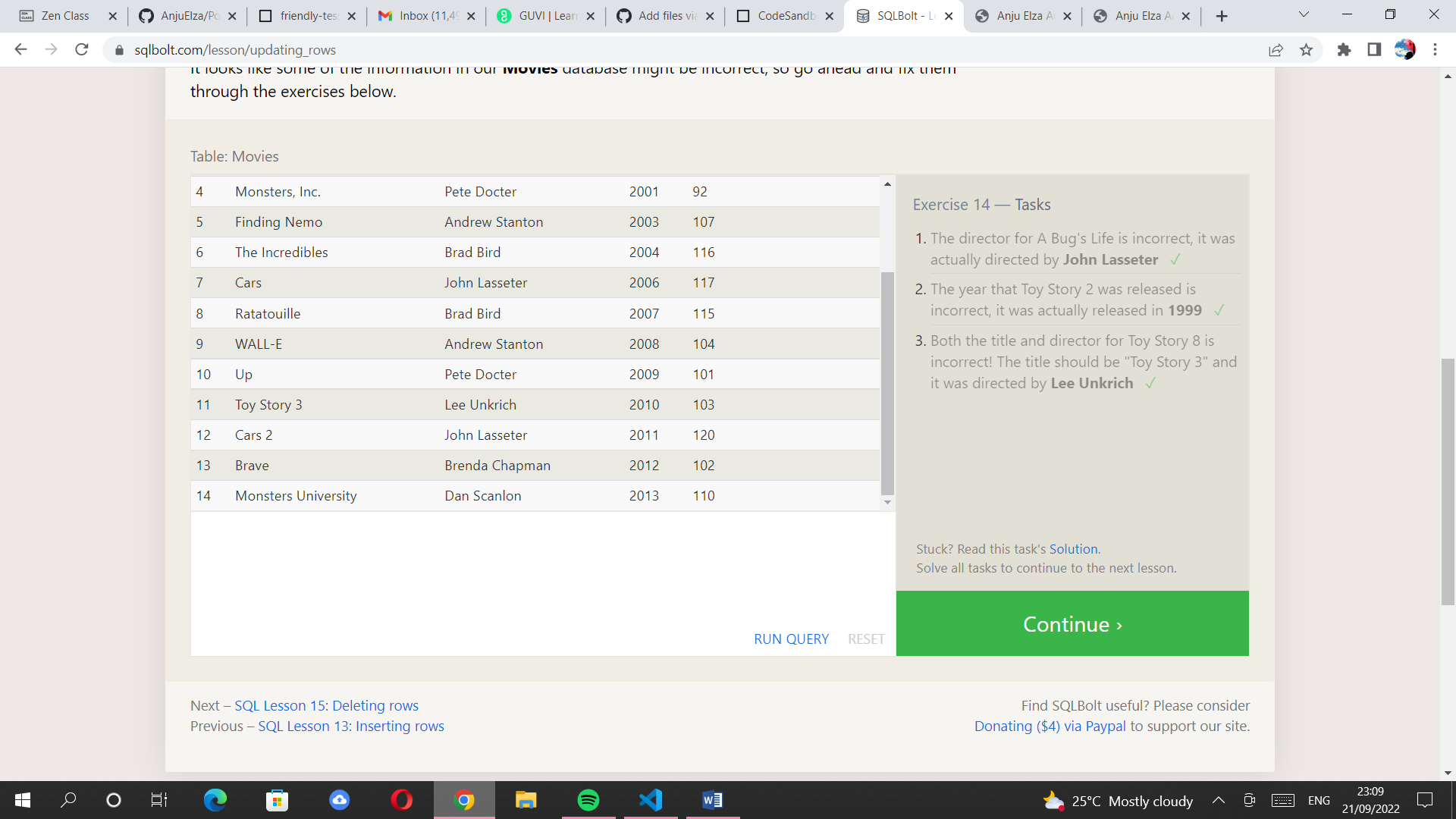
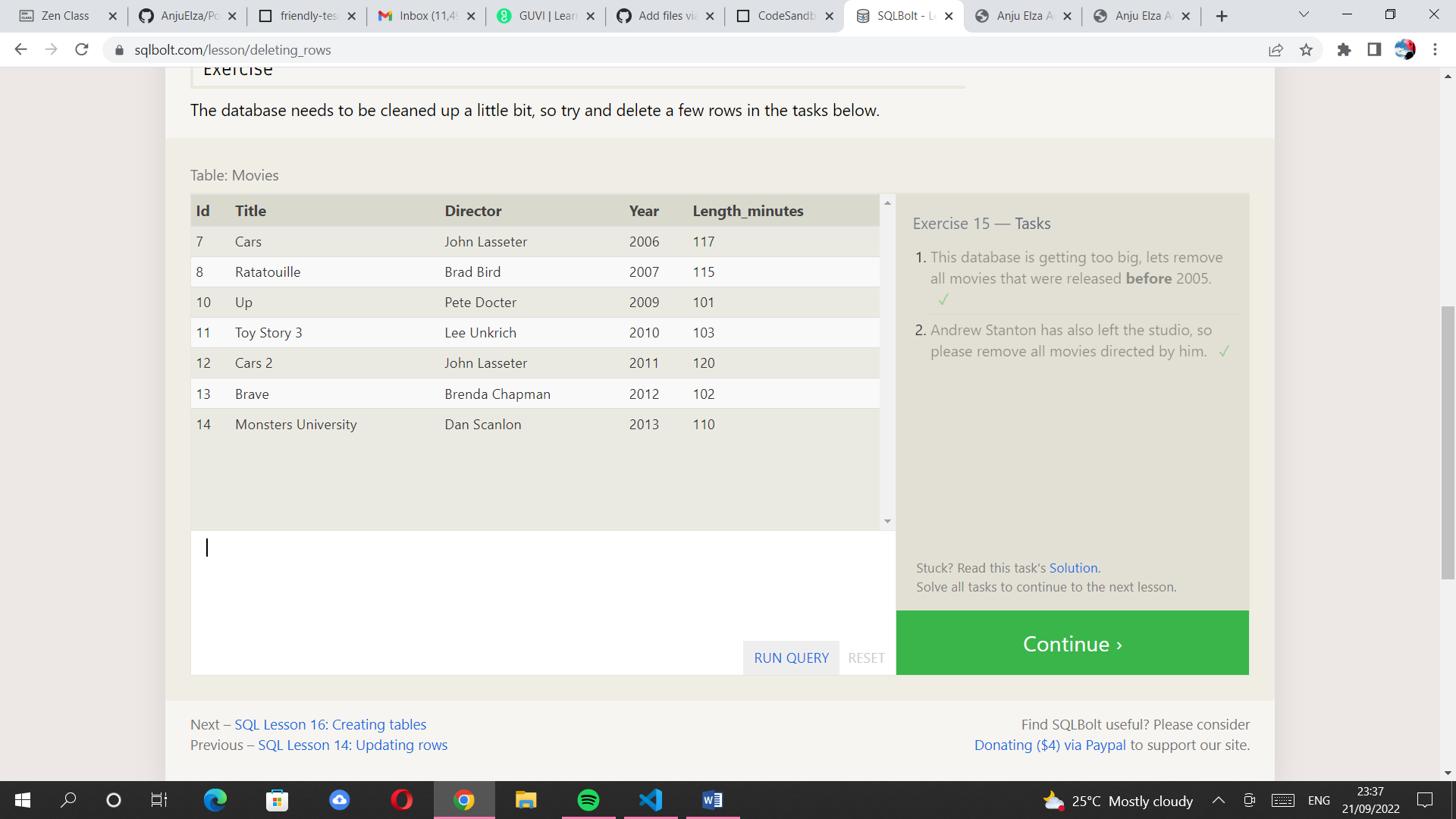
DAY 36 TASK: SQL BOLTS

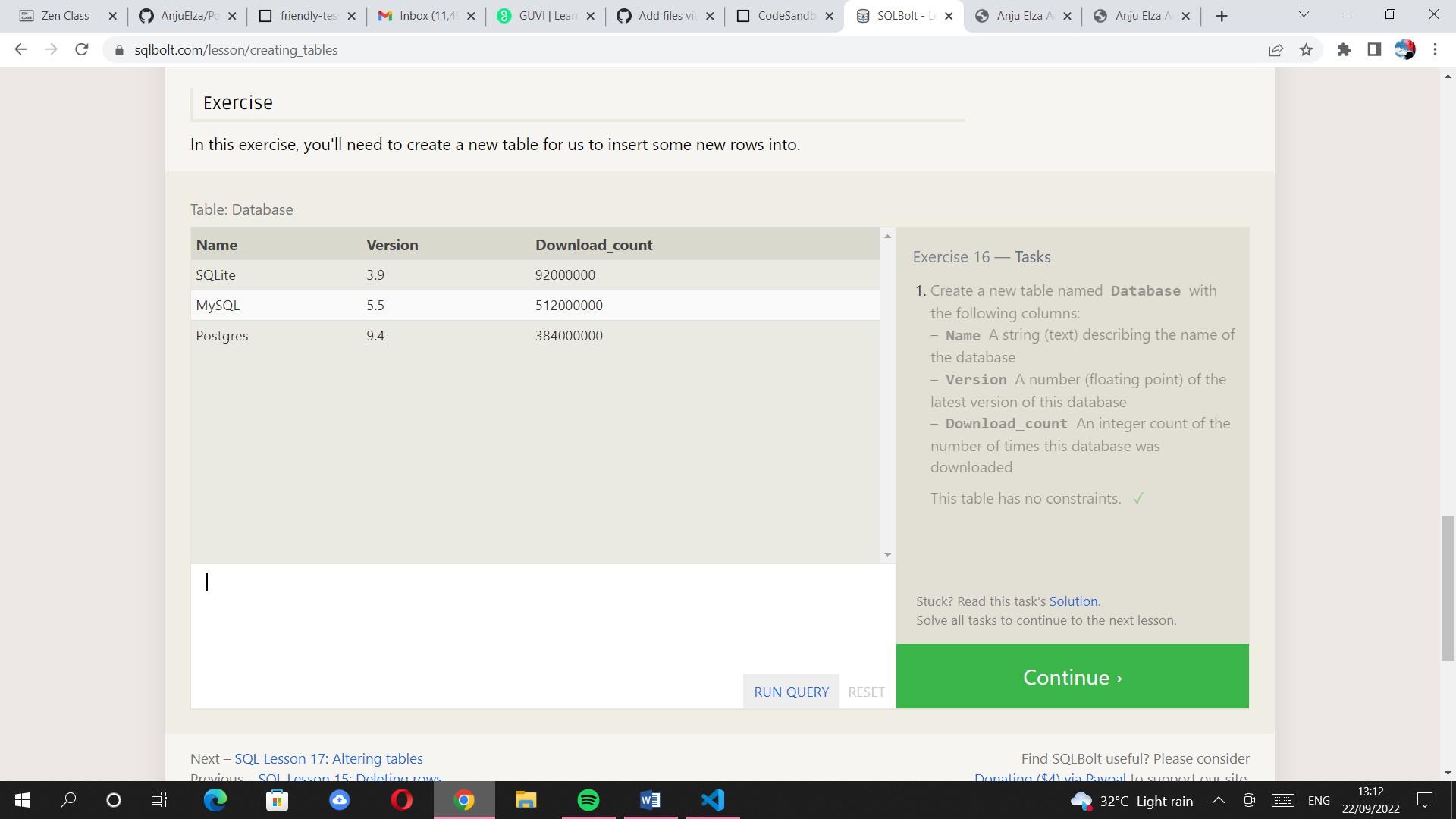
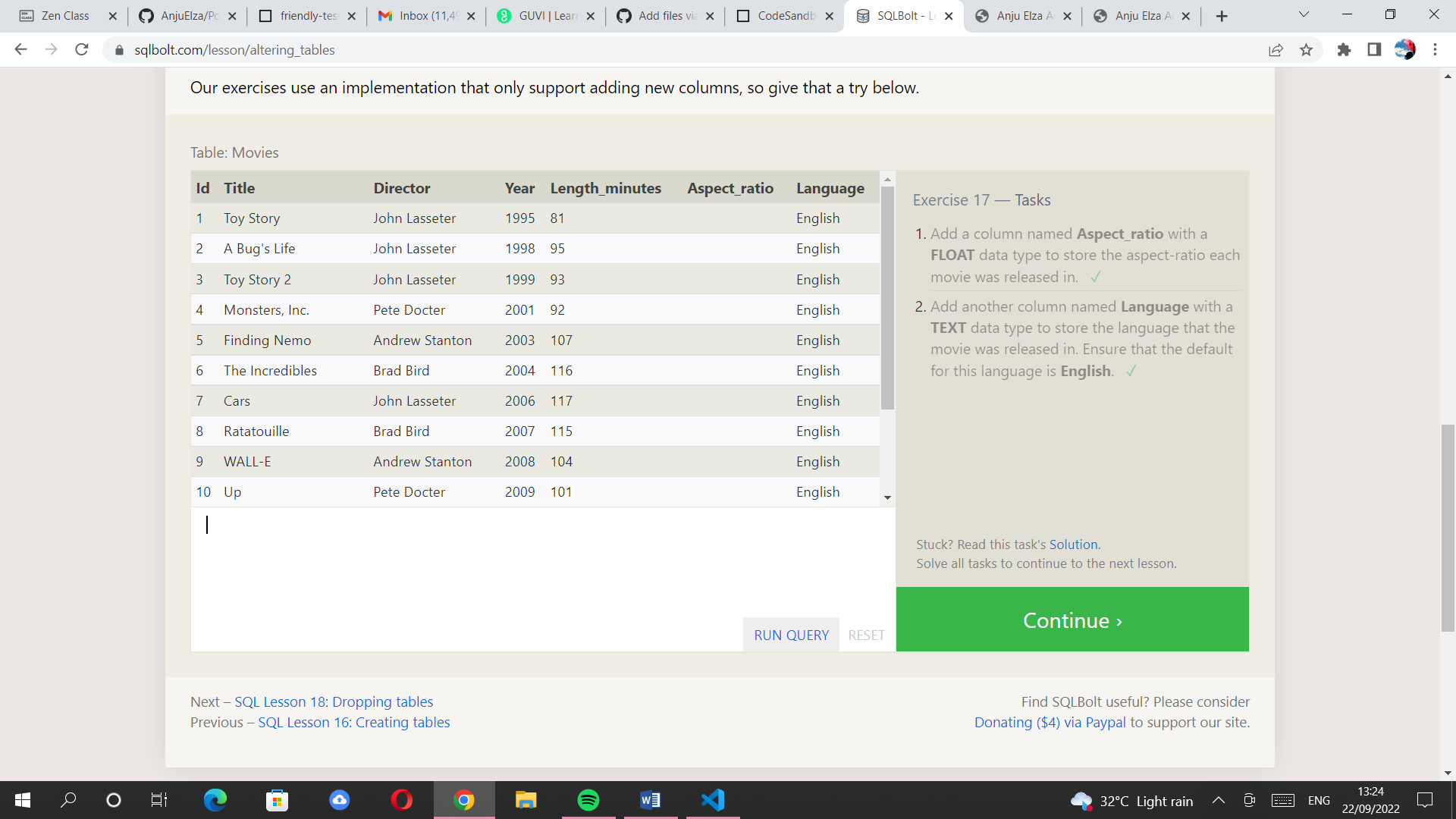
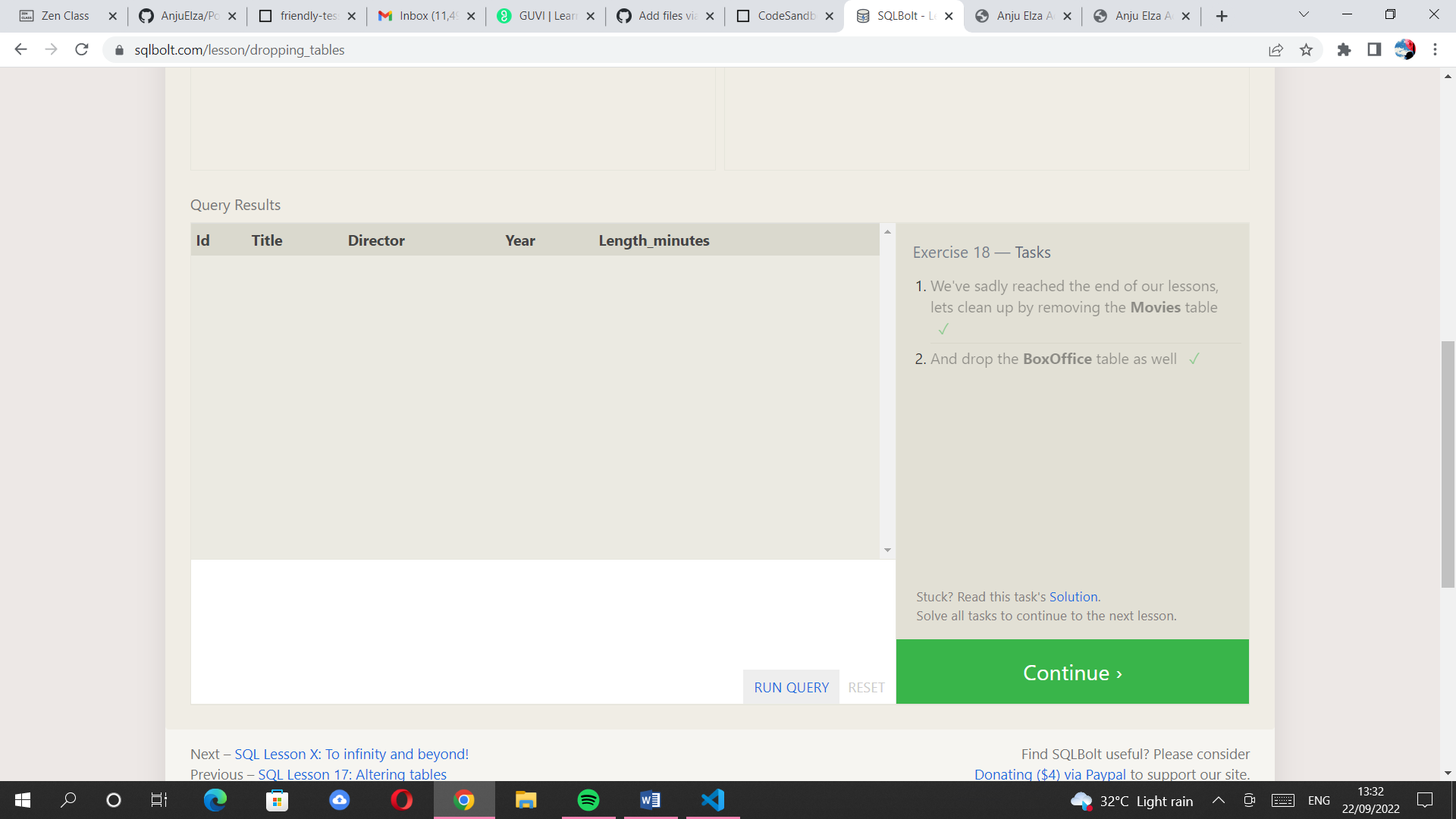
  

 . 

 . 

.  

. . 

//Day 36 task- bolt sql

//EXERCISE-1

//qn1

SELECT Title FROM movies;

//qn2

SELECT Director FROM movies;

//qn3

SELECT Title, Director FROM movies;

//qn4

SELECT Title, Year FROM movies;

//qn5

SELECT \* FROM movies;

//EXERCISE-2

//Qn1

SELECT Title FROM movies WHERE Id=6;

//QN2

SELECT Title FROM movies WHERE Year BETWEEN 2000 AND 2010;

//QN3

SELECT Title FROM movies WHERE Year NOT BETWEEN 2000 AND 2010;

//QN4

SELECT title, year FROM movies

WHERE year <= 2003;

//EXERCISE-3

//QN1

SELECT title, director FROM movies

WHERE title LIKE "Toy Story%";

//qn2

SELECT title FROM movies

WHERE Director = "John Lasseter";

//qn3

SELECT title, Director FROM movies

WHERE Director != "John Lasseter";

//qn4

SELECT \* FROM movies

WHERE Title LIKE "WALL-%";

//EXERCISE 4

//QN1

SELECT DISTINCT Director FROM movies

ORDER BY Director ASC;

//QN2

SELECT Title FROM movies

ORDER BY Year DESC

LIMIT 4;

//QN3

SELECT Title FROM movies

ORDER BY Title ASC

LIMIT 5;

//QN4

SELECT title FROM movies

ORDER BY title ASC

LIMIT 5 OFFSET 5;

//REVIEW1

//QN1

SELECT City, Population FROM north\_american\_cities

WHERE Country="Canada";

//QN2

SELECT city, latitude FROM north\_american\_cities

WHERE country = "United States"

ORDER BY latitude DESC;

//QN3

SELECT city, longitude FROM north\_american\_cities

WHERE longitude < -87.629798

ORDER BY longitude ASC;

//QN4

SELECT city, population FROM north\_american\_cities

WHERE country LIKE "Mexico"

ORDER BY population DESC

LIMIT 2;

//QN5

SELECT city, population FROM north\_american\_cities

WHERE country LIKE "United States"

ORDER BY population DESC

LIMIT 2 OFFSET 2;

//EXERCISE-6

//QN1

SELECT Title, Domestic\_sales, International\_sales

FROM movies

INNER JOIN Boxoffice

ON Movies.Id= Boxoffice.Movie\_id;

//QN2

SELECT Title, Domestic\_sales, International\_sales

FROM movies

INNER JOIN Boxoffice

ON Movies.Id= Boxoffice.Movie\_id

WHERE International\_sales>Domestic\_sales;

//qn3

SELECT Title, Rating

FROM movies

INNER JOIN Boxoffice

ON Movies.Id= Boxoffice.Movie\_id

ORDER BY Rating DESC;

//EXERCISE-7

//QN1

SELECT DISTINCT building FROM employees;

//QN2

SELECT DISTINCT Building\_name, Capacity

FROM Buildings;

//qn3

SELECT DISTINCT Building\_name, Capacity

FROM Buildings;

//EXERCISE-8

//QN1

SELECT NAME,ROLE

FROM employees

WHERE Building IS NULL;

//QN2

SELECT DISTINCT building\_name

FROM buildings

  LEFT JOIN employees

    ON building\_name = building

WHERE role IS NULL;

//exercise-9

//qn1

SELECT title, (domestic\_sales + international\_sales) / 1000000 AS gross\_sales\_millions

FROM movies

  JOIN boxoffice

    ON movies.id = boxoffice.movie\_id;

//qn2

SELECT title, rating\*10 AS rate\_percentage

FROM movies

  JOIN boxoffice

    ON movies.id = boxoffice.movie\_id;

//qn3

SELECT title  AS movies\_released\_in\_evenno\_years

FROM movies

  WHERE YEAR%2.0=0;

//EXERCISE 10

//QN1

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

FROM employees;

//QN2

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

FROM employees

GROUP BY role;

//qn3

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

FROM employees

GROUP BY building;

//EXERCISE-11

//QN1

SELECT role, COUNT(\*) AS Number\_of\_artists

FROM employees

WHERE role = "Artist";

//QN2

SELECT role, COUNT(\*) as Number

FROM employees

GROUP BY role;

//qn3

SELECT role, SUM(years\_employed)

FROM employees

GROUP BY role

HAVING role = "Engineer";

//EXERCISE-12

//QN1

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

FROM movies

GROUP BY director;

//QN2

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;

//EXERCISE-13

//qn1

INSERT INTO Movies

VALUES(4, Toy Story 4, John Mathew, 98);

//QN2

INSERT INTO Boxoffice

VALUES( 4, 8.7, 340000000, 270000000);

EXERCISE-14

//QN1

UPDATE Movies

SET Director="John Lasseter"

WHERE Id=2;

//qn2

UPDATE Movies

SET Year=1999

WHERE Id=3;

//qn3

UPDATE Movies

SET Title="Toy Story 3",

    Director="Lee Unkrich"

    WHERE Id=11;

//EXERCISE-15

//QN1

DELETE FROM movies

where year < 2005;

//QN2

DELETE FROM movies

where Director= "Andrew Stanton";

EXERCISE-16

//QN1

 CREATE TABLE Database (

    Name TEXT,

    Version FLOAT,

    Download\_count INTEGER

);

//EXERCISE-17

//qn1

ALTER TABLE Movies

  ADD COLUMN Aspect\_ratio FLOAT ;

  //qn2

  ALTER TABLE Movies

  ADD COLUMN Language TEXT DEFAULT "English";

//EXERCISE-18

//qn1

DROP TABLE Movies;

//qn2

DROP TABLE BoxOffice;