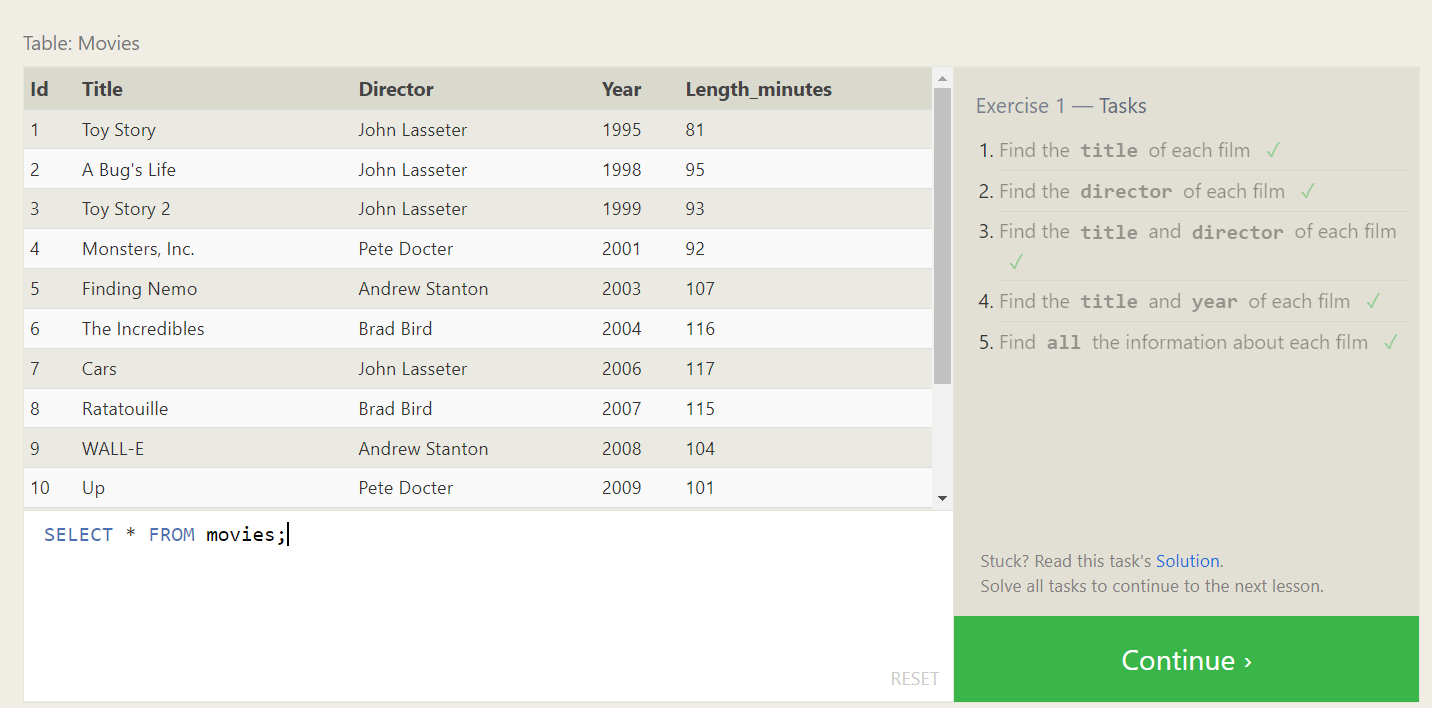
**SQL Lesson 1: SELECT queries**



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;

**SQL Lesson 2: Queries with constraints**

1. SELECT \* FROM movies

WHERE ID = 6;

2. SELECT \* FROM movies

WHERE year BETWEEN 2000 AND 2010;

3. SELECT \* FROM movies

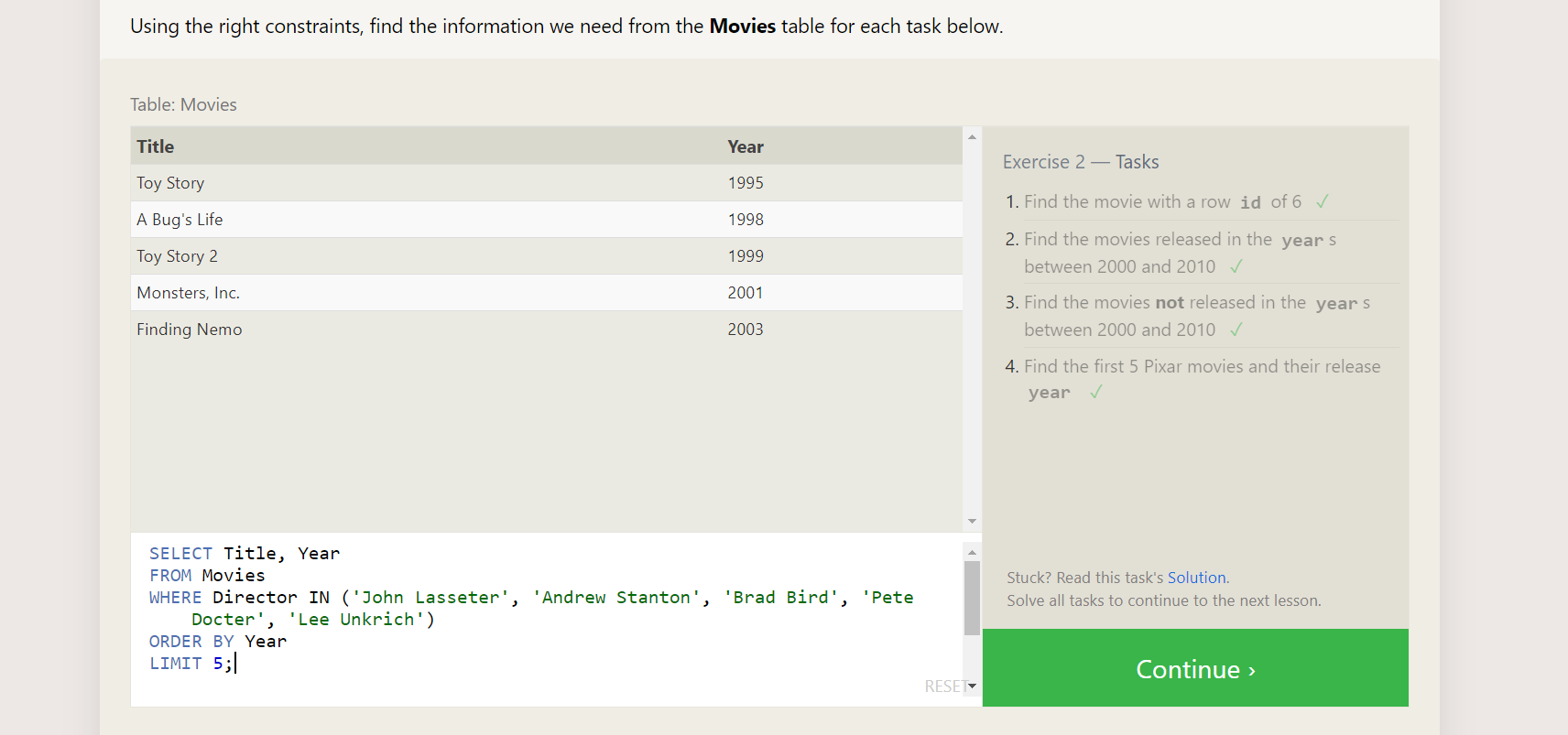
WHERE year NOT BETWEEN 2000 AND 2010;

4. SELECT Title, Year FROM movies

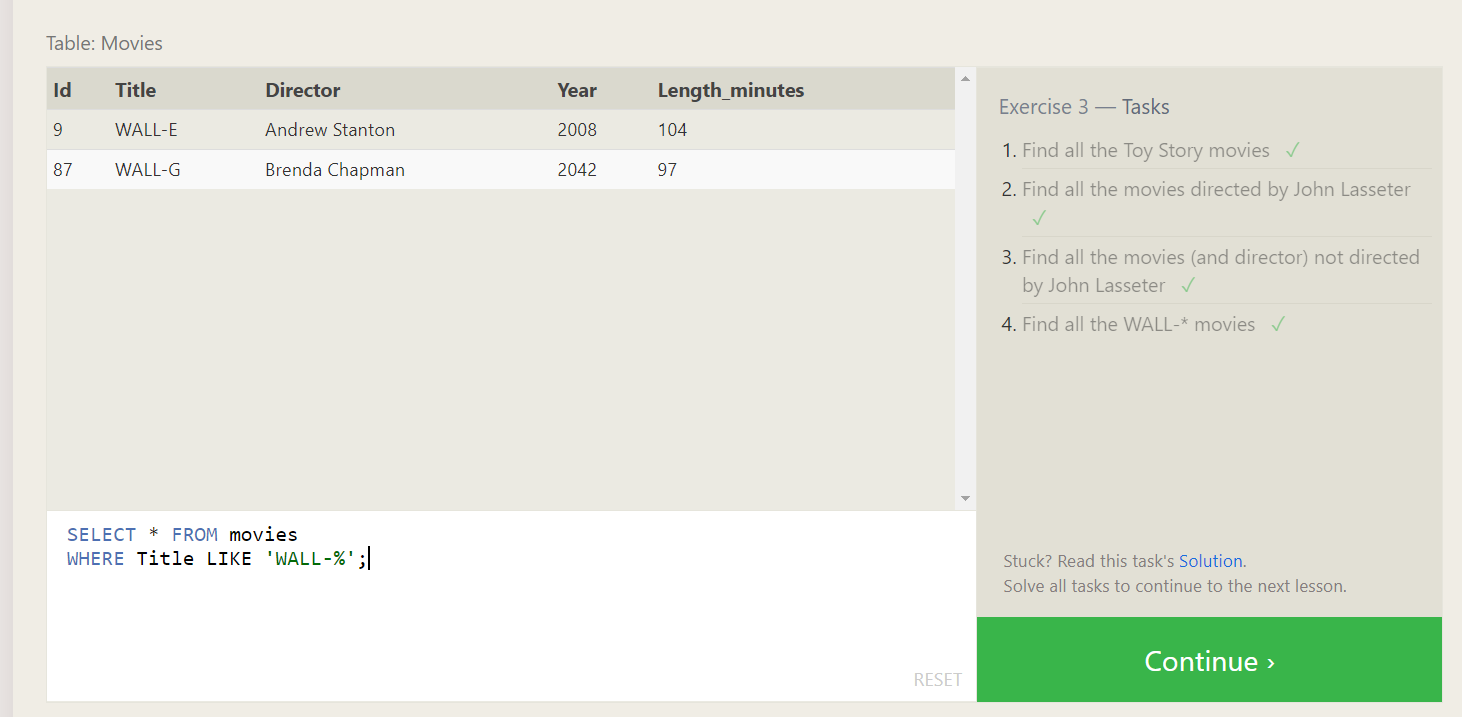
WHERE Director IN ('John Lasseter', 'Andrew Stanton', 'Brad Bird', 'Pete Docter', 'Lee Unkrich')

ORDER BY Year

LIMIT 5;



**SQL Lesson 3: Queries with constraints**



1. SELECT \* FROM movies

WHERE Title LIKE 'Toy Story%';

2. SELECT \* FROM movies

WHERE director LIKE 'John Lasseter%';

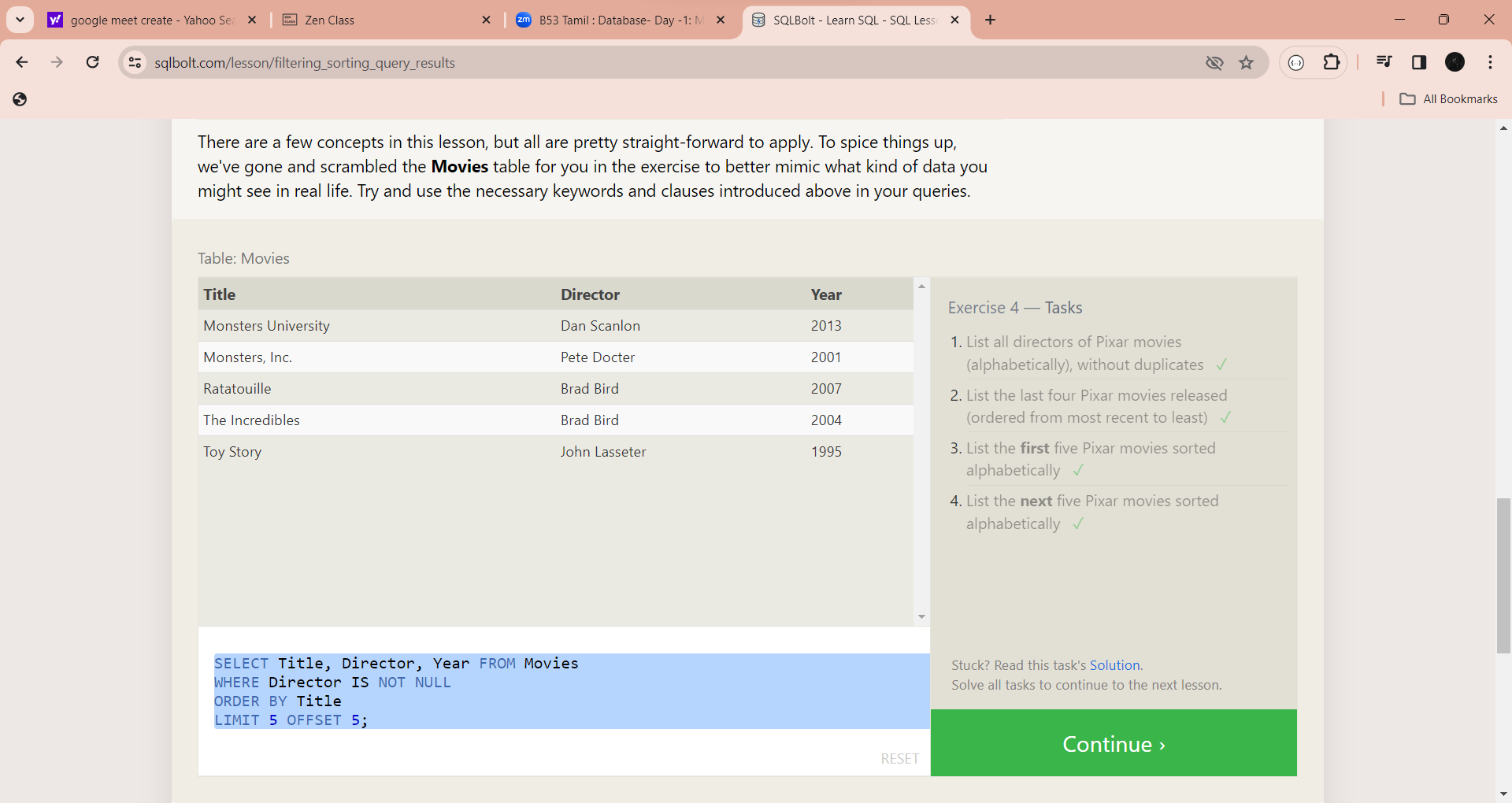
3. SELECT \* FROM movies

WHERE director NOT LIKE 'John Lasseter%';

4. SELECT \* FROM movies

WHERE Title LIKE 'WALL-%';

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

****

1. SELECT DISTINCT Director FROM Movies

WHERE Director IS NOT NULL

ORDER BY Director;

2. SELECT Title, Year FROM Movies

WHERE Director IS NOT NULL

ORDER BY Year DESC

LIMIT 4;

3. SELECT Title, Director, Year FROM Movies

WHERE Director IS NOT NULL

ORDER BY Title

LIMIT 5;

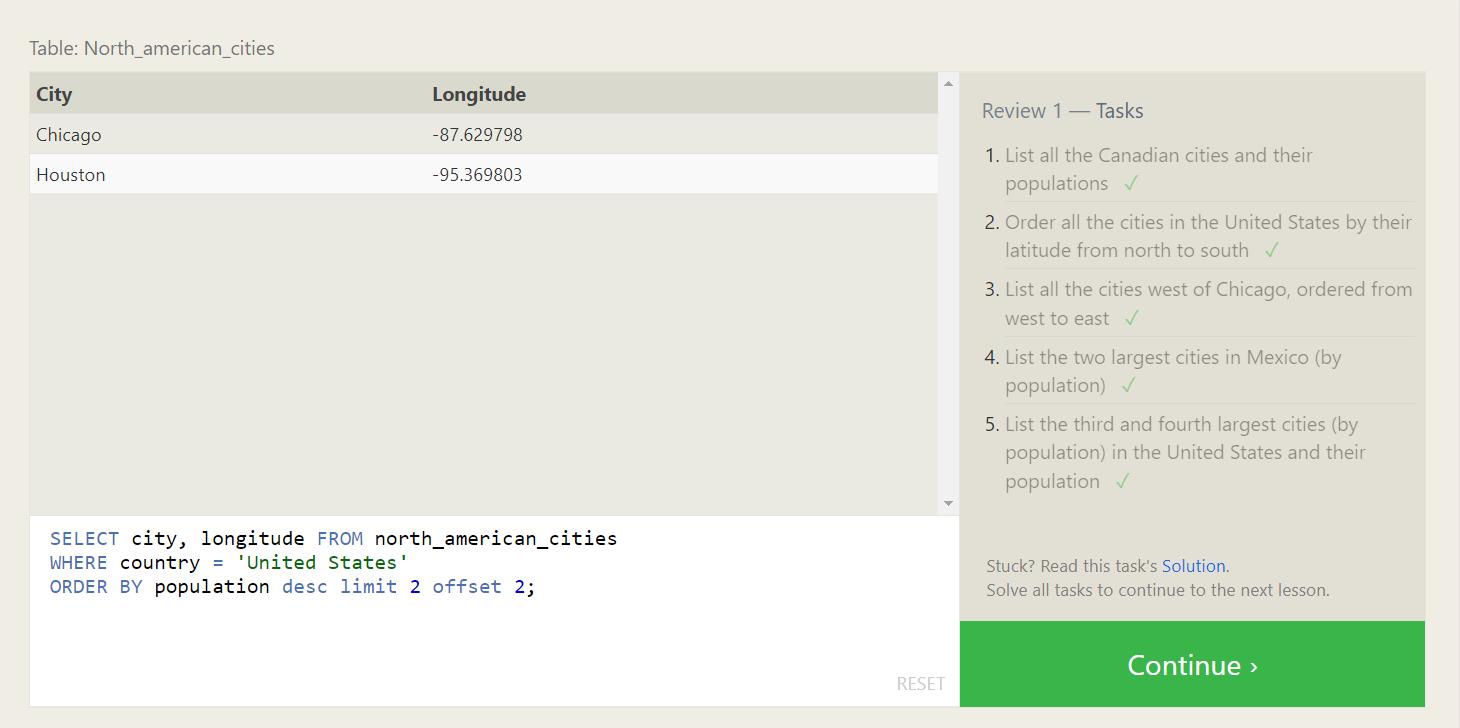
4. SELECT Title, Director, Year FROM Movies

WHERE Director IS NOT NULL

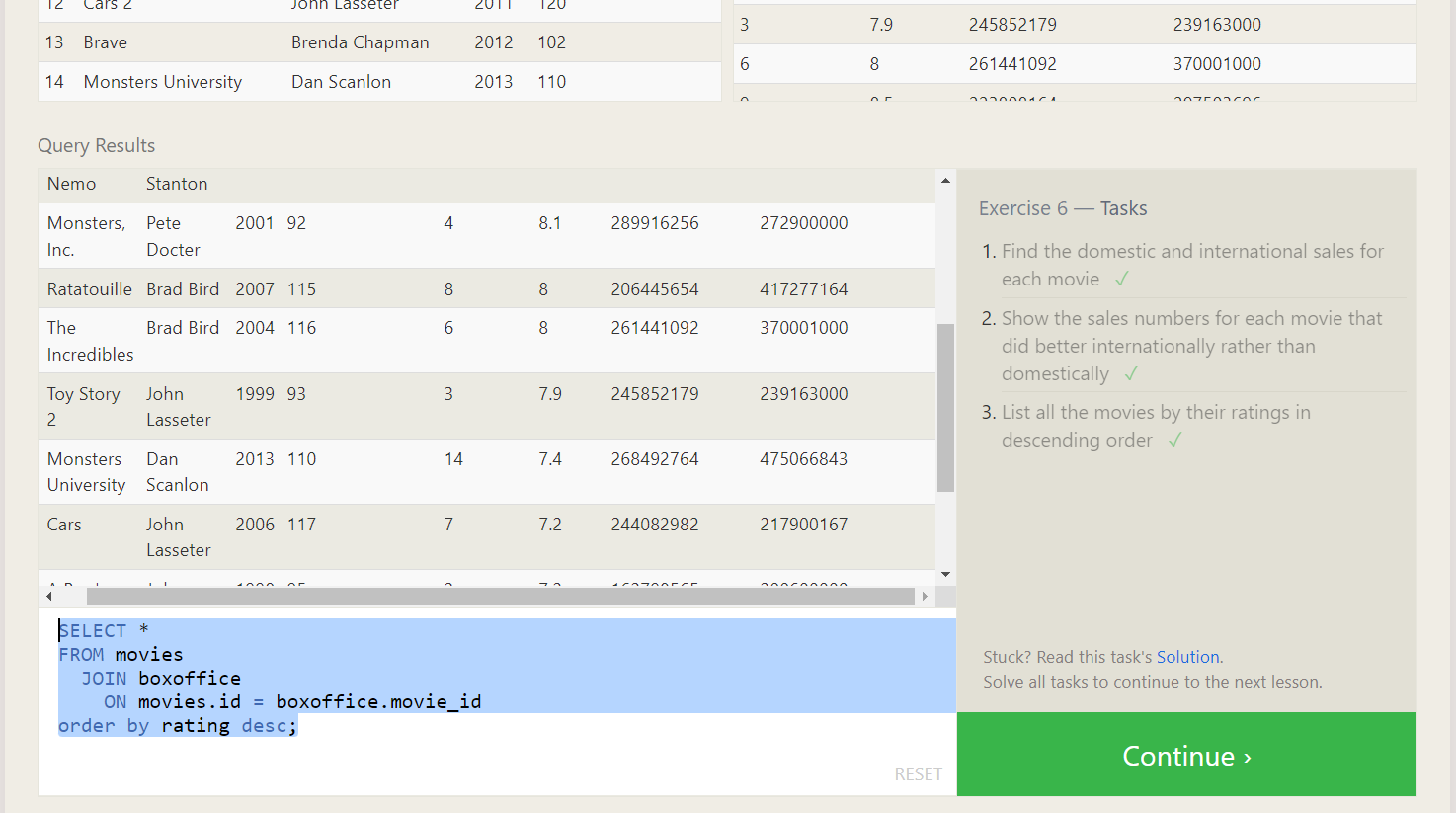
ORDER BY Title

LIMIT 5 OFFSET 5;

**SQL Review 5 : Simple SELECT Queries**



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

****

1. SELECT title, domestic\_sales, international\_sales

FROM movies

JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

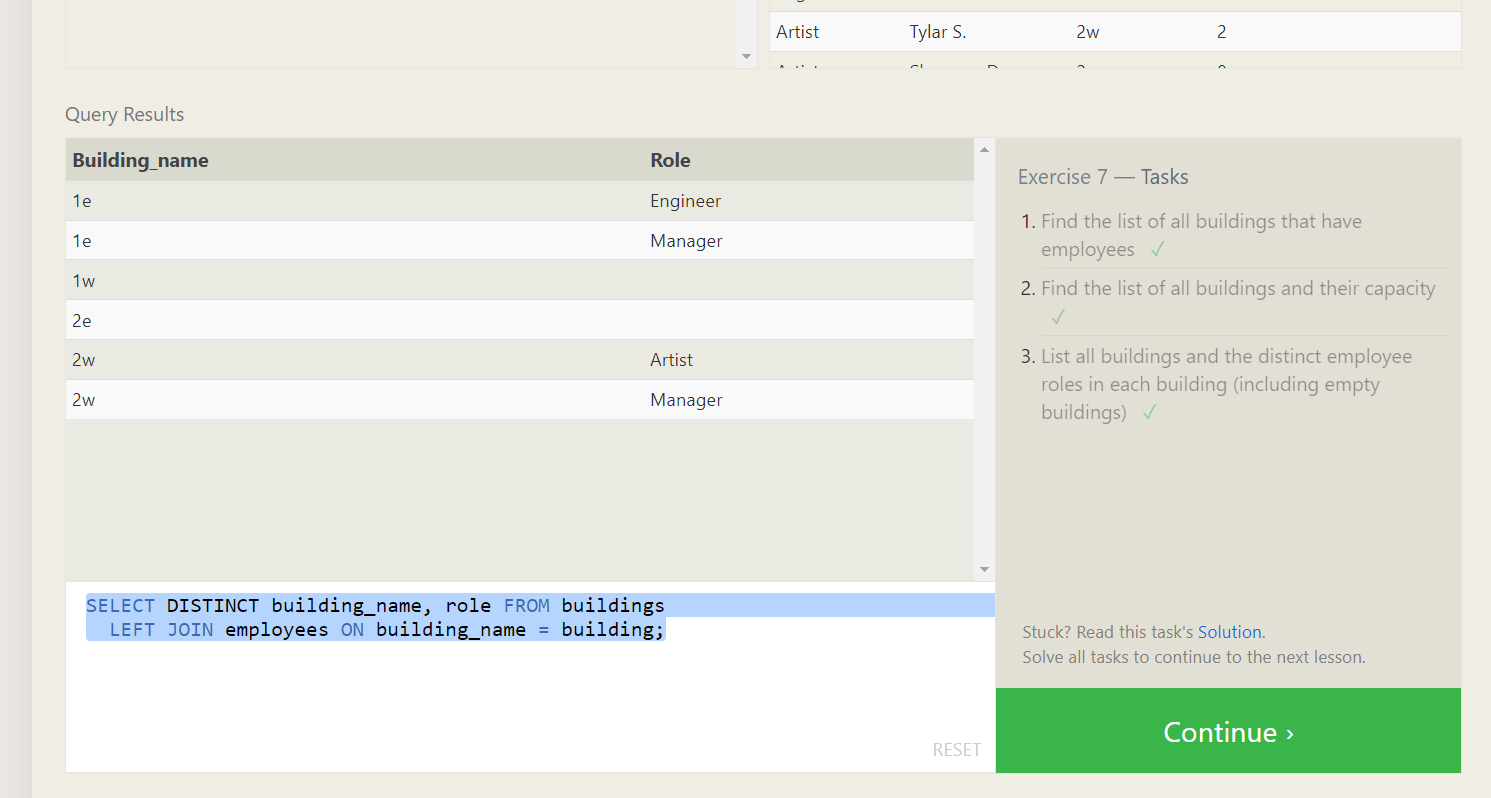
2. SELECT title, domestic\_sales, international\_sales

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;

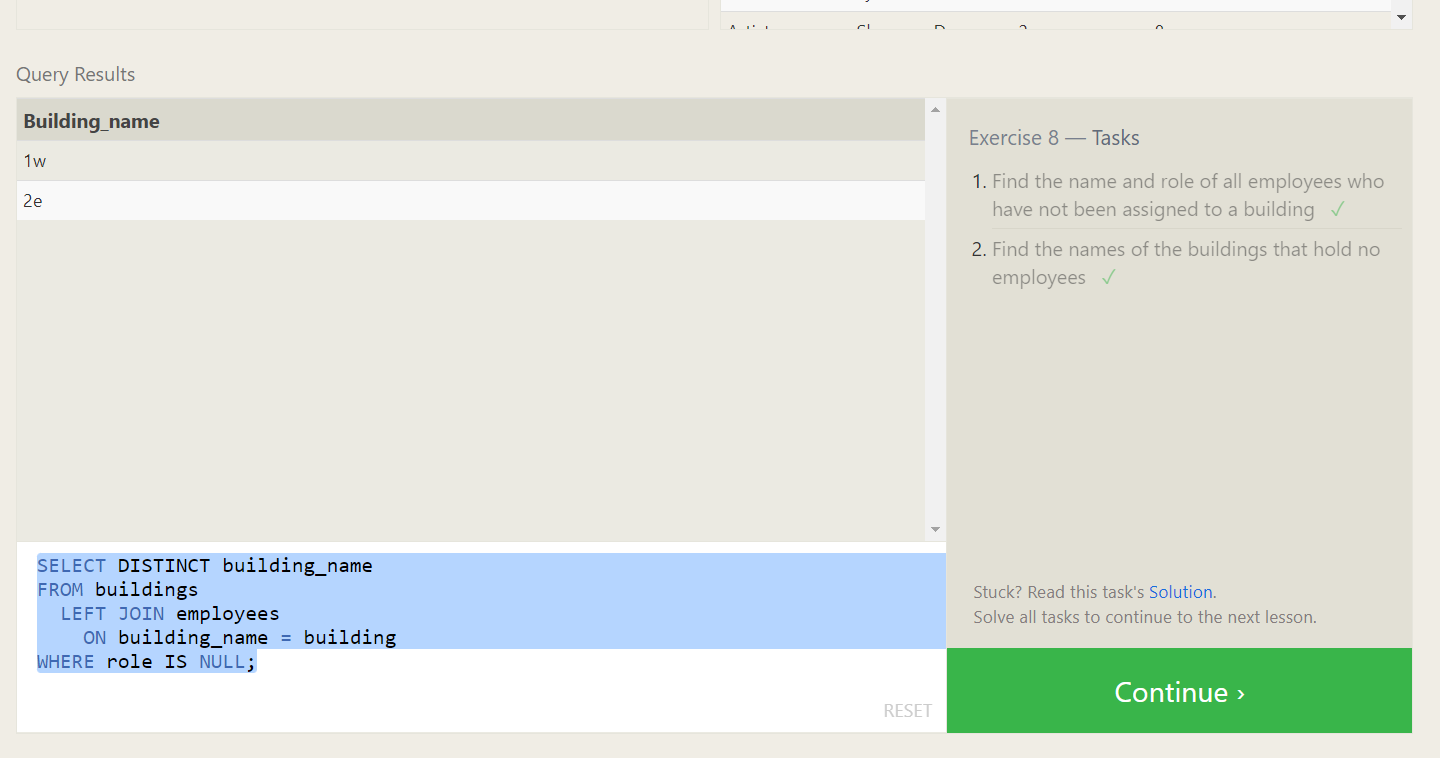
**SQL Lesson 7: OUTER JOINs**

****

1. **SELECT DISTINCT building 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 name, role FROM employees**

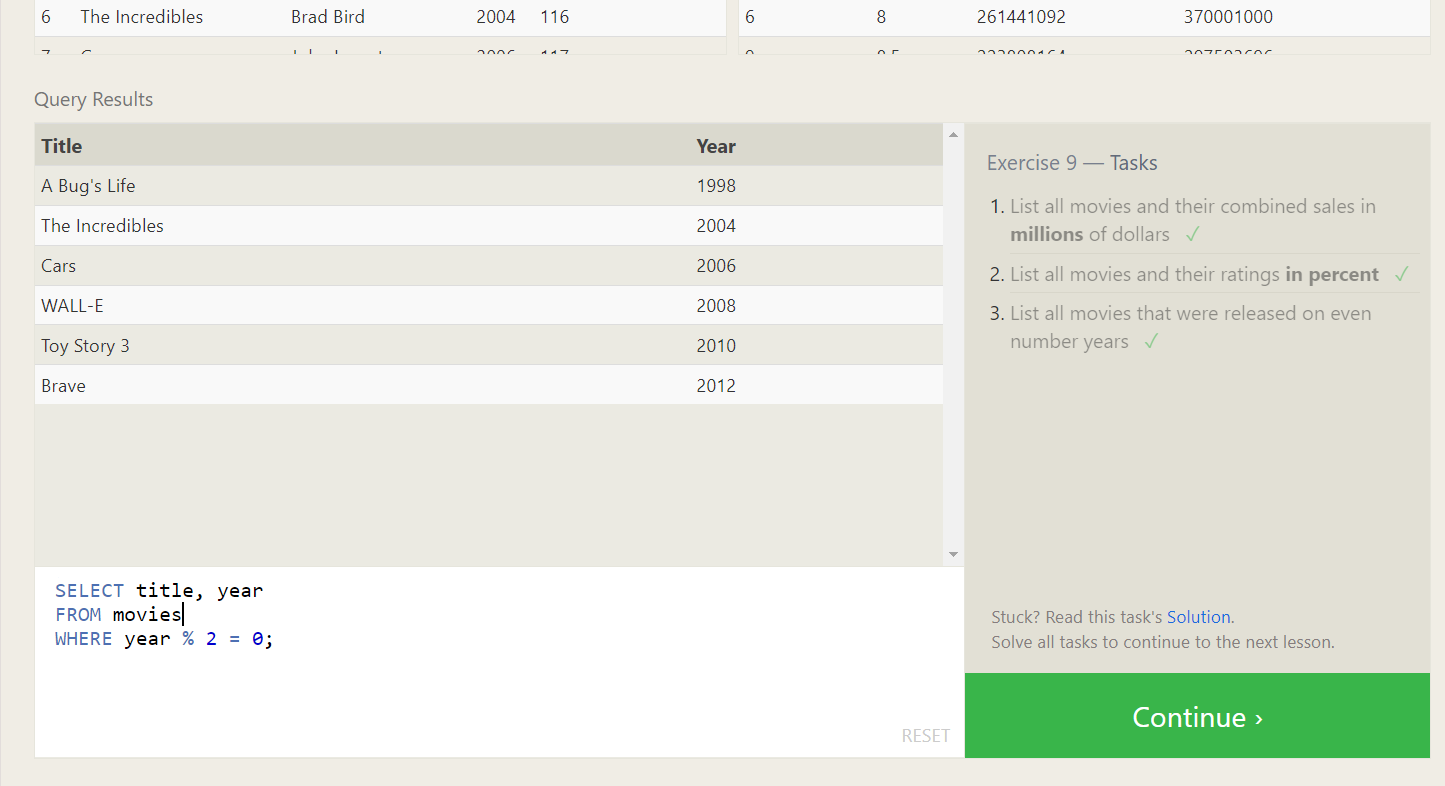
**WHERE building IS NULL;**

**2. SELECT DISTINCT building\_name FROM buildings**

**LEFT JOIN employees ON building\_name = building**

**WHERE role IS NULL;**

**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**

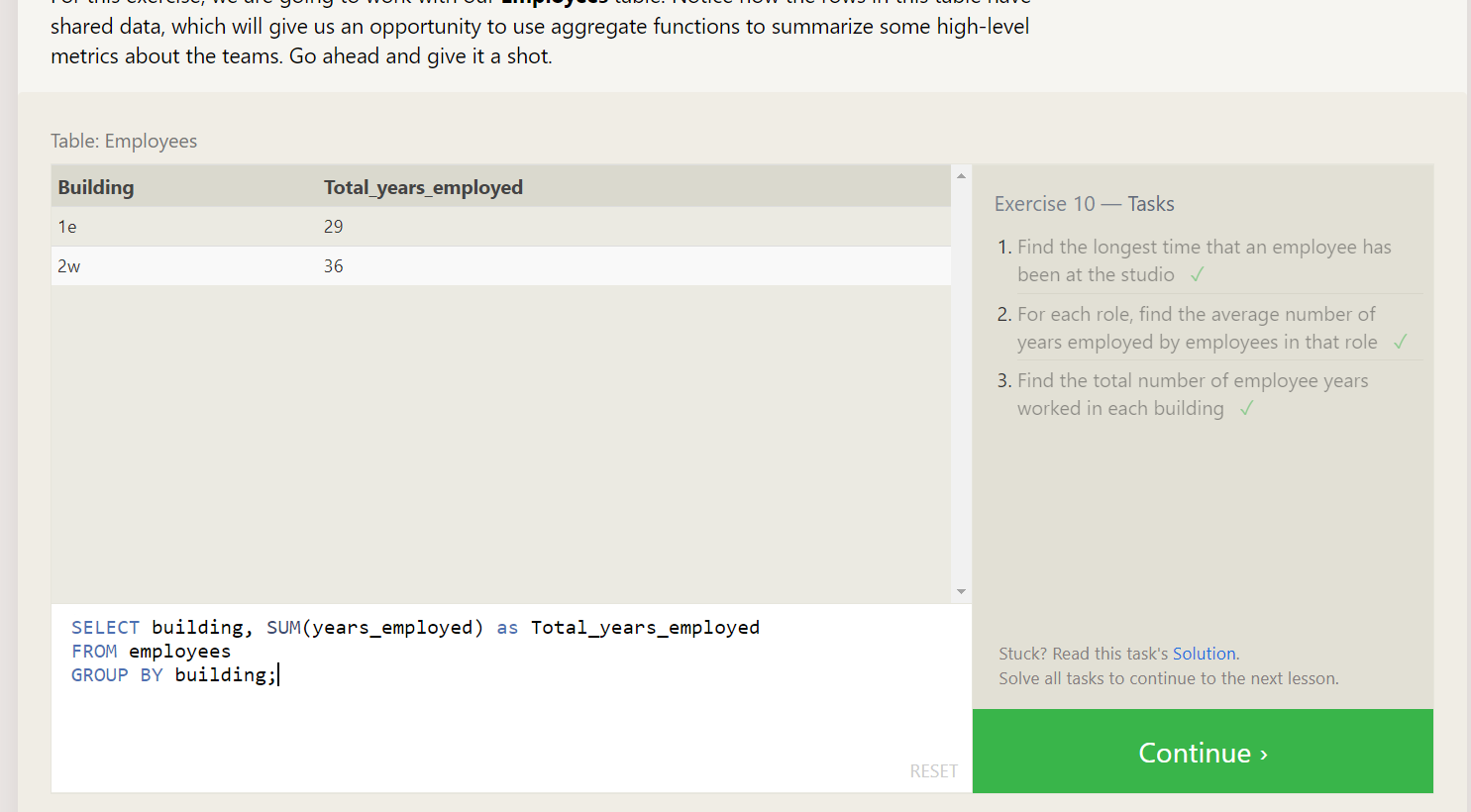
**ON movies.id = boxoffice.movie\_id;**

**3.** **SELECT title, year**

**FROM movies**

**WHERE year % 2 = 0;**

**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**

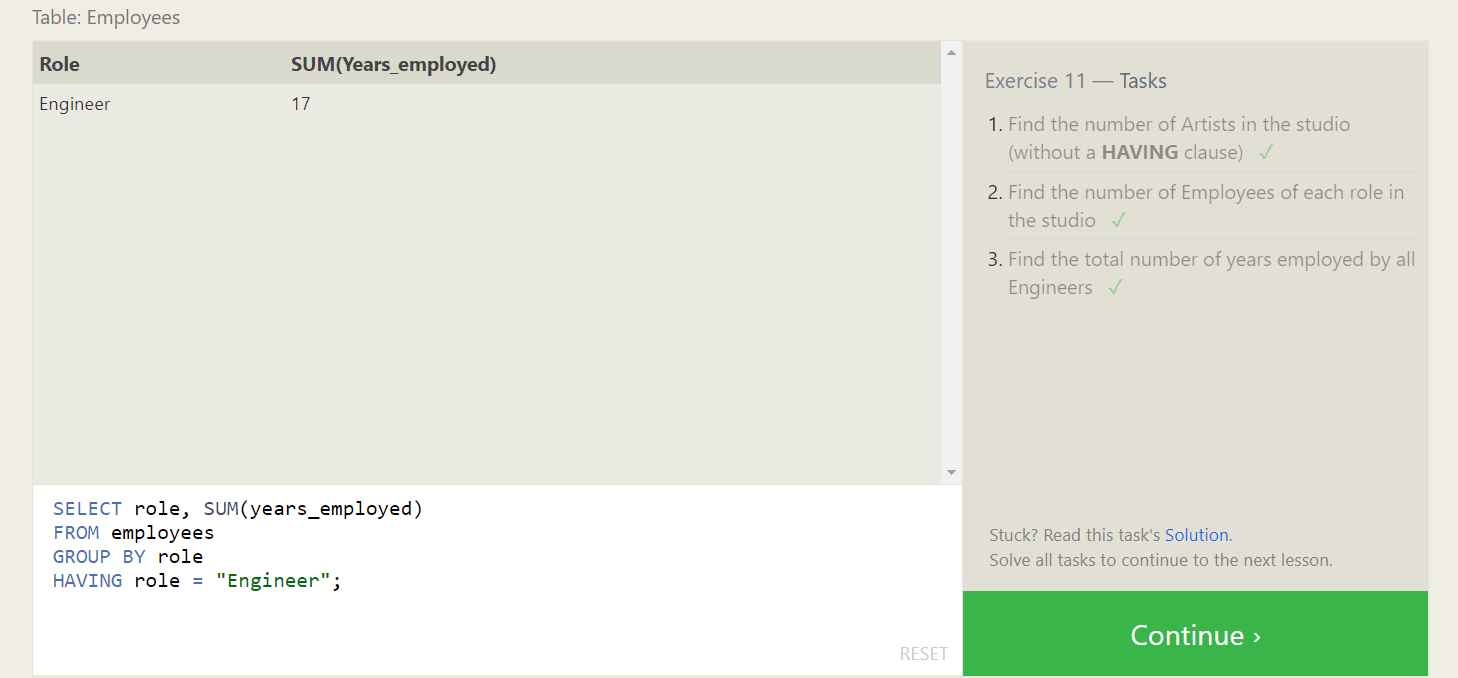
**GROUP BY role;**

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

**FROM employees**

**GROUP BY building;**

**SQL Lesson 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;**

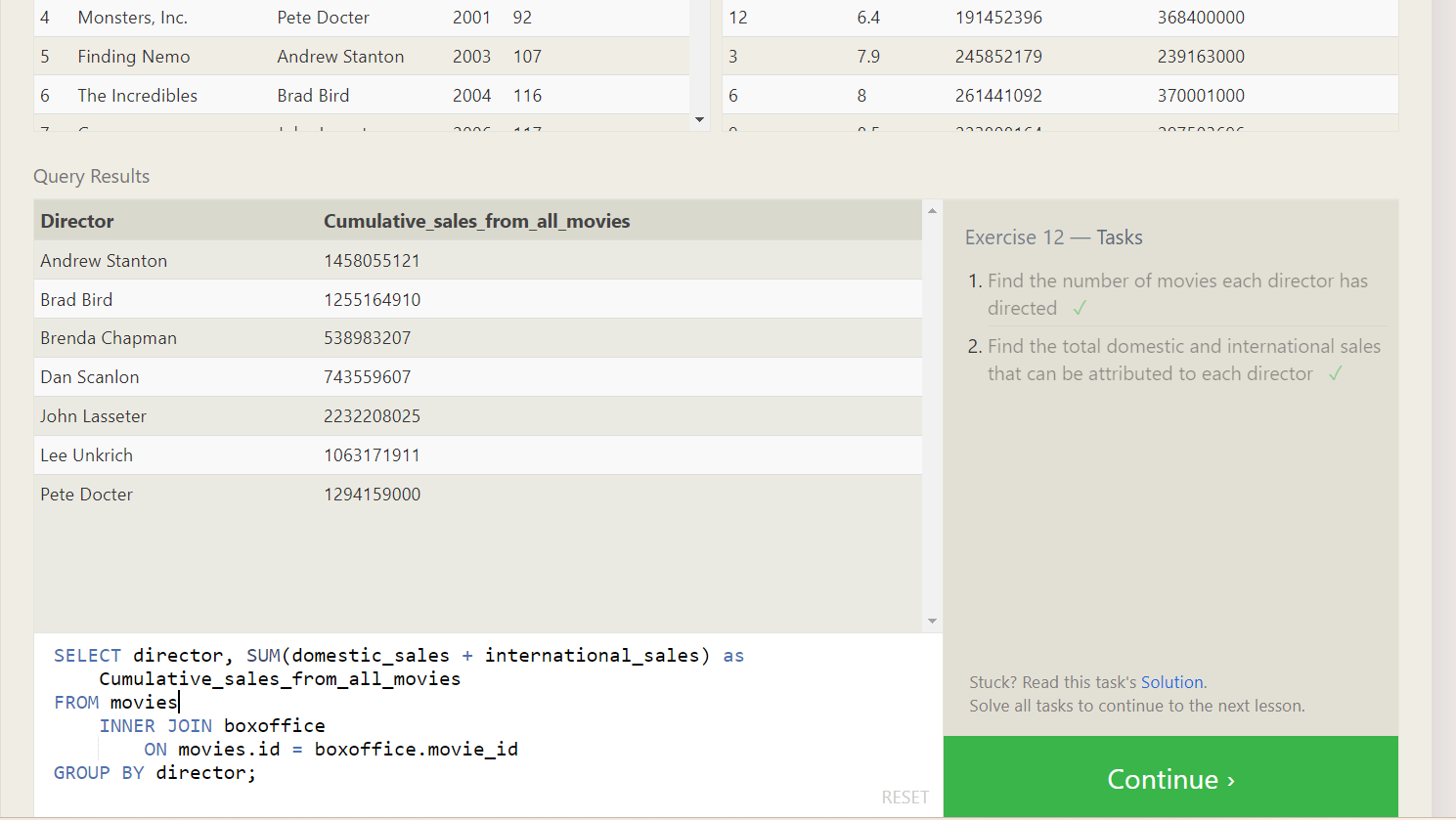
**3.** **SELECT role, SUM(years\_employed)**

**FROM employees**

**GROUP BY role**

**HAVING role = "Engineer";**

**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**

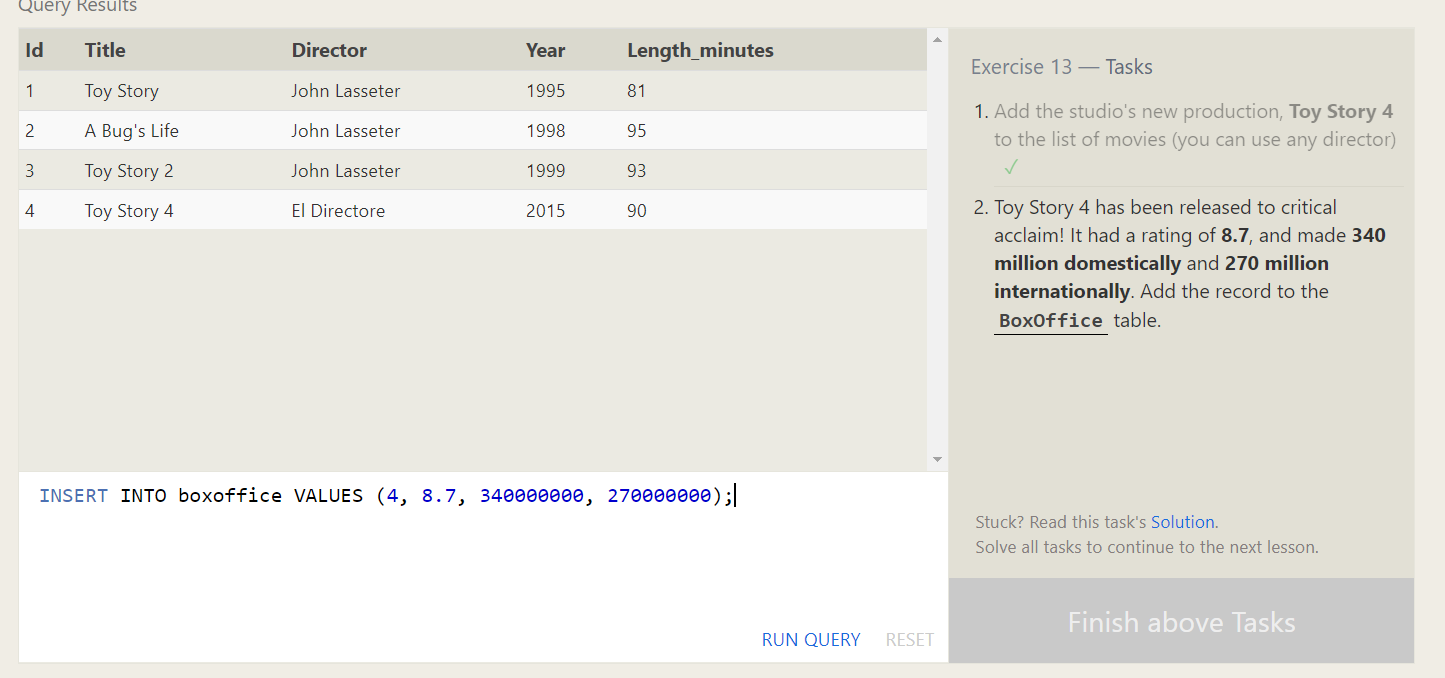
**FROM movies**

**INNER JOIN boxoffice**

**ON movies.id = boxoffice.movie\_id**

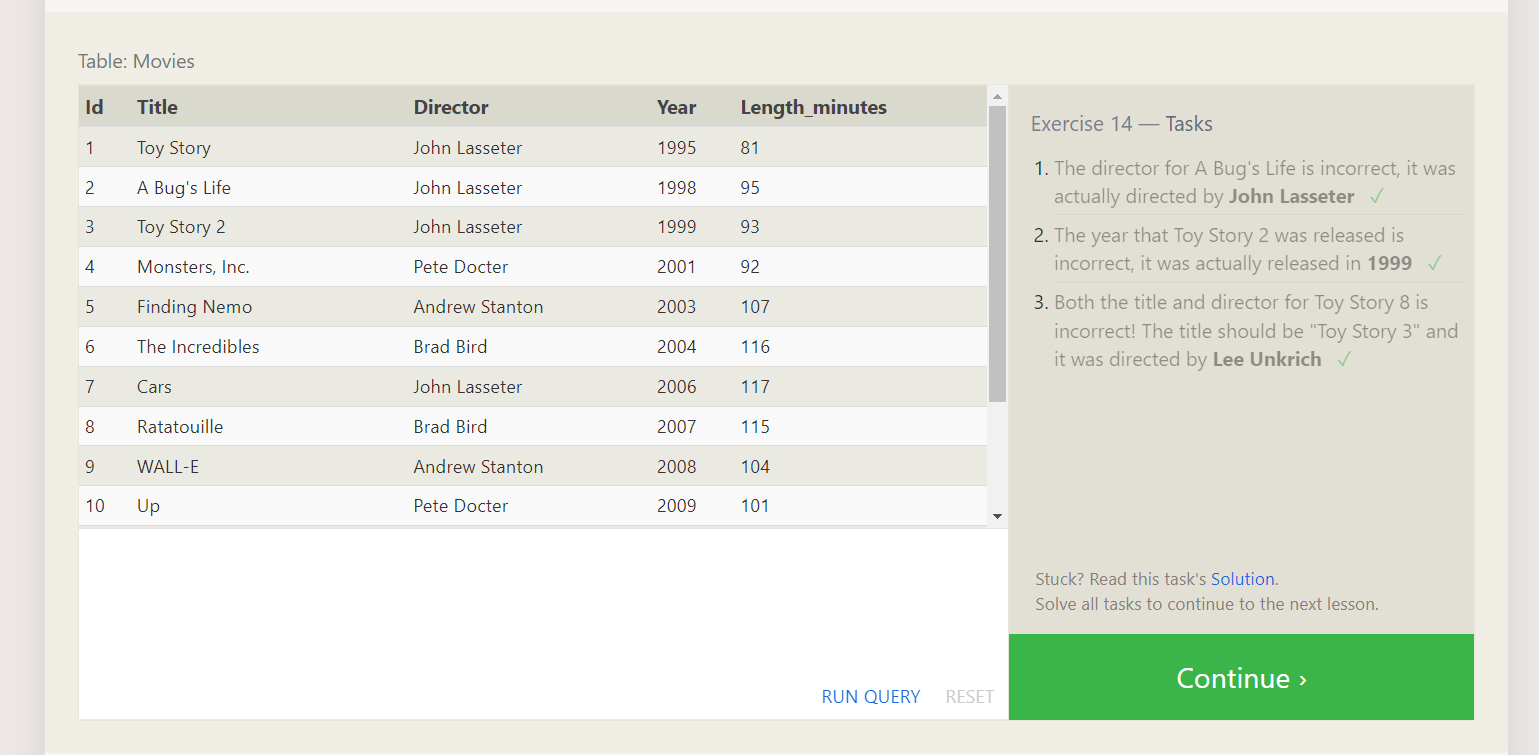
**GROUP BY director;**

**SQL Lesson 13: Inserting rows**

****

1. **INSERT INTO movies VALUES (4, "Toy Story 4", "El Directore", 2015, 90);**
2. INSERT INTO boxoffice VALUES (4, 8.7, 340000000, 270000000);

**SQL Lesson 14: Updating rows**



1. UPDATE movies

SET director = "John Lasseter"

WHERE id = 2;

2. UPDATE movies

SET year = 1999

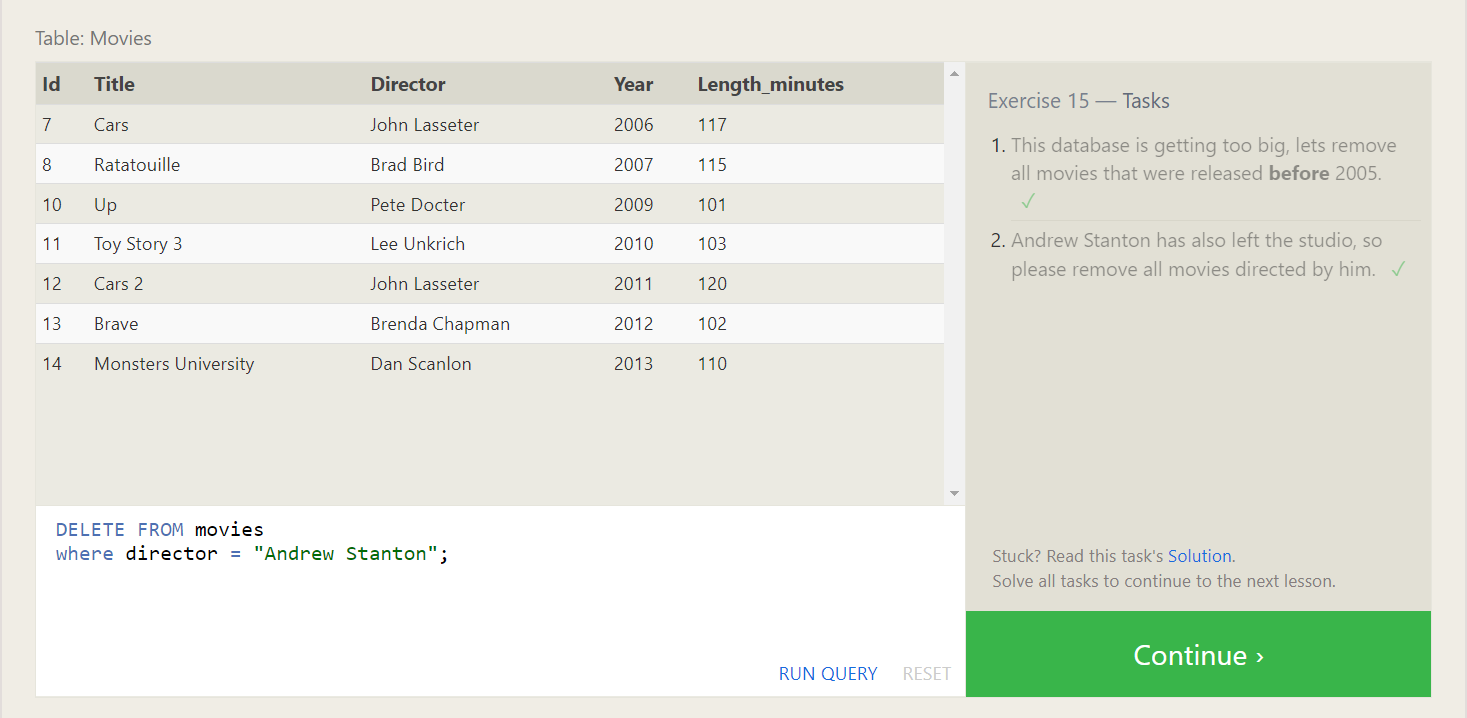
WHERE id = 3;

3. UPDATE movies

SET title = "Toy Story 3", director = "Lee Unkrich"

WHERE id = 11;

**SQL Lesson 15: Deleting rows**

****

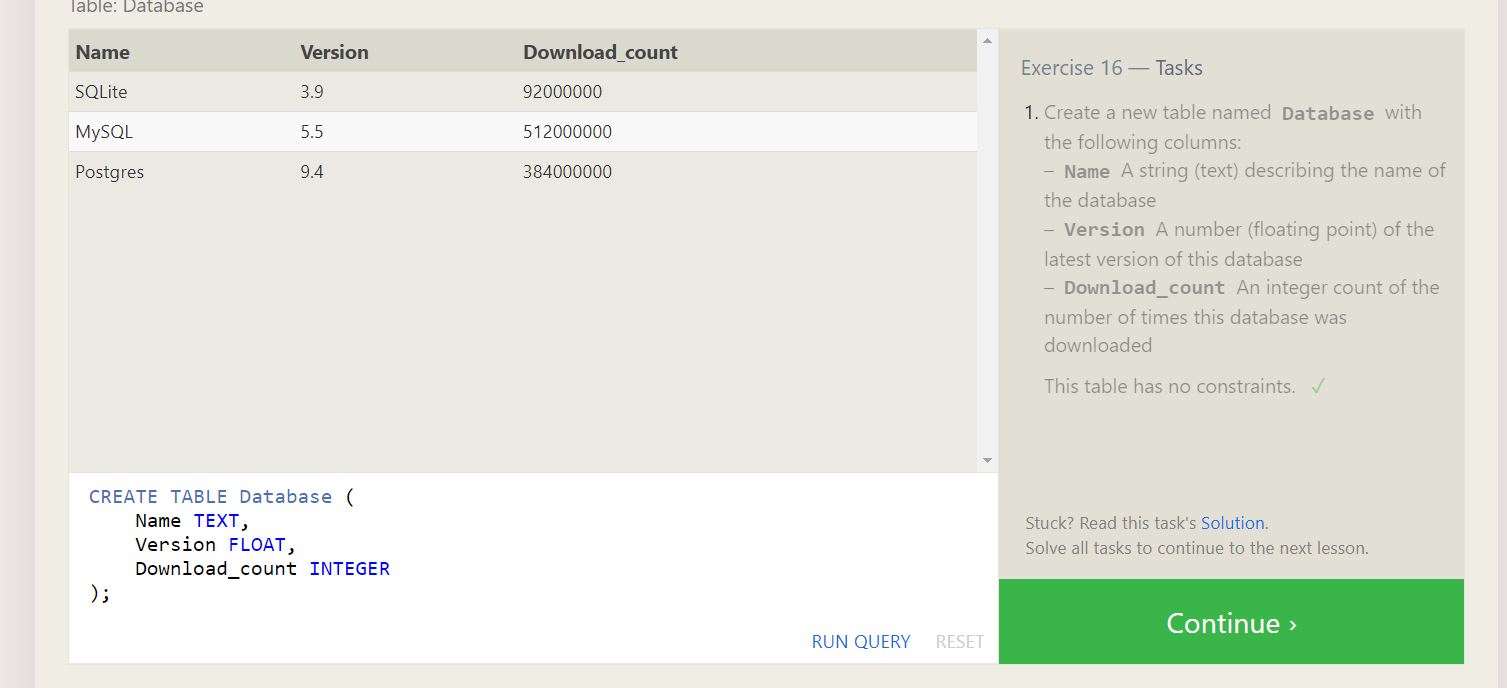
1. DELETE FROM movies

where year < 2005;

1. DELETE FROM movies

where director = "Andrew Stanton";

**SQL Lesson 16: Creating tables**



CREATE TABLE Database (

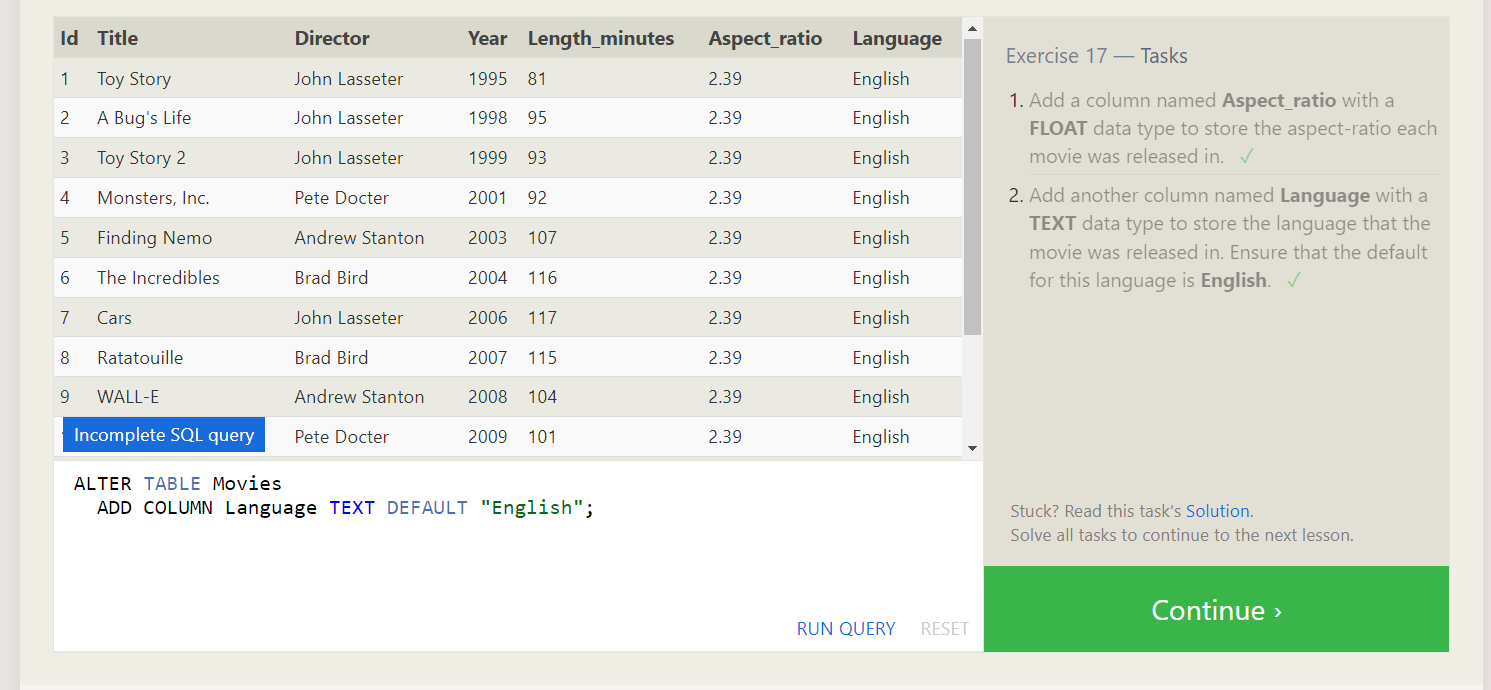
Name TEXT,

Version FLOAT,

Download\_count INTEGER

);

**SQL Lesson 17: Altering tables**

****

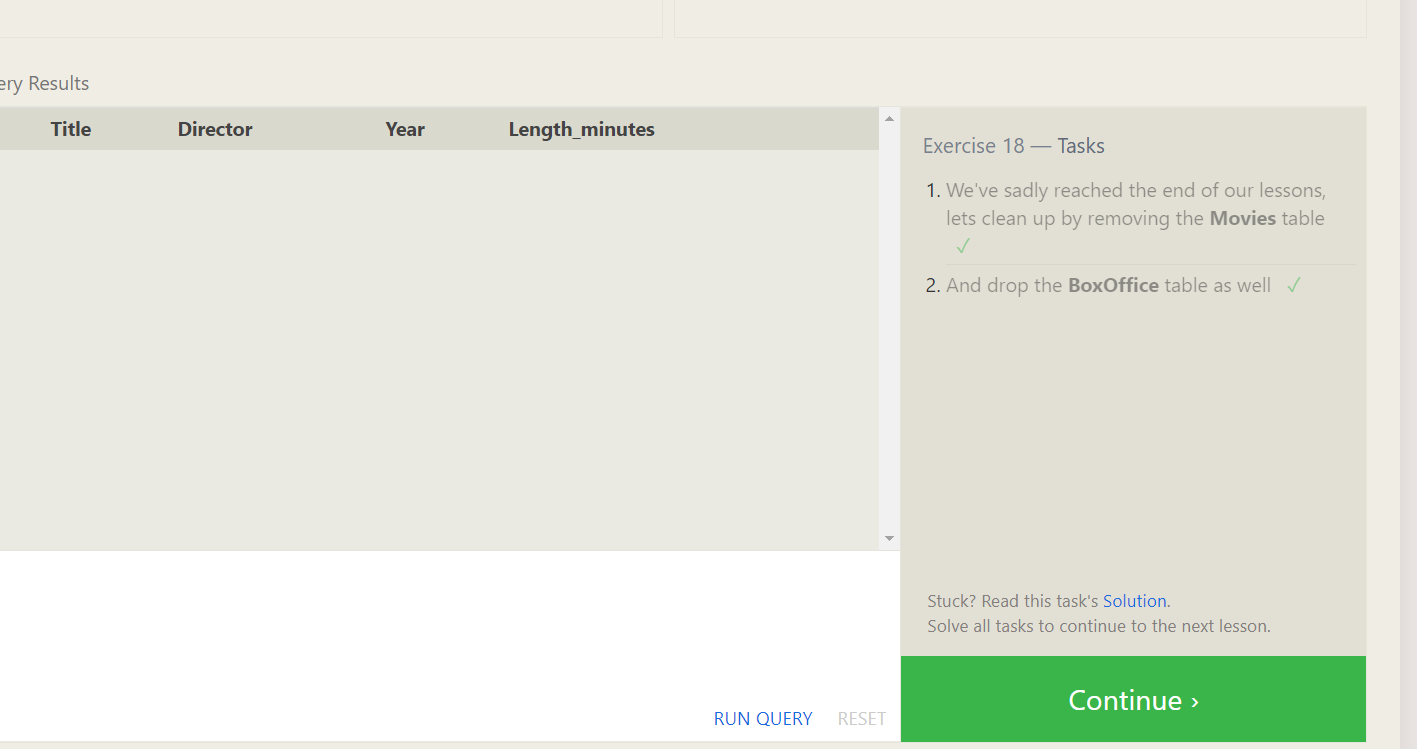
1. ALTER TABLE Movies

ADD COLUMN Aspect\_ratio FLOAT DEFAULT 2.39;

1. ALTER TABLE Movies

ADD COLUMN Language TEXT DEFAULT "English";

**SQL Lesson 18: Dropping tables**

****

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

