**SQL Lesson 1: SELECT queries 101**

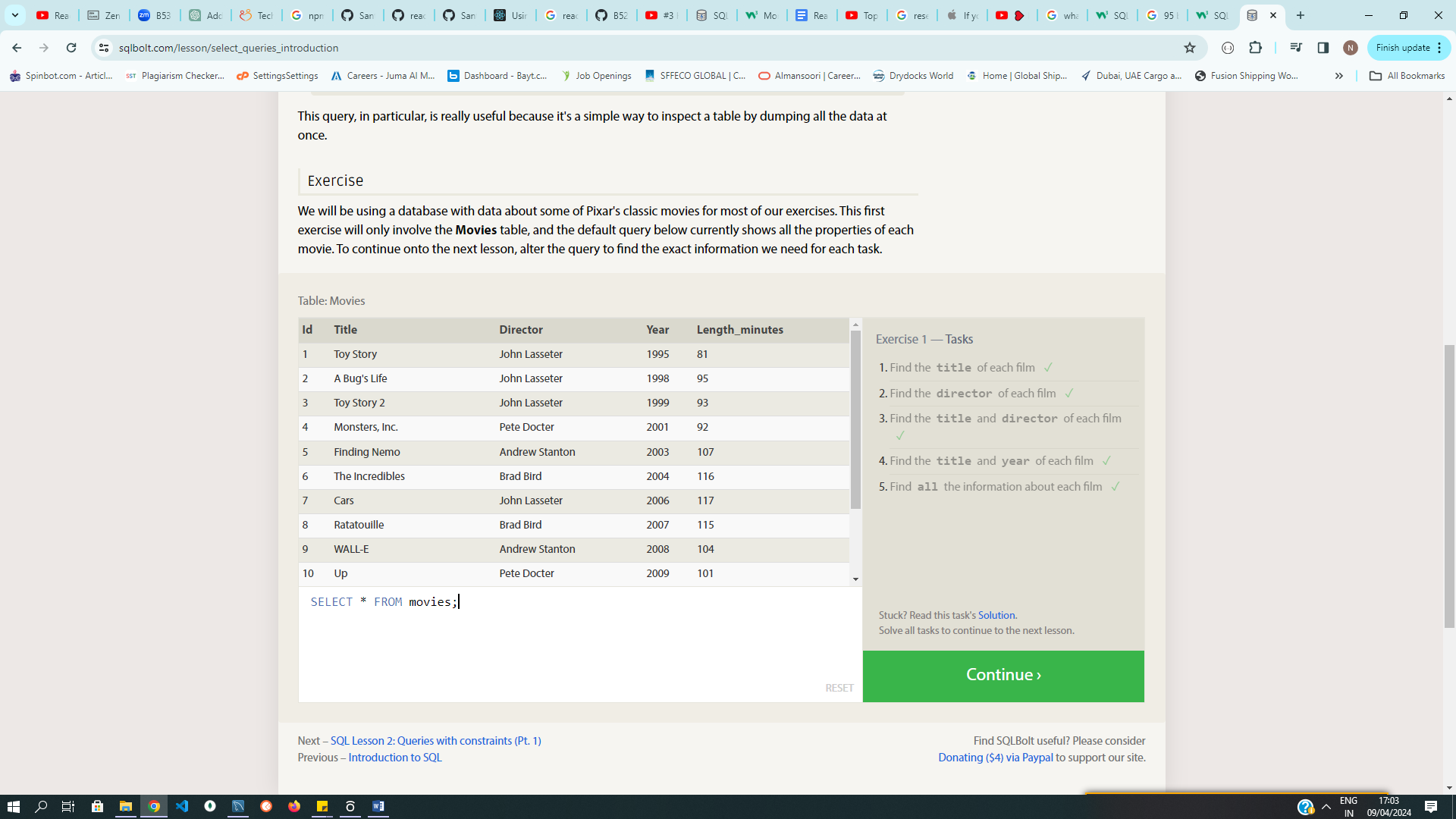
1.SELECT title FROM movies;

2.SELECT director FROM movies;

3.SELECT title,director FROM movies;

4.SELECT title,year FROM movies;

5.SELECT \* FROM movies;



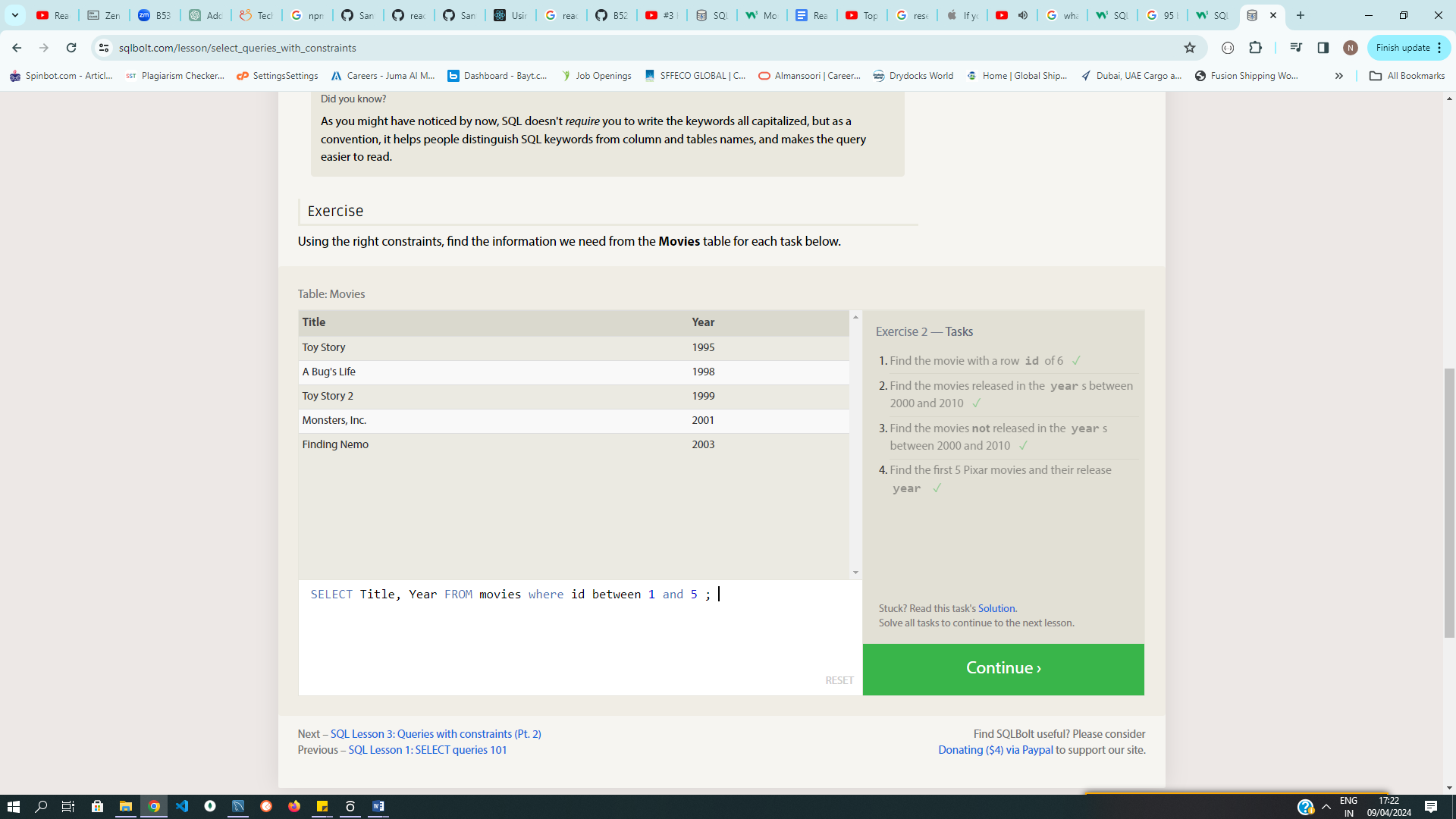
**SQL Lesson 2: Queries with constraints (Pt. 1)**

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 Title, Year FROM movies where id between 1 and 5 ;

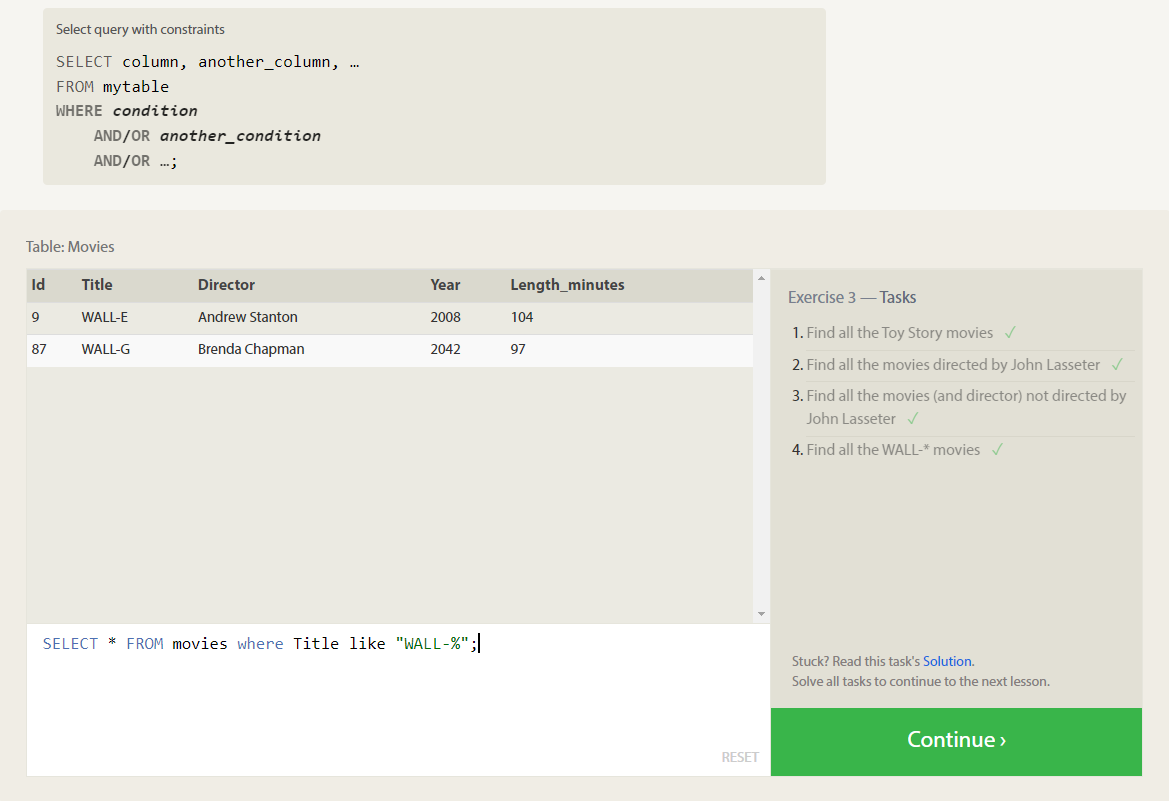


**SQL Lesson 3: Queries with constraints (Pt. 2)**

1.SELECT \* FROM movies where Title like "Toy Story%";

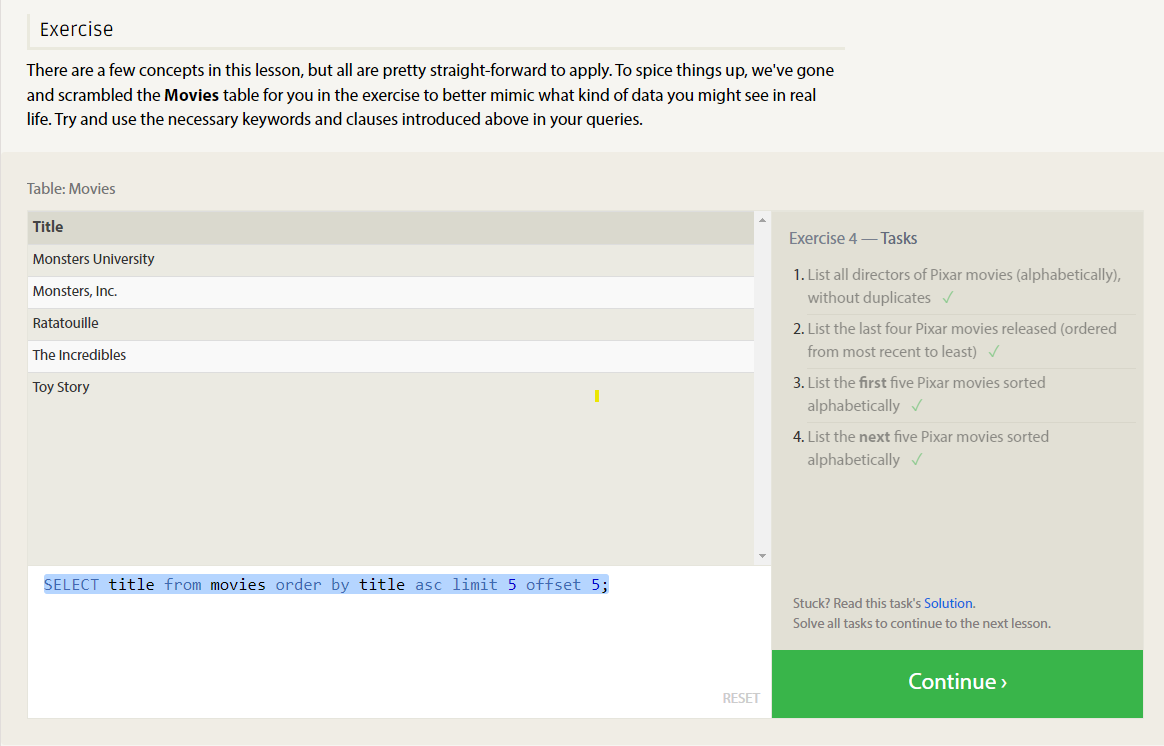
2.SELECT \* FROM movies where Director like "John Lasseter";

3.SELECT title, director FROM movies where Director != "John Lasseter";

4.SELECT \* FROM movies where Title like "WALL-%";  
  


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

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

2.SELECT title 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;  
  


**SQL Review: Simple SELECT Queries**

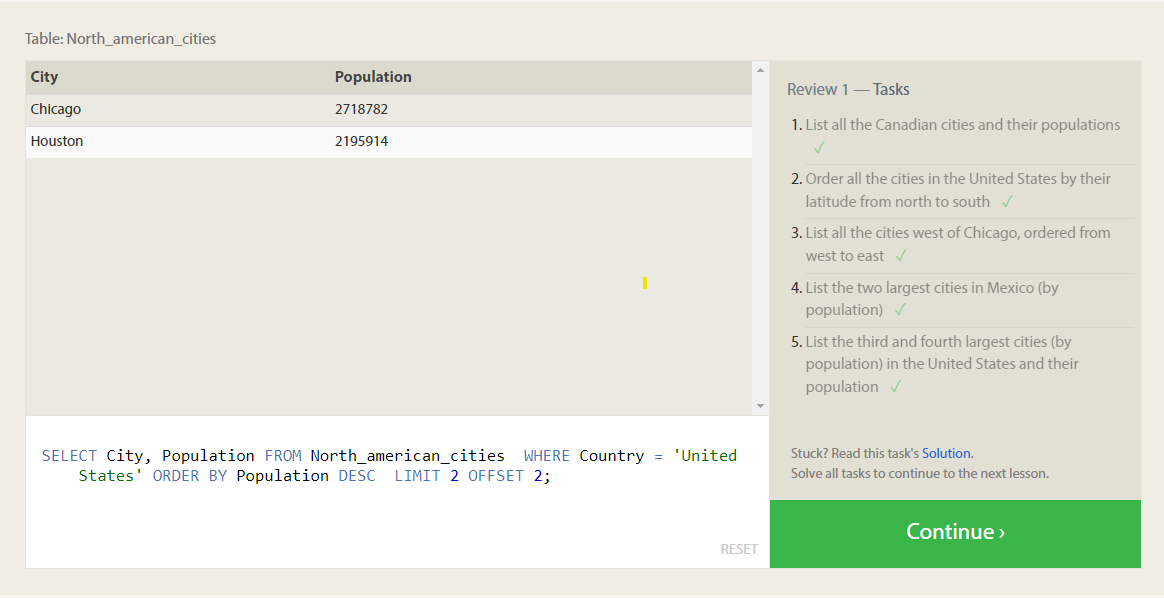
1.SELECT City, Population FROM North\_american\_cities WHERE Country = 'Canada';

2.SELECT city FROM north\_american\_cities where country= "United States" order by latitude desc;

3.SELECT city FROM north\_american\_cities where longitude < (select longitude from north\_american\_cities where city="Chicago") order by longitude asc;

4.SELECT city,population FROM north\_american\_cities where country ="Mexico" order by population desc limit 2;

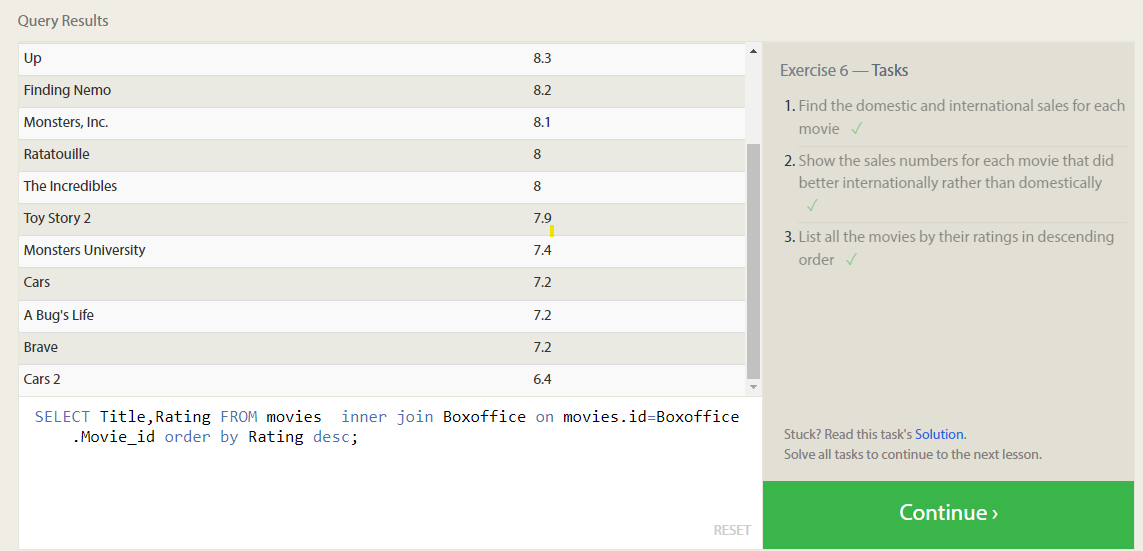
5.SELECT City, Population FROM North\_american\_cities WHERE Country = 'United States' ORDER BY Population DESC LIMIT 2 OFFSET 2;



**SQL lesson 6: Multi-table queries with joins:**  
  
1.SELECT Title,Domestic\_sales,International\_sales FROM movies inner join Boxoffice on movies.id=Boxoffice.Movie\_id;

2.SELECT Title,International\_sales,Domestic\_sales FROM movies inner join Boxoffice on movies.id=Boxoffice.Movie\_id where International\_sales > Domestic\_sales;

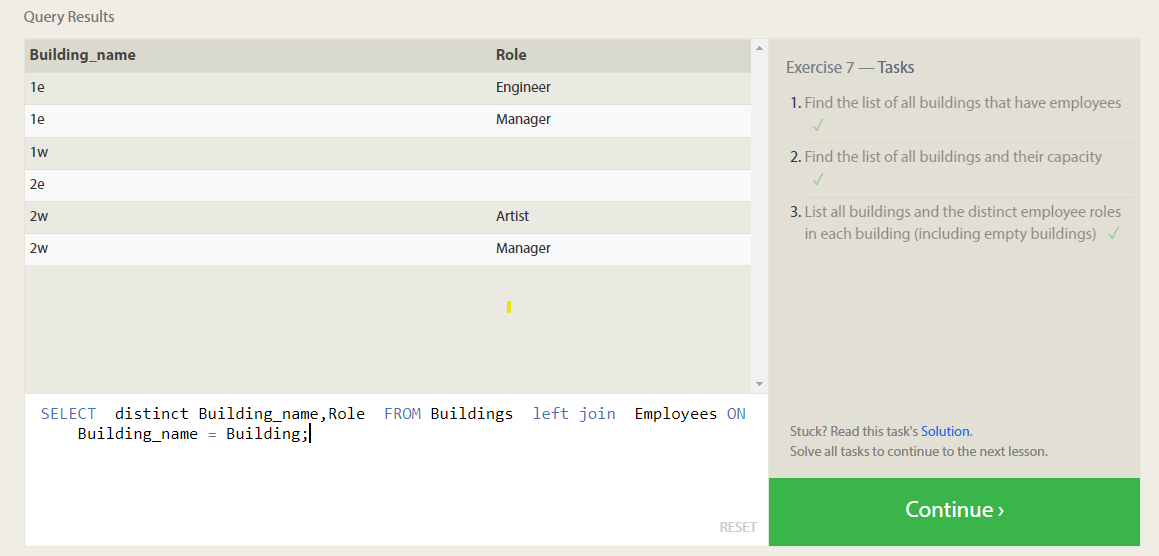
3.SELECT Title,Rating FROM movies inner join Boxoffice on movies.id=Boxoffice.Movie\_id order by Rating desc;



**SQL Lesson 7: Outer Joins:**

1.SELECT Distinct Building,capacity FROM employees inner join Buildings on Building\_name = Building;

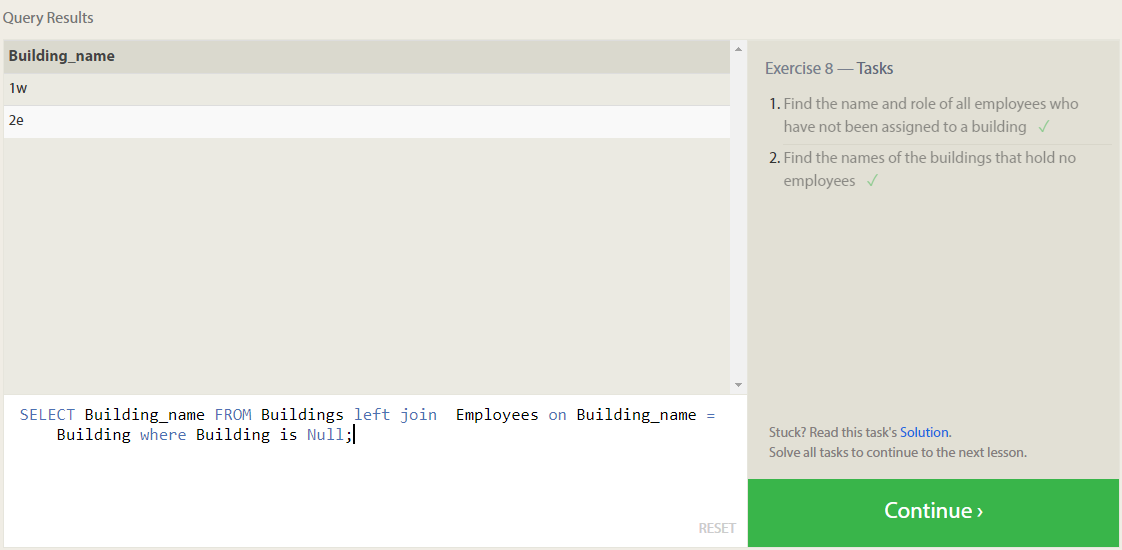
2.SELECT distinct Building\_name,capacity FROM Buildings left join Employees ON Building\_name = Building;  
  
3.SELECT distinct Building\_name,Role FROM Buildings left join Employees ON Building\_name = Building;



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

1.SELECT Name, Role FROM employees where Building is Null;

2.SELECT Building\_name FROM Buildings left join Employees on Building\_name = Building where Building is Null;



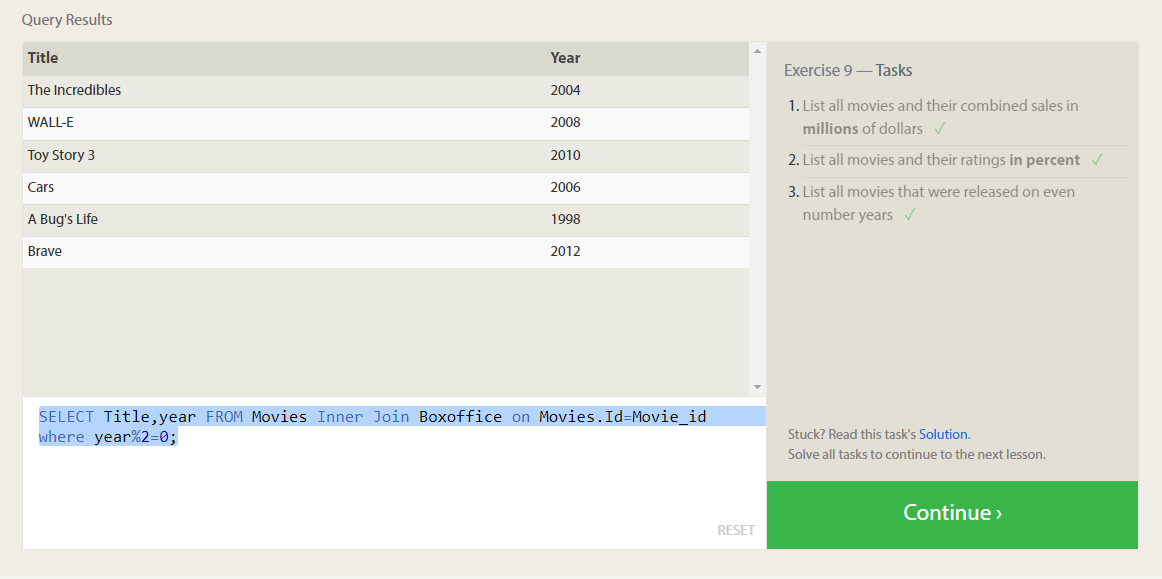
**SQL Lesson 9 : Queries with Expresions:**

1.SELECT Title,Domestic\_sales,International\_sales,(Domestic\_sales + International\_sales)/1000000 AS Combined\_Sales FROM Movies Inner Join Boxoffice on Movies.Id=Movie\_id ;

2.SELECT Title,Rating,(Rating\*10) AS Ratings\_percent FROM Movies Inner Join Boxoffice on Movies.Id=Movie\_id;

3.SELECT Title,year FROM Movies Inner Join Boxoffice on Movies.Id=Movie\_id

where year%2=0;

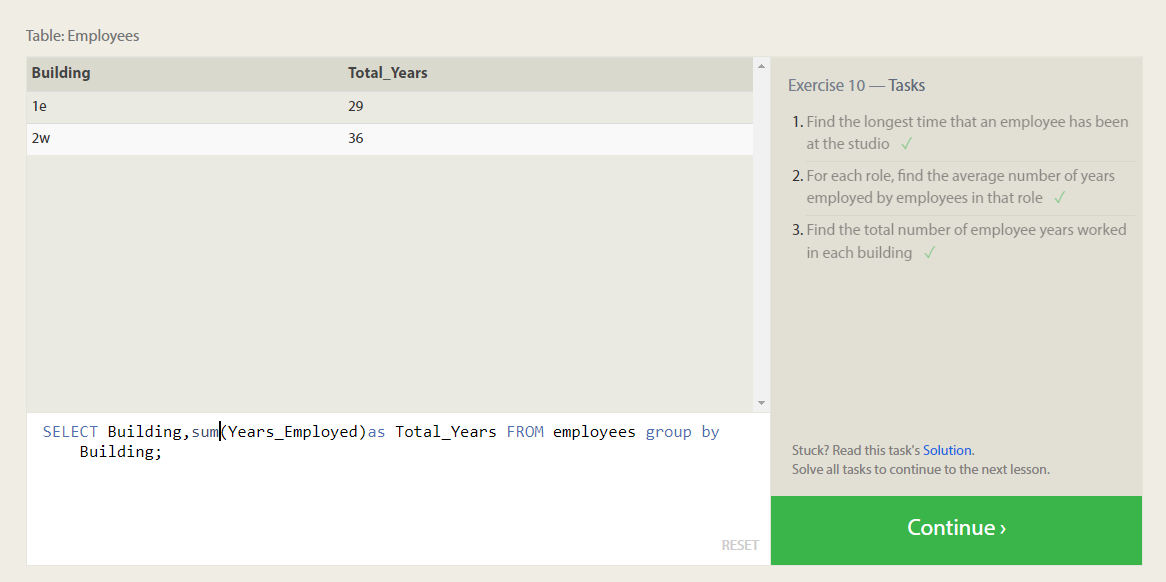


**SQL Lesson 10: Queries with aggregates (Pt. 1)**

1.SELECT MAX(Years\_Employed) FROM employees;

2.SELECT Name, Role,Years\_Employed,Avg(Years\_Employed)as Avg\_Year FROM employees Group by Role;

3.SELECT Building,sum(Years\_Employed)as Total\_Years FROM employees group by Building;

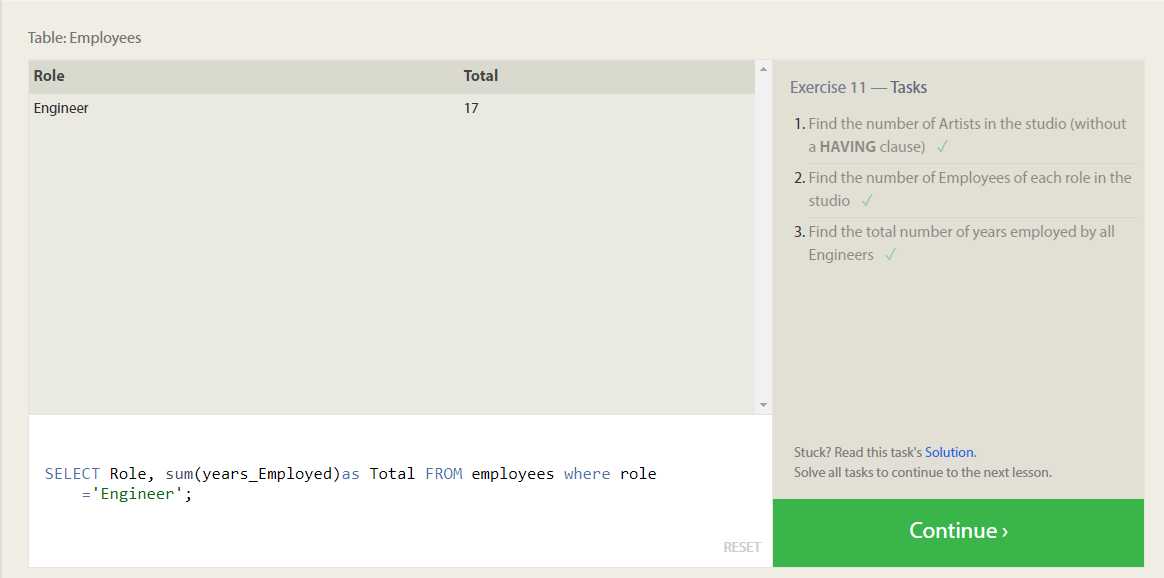


**SQL Lesson 11: Queries with aggregates (Pt. 2)**

1.SELECT Building, COUNT(\*) AS Total\_Artists FROM Employees WHERE Role = 'Artist';

2.SELECT Role, Count(Name)as Total FROM employees group by role;

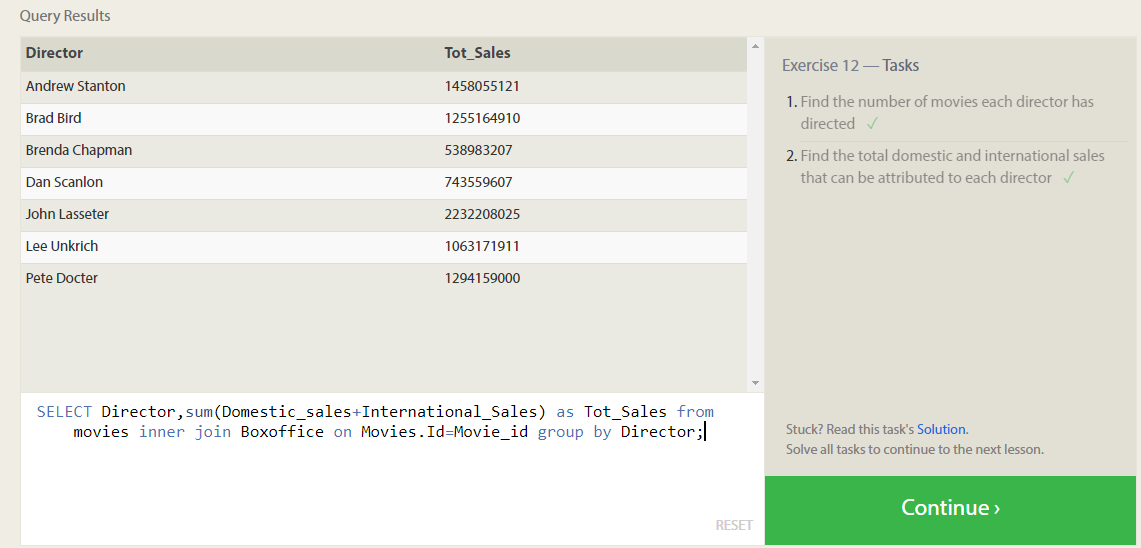
3.SELECT Role, sum(years\_Employed)as Total FROM employees where role='Engineer';



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

1.SELECT Director,Count(Title) as Num\_Movies from movies group by Director;

2.SELECT Director,sum(Domestic\_sales+International\_Sales) as Tot\_Sales from movies inner join Boxoffice on Movies.Id=Movie\_id group by Director;



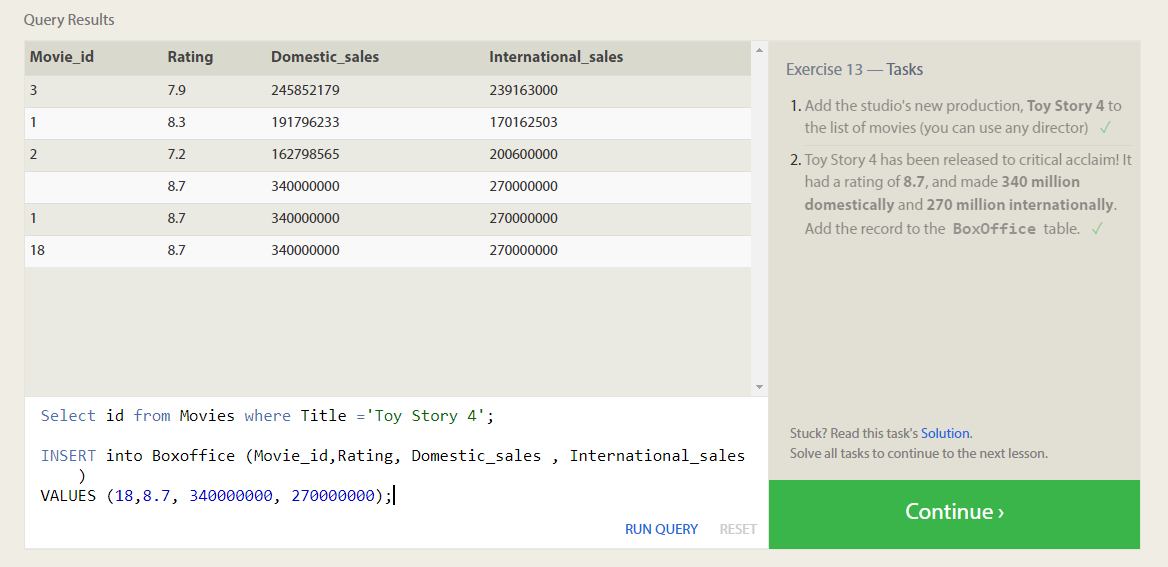
**SQL Lesson 13: Inserting rows**

1.INSERT into Movies (Title, Director, Year, Length\_minutes) VALUES ('Toy Story 4', 'any director', 2024, 100);

2.Select id from Movies where Title ='Toy Story 4';

INSERT into Boxoffice (Movie\_id,Rating, Domestic\_sales , International\_sales)

VALUES (18,8.7, 340000000, 270000000);

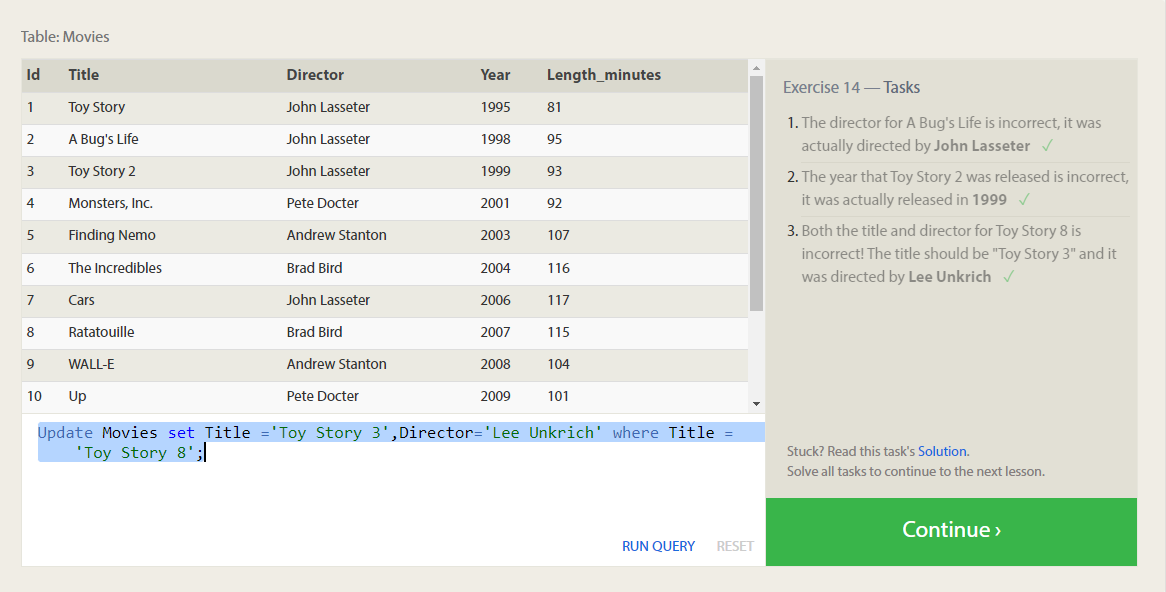


**SQL Lesson 14: Updating rows**

1.Update Movies set Director ='John Lasseter' where Title = "A Bug's Life";

2.Update Movies set Year ='1999' where Title = "Toy Story 2";

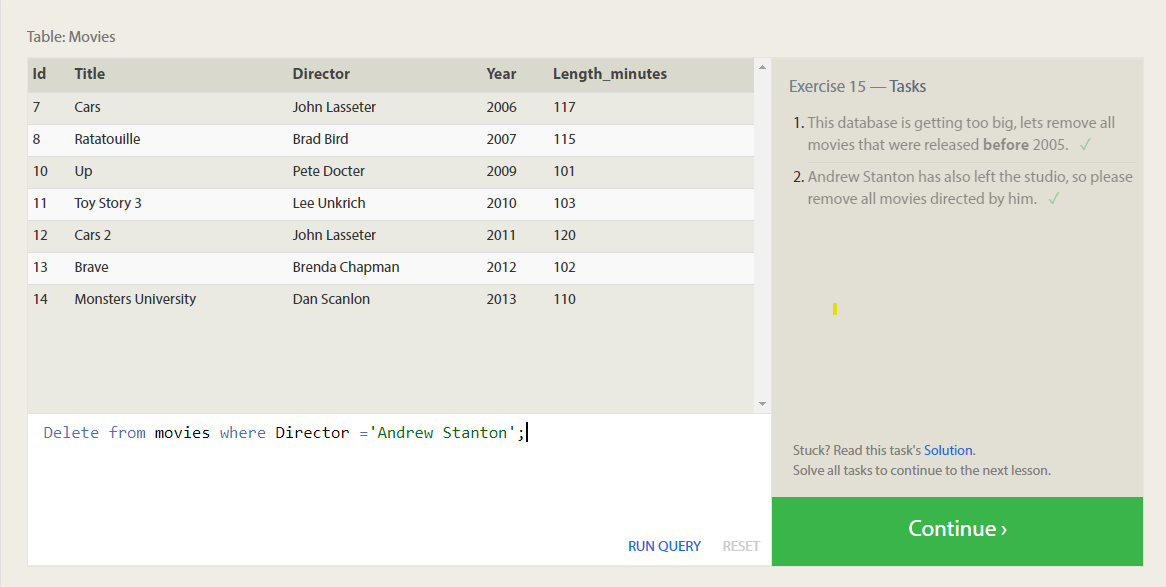
3.Update Movies set Title ='Toy Story 3',Director='Lee Unkrich' where Title = 'Toy Story 8';



**SQL Lesson 15: Deleting rows**

1.Delete from movies where year<2005;

2.Delete from movies where Director ='Andrew Stanton';



**SQL Lesson 16: Creating tables**

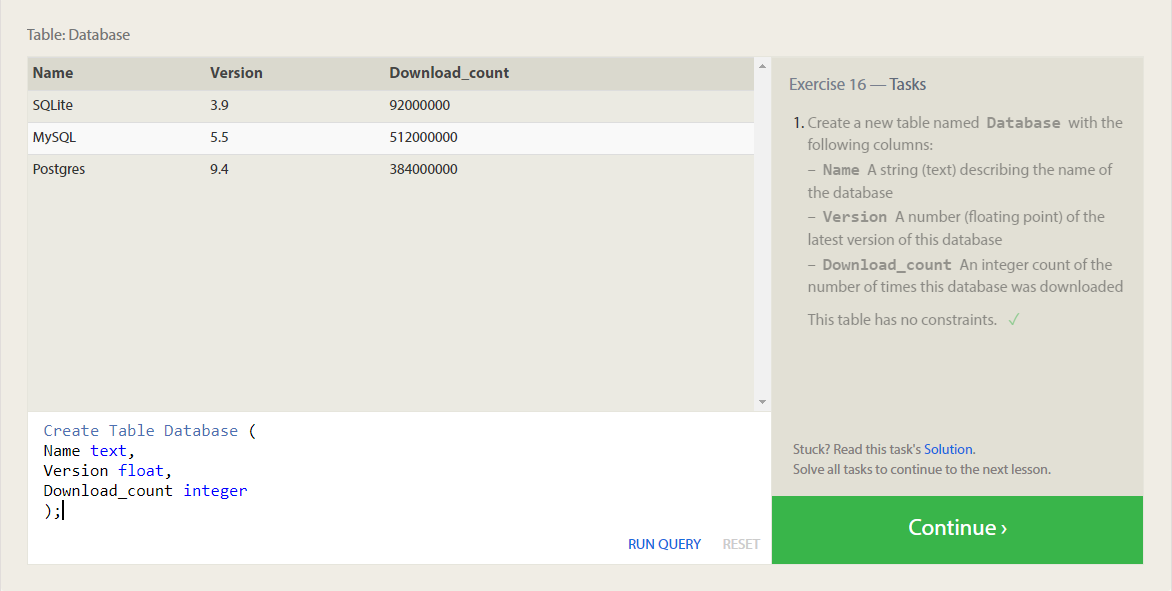
Create Table Database (

Name text,

Version float,

Download\_count integer

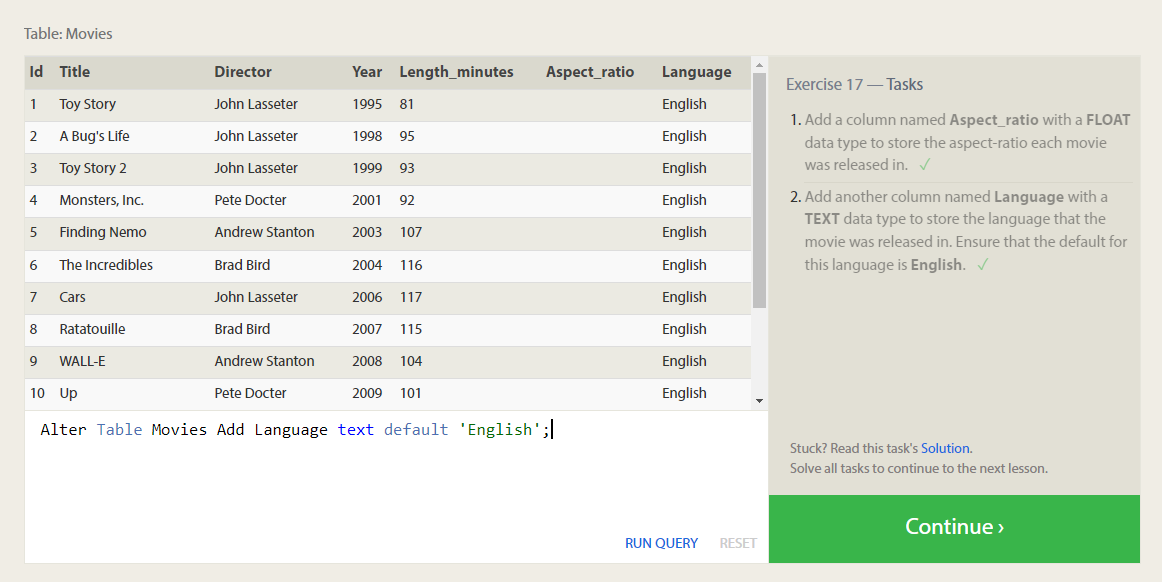
);



**SQL Lesson 17: Altering tables**

1.Alter Table Movies Add Aspect\_ratio float;

2.Alter Table Movies Add Language text default 'English';



**SQL Lesson 18: Dropping tables**

1.Drop table if exists Movies;

2.Drop table if exists Boxoffice;

