Exercise 1

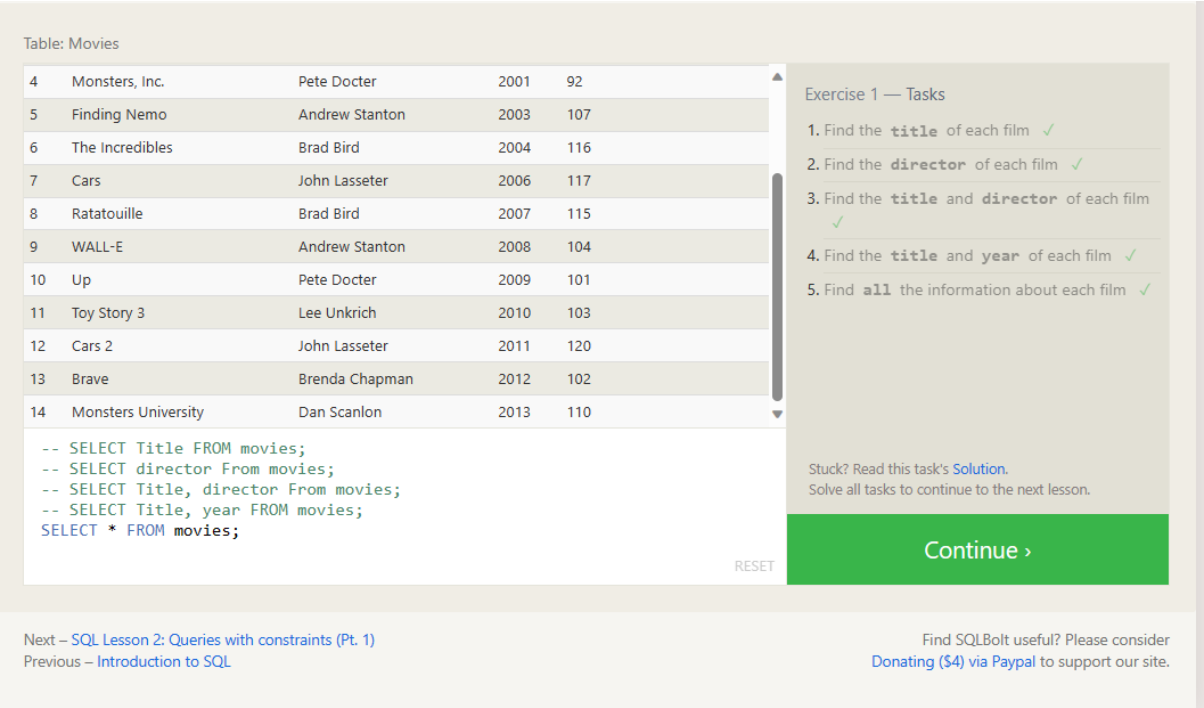


Table: Movies

4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101
11	Toy Story 3	Lee Unkrich	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

```
-- SELECT Title FROM movies;
-- SELECT director From movies;
-- SELECT Title, director From movies;
-- SELECT Title, year FROM movies;
SELECT * FROM movies;
```

RESET

Exercise 1 — Tasks

1. Find the `title` of each film ✓
2. Find the `director` of each film ✓
3. Find the `title` and `director` of each film ✓
4. Find the `title` and `year` of each film ✓
5. Find `all` the information about each film ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

Next – SQL Lesson 2: Queries with constraints (Pt. 1)
Previous – Introduction to SQL

Find SQLBolt useful? Please consider
Donating ($4) via Paypal to support our site.

Query

SELECT Title FROM movies;

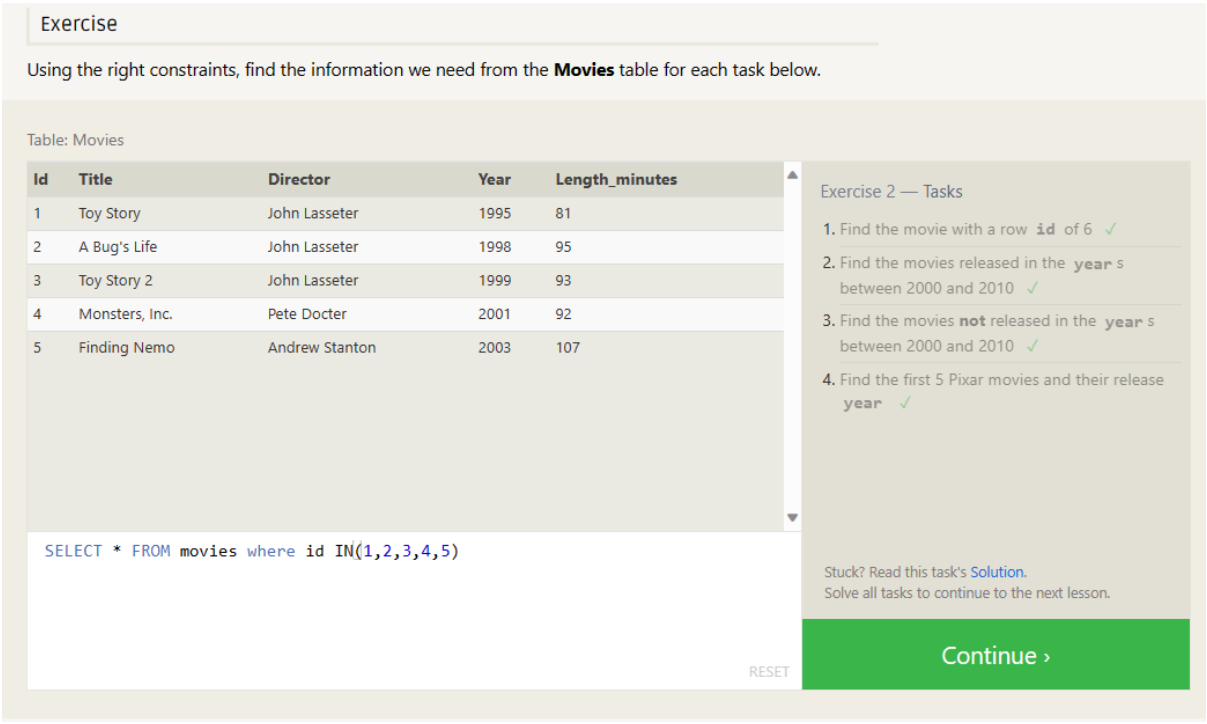
SELECT director From movies;

SELECT Title, director From movies;

SELECT Title, year FROM movies;

SELECT * FROM movies;

Exercise 2



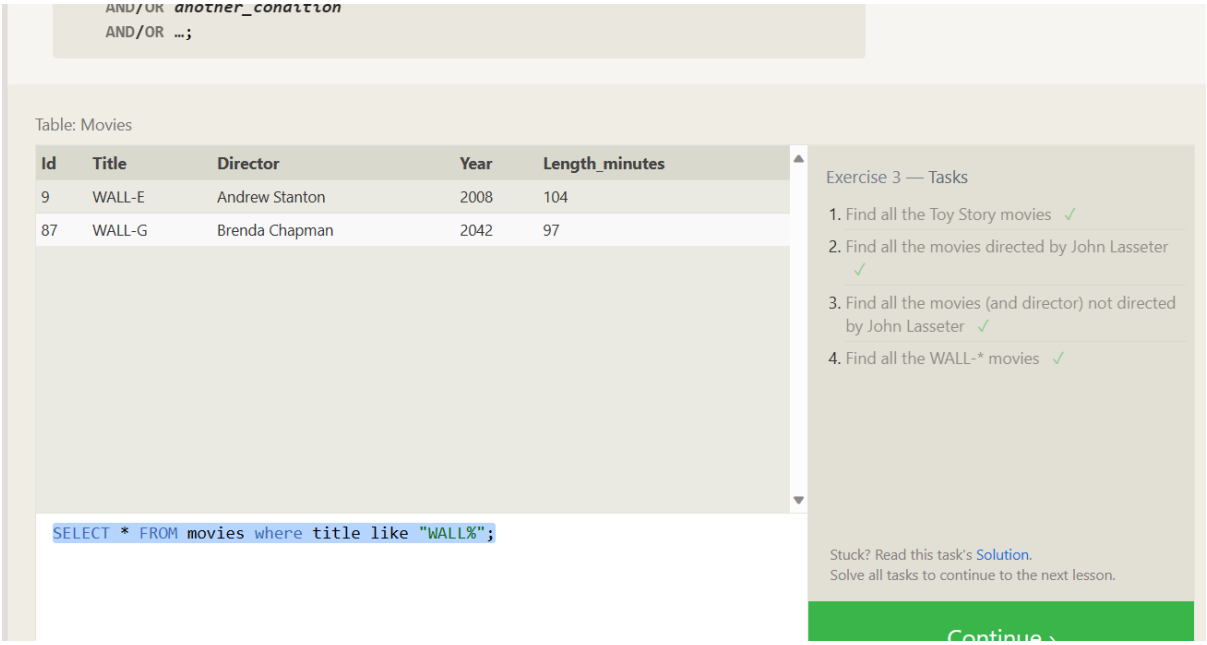
SELECT * FROM movies where id=6;

SELECT Title FROM movies where year BETWEEN 2000 AND 2010;

SELECT Title FROM movies where year NOT BETWEEN 2000 AND 2010;

SELECT * FROM movies where id IN(1,2,3,4,5)

Exercise 3



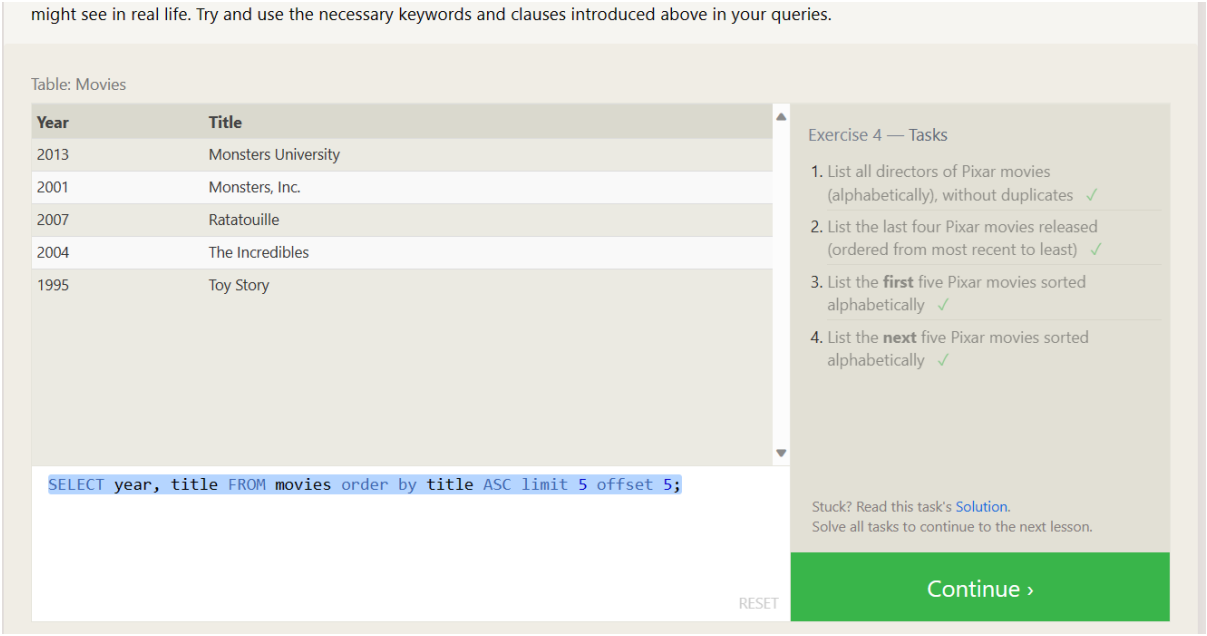
SELECT * FROM movies where title like "Toy%";

SELECT * FROM movies where director="John Lasseter";

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

SELECT * FROM movies where title like "WALL%";

Exercise 4



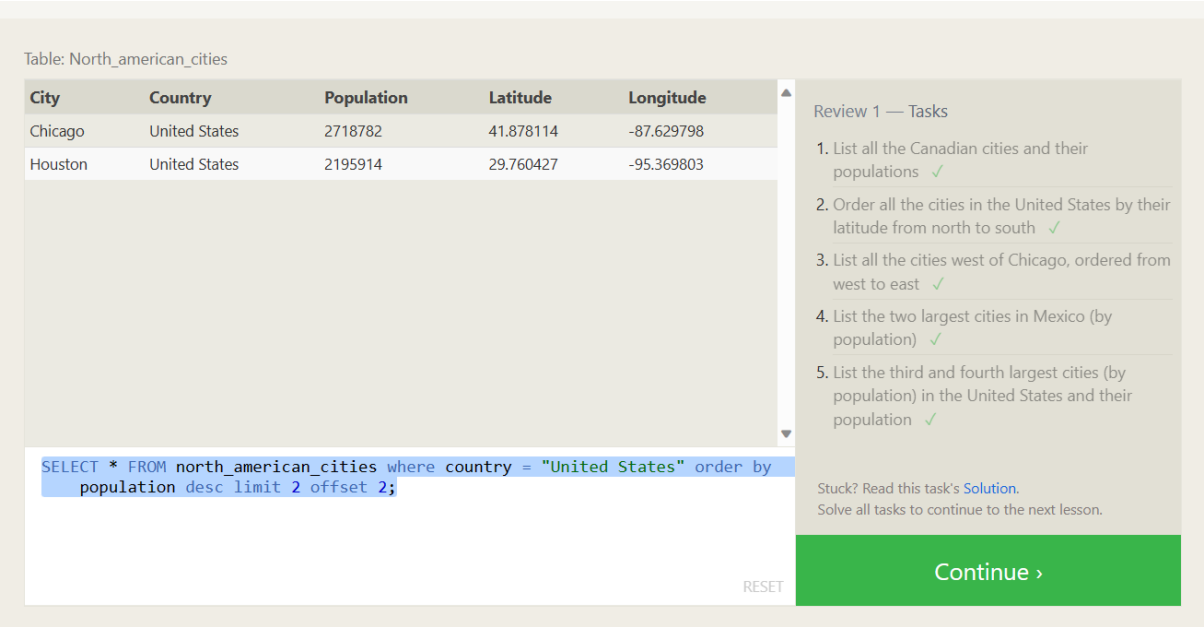
SELECT DISTINCT director FROM movies order by director ASC;

SELECT year, title FROM movies order by year desc limit 4;

SELECT year, title FROM movies order by title ASC limit 5;

SELECT year, title FROM movies order by title ASC limit 5 offset 5;

Exercise 5



SELECT city,country, population FROM north_american_cities where country like "canada";

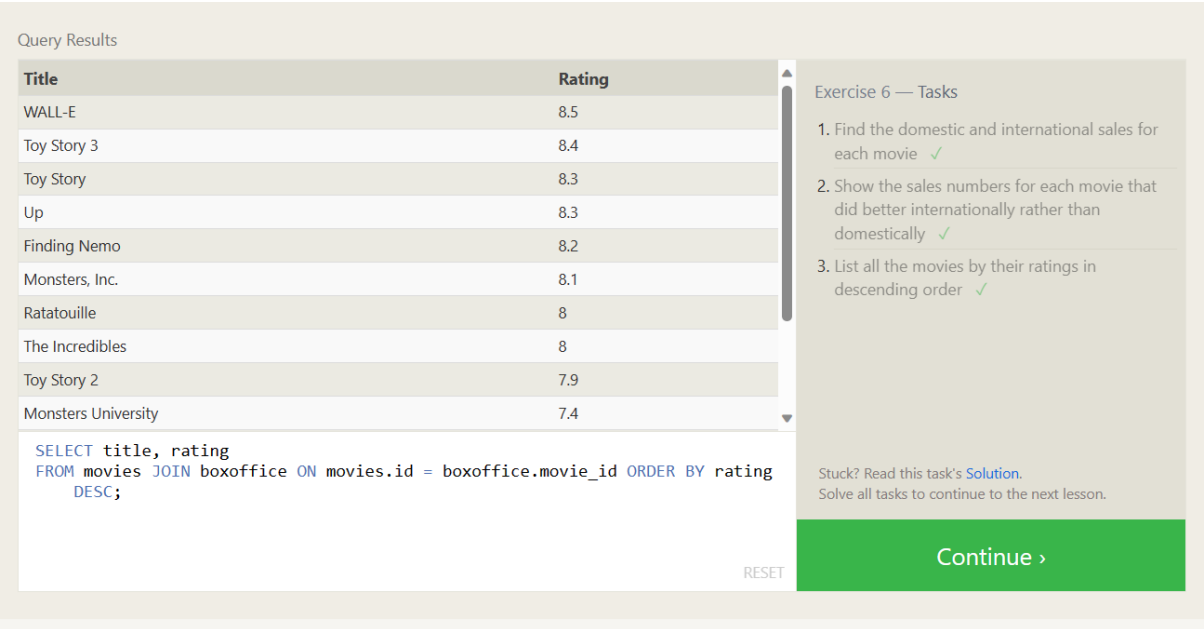
SELECT * FROM north_american_cities where country = "United States" order by latitude DESC;

SELECT * FROM north_american_cities where longitude <-87.629798 order by longitude asc;

SELECT * FROM north_american_cities where country = "Mexico" order by population desc limit 2;

SELECT * FROM north_american_cities where country = "United States" order by population desc limit 2 offset 2;

Exercise 6

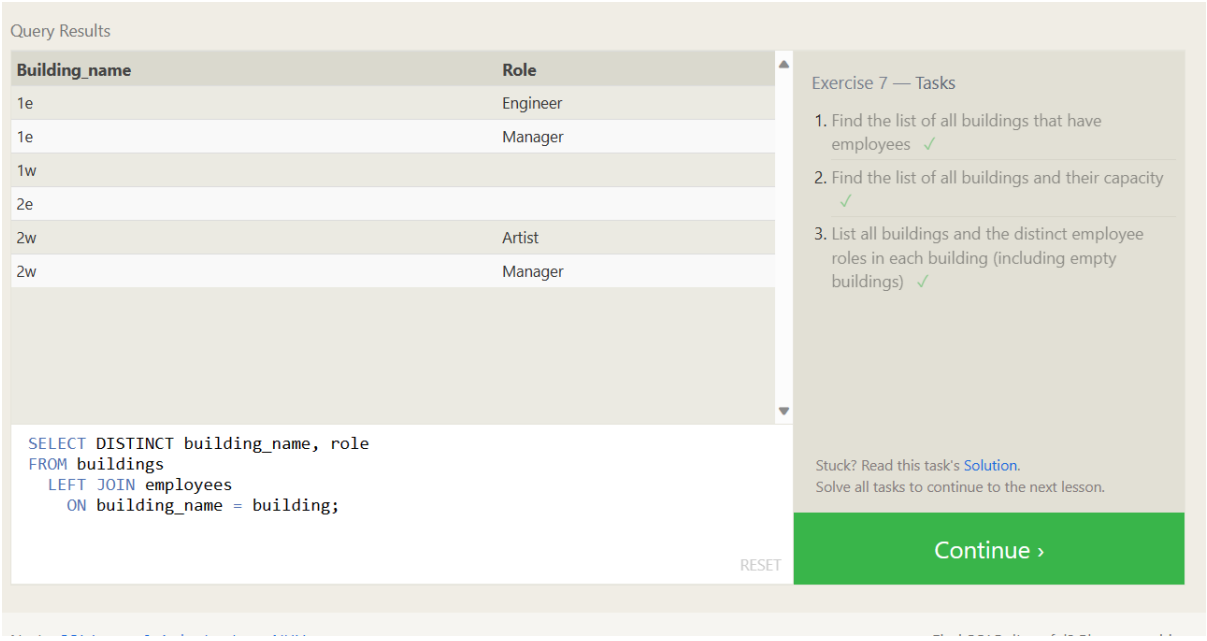


```
SELECT title, domestic_sales, international_sales
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id;
```

```
SELECT title, domestic_sales, international_sales
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id
WHERE international_sales > domestic_sales;
```

```
SELECT title, rating
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id
ORDER BY rating DESC;
```

Exercise 7




```
SELECT DISTINCT building FROM employees;
```

```
SELECT * FROM buildings;
```

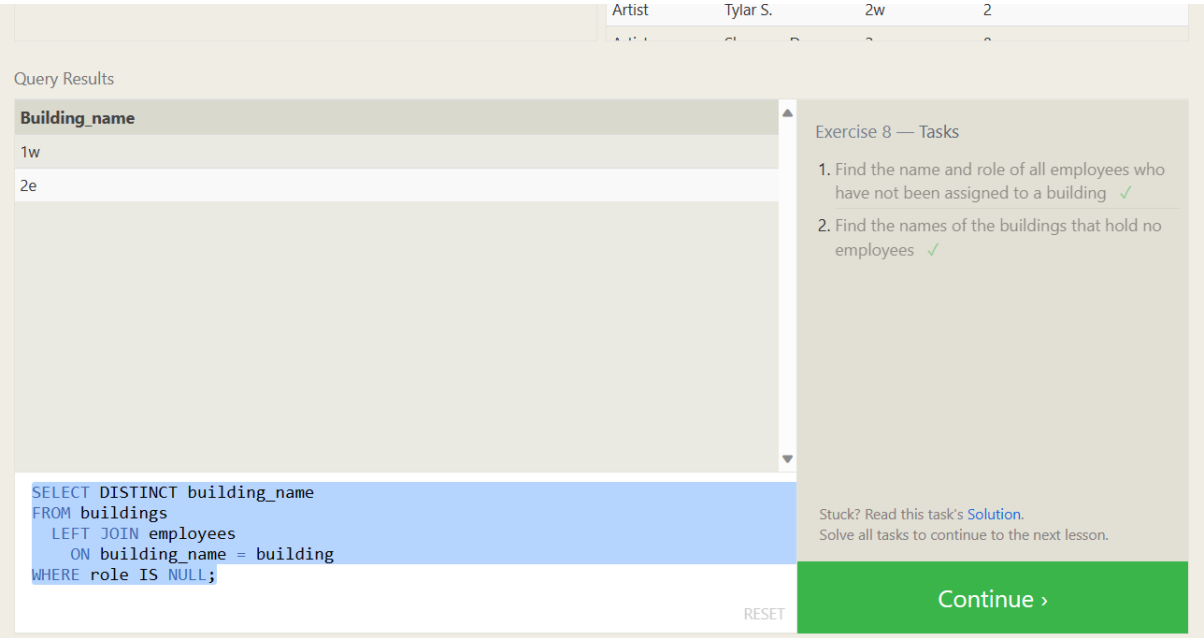
```
SELECT DISTINCT building_name, role
```

FROM buildings

LEFT JOIN employees

ON building_name = building;

Exercise 8



SELECT name, role FROM employees

WHERE building IS NULL;

SELECT DISTINCT building_name

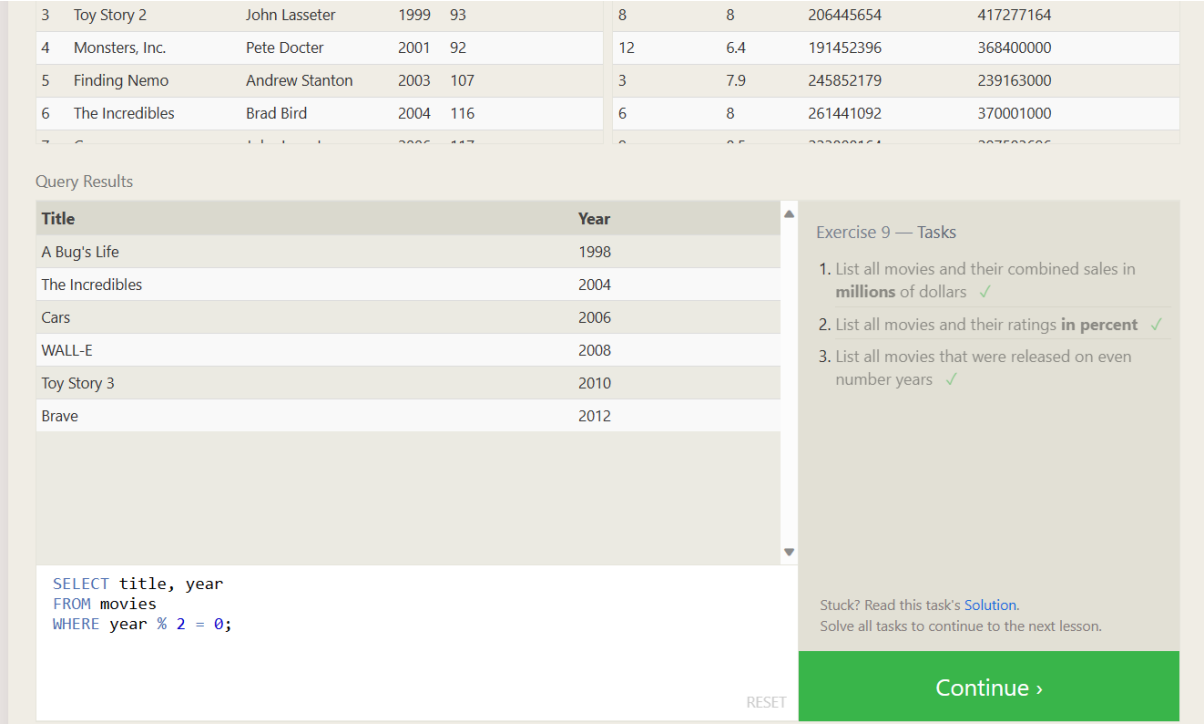
FROM buildings

LEFT JOIN employees

ON building_name = building

WHERE role IS NULL;

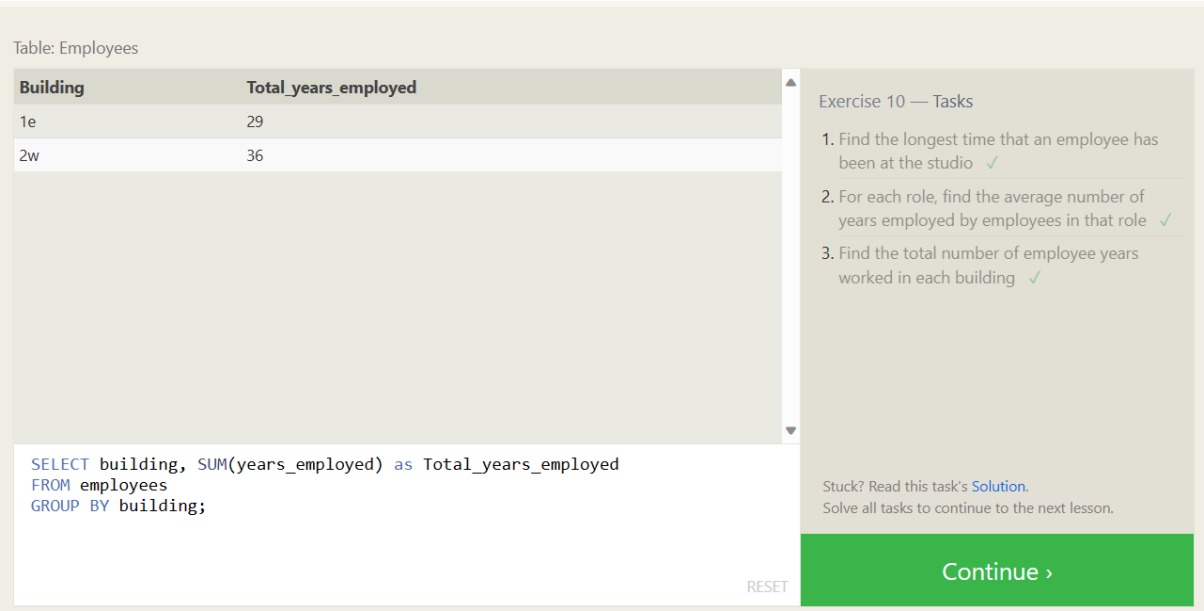
Exercise 9



3	Toy Story 2	John Lasseter	1999	93	8	8	206445654	417277164
4	Monsters, Inc.	Pete Docter	2001	92	12	6.4	191452396	368400000
5	Finding Nemo	Andrew Stanton	2003	107	3	7.9	245852179	239163000
6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000


```
SELECT title, (domestic_sales + international_sales) / 1000000 AS gross_sales_millions
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id;
SELECT title, rating * 10 AS rating_percent
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id;
SELECT title, year
FROM movies
WHERE year % 2 = 0;
```

Exercise 10



SELECT MAX(years_employed) as Max_years_employed

FROM employees;

SELECT role, AVG(years_employed) as Average_years_employed

FROM employees

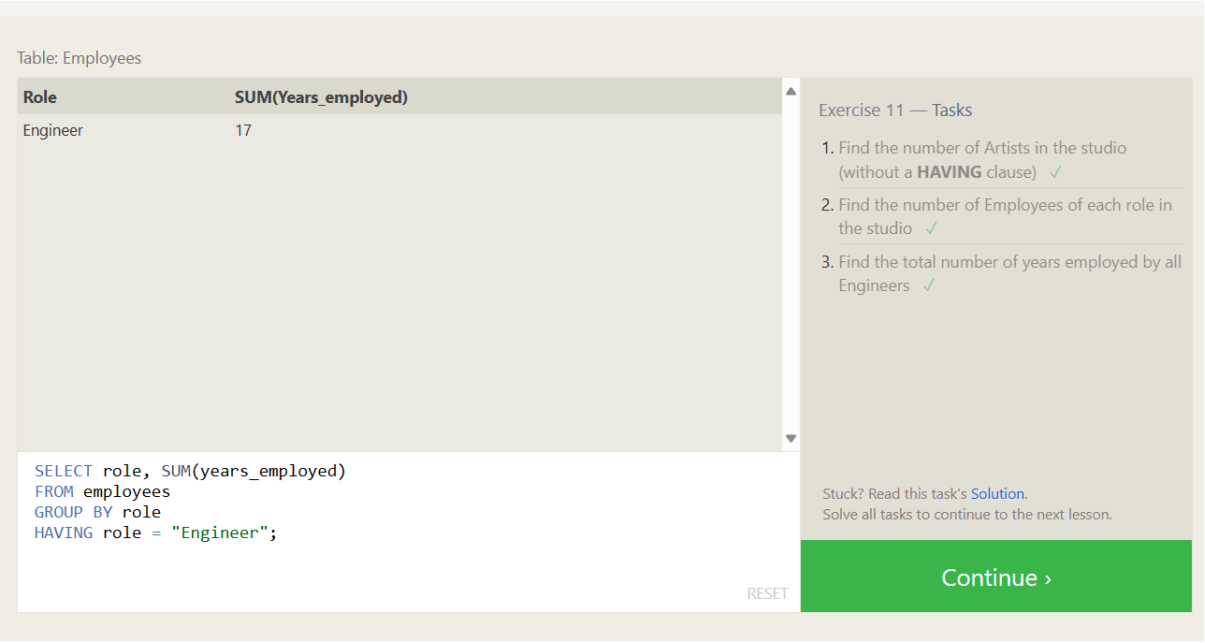
GROUP BY role;

SELECT building, SUM(years_employed) as Total_years_employed

FROM employees

GROUP BY building;

Exercise 11



SELECT role, COUNT(*) as Number_of_artists

FROM employees

WHERE role = "Artist";

SELECT role, COUNT(*)

FROM employees

GROUP BY role;

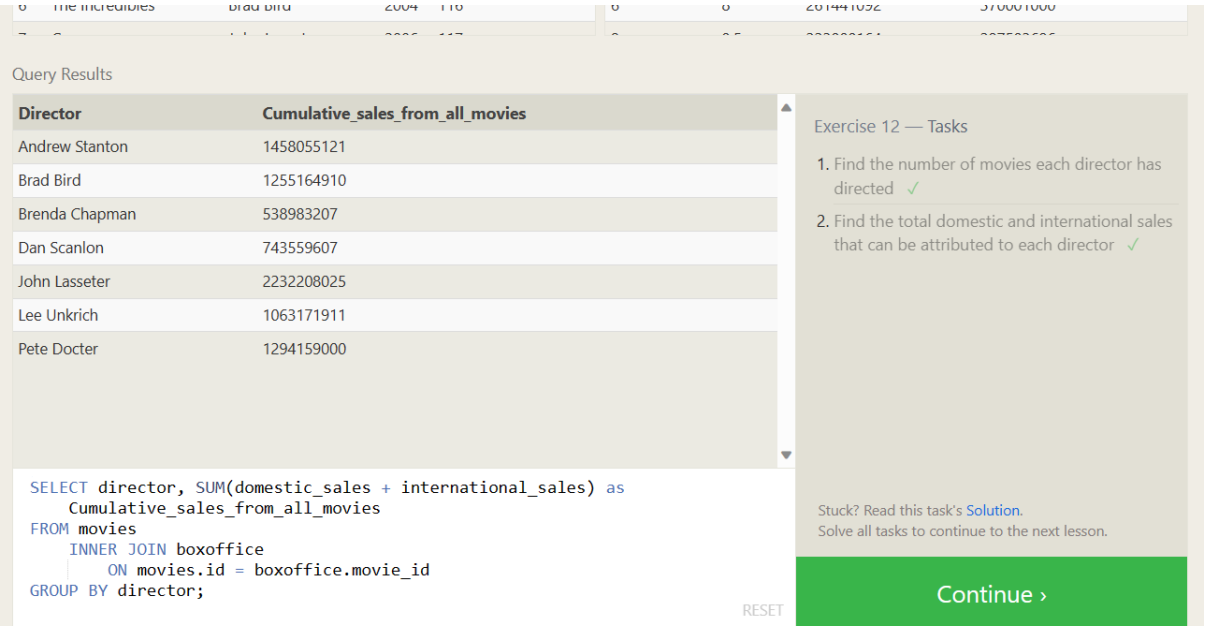
SELECT role, SUM(years_employed)

FROM employees

GROUP BY role

HAVING role = "Engineer";

Exercise 12



SELECT director, COUNT(id) as Num_movies_directed

FROM movies

GROUP BY director;

SELECT director, SUM(domestic_sales + international_sales) as Cumulative_sales_from_all_movies

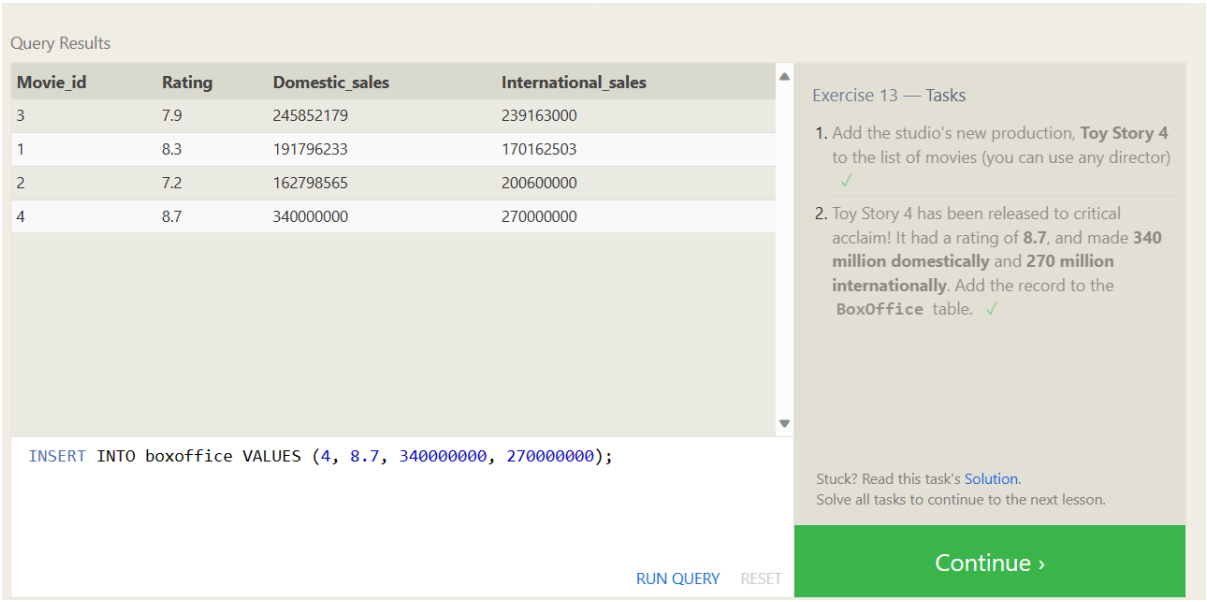
FROM movies

INNER JOIN boxoffice

ON movies.id = boxoffice.movie_id

GROUP BY director;

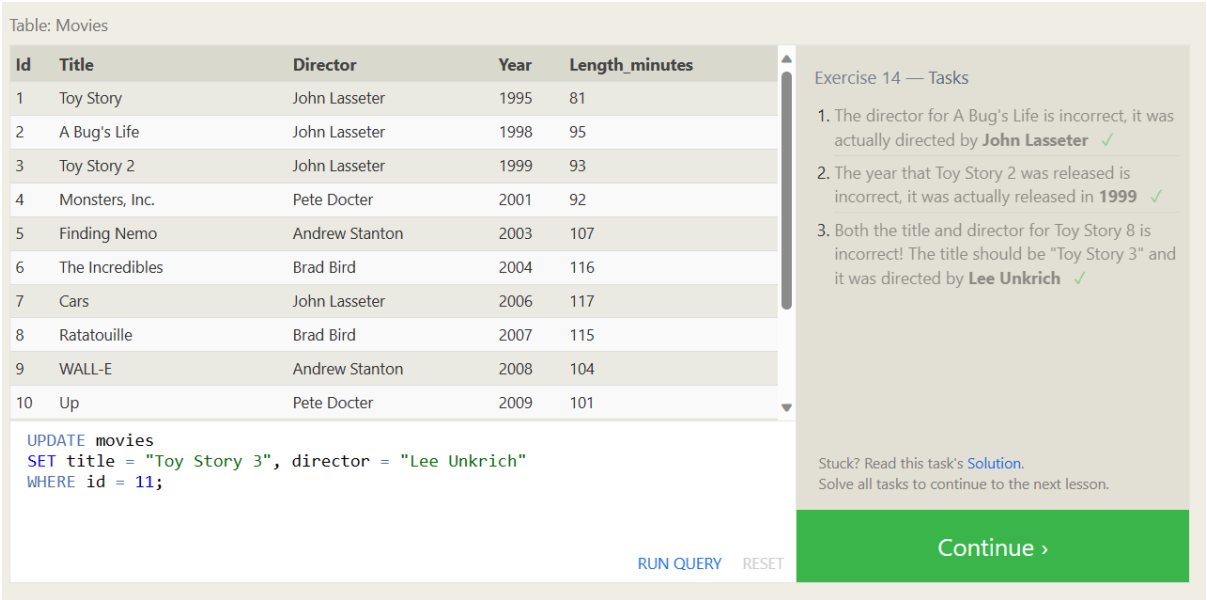
Exercise 13



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

INSERT INTO boxoffice VALUES (4, 8.7, 340000000, 270000000);

Exercise 14



UPDATE movies

SET director = "John Lasseter"

WHERE id = 2;

UPDATE movies

SET year = 1999

WHERE id = 3;

UPDATE movies

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

WHERE id = 11;

Exercise 15

Table: Movies

Id	Title	Director	Year	Length_minutes
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
10	Up	Pete Docter	2009	101
11	Toy Story 3	Lee Unkrich	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

DELETE FROM movies

where year < 2005;

DELETE FROM movies

where director = "Andrew Stanton";

Exercise 16

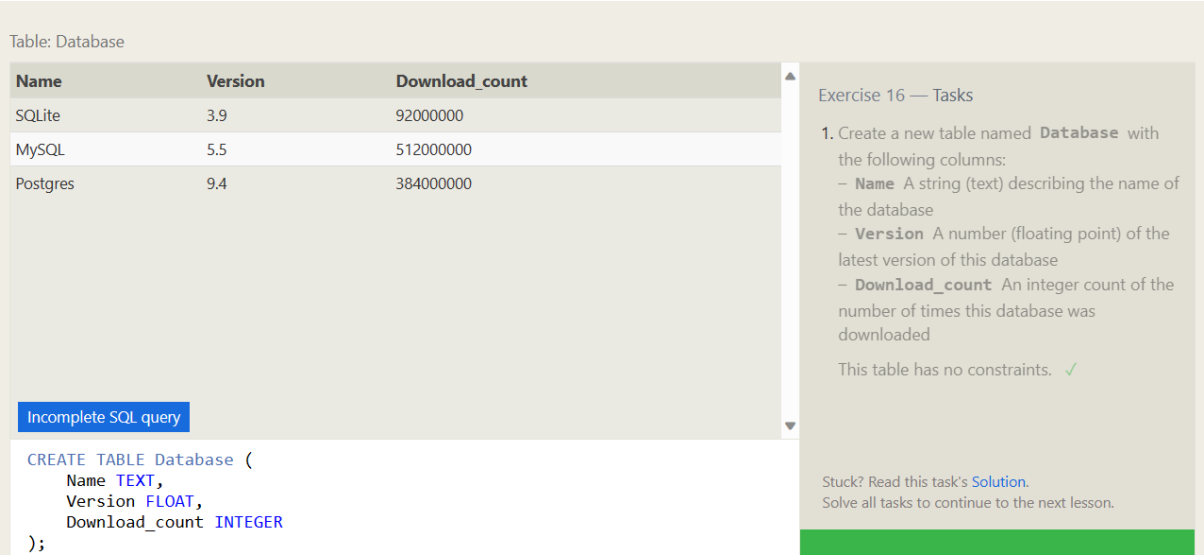


Table: Database

Name	Version	Download_count
SQLite	3.9	92000000
MySQL	5.5	512000000
Postgres	9.4	384000000

CREATE TABLE Database (

Name TEXT,

Version FLOAT,

Download_count INTEGER

);

Exercise 17

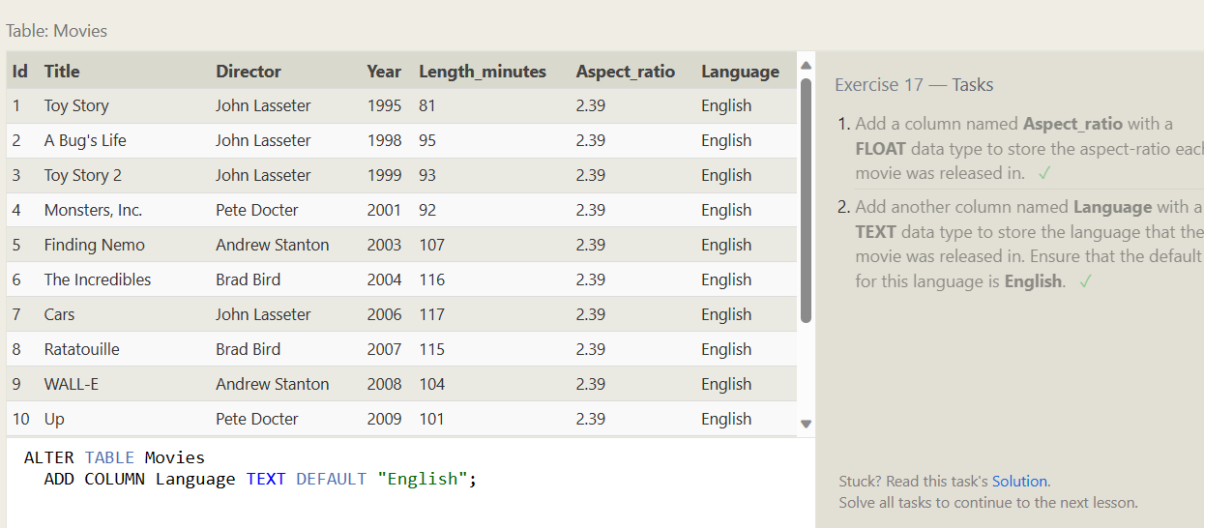


Table: Movies

Id	Title	Director	Year	Length_minutes	Aspect_ratio	Language
1	Toy Story	John Lasseter	1995	81	2.39	English
2	A Bug's Life	John Lasseter	1998	95	2.39	English
3	Toy Story 2	John Lasseter	1999	93	2.39	English
4	Monsters, Inc.	Pete Docter	2001	92	2.39	English
5	Finding Nemo	Andrew Stanton	2003	107	2.39	English
6	The Incredibles	Brad Bird	2004	116	2.39	English
7	Cars	John Lasseter	2006	117	2.39	English
8	Ratatouille	Brad Bird	2007	115	2.39	English
9	WALL-E	Andrew Stanton	2008	104	2.39	English
10	Up	Pete Docter	2009	101	2.39	English

```
ALTER TABLE Movies
  ADD COLUMN Language TEXT DEFAULT "English";
```

Exercise 17 — Tasks

1. Add a column named **Aspect_ratio** with a **FLOAT** data type to store the aspect-ratio each movie was released in. ✓
2. Add another column named **Language** with a **TEXT** data type to store the language that the movie was released in. Ensure that the default for this language is **English**. ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

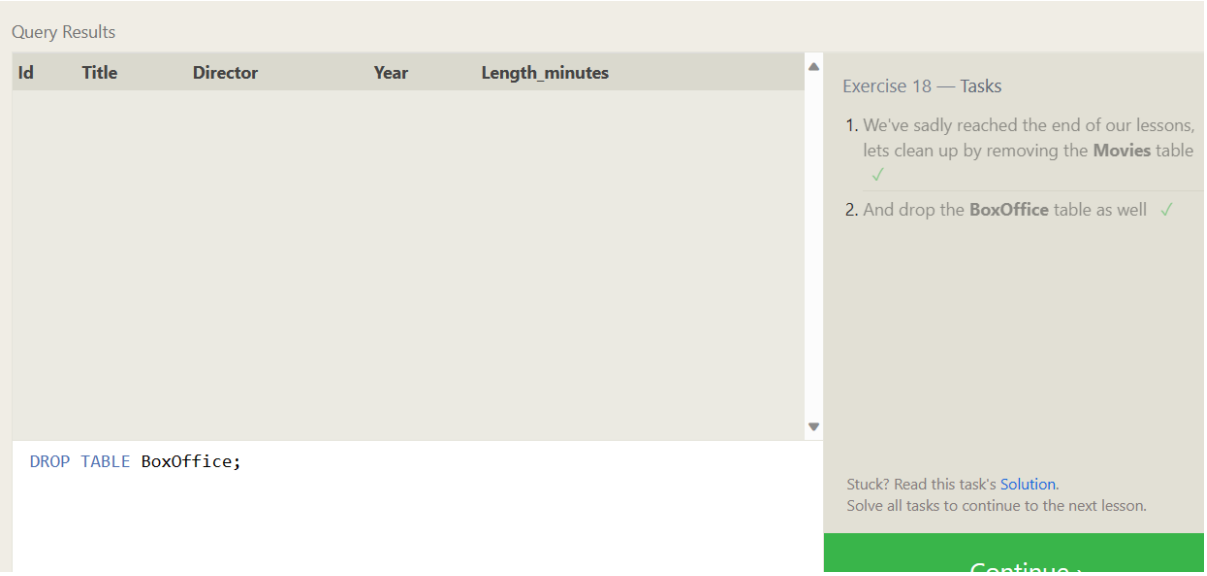
ALTER TABLE Movies

ADD COLUMN Aspect_ratio FLOAT DEFAULT 2.39;

ALTER TABLE Movies

ADD COLUMN Language TEXT DEFAULT "English";

Exercise 18



Query Results

Id	Title	Director	Year	Length_minutes

```
DROP TABLE BoxOffice;
```

Exercise 18 — Tasks

1. We've sadly reached the end of our lessons, lets clean up by removing the **Movies** table ✓
2. And drop the **BoxOffice** table as well ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue

DROP TABLE Movies;

DROP TABLE BoxOffice;

Certification