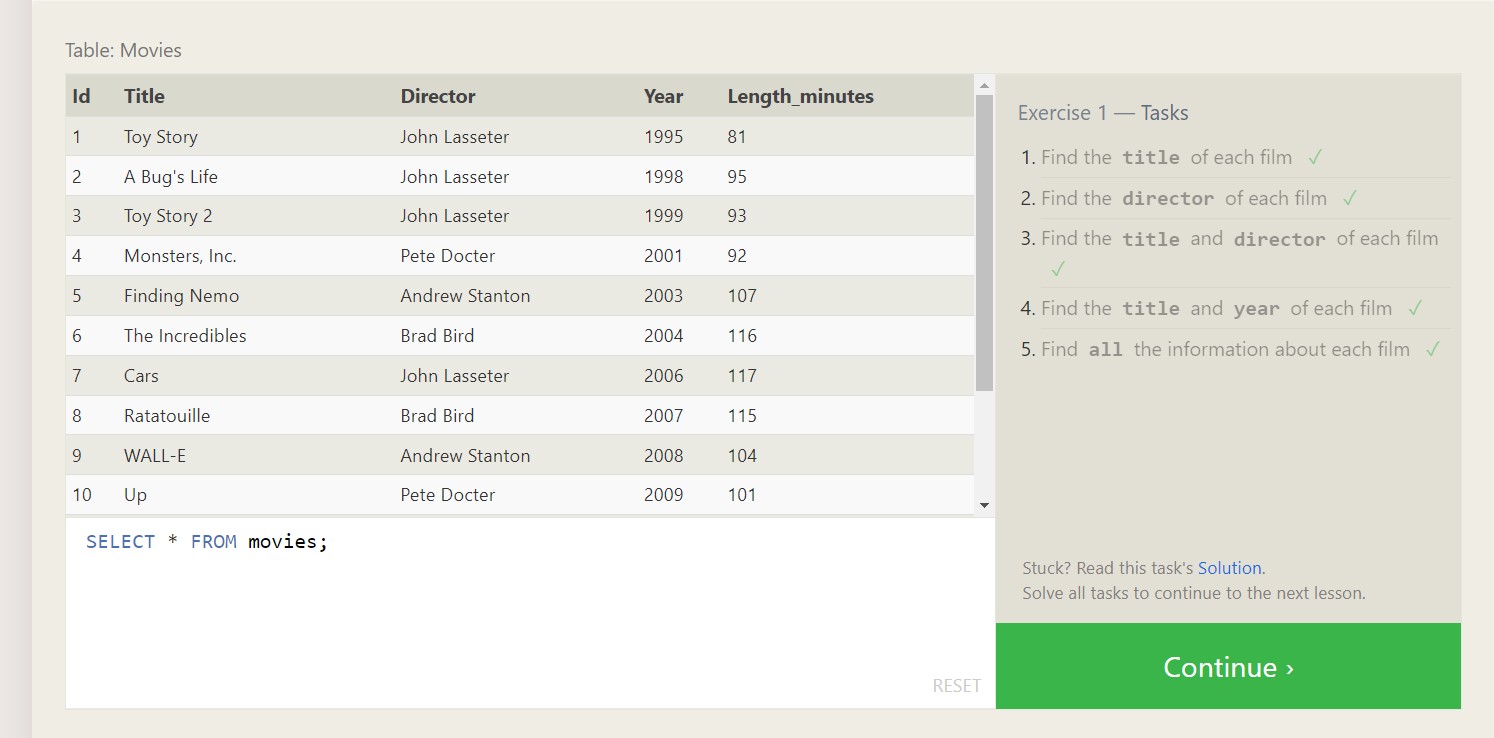
**SQL Lesson 1: SELECT queries**



1.SELECT title FROM movies;

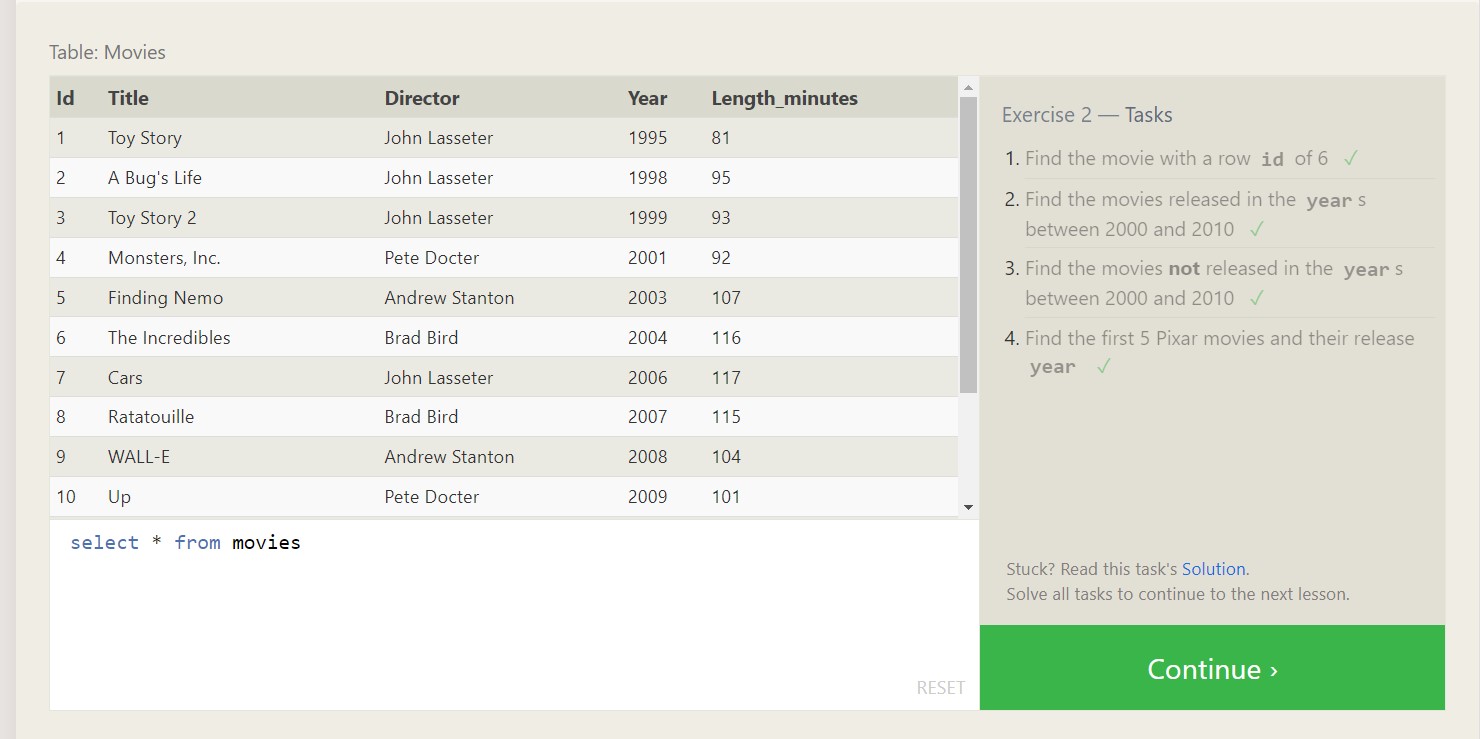
2.SELECT director FROM movies;

3.SELECT title, director FROM movies;

4.SELECT title, year FROM movies;

5. SELECT \* FROM movies;

**SQL Lesson 2: Queries with constraints**



1.SELECT id, title FROM movies

Where id=6;

2.SELECT title, year FROM movies

Where year between 2000 and 2010;

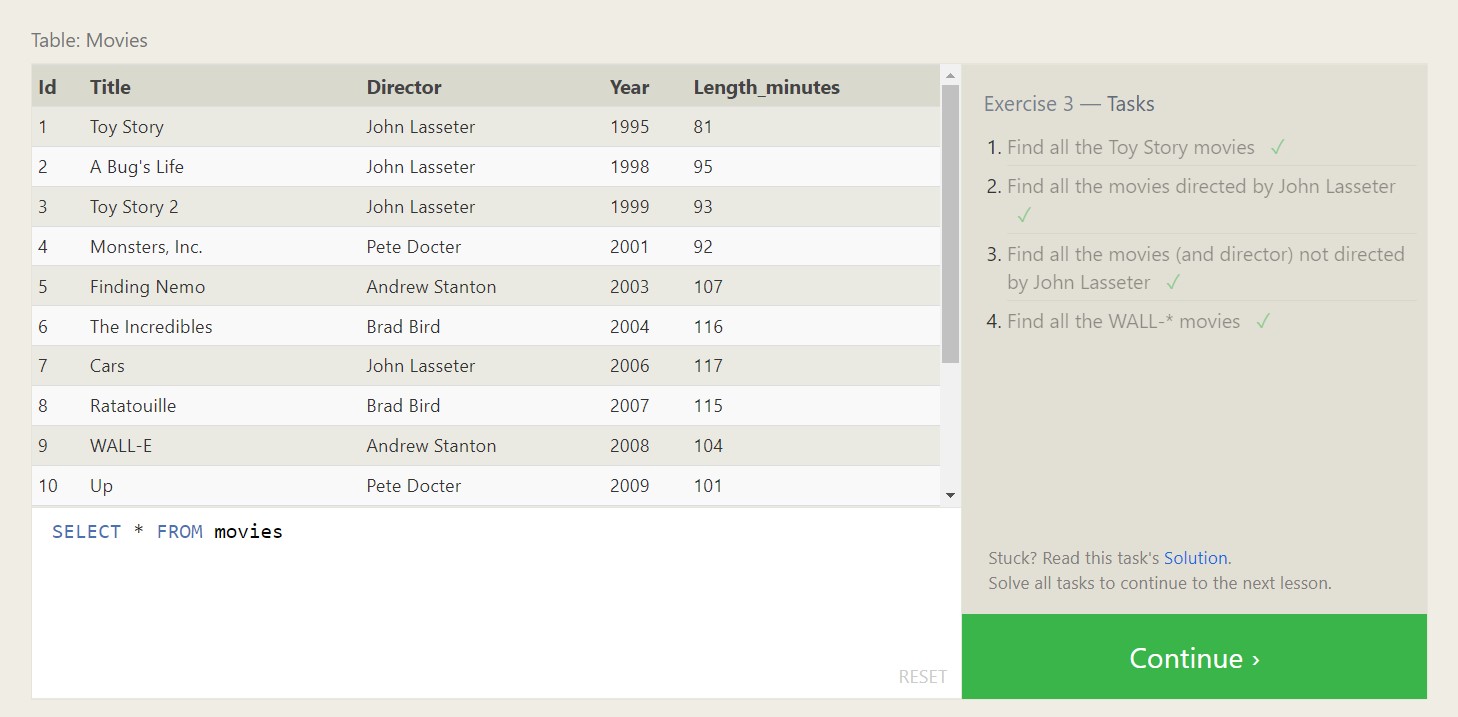
3.select title, year from movies

Where year not between 2000 and 2010;

4. select title, year from movies

Where year <=2003;

**SQL Lesson 3: Queries with constraints**

****

1.select title, director from movies

Where title like “toy story%”;

2. select title, director from movies

where director like "john lasseter";

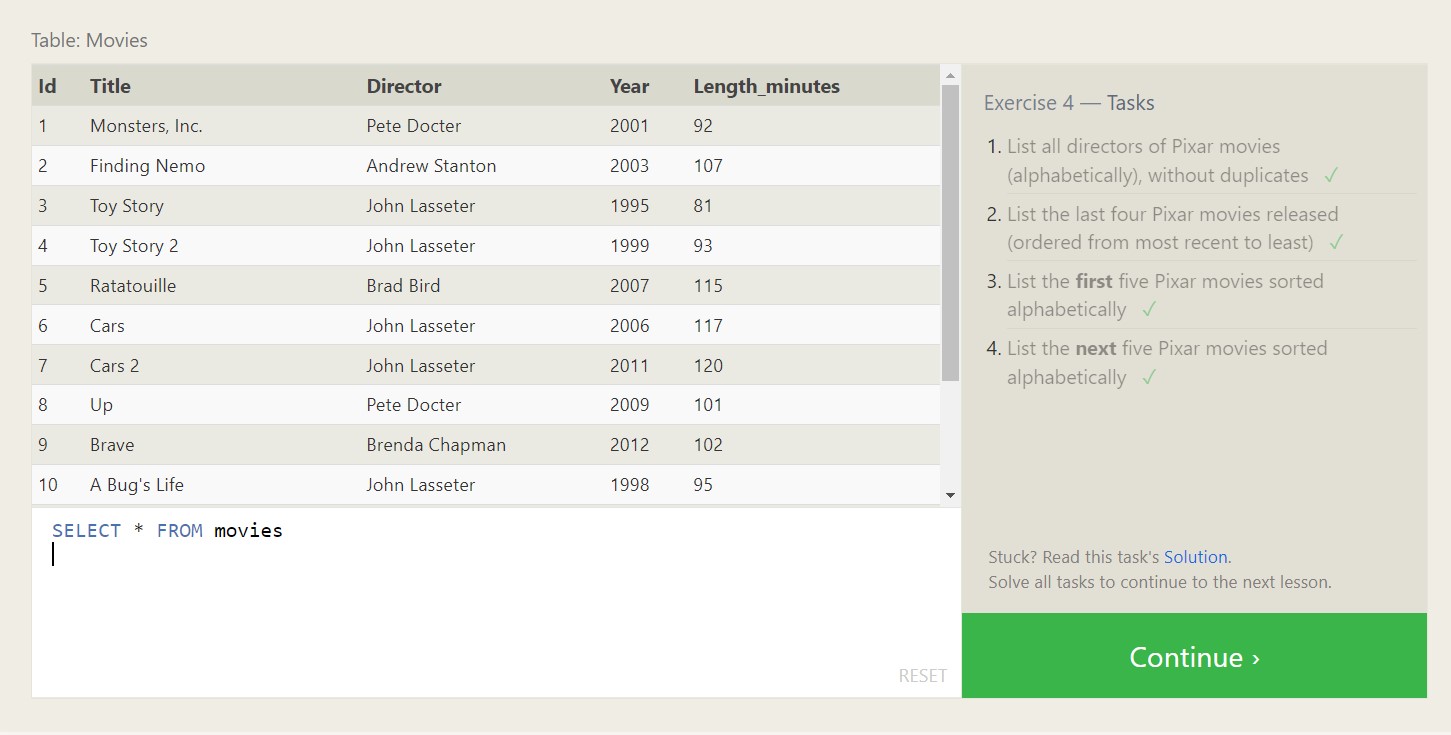
3. select title, director from movies

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, year from movies

Order by year desc

Limit 4;

3. select title from movies

order by title asc

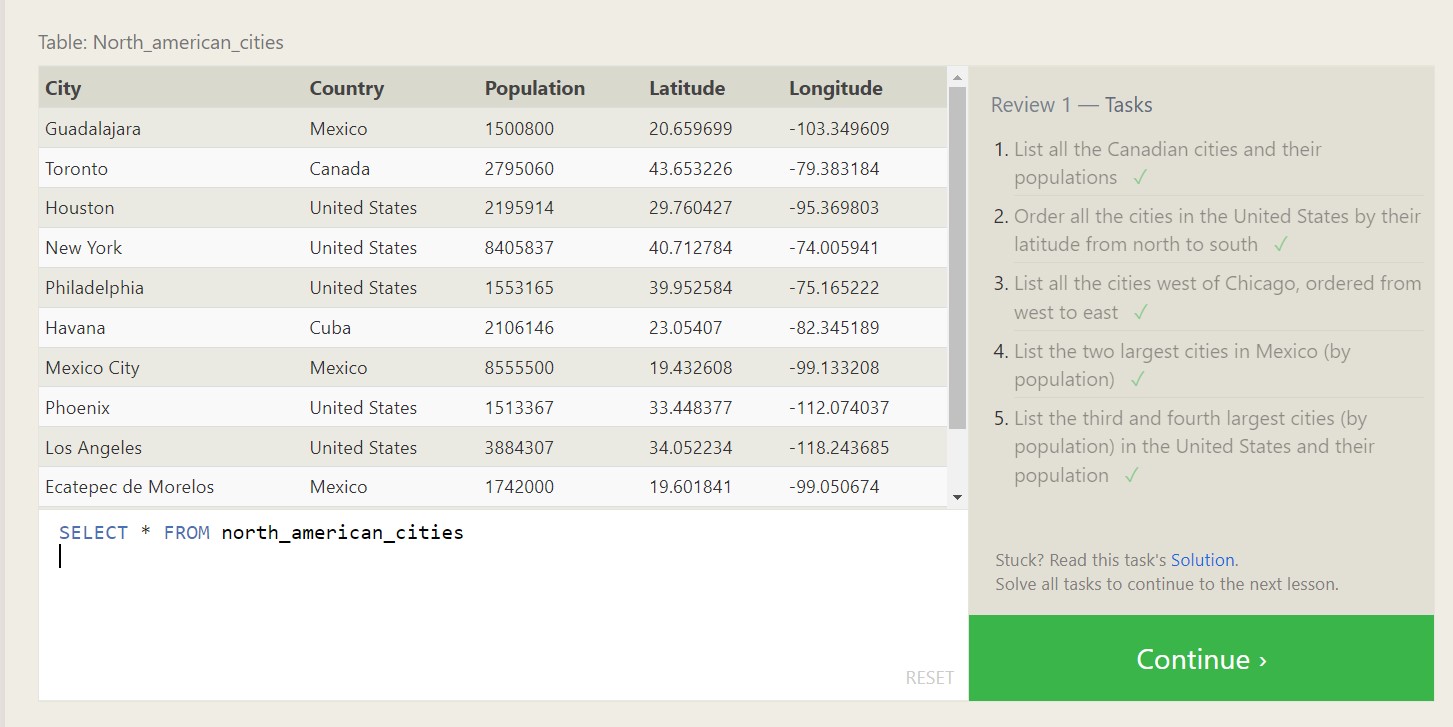
limit 5;

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, latitude from north\_american\_cities

where country=”united states”

order by latitude desc;

3. select city, longitude from north\_american\_cities

where longitude<-87.629798

order by longitude asc;

4.select city, population from north\_american\_cities

where country like “Mexico”

order by population desc

limit 2;

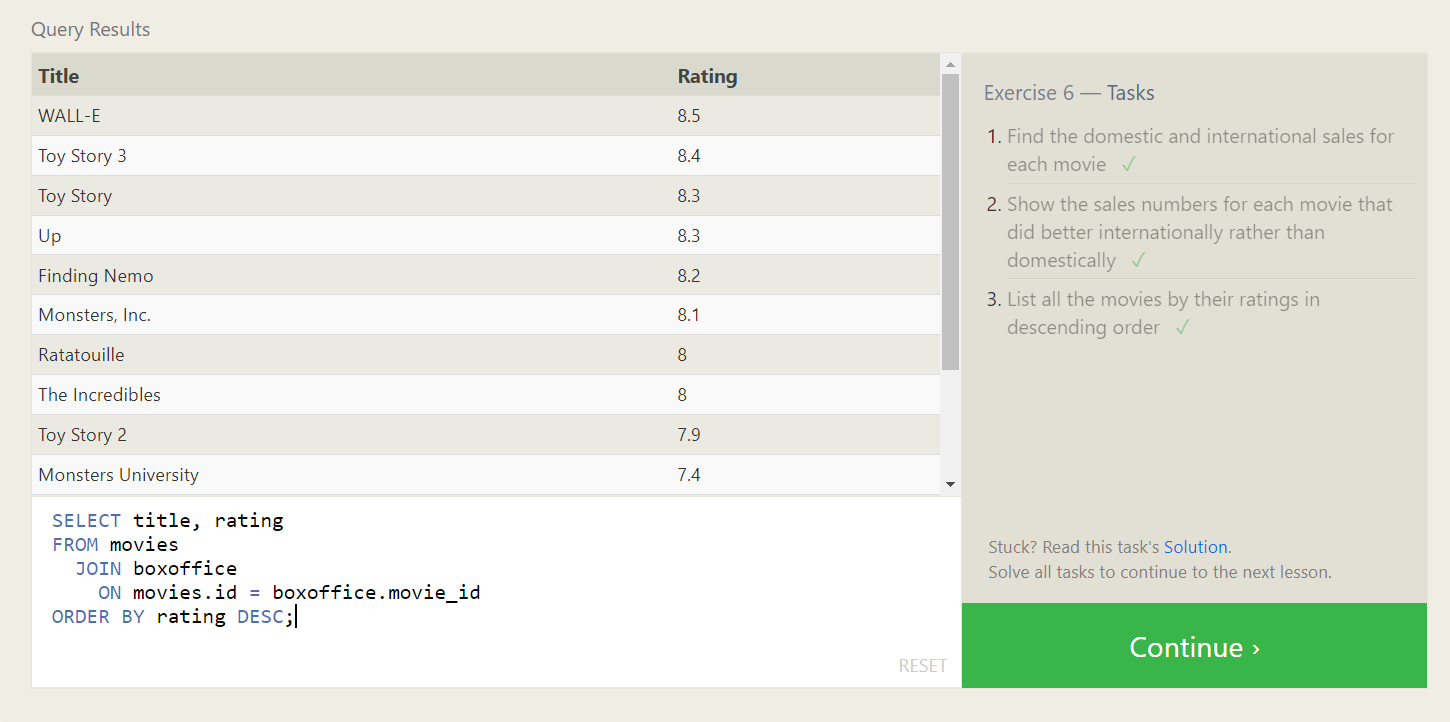
5. select city, population from north\_american\_cities

where country like “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

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 title, rating

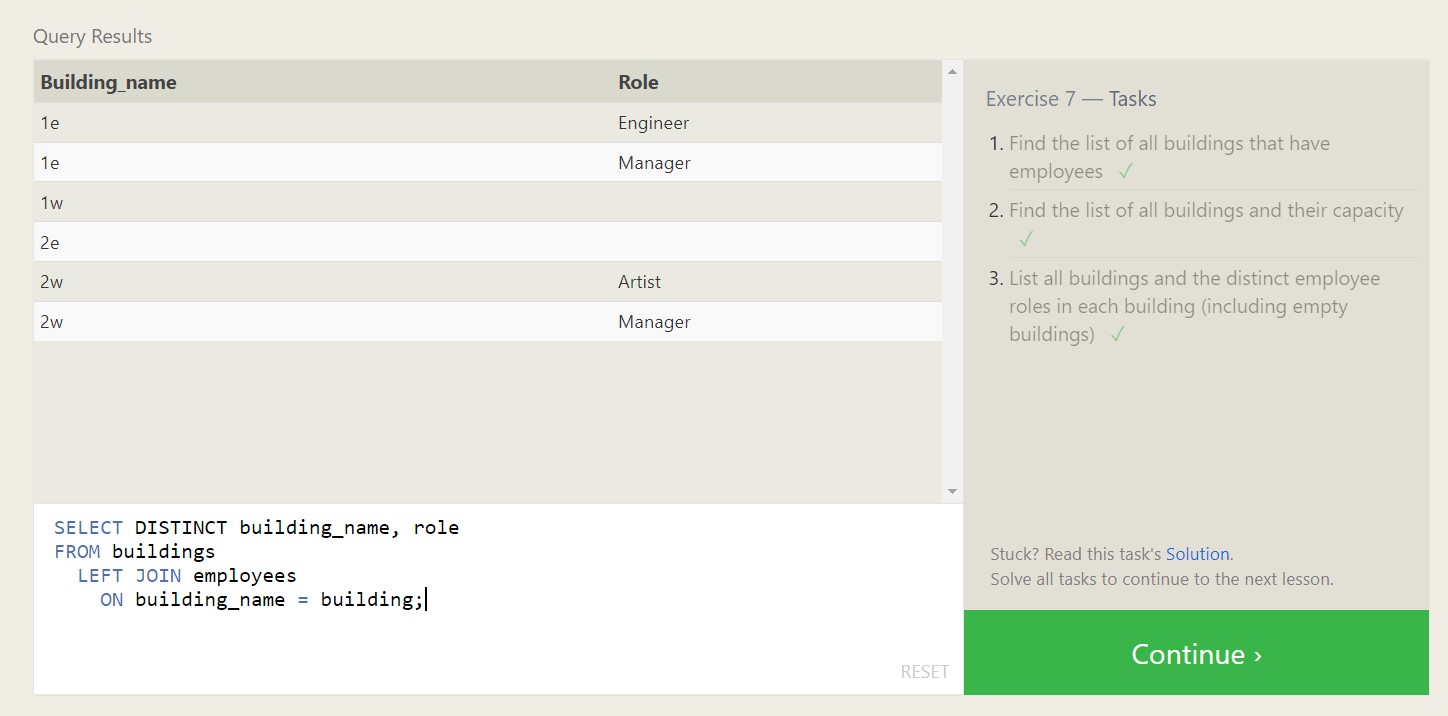
from movies

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;

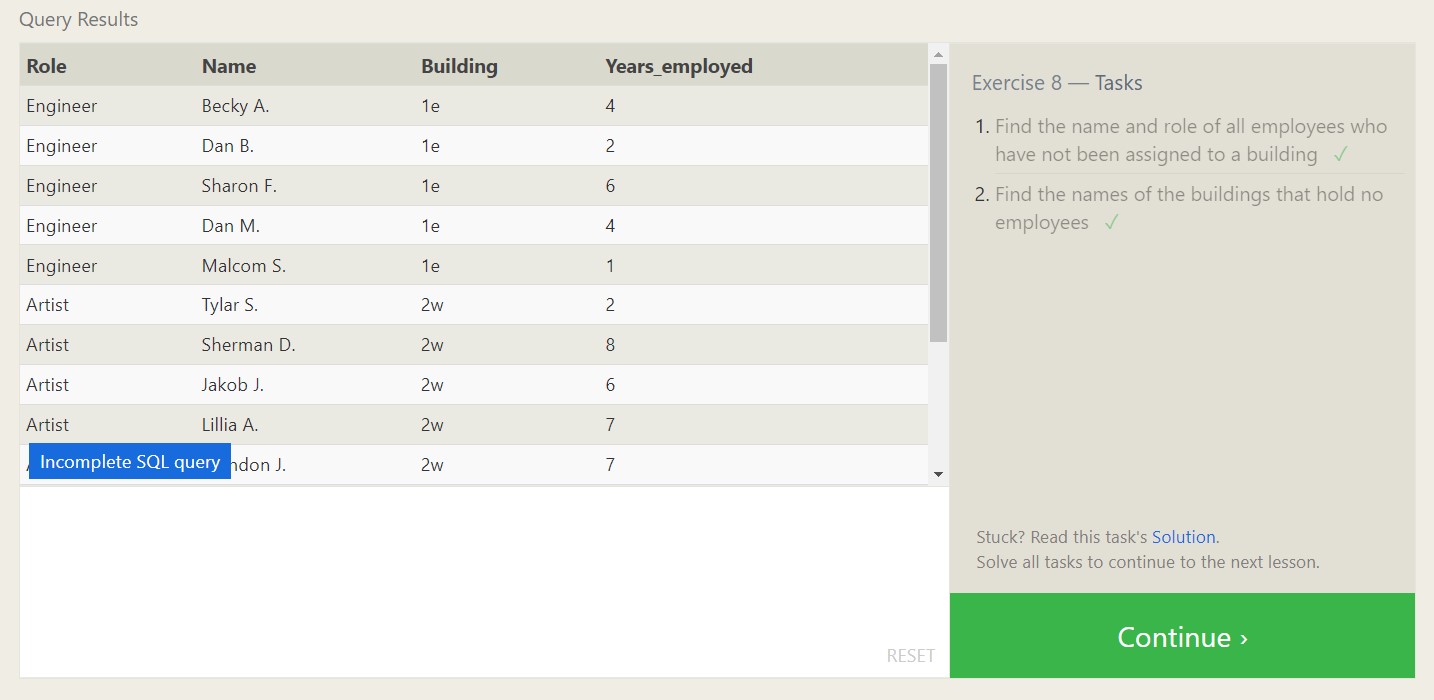
3.select distinct building\_name, role

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

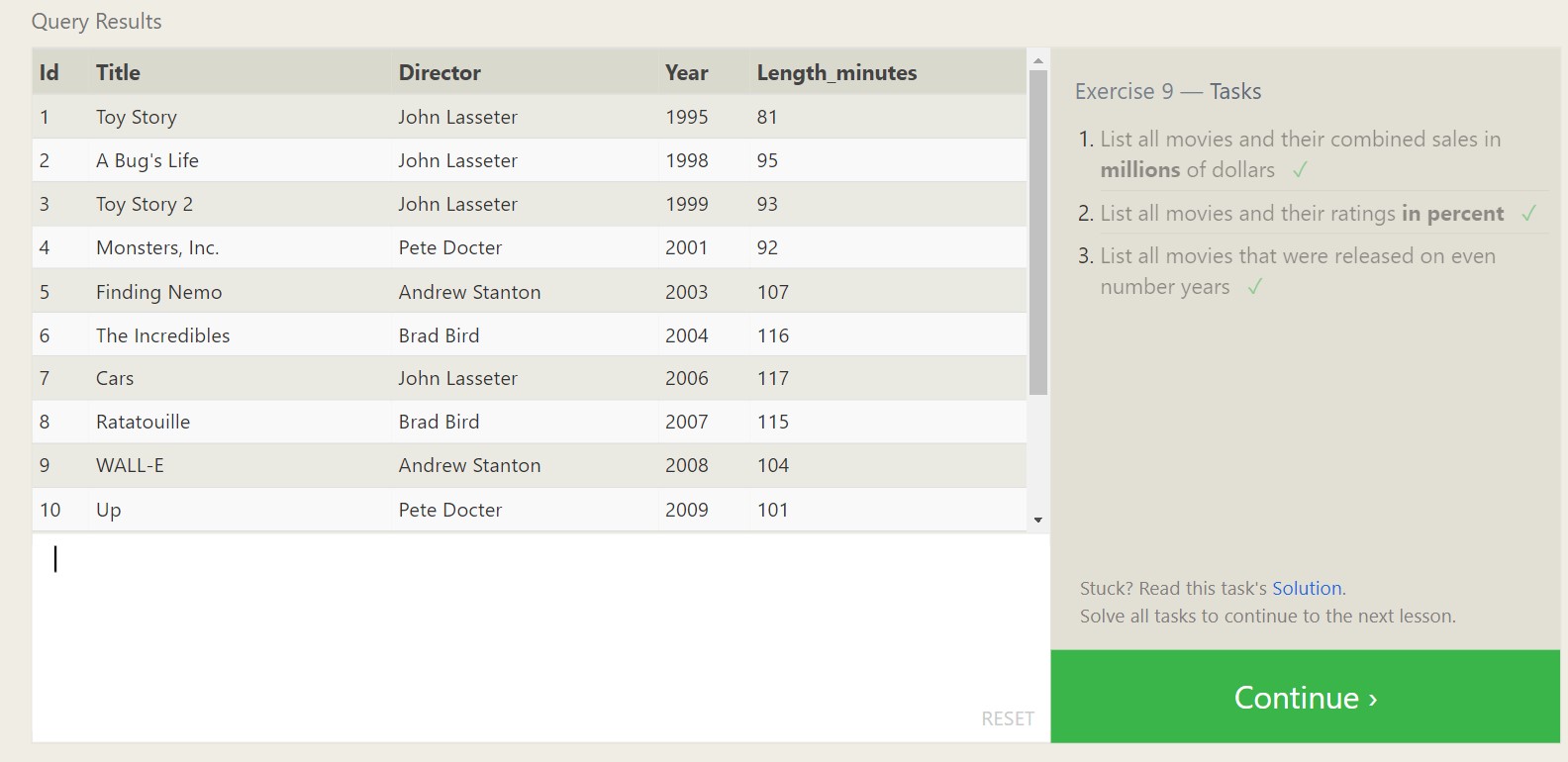
From buldings

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

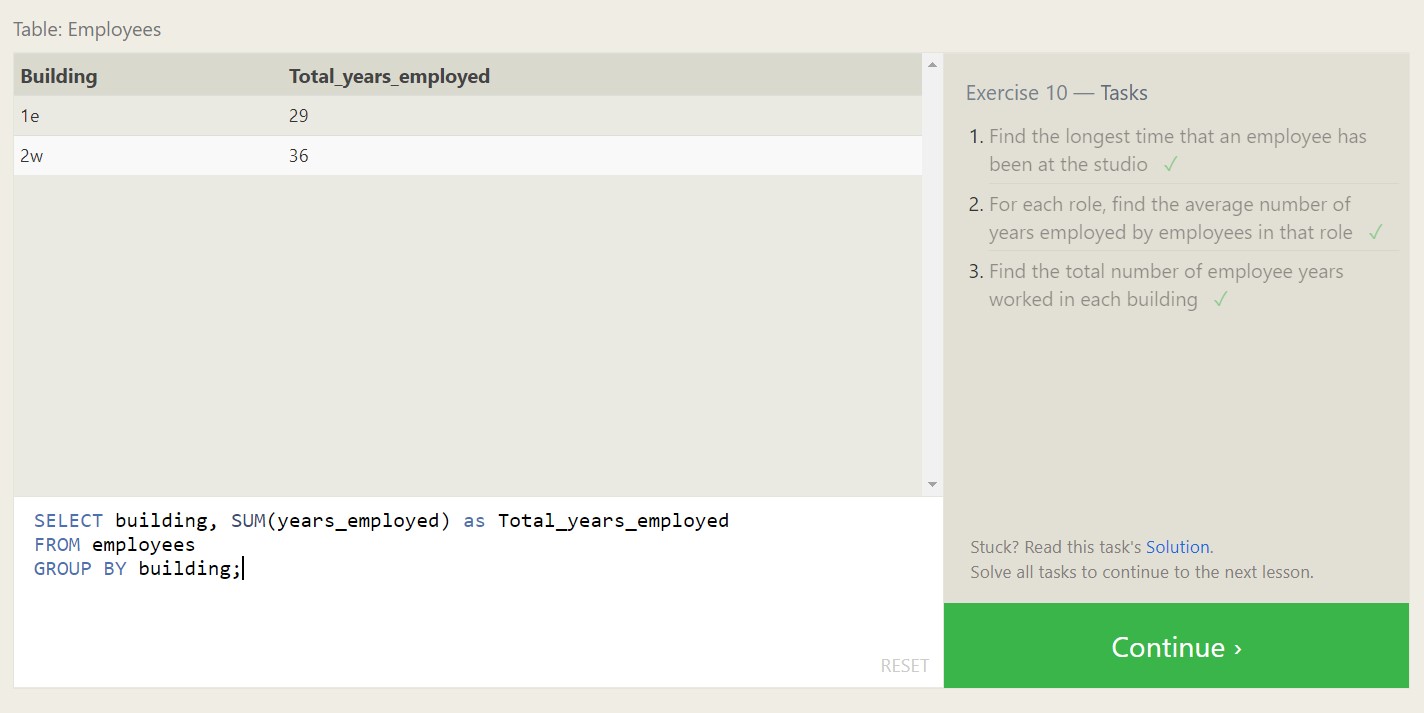
On movies.id = boxoffice.movie\_id;

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

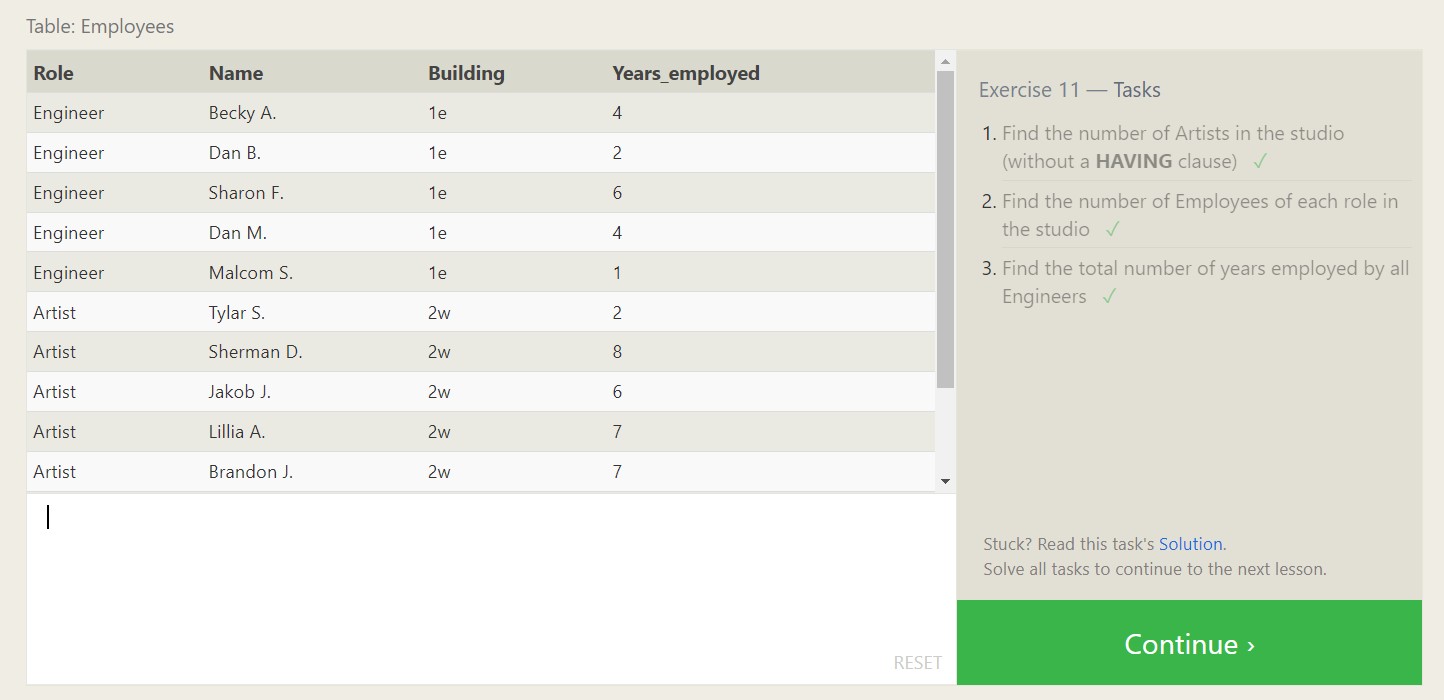
Group by role;

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

From employees

Group by building;

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



1. select role, count(\*) as Number\_of\_artists

From employees

Where role = "Artist";

2.select role, count(\*)

From employees

Group by role;

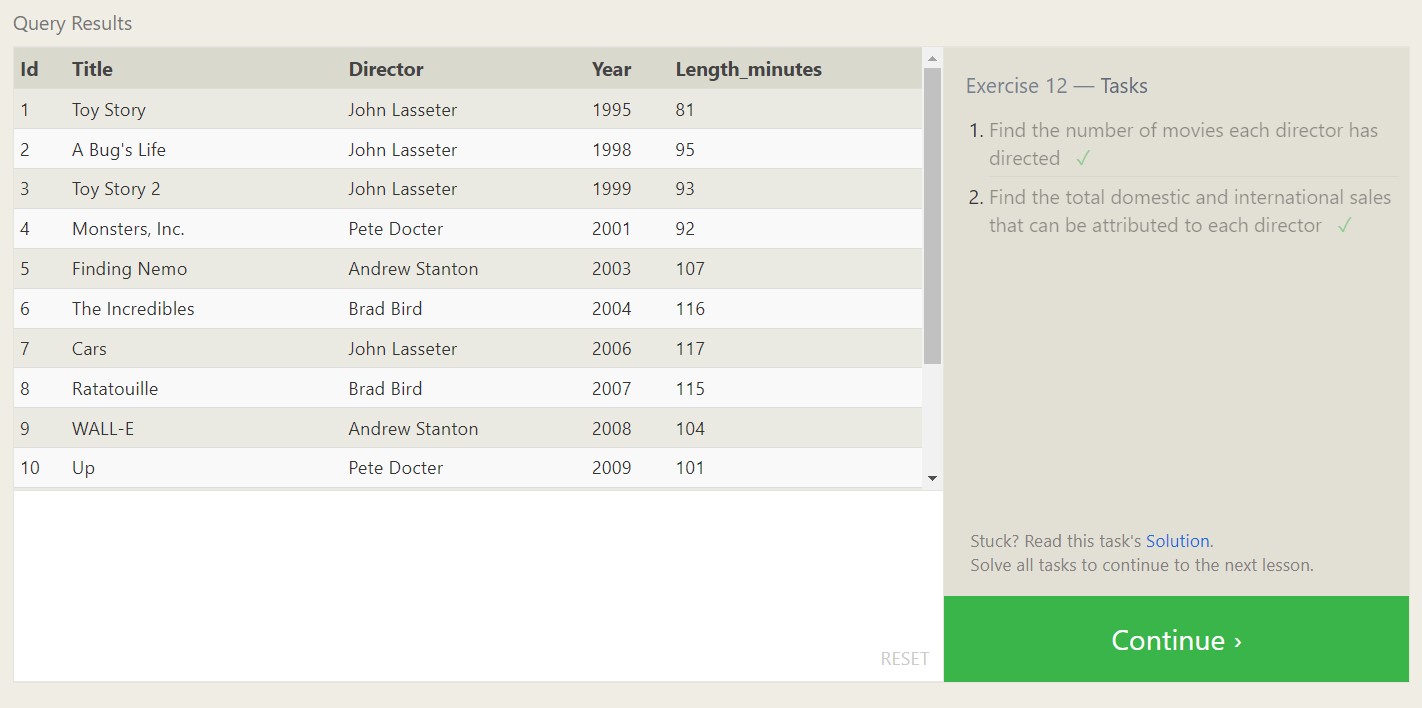
3.select role, SUM(years\_employed)

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

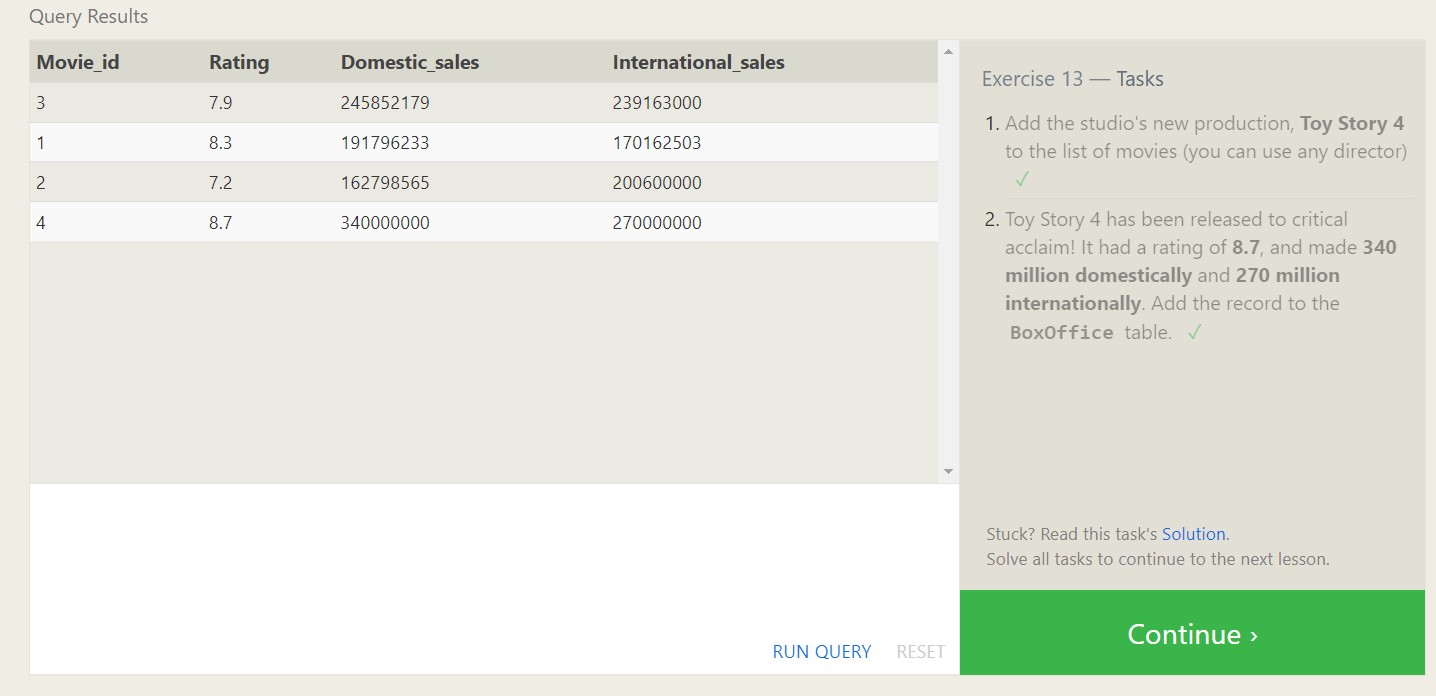
From movies

Inner join boxoffice

On movies.id = boxoffice.movie\_id

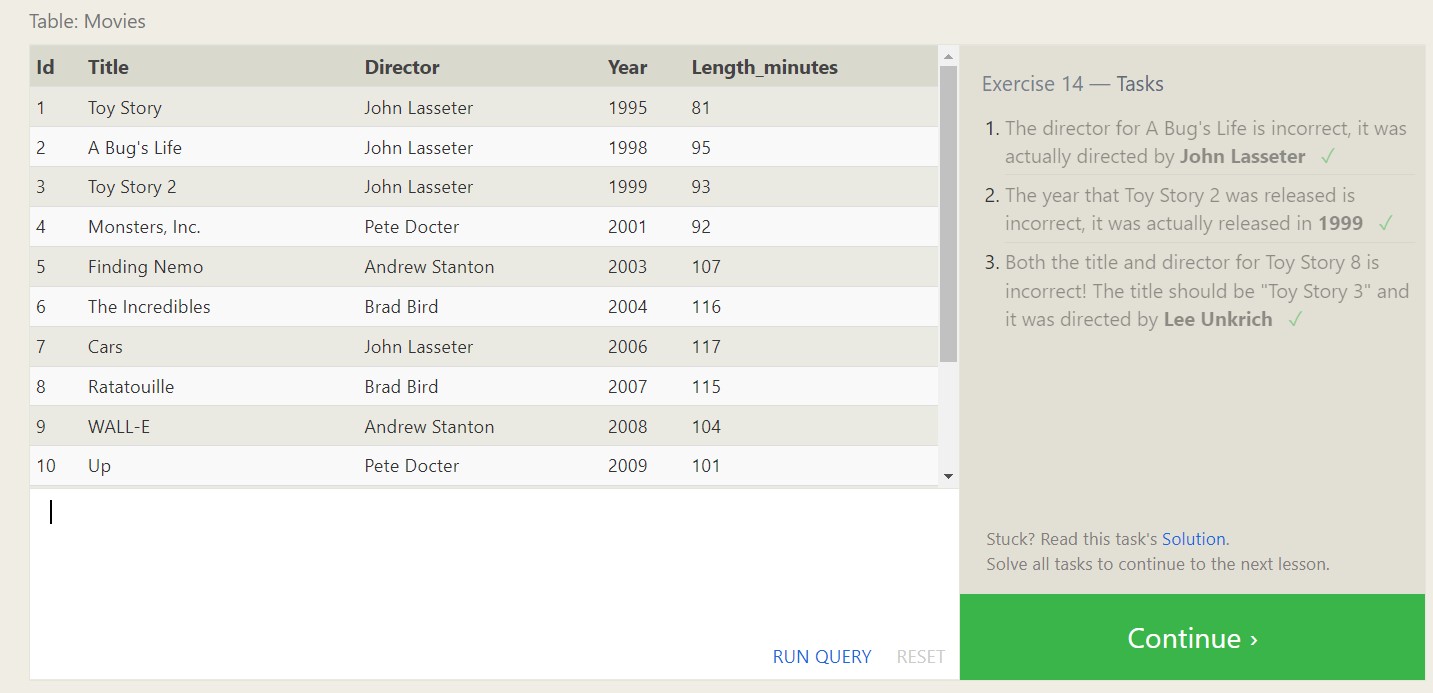
Group by director;

**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

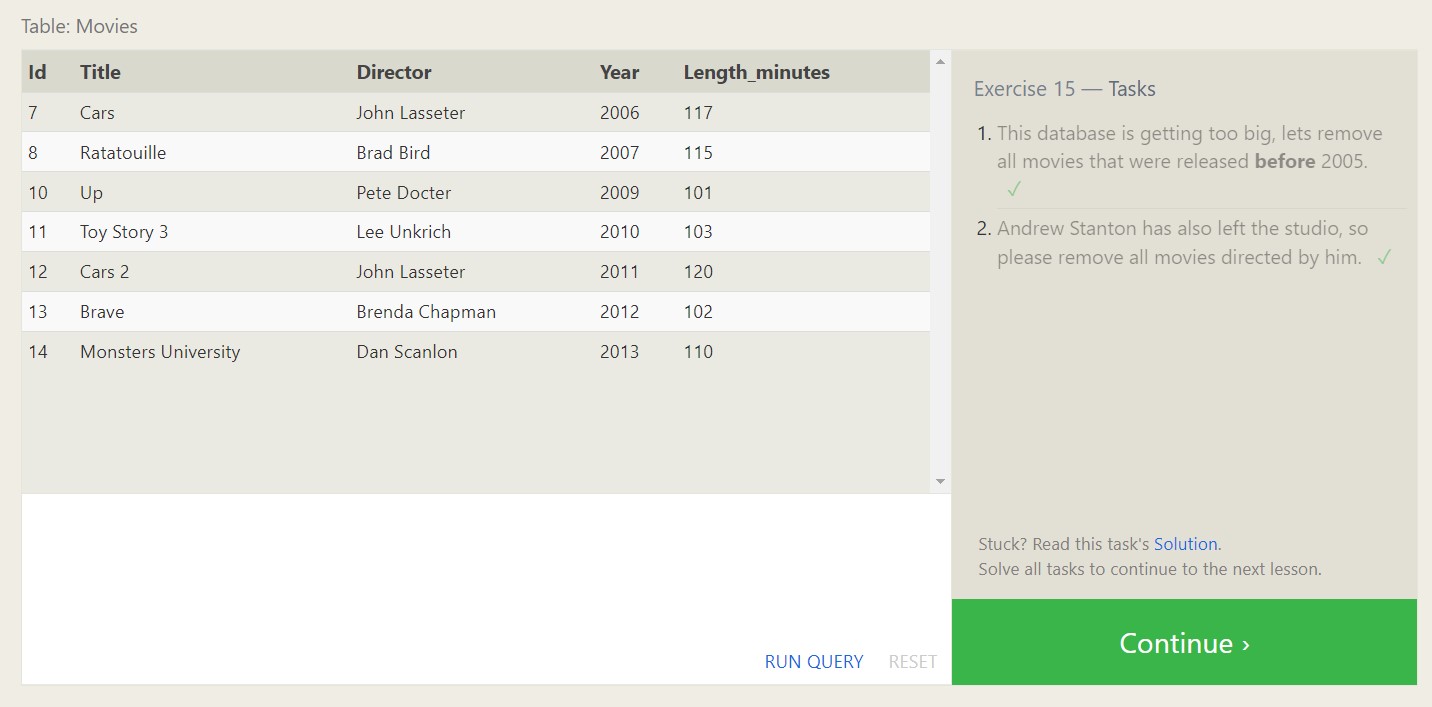
Where id = 3;

3. update movies

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

Where id = 11;

**SQL Lesson 15: Deleting rows**



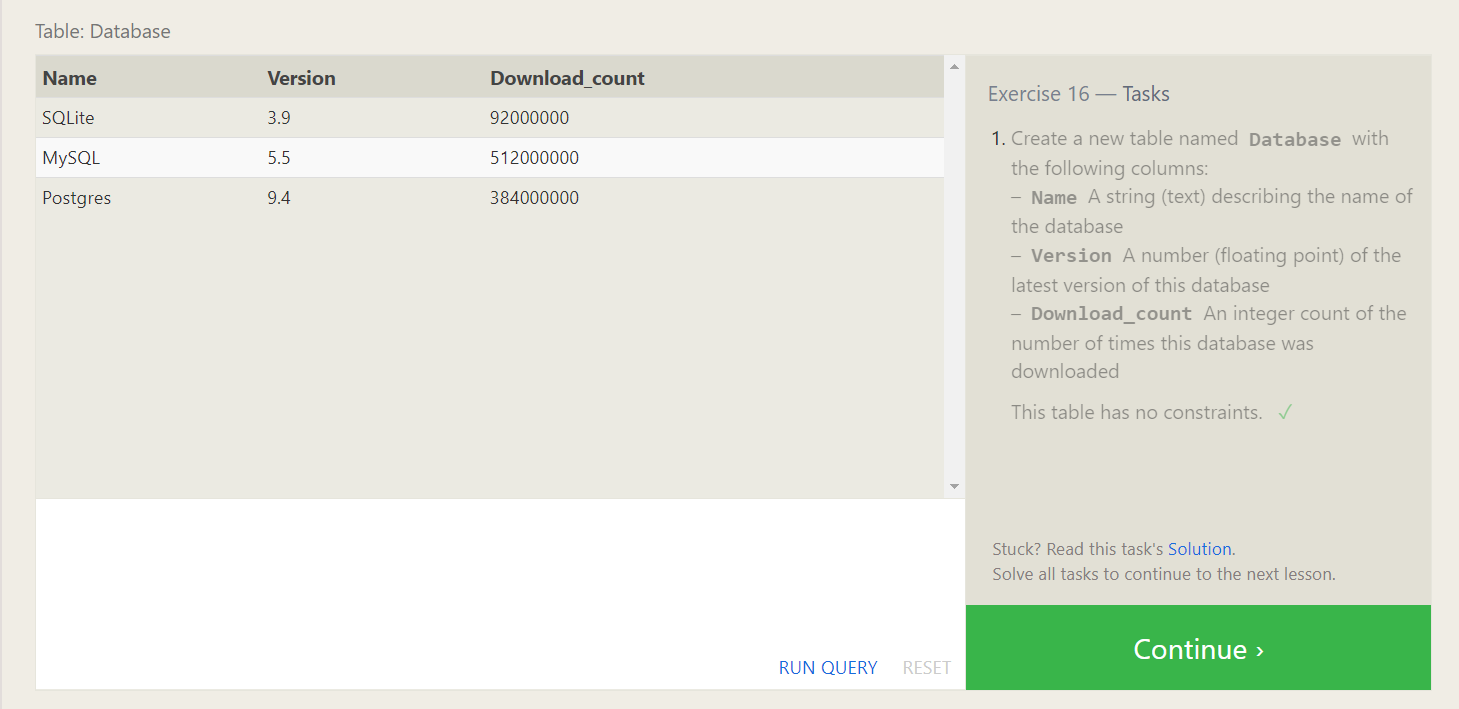
1. delete from movies

where year < 2005;

2. delete from movies

where director = "Andrew Stanton";

**SQL Lesson 16: Creating tables**

****

1. create table Database (

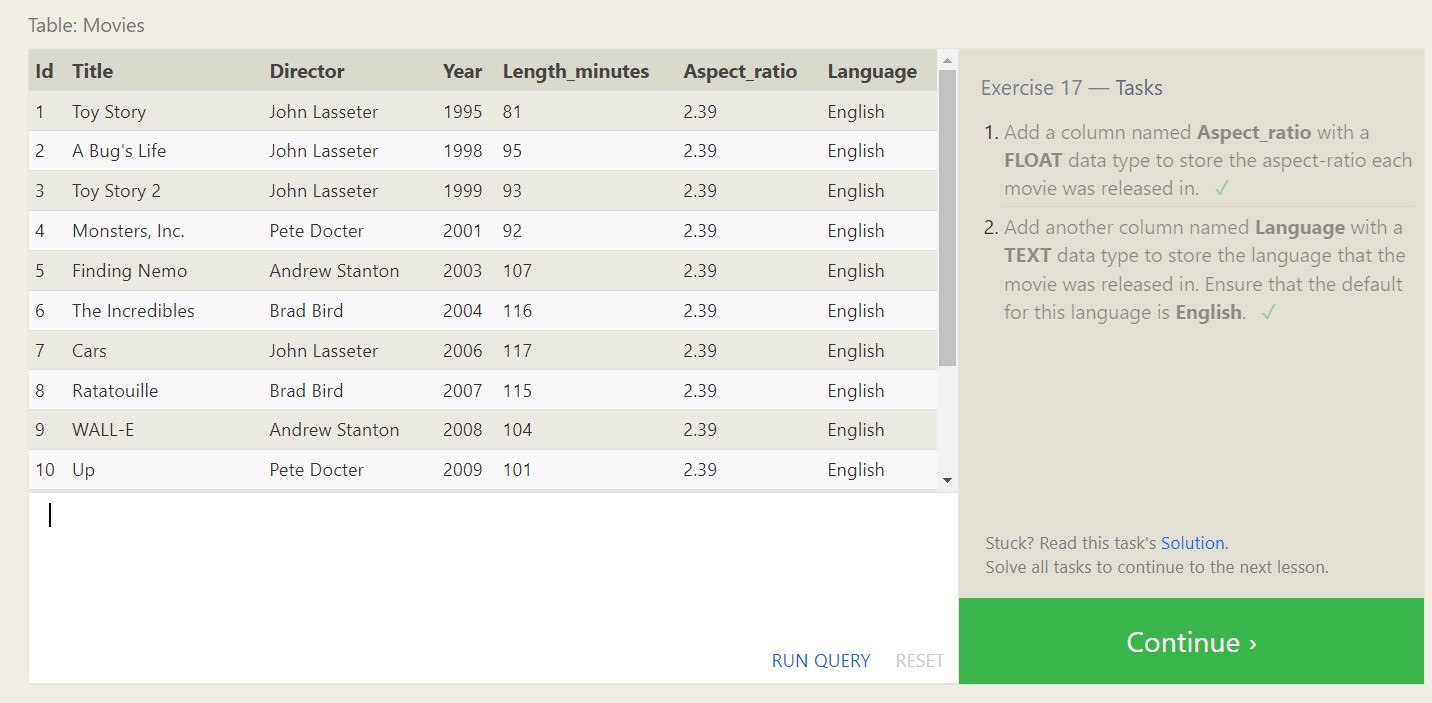
Name text,

Version float,

Download\_count integer

);

**SQL Lesson 17: Altering tables**



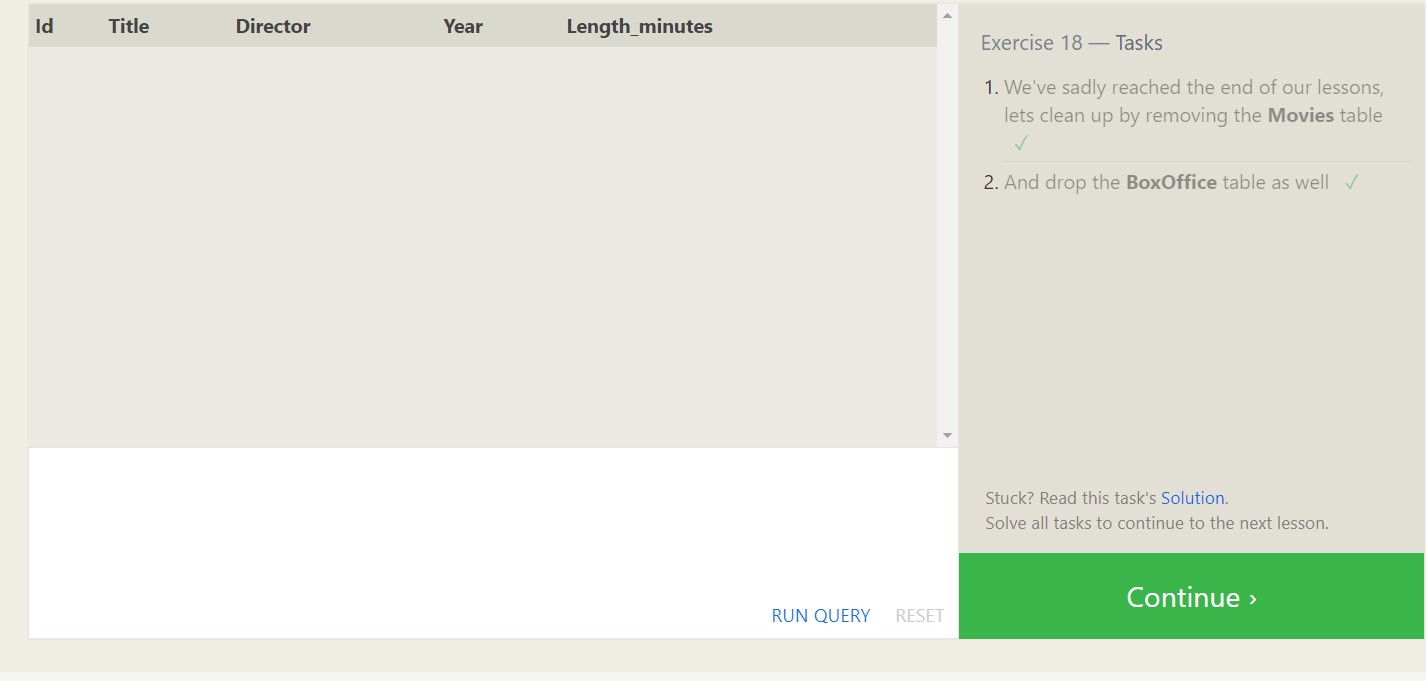
1. alter table Movies

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;

