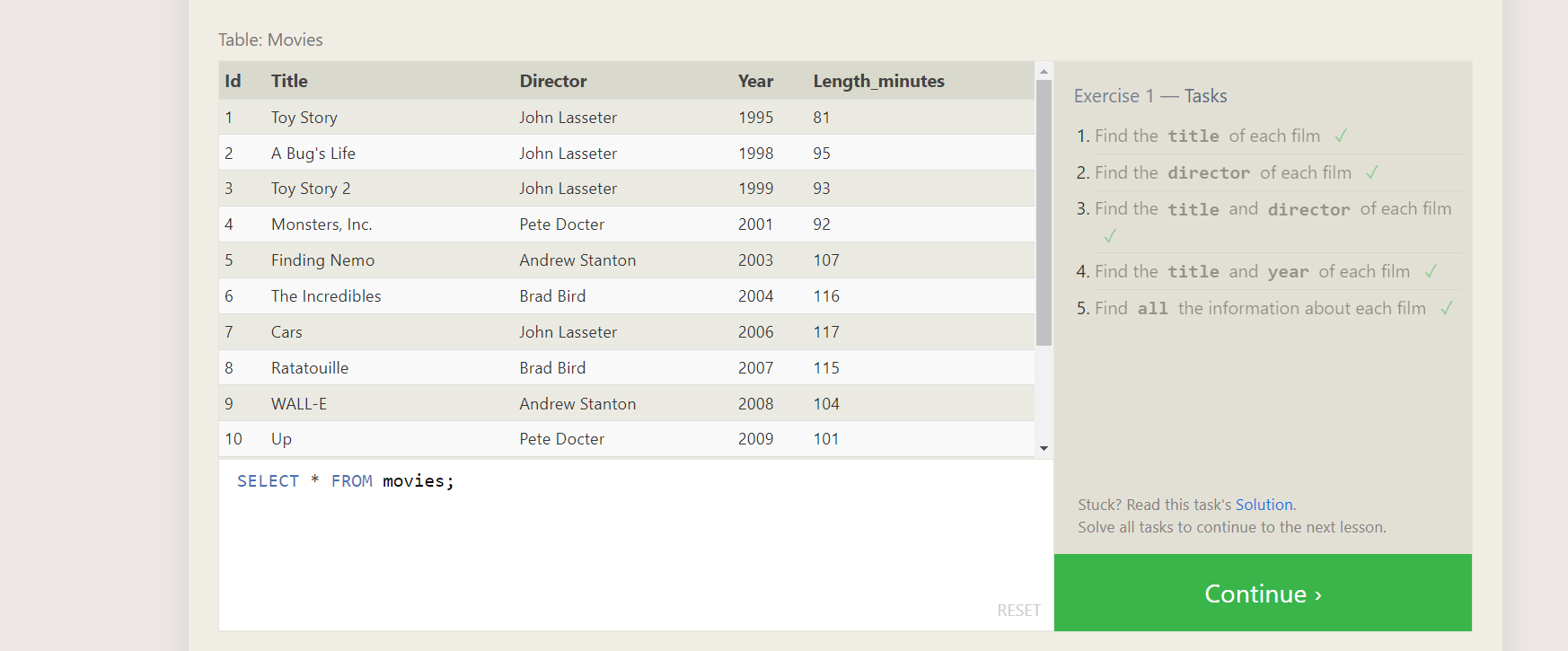
**MY SQL**

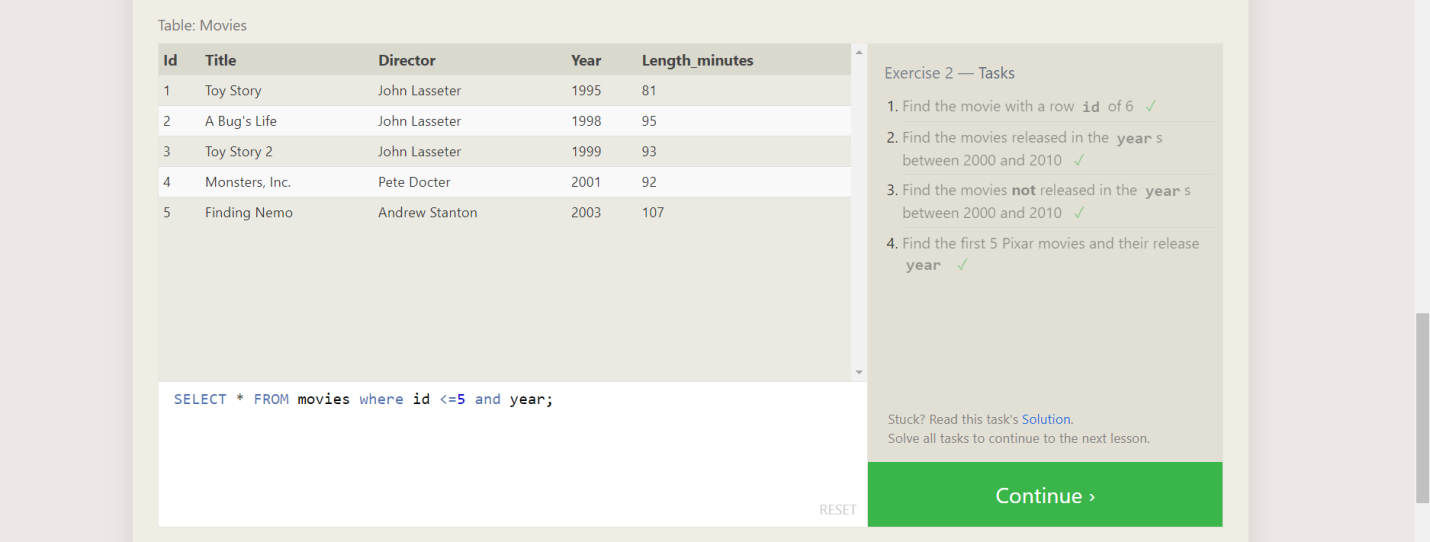
**Task 1:**

SQL Lesson 1: SELECT queries 101:



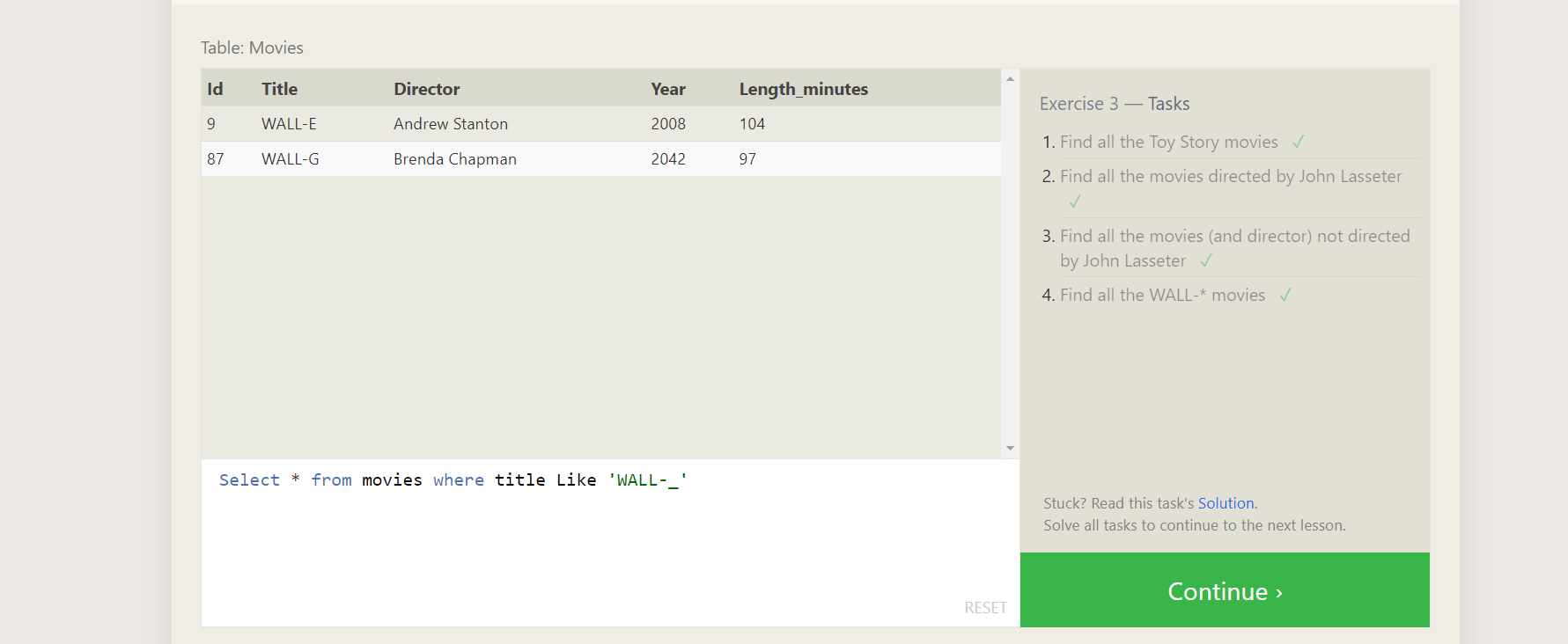
1. SELECT TITLE FROM MOVIES;
2. SELECT DIRECTOR FROM MOVIES;
3. SELECT TITLE,DIRECTOR FROM MOVIES;
4. SELECT TITLE,YEAR FROM MOVIES;
5. SELECT \* FROM MOVIES;

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



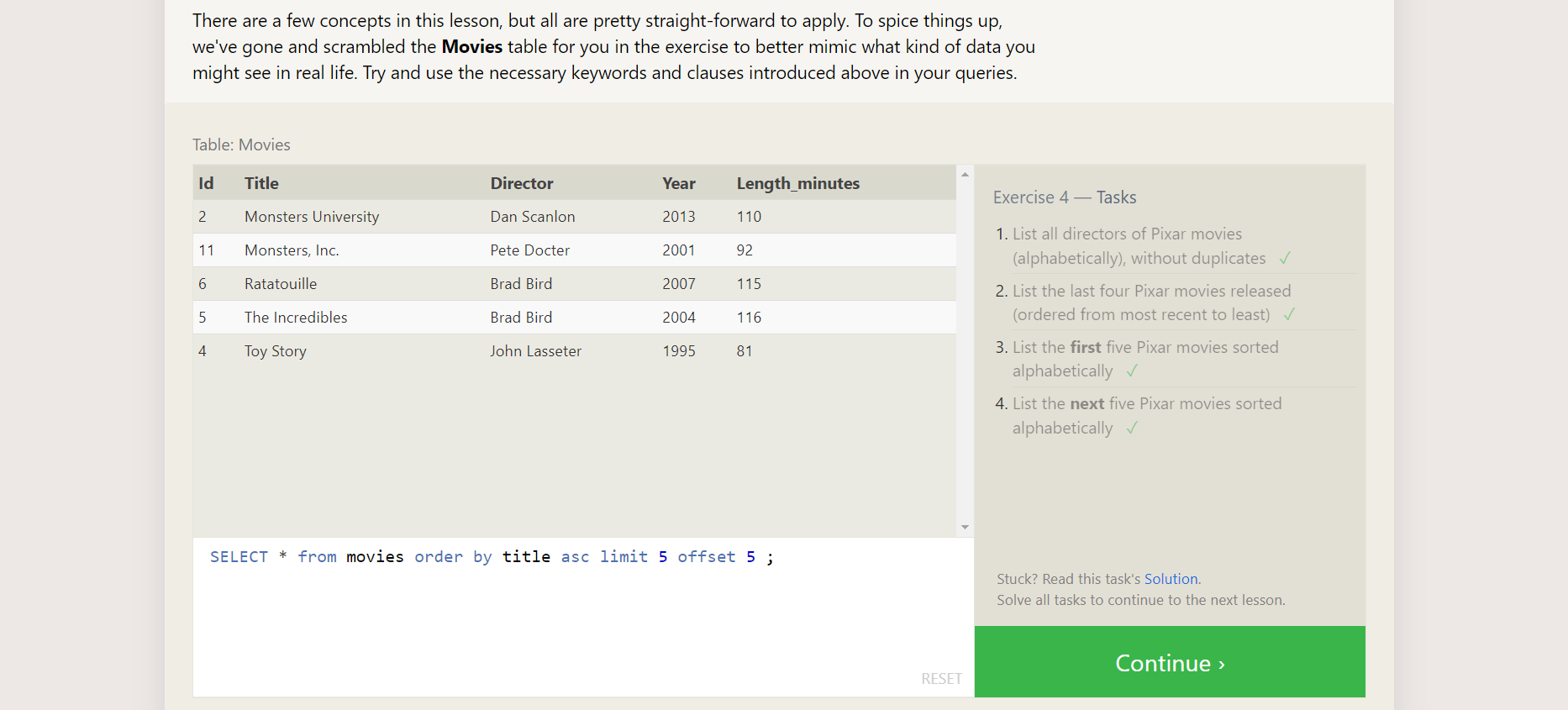
1. SELECT \* FROM MOVIES WHERE ID =6;
2. SELECT \* FROM MOVIES WHERE YEAR BETWEEN 2000 AND 2010;
3. SELECT \* FROM MOVIES WHERE YEAR NOT BETWEEN 2000 AND 2010;
4. SELECT \* FROM MOVIES WHERE ID <=5 AND YEAR;

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 not director like ‘John Lasseter’
4. Select \* from movies where title like ‘WALL-\_’

SQL Lesson 4: Filtering and sorting Query results



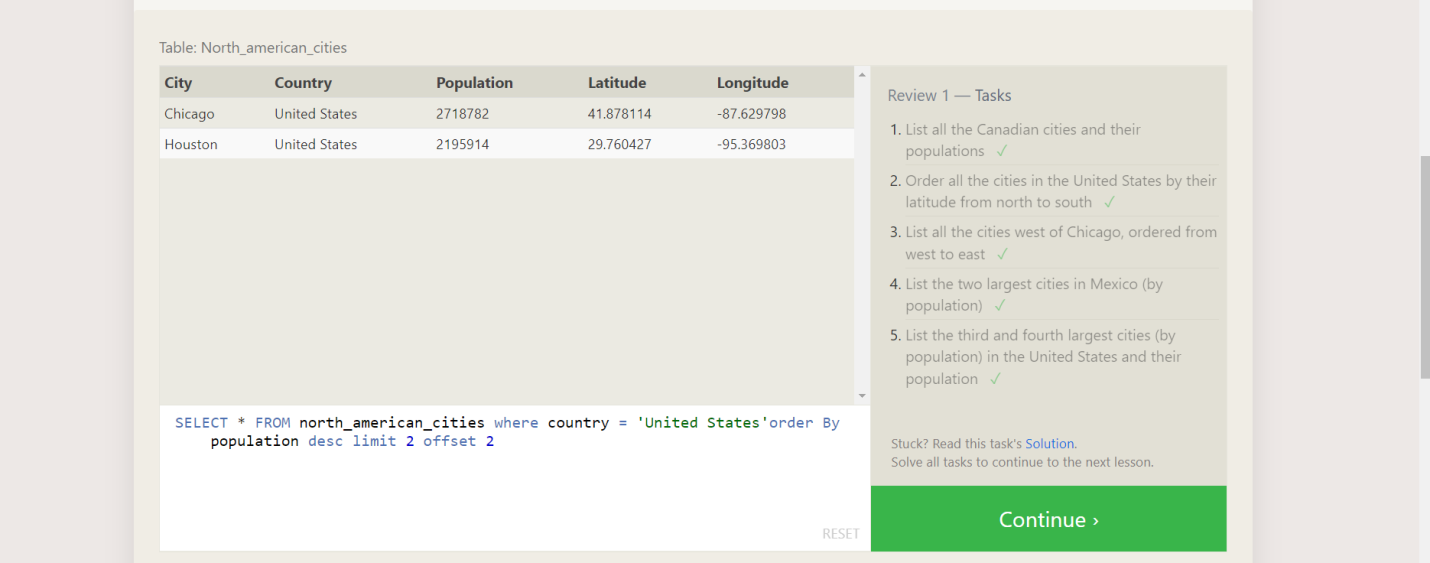
1.SELECT DISTINCT(DIRECTOR) FROM MOVIES ORDER BY DIRECTOR ;

2. SELECT \* FROM MOVIES ORDER BY YEAR DESC LIMIT 4;

3.SELECT \* FROM MOVIES ORDERBY TITLE ASC LIMIT 5;

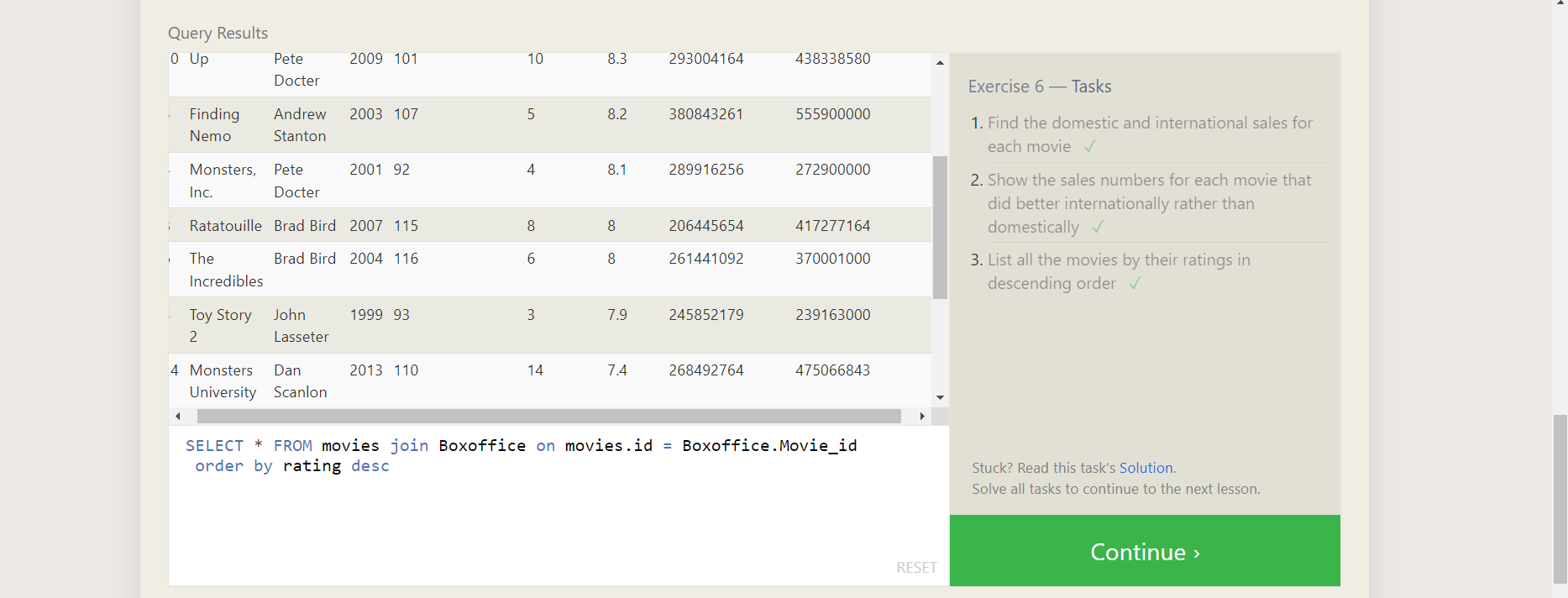
4.SELECT \* FROM MOVIES ORDER BY TITLE ASC LIMIT 5 OFFSET 5;

SQL Review: Simple SELECT Queries



1. Select \* from north\_americans \_ citizens where country = ‘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 asc;
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;

SQL Lesson 6: Multi-table queries with JOINs



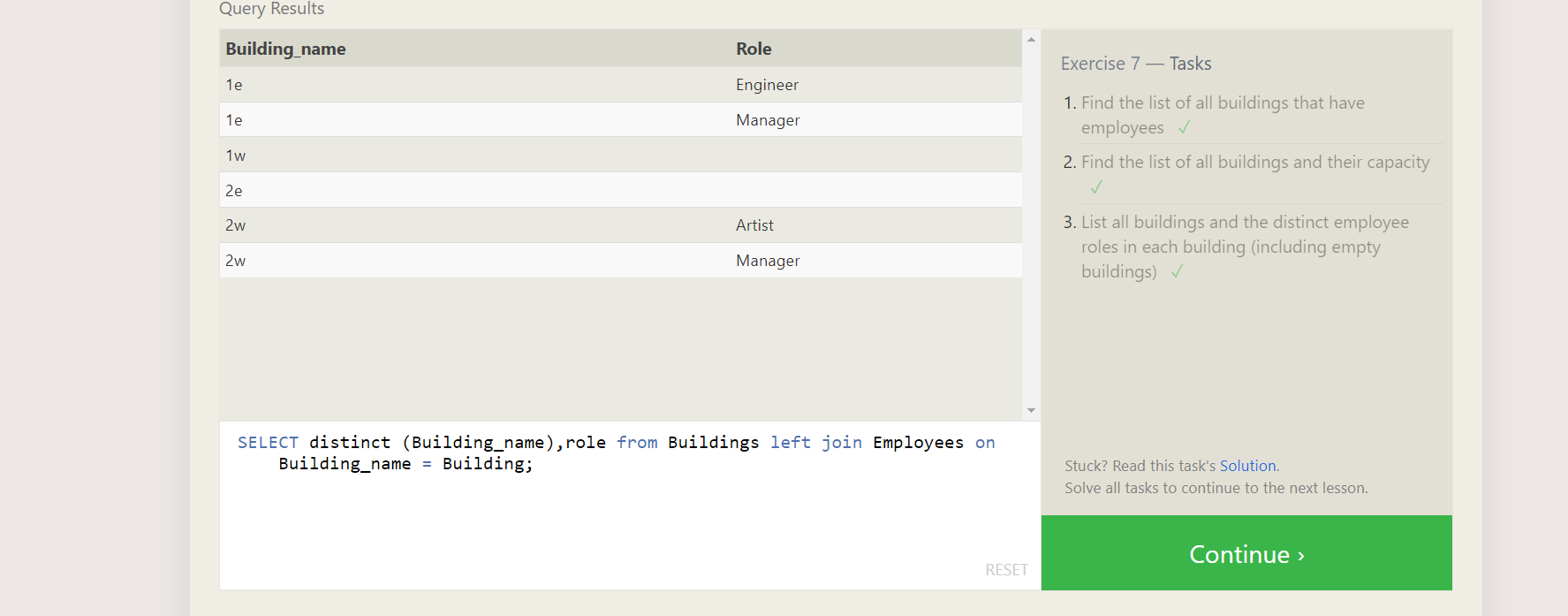
1. SELECT \* FROM movies join Boxoffice on movies.id = Boxoffice.Movie\_id;

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;

SQL Lesson 7: OUTER JOINs

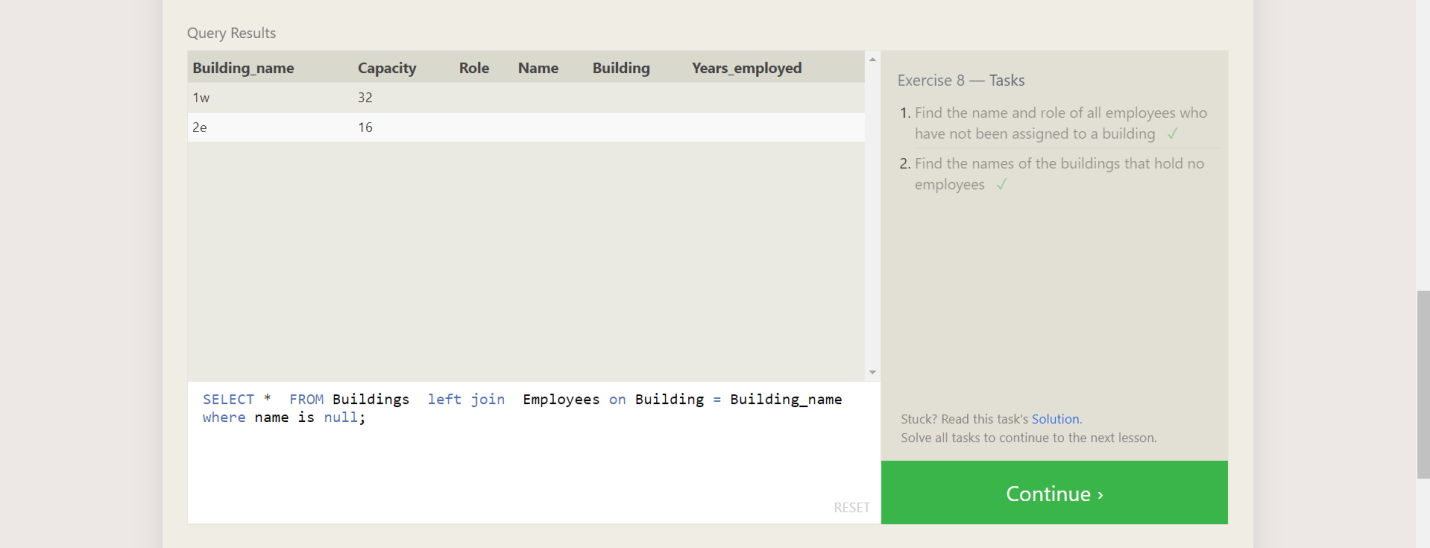


1. SELECT distinct(Buildings) from employees;

2. SELECT \*from Buildings;

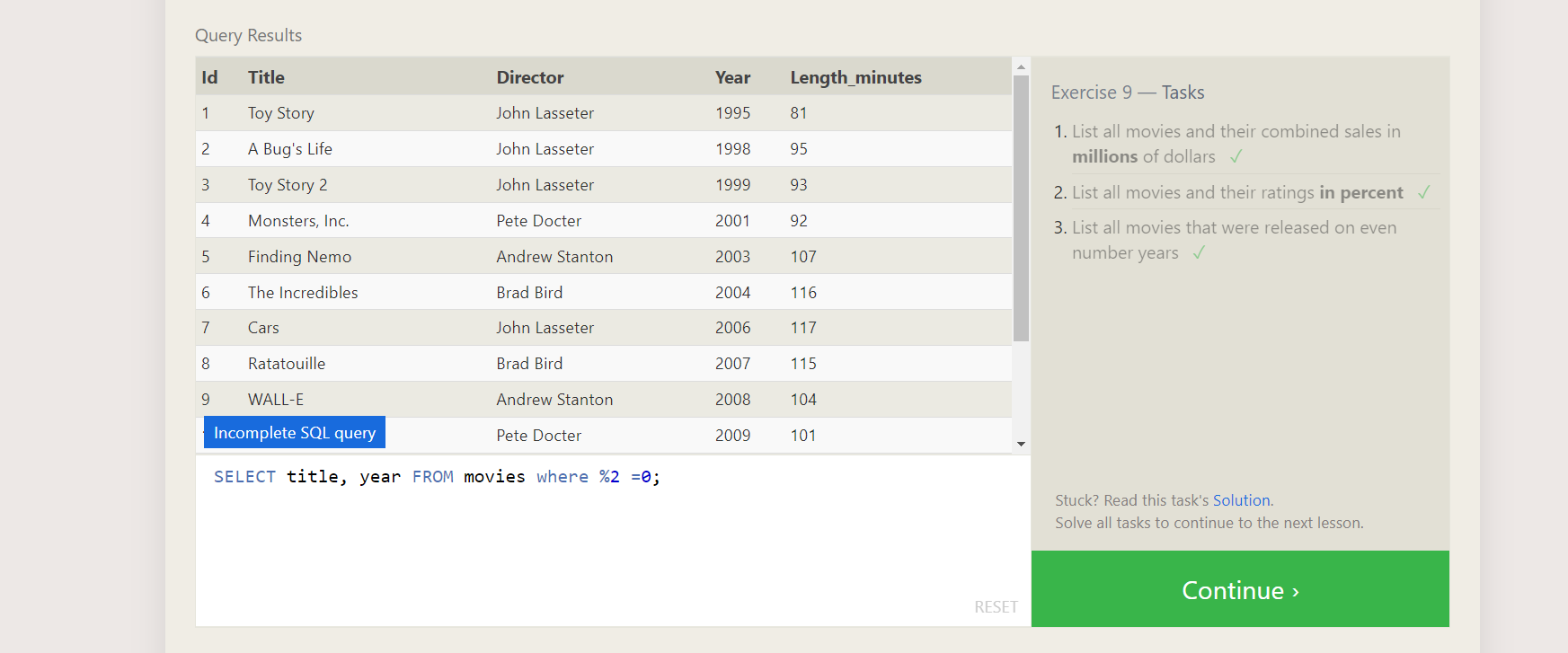
3. SELECT distinct (Building\_name),role from Buildings left join Employees on Building\_name = Building;

SQL Lesson 8: A short note on NULLs



1. SELECT \* FROM employees where Building is Null;
2. SELECT \* FROM Buildings left join Employees on Building = Building\_name where name is null;

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

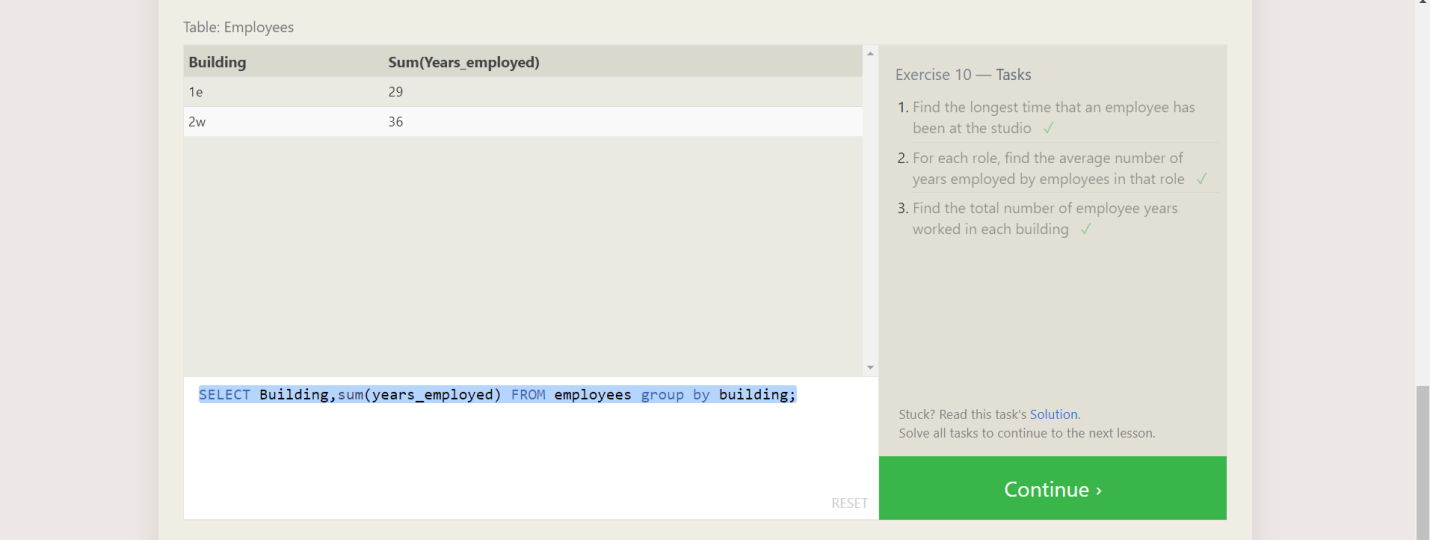
FROM movies

JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

1. SELECT title, year FROM movies where %2 = 0;

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

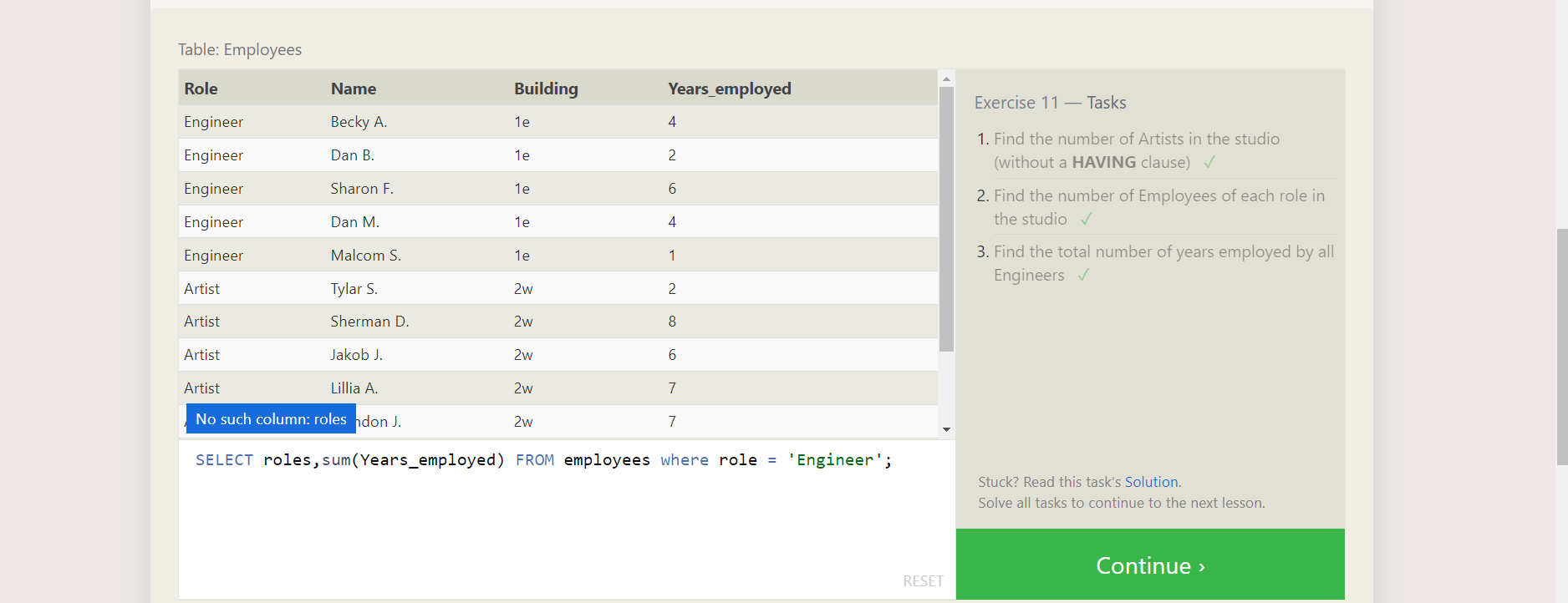


1. SELECT max(years\_employed) FROM employees;

2. SELECT role,avg(Years\_employed),count()from Employees group by role;

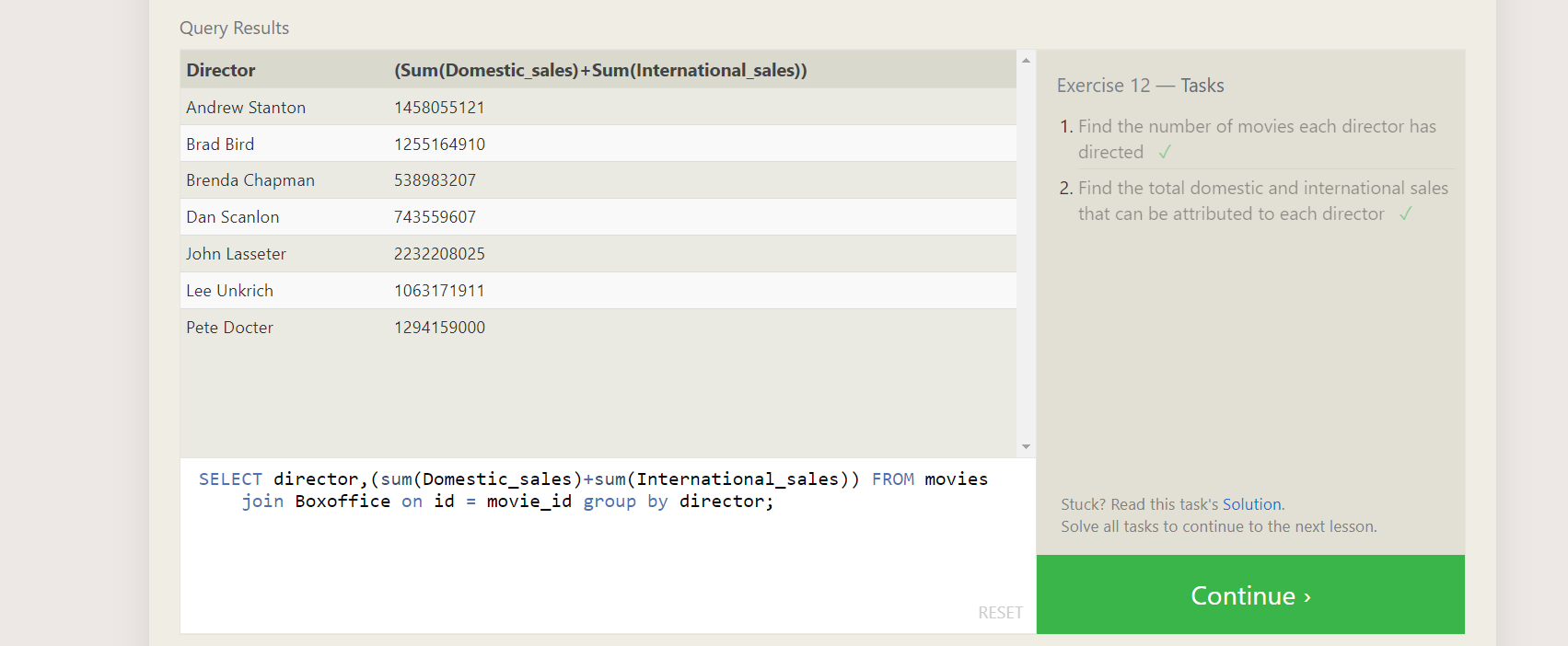
3. SELECT Building,sum(years\_employed) FROM employees group by building;

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



1. Select count(\*) from Employees where role ='Artist';
2. SELECT role,count(\*) from employees group by role;
3. SELECT roles,sum(Years\_employed) FROM employees where role = 'Engineer';

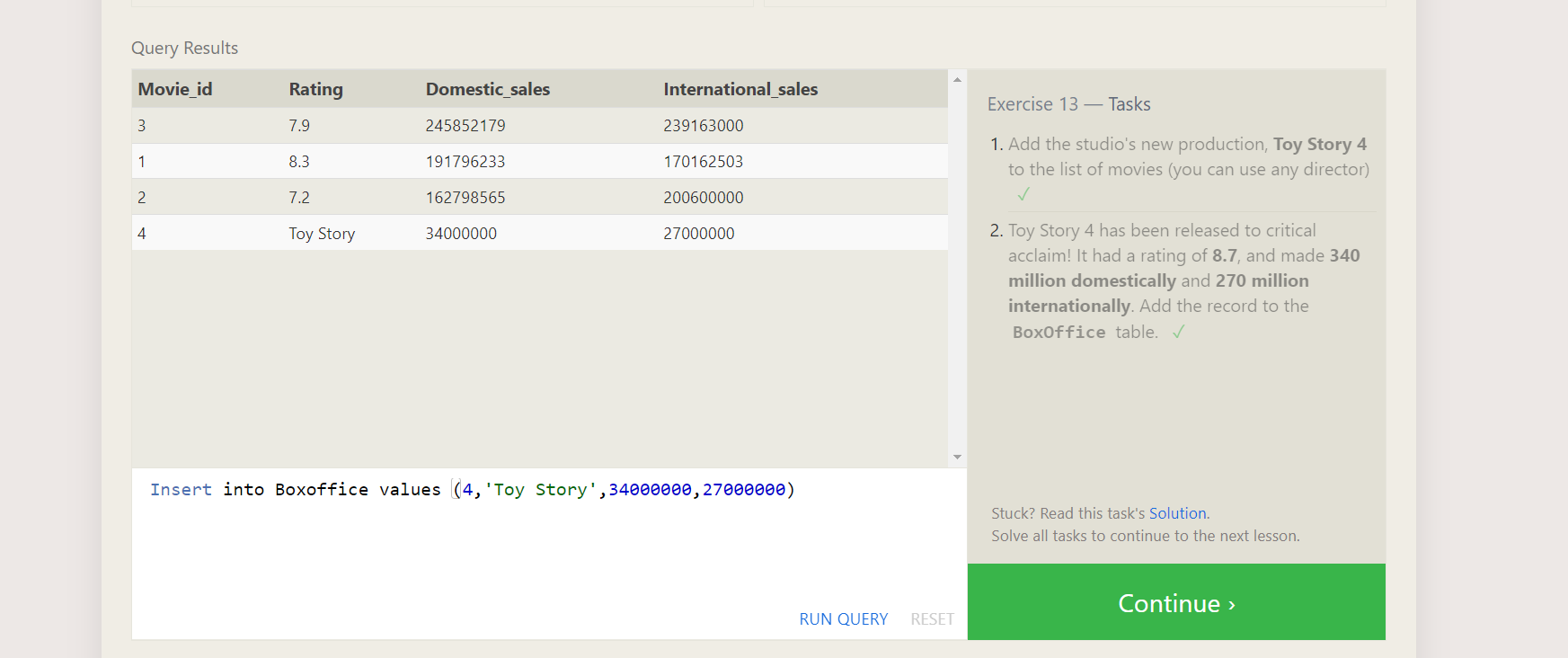
SQL Lesson 12: Order of execution of a Query



1. SELECT director,count()FROM movies group by director;

2. SELECT director,(sum(Domestic\_sales)+sum(International\_sales)) FROM movies join Boxoffice on id = movie\_id group by director;

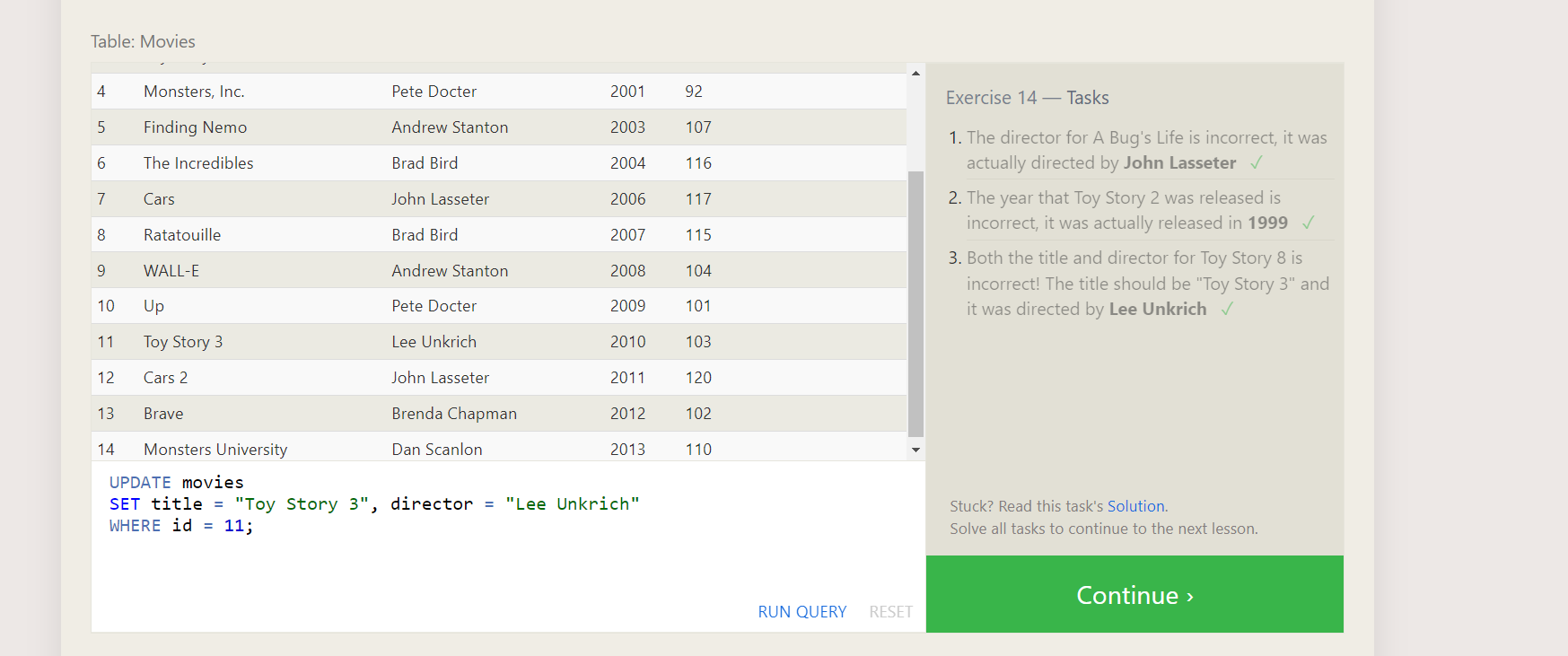
SQL Lesson 13: Inserting rows:



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

2. Insert into Boxoffice values (4,'Toy Story',34000000,27000000);

SQL Lesson 14: Updating rows:



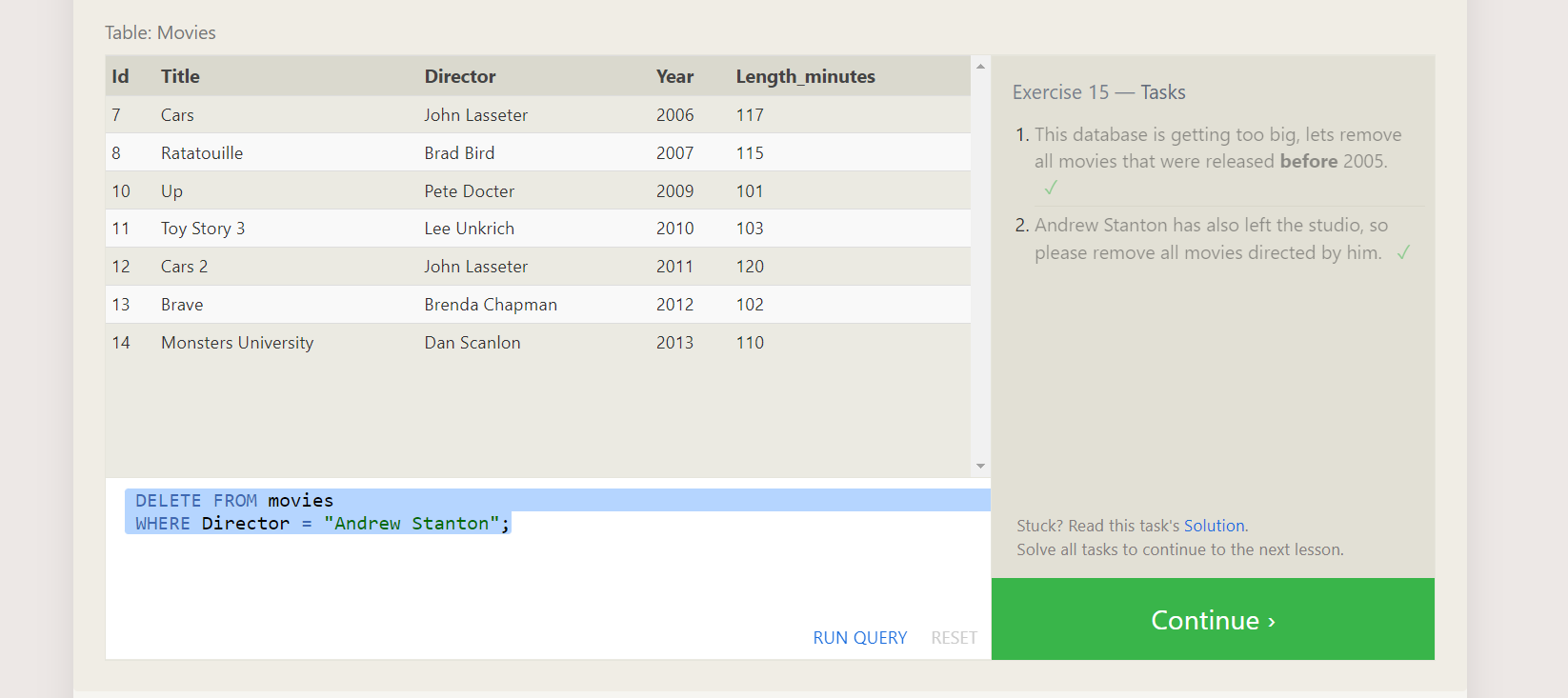
1. update Movies set Director = "John Lasseter" where id = 2;

2. update Movies SET year = 1999 WHERE id = 3;

3. UPDATE movies SET title = "Toy Story 3", director = "Lee Unkrich"

WHERE id = 11;

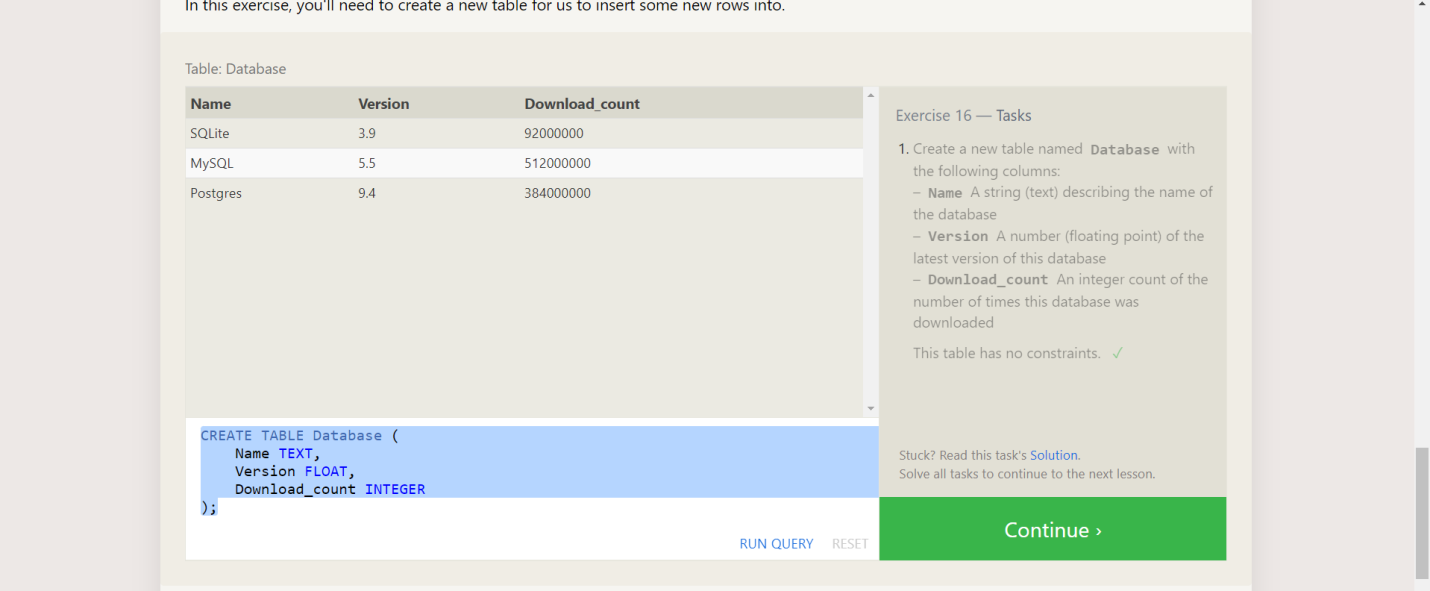
SQL Lesson 15: Deleting rows:



1. DELETE FROM movies where year < 2005;

2. DELETE FROM movies WHERE Director = "Andrew Stanton";

SQL Lesson 16: Creating tables



CREATE TABLE Database (

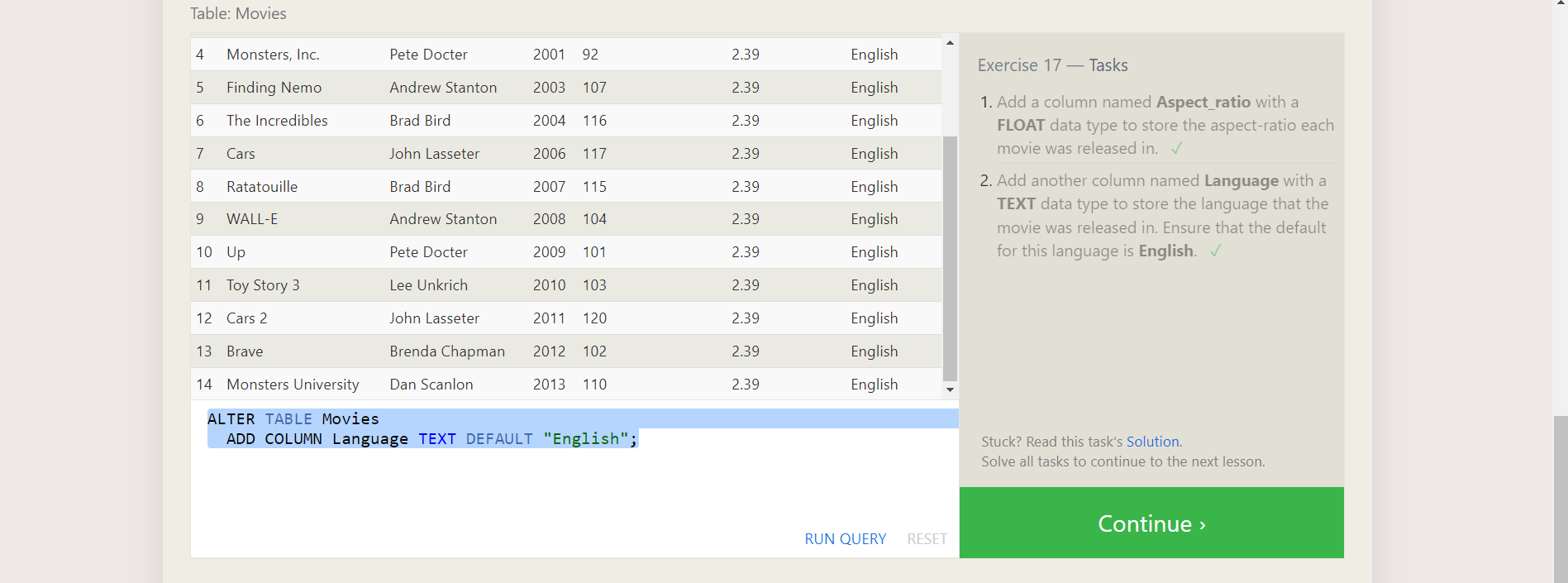
Name TEXT,

Version FLOAT,

Download\_count INTEGER

);

SQL Lesson 17: Altering tables



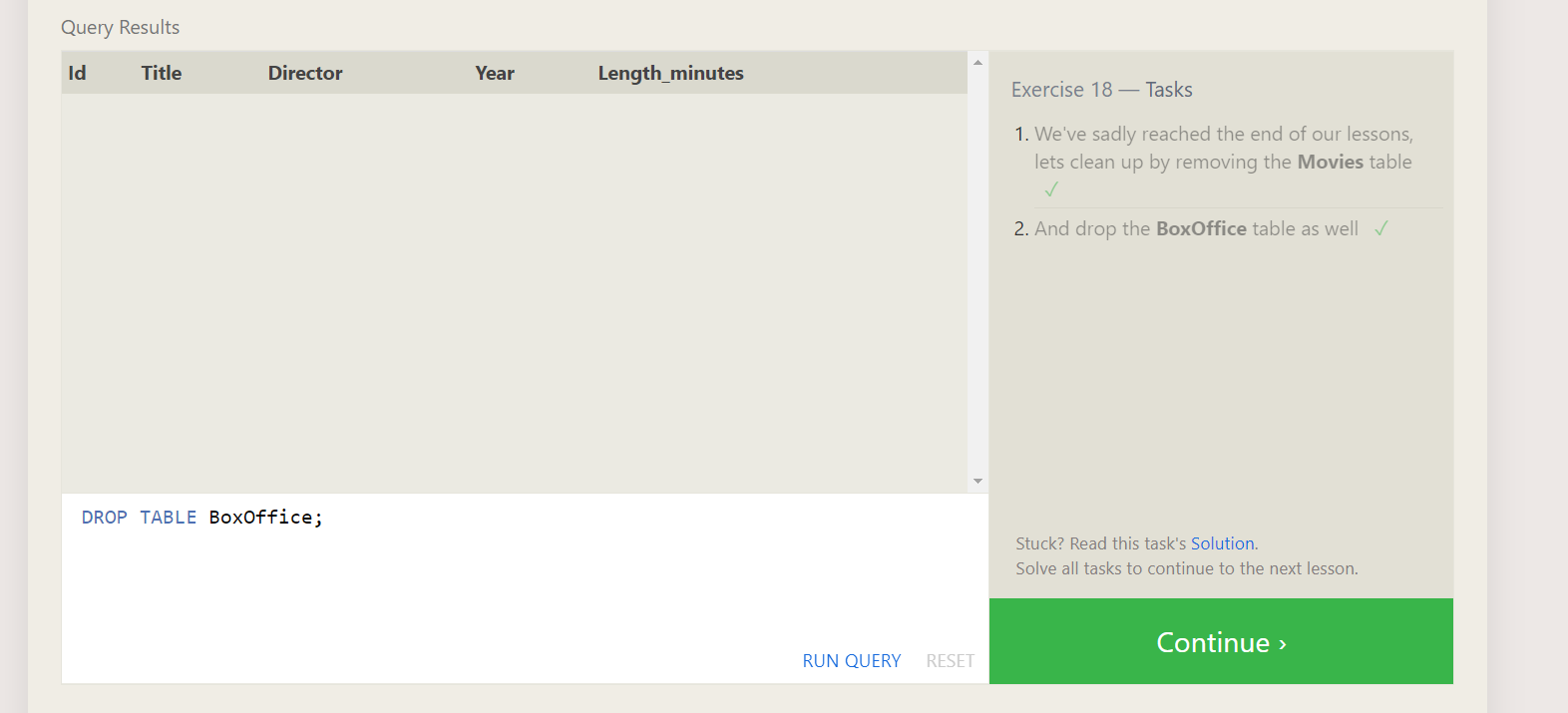
1. ALTER TABLE Movies

ADD COLUMN Aspect\_ratio FLOAT DEFAULT 2.39;

2. ALTER TABLE Movies

ADD COLUMN Language TEXT DEFAULT "English";

SQL Lesson 18: Dropping tables:



1. DROP TABLE Movies;

2. DROP TABLE BoxOffice;