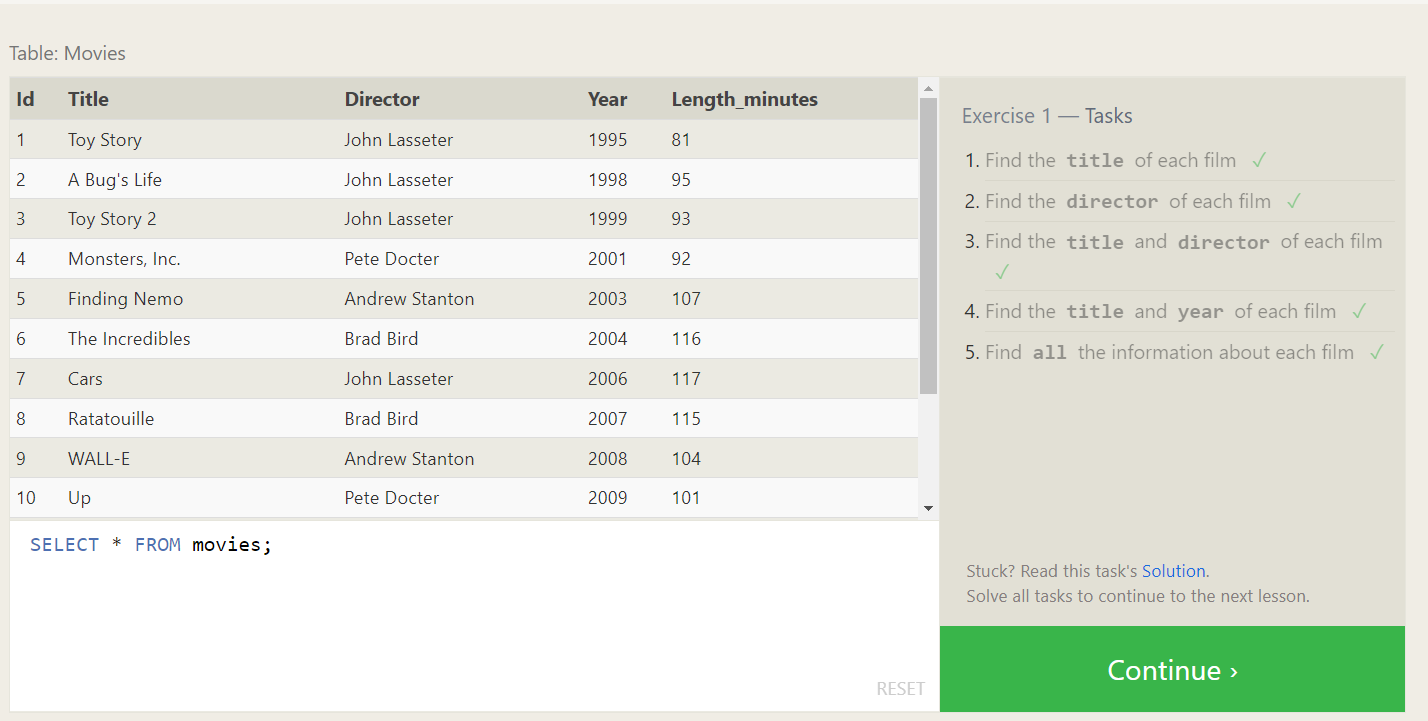
**MYSQL – TASK 1**

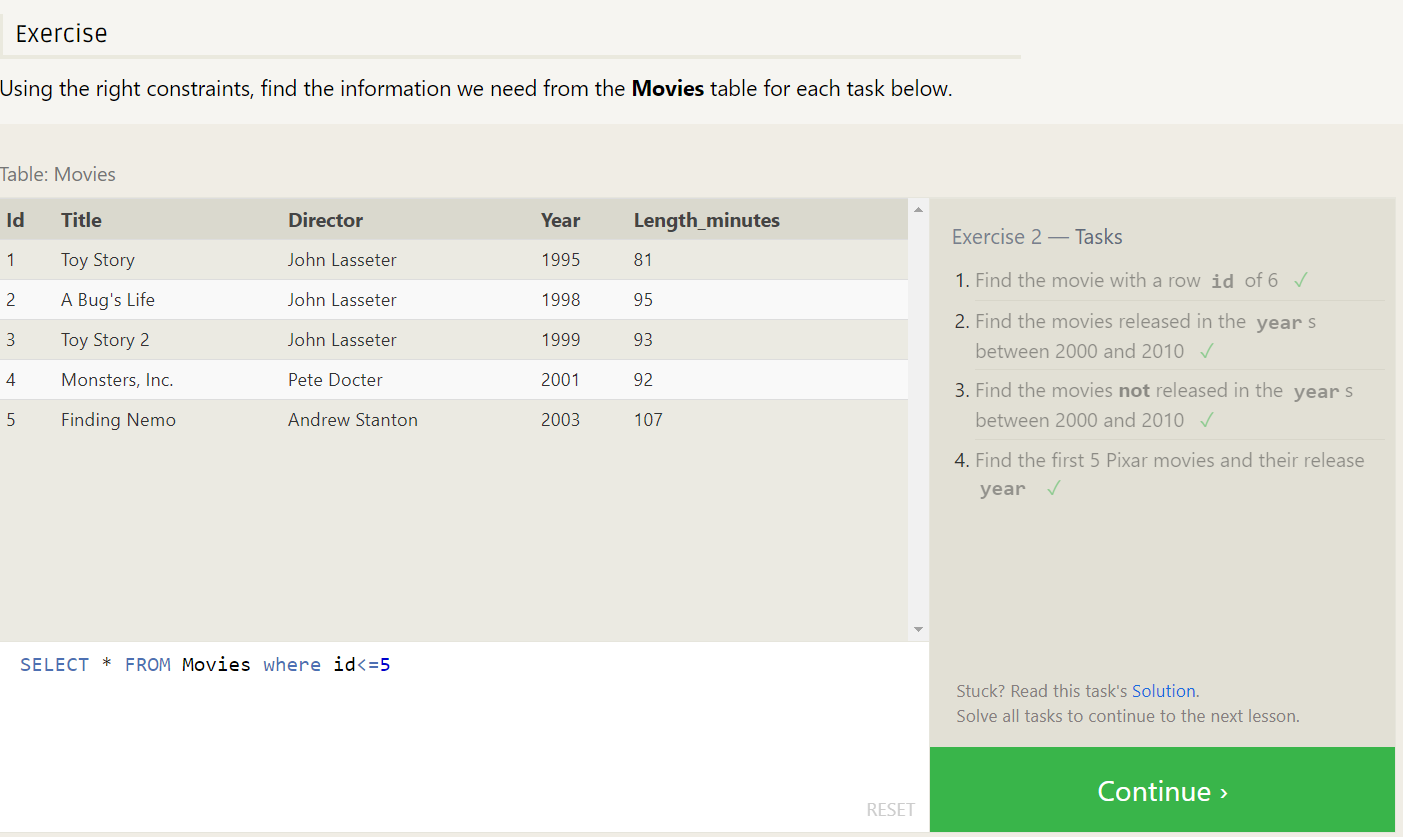
**Exercise 1: Queries**

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;



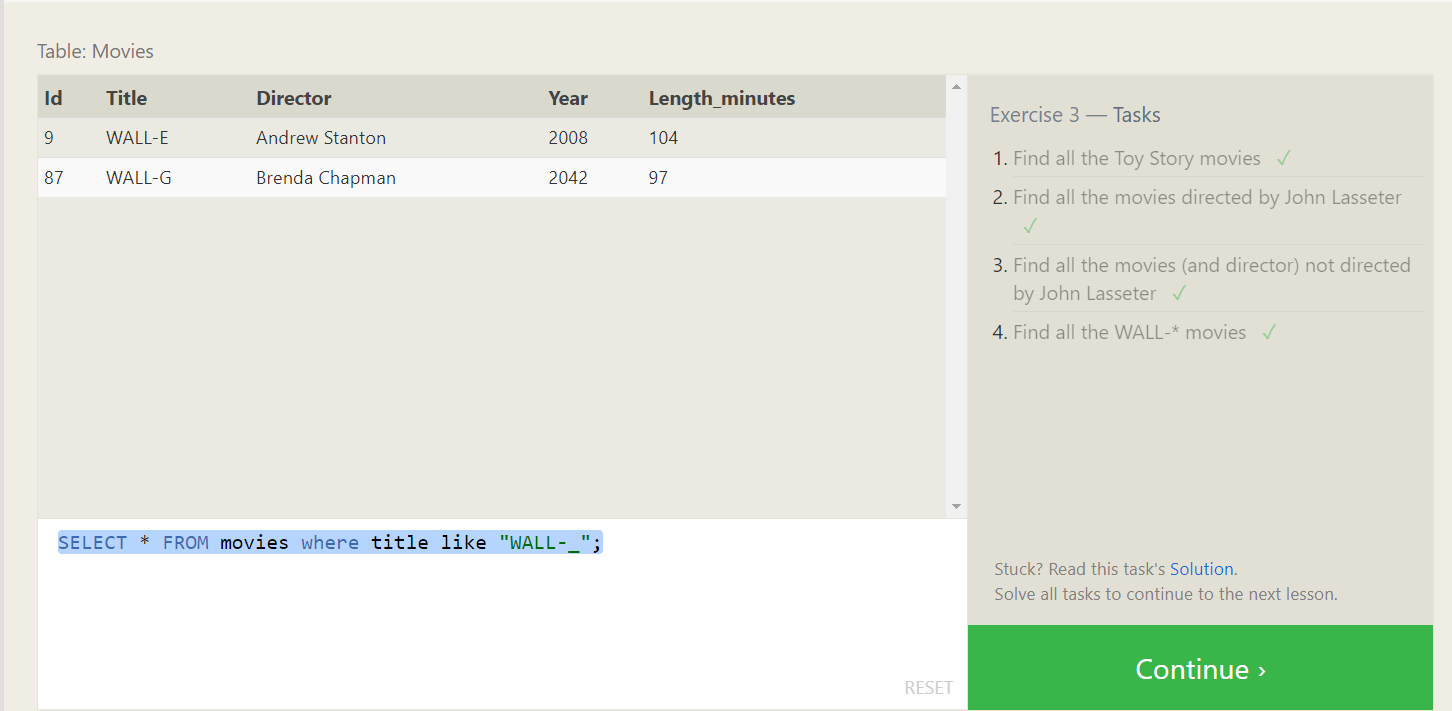
**Exercise 2: Selected Queries**

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;

****

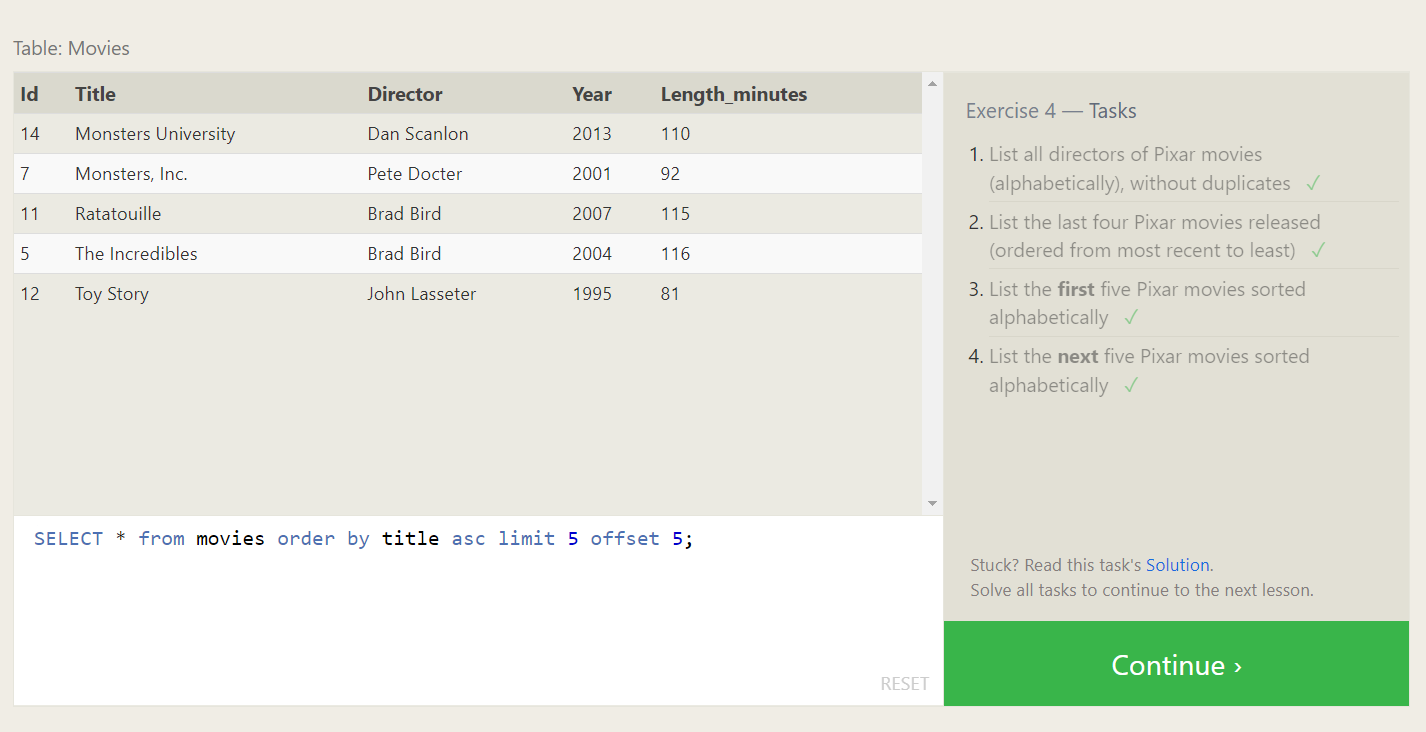
**Exercise 3: Queries with Constraints**

1. SELECT \* FROM movies where title like 'Toy Story%';
2. SELECT \* FROM movies where director='John Lasseter'
3. SELECT \* FROM movies where director!='John Lasseter'
4. SELECT \* FROM movies where title like "WALL-\_";

****

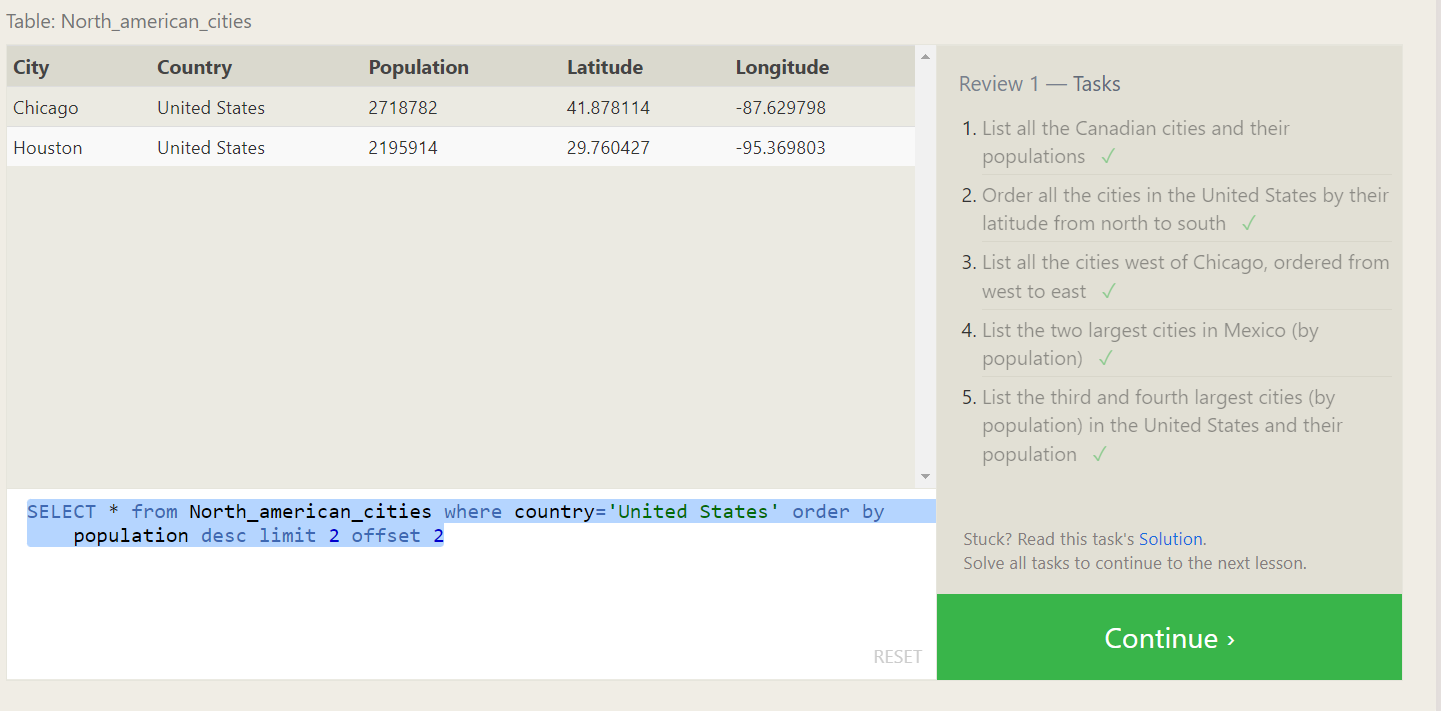
**Exercise 4: Filtering and Sorting Query Results**

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;



**Exercise 5: Simple Select Queries**

1. SELECT City, population FROM north\_american\_cities where country like 'Canada';
2. SELECT \* from North\_american\_cities where country='United States' order by latitude desc
3. SELECT \* from North\_american\_cities where longitude<-87.629798 order by longitude
4. SELECT \* from North\_american\_cities where country='Mexico' order by population desc limit 2
5. SELECT \* from North\_american\_cities where country='United States' order by population desc limit 2 offset 2

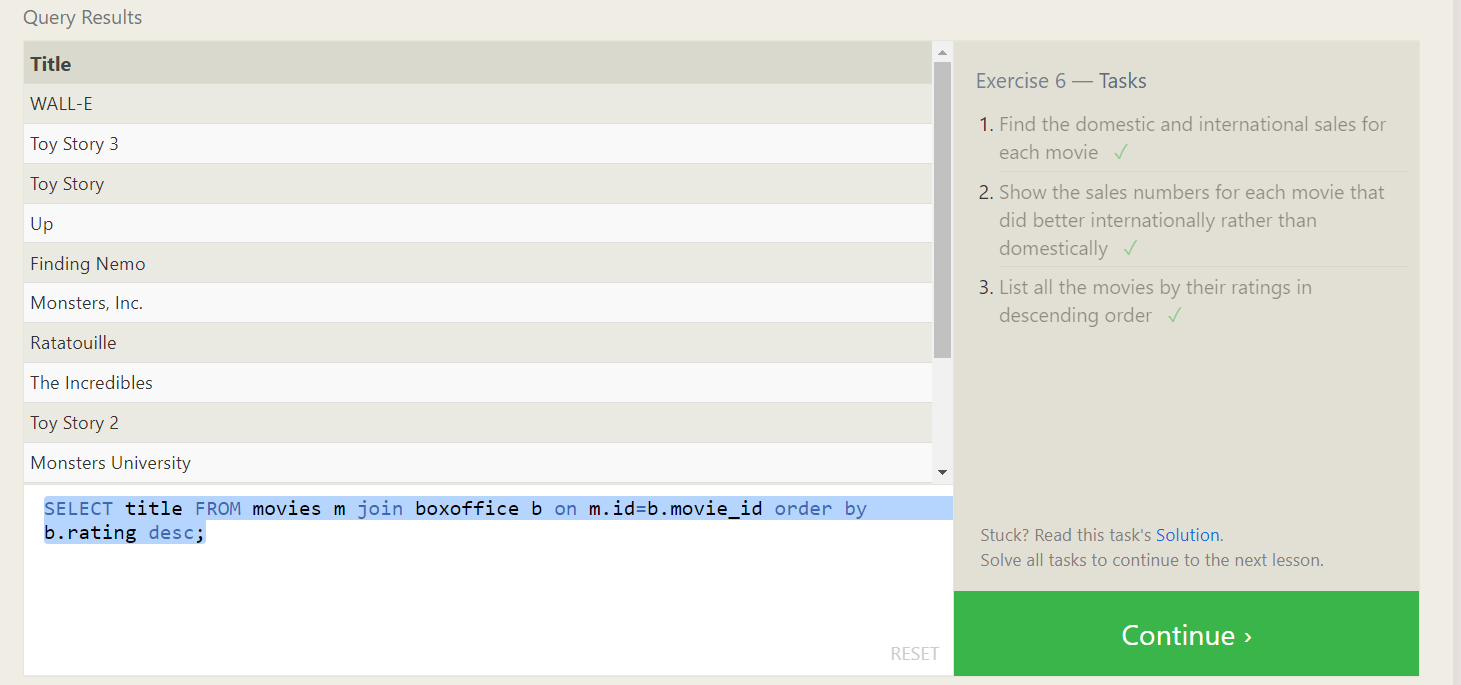


**Exercise 6: Multi table queries with joins**

1. SELECT \* FROM movies m join boxoffice b on m.id=b.movie\_id;
2. SELECT \* FROM movies m join boxoffice b on m.id=b.movie\_id where

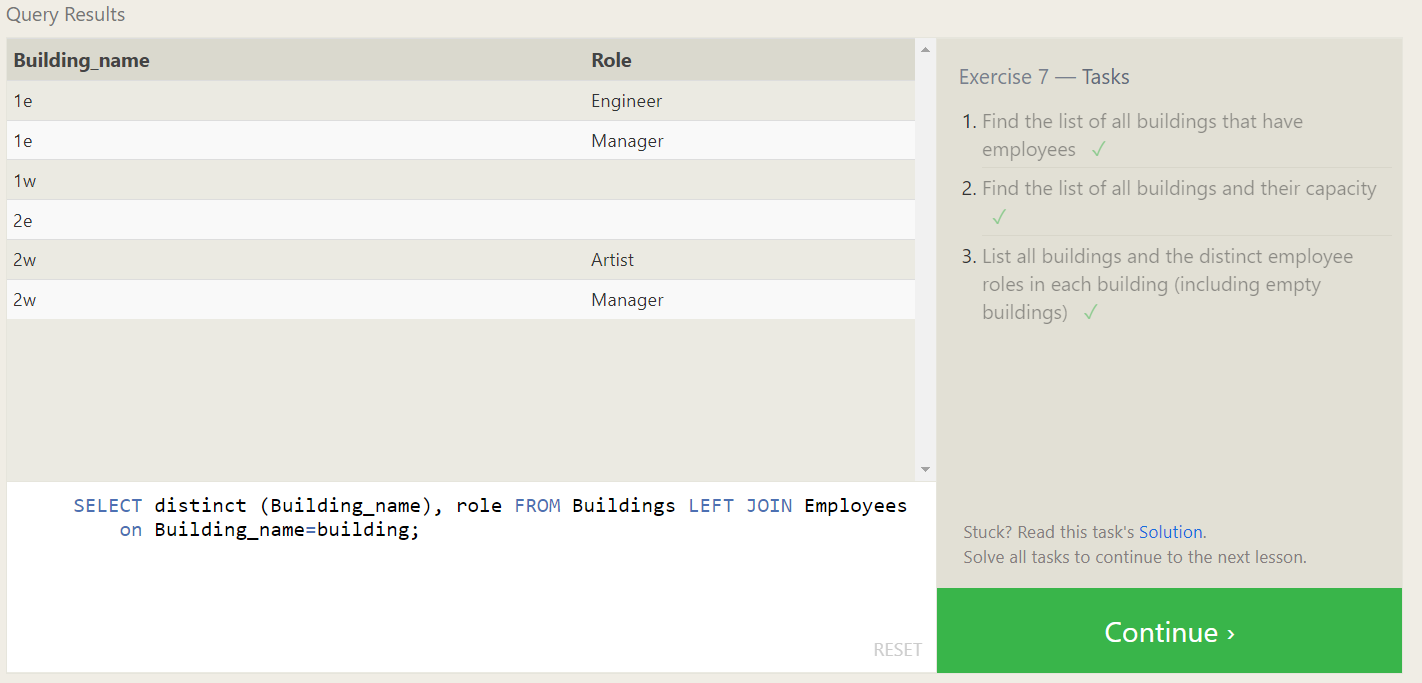
b.International\_sales>b.Domestic\_sales;

3. SELECT title FROM movies m join boxoffice b on m.id=b.movie\_id order by b.rating desc;



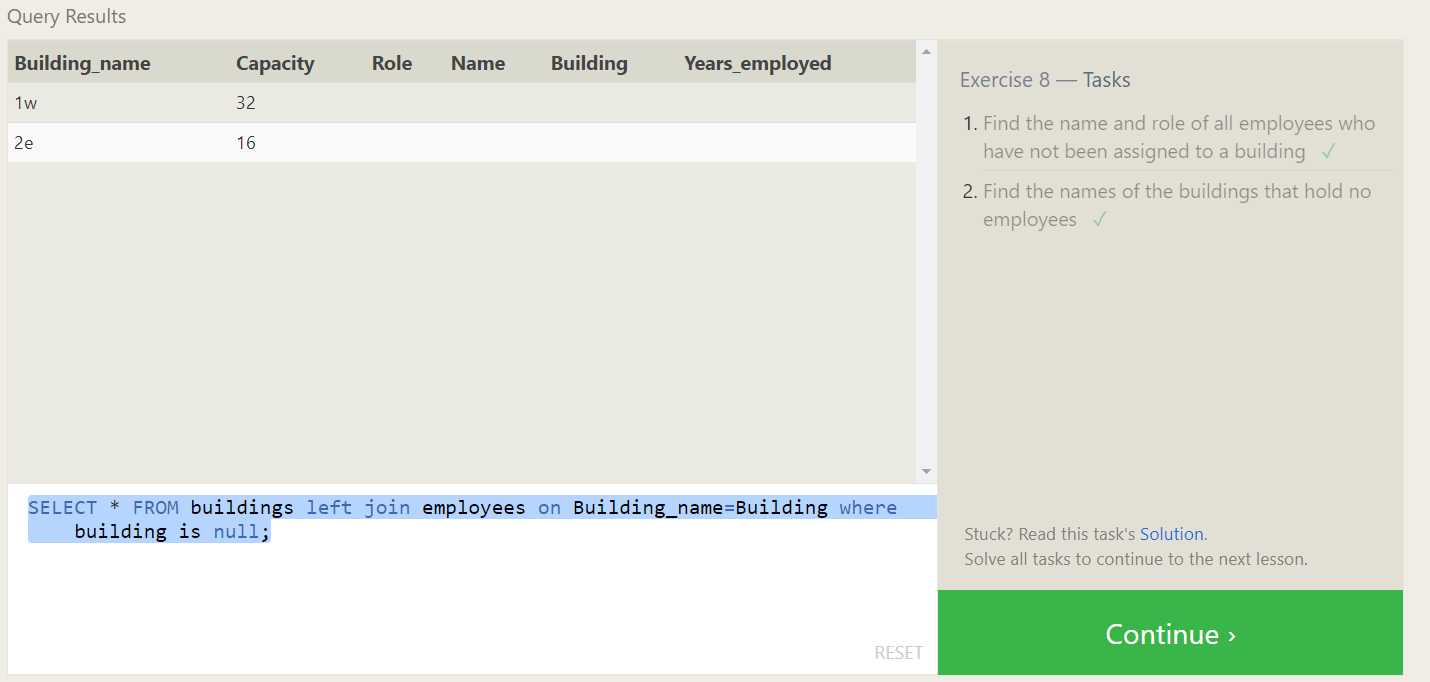
**Exercise 7: Outer Joins**

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;



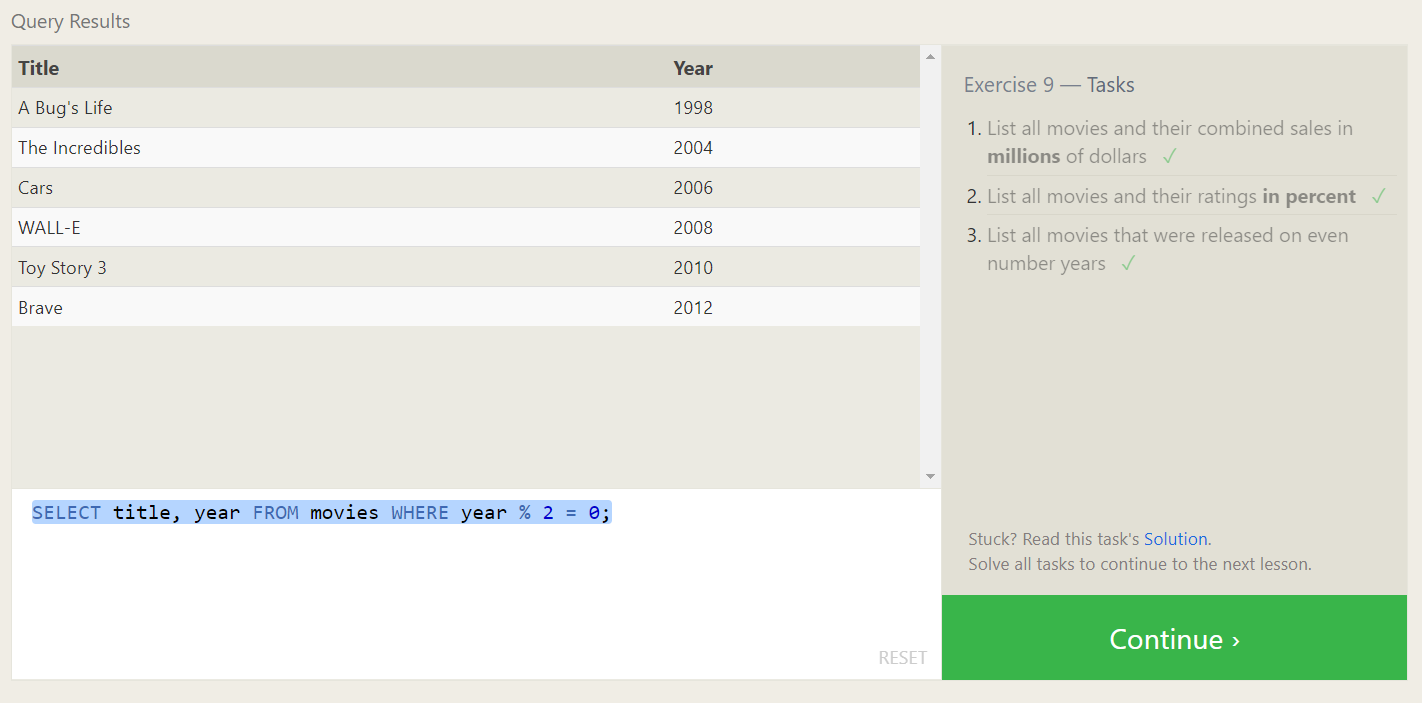
**Exercise 8: A short note on nulls**

1. SELECT \* FROM employees where building is null;
2. SELECT \* FROM buildings left join employees on Building\_name=Building where building is null;



**Exercise 9: Queries with Expresssions**

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;



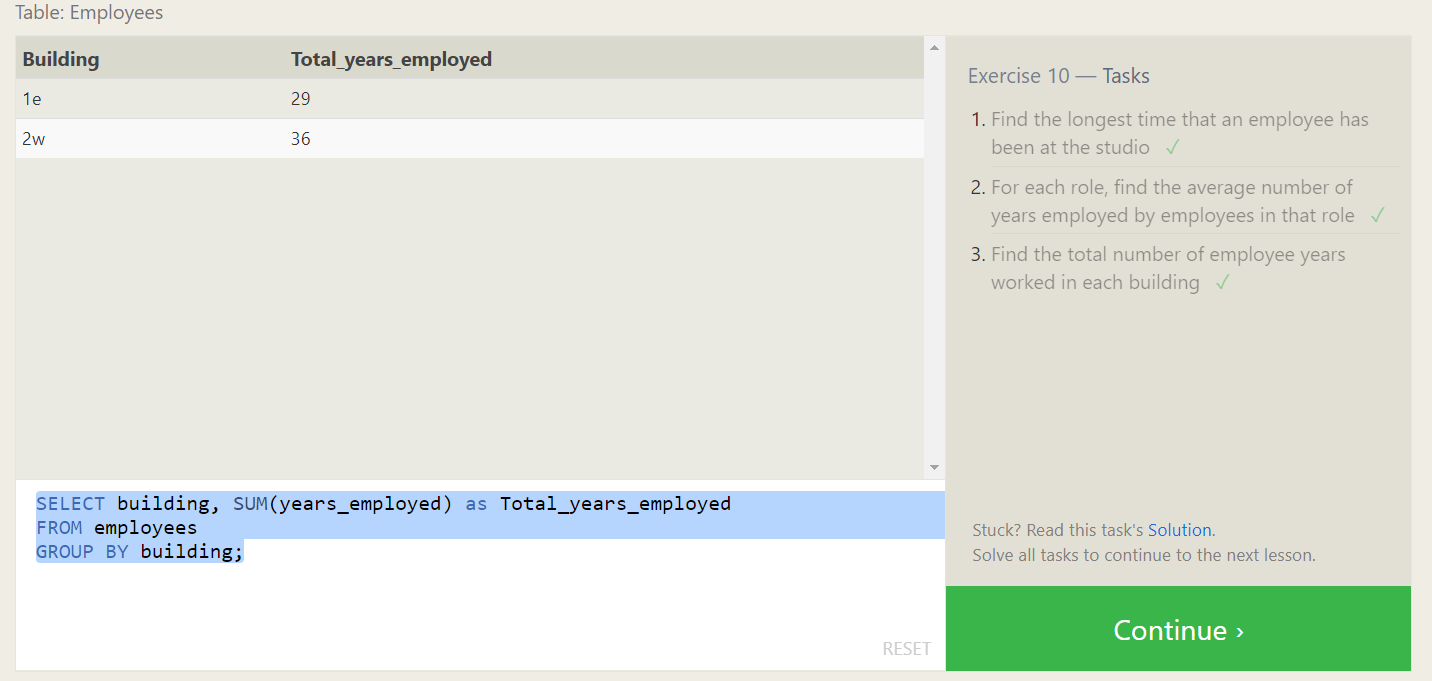
**Exercise 10: Queries with Aggregates**

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;

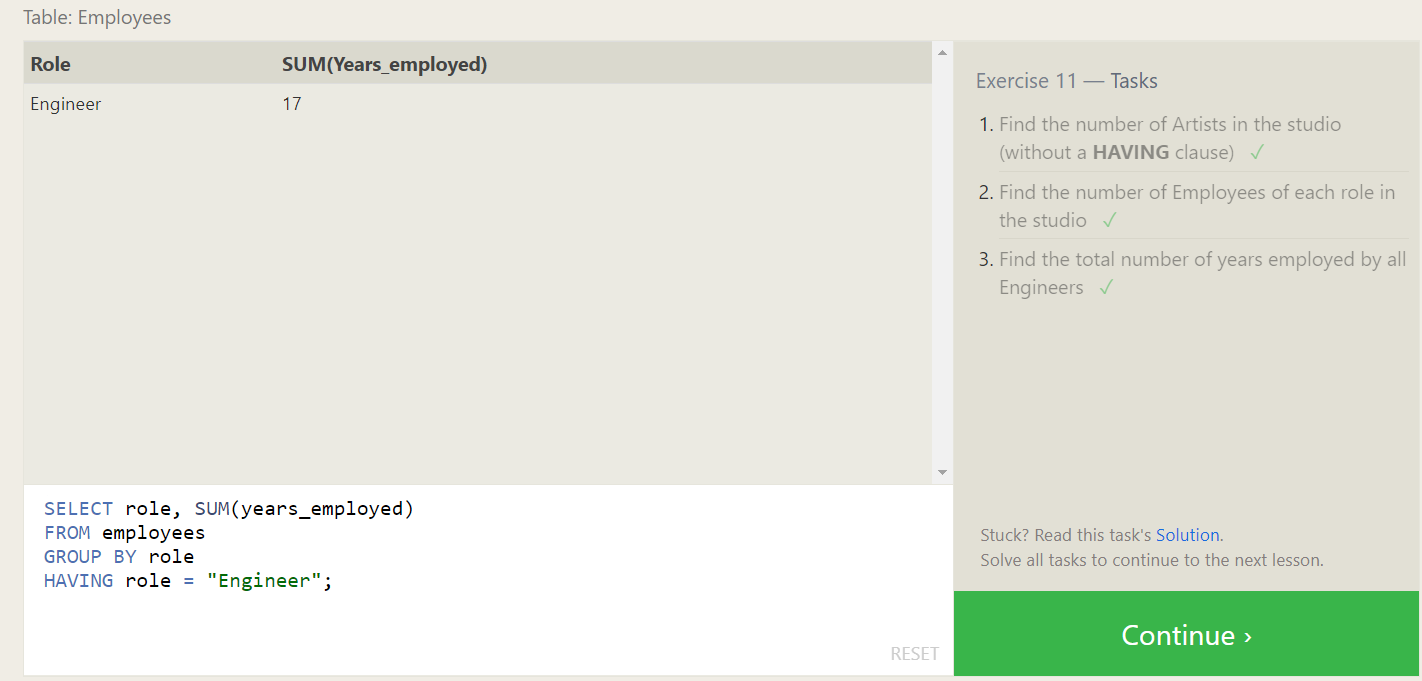


**Exercise 11: Queries with aggregates**

1. SELECT role, COUNT(\*) as Number\_of\_artists FROM employees WHERE role = "Artist";

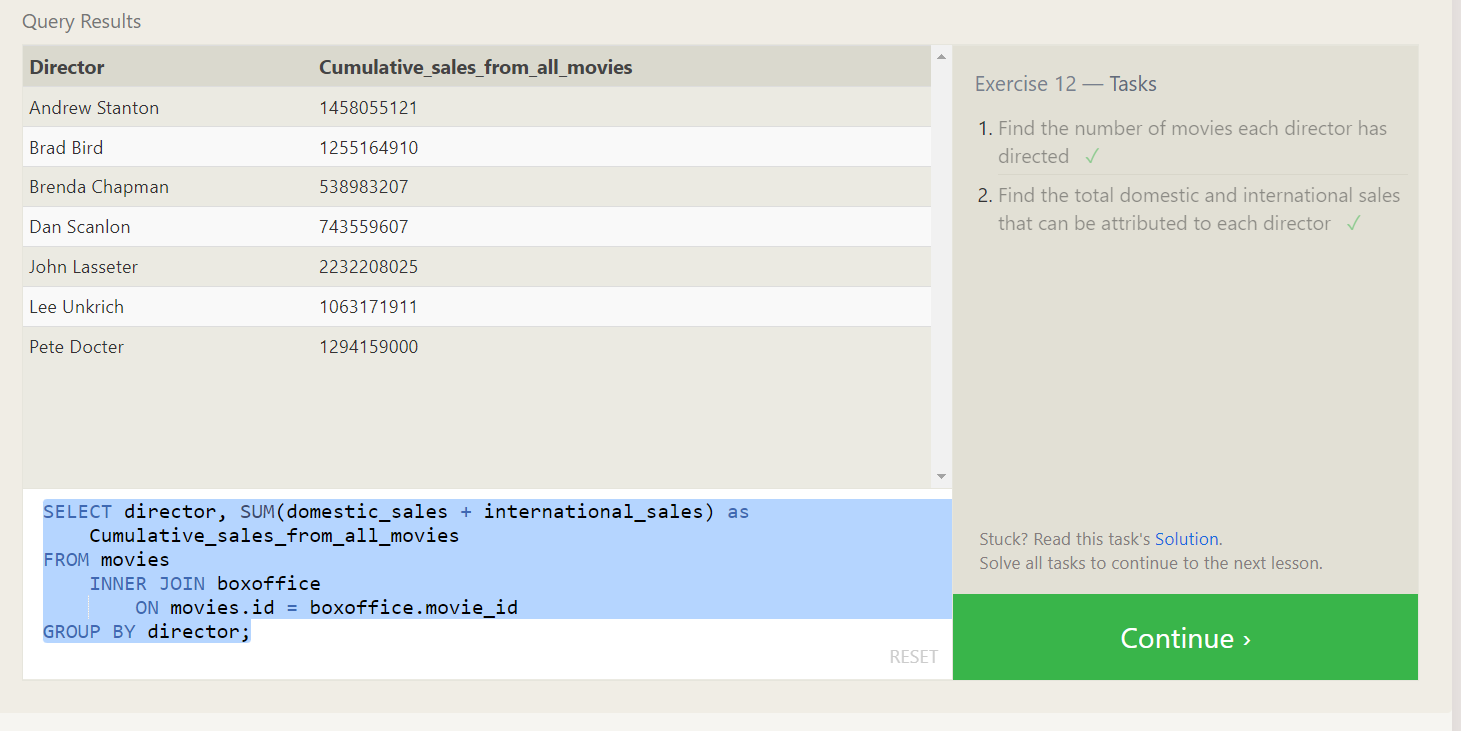
2. SELECT role, COUNT(\*) FROM employees GROUP BY role;

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



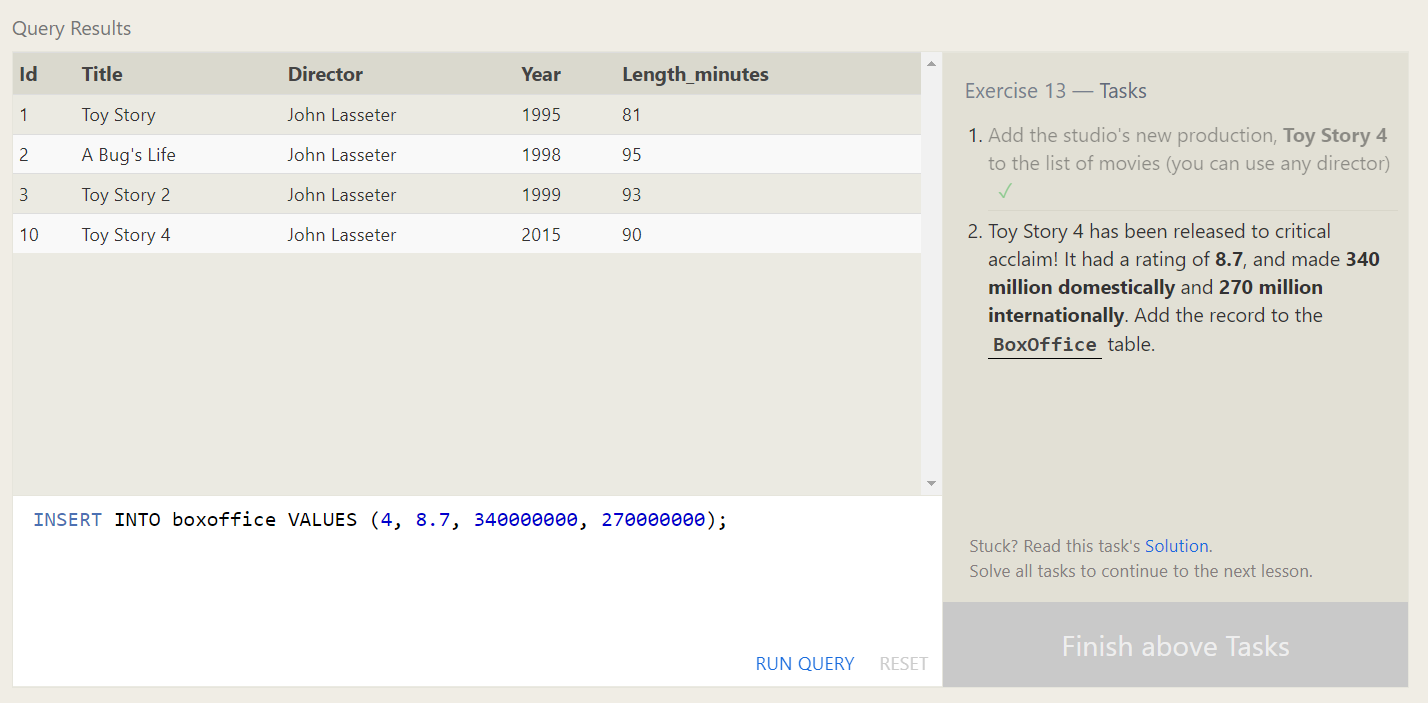
**Exercise 12: Order of execution of a Query**

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;



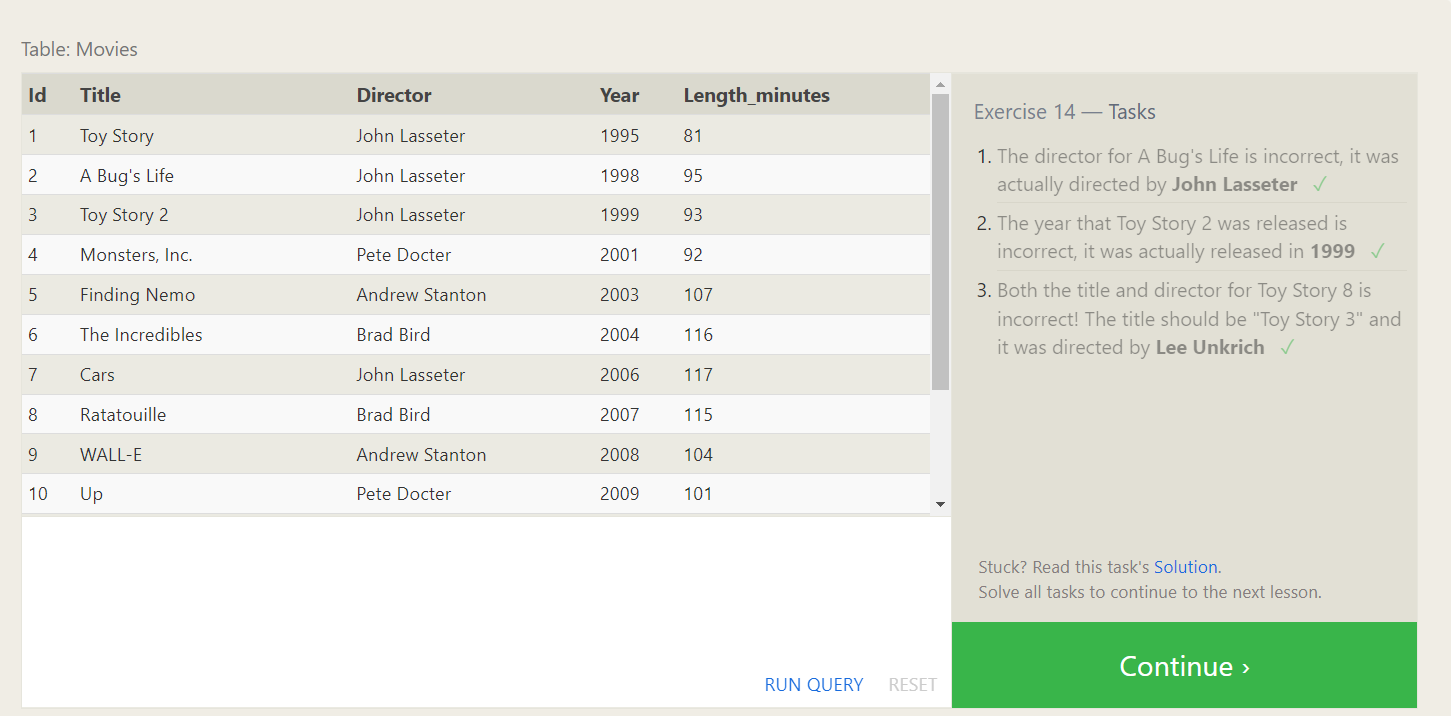
**Exercise 13: Inserting Rows**

1. INSERT INTO movies VALUES (10, "Toy Story 4", "John Lasseter", 2015, 90);
2. INSERT INTO boxoffice VALUES (4, 8.7, 340000000, 270000000);



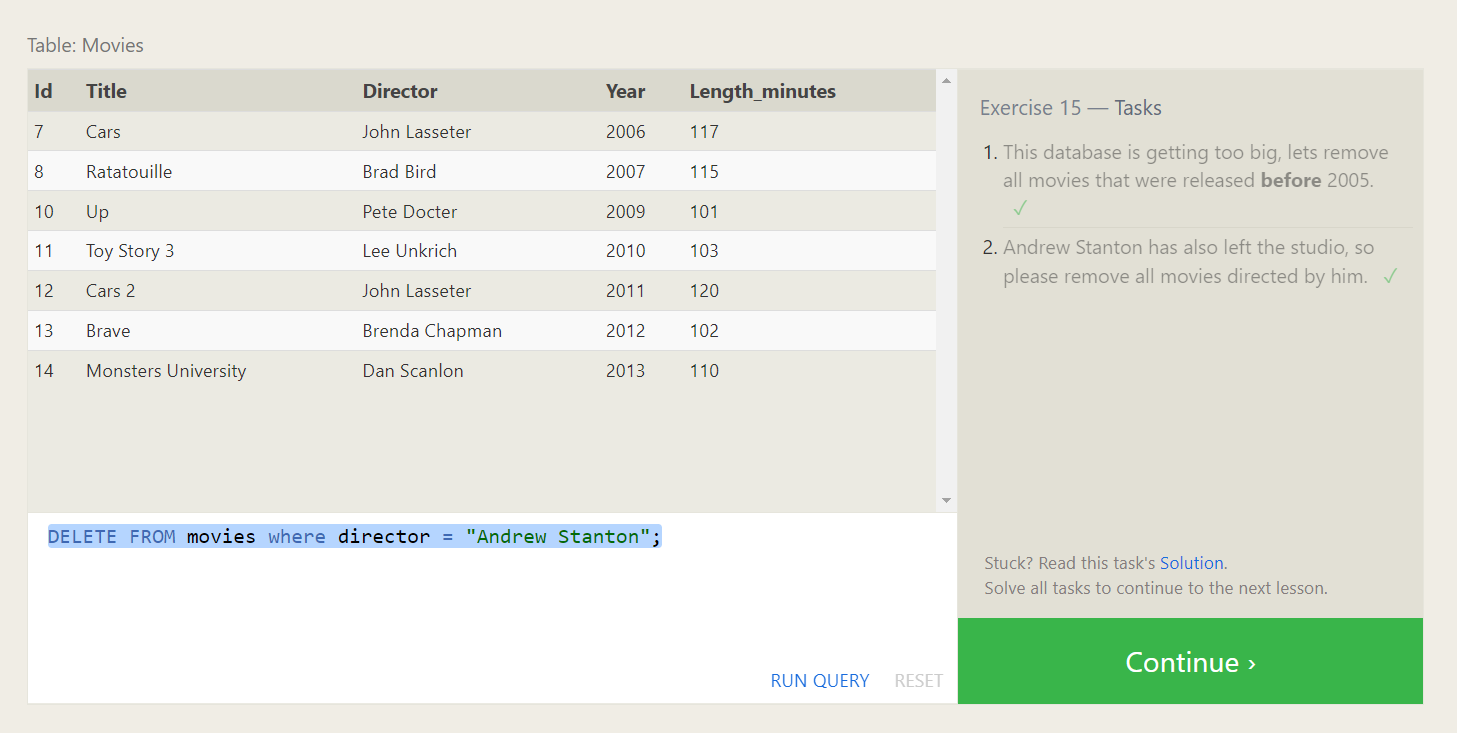
**Exercise 14: Updating Rows**

1. UPDATE movies SET director = "John Lasseter" WHERE id = 2;
2. UPDATE movies SET year = 1999 WHERE id = 3;
3. UPDATE movies SET title = "Toy Story 3", director = "Lee Unkrich" WHERE id = 11;



**Exercise 15: Deleting Rows**

1. DELETE FROM movies where year < 2005;
2. DELETE FROM movies where director = "Andrew Stanton";



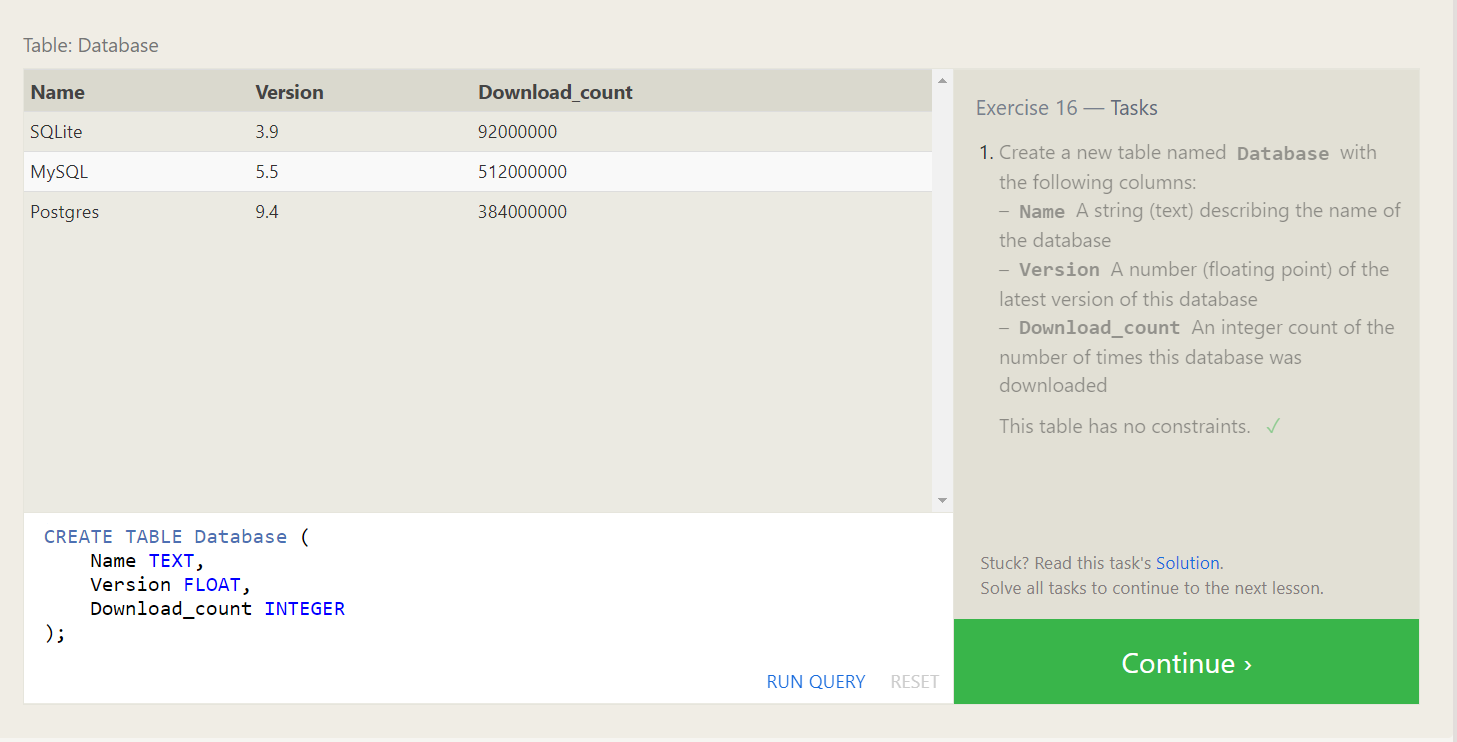
**Exercise 16: Creating Tables**

1. CREATE TABLE Database (

Name TEXT,

Version FLOAT,

Download\_count INTEGER

);  


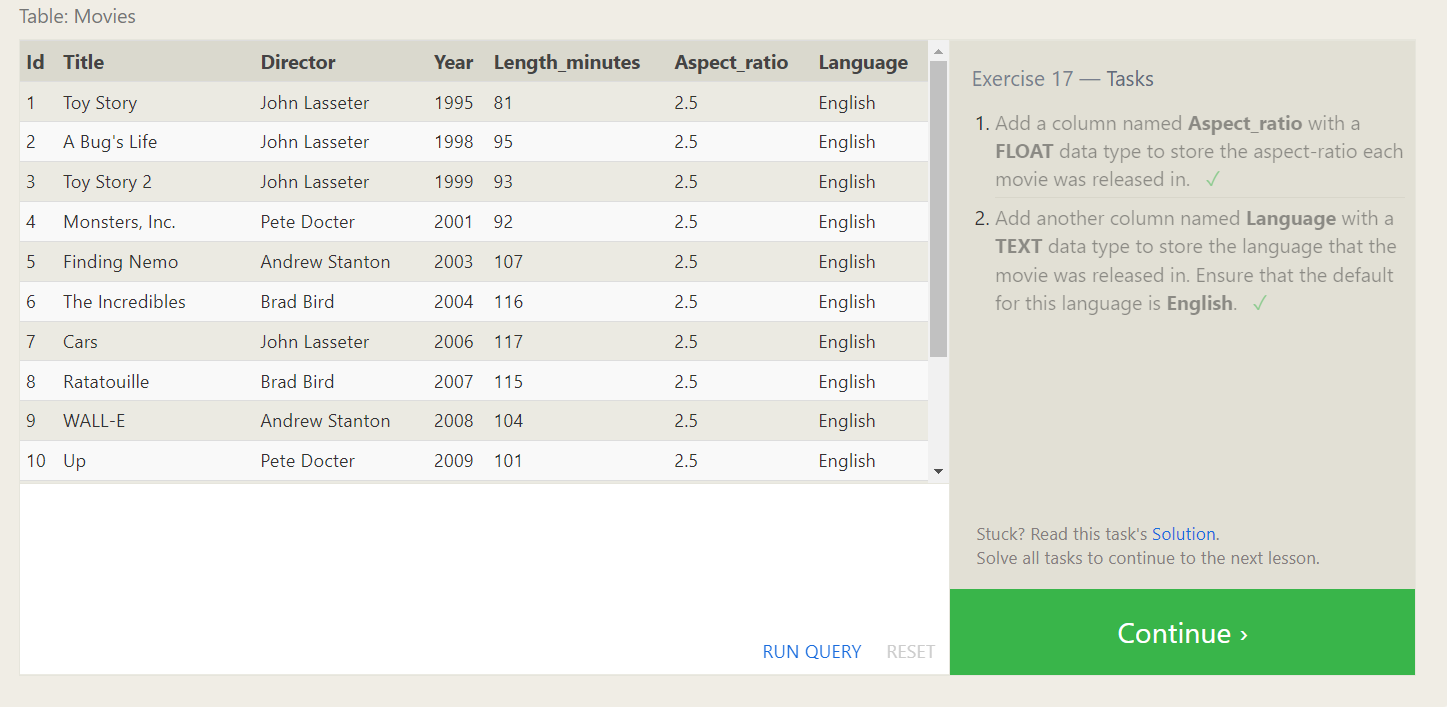
**Exercise 17: Altering Tables**

1. ALTER TABLE Movies

ADD COLUMN Aspect\_ratio FLOAT DEFAULT 2.5;

2. ALTER TABLE Movies

ADD COLUMN Language TEXT DEFAULT "English";



**Exercise 18: Dropping Tables**

1. DROP TABLE Movies;
2. DROP TABLE BoxOffice;

