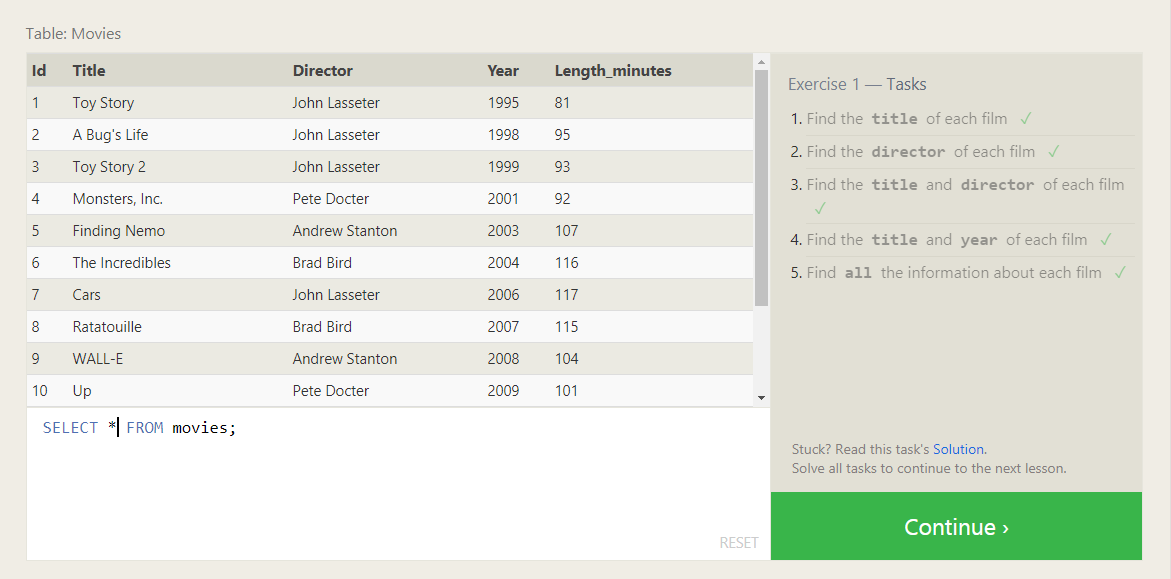
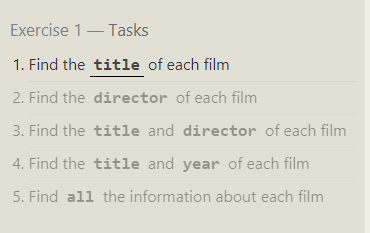
**SQL BOLT QUERIES TASK**

SQL Lesson 1:



Que:

Queries:

1.SELECT title FROM movies;

2.SELECT director FROM movies;

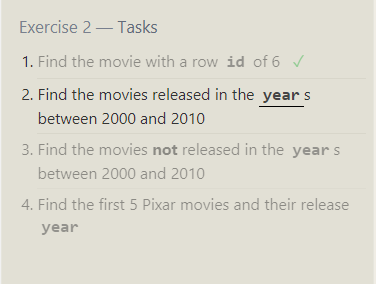
3.SELECT title, director FROM movies;

4.SELECT title, year FROM movies;

5.SELECT \* FROM movies;

SQL Lesson 2:



Que:

Queries:

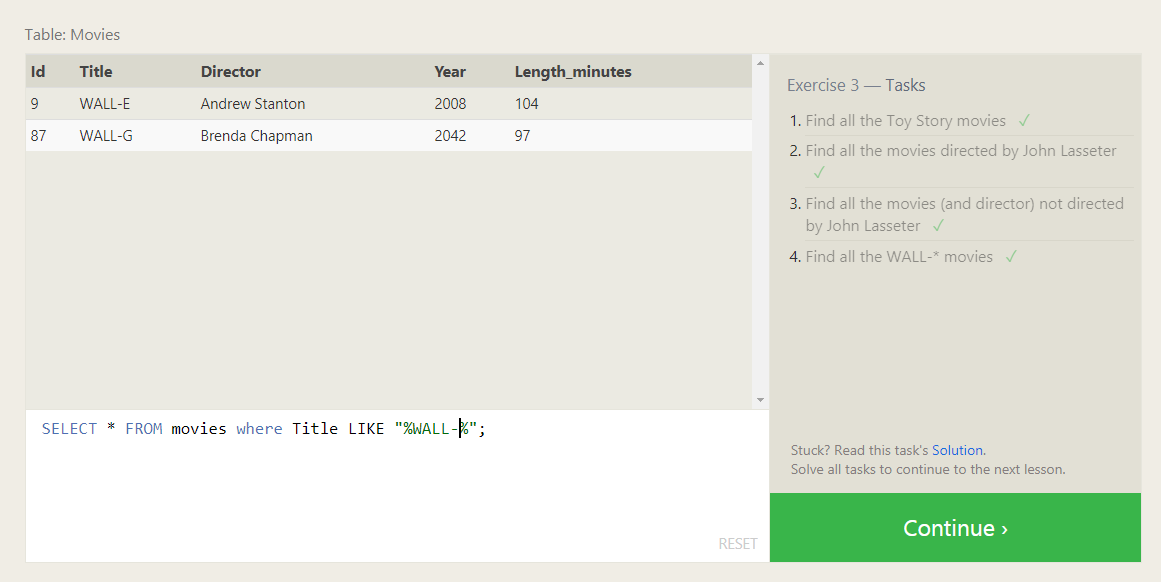
1.SELECT Title FROM movies where id=6;

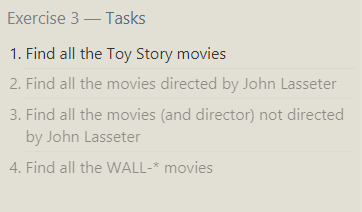
2. SELECT Title FROM movies where year between 2000 and 2010;

3. SELECT Title FROM movies where year NOT BETWEEN 2000 and 2010;

4.SELECT Title,Year FROM movies where year LIMIT 5;

SQL Lesson 3:

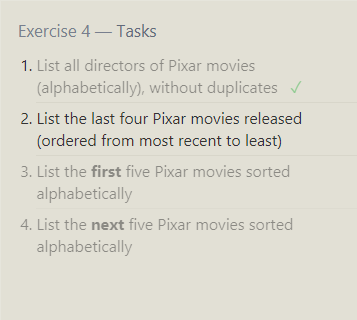


Que:

Queries:SELECT title FROM movies where title like "WALL-%";

SQL Lesson 4:



Que:

Queries:

1. SELECT distinct director FROM movies ORDER BY director ASC;

2. SELECT title,year FROM movies ORDER BY year DESC limit 4;

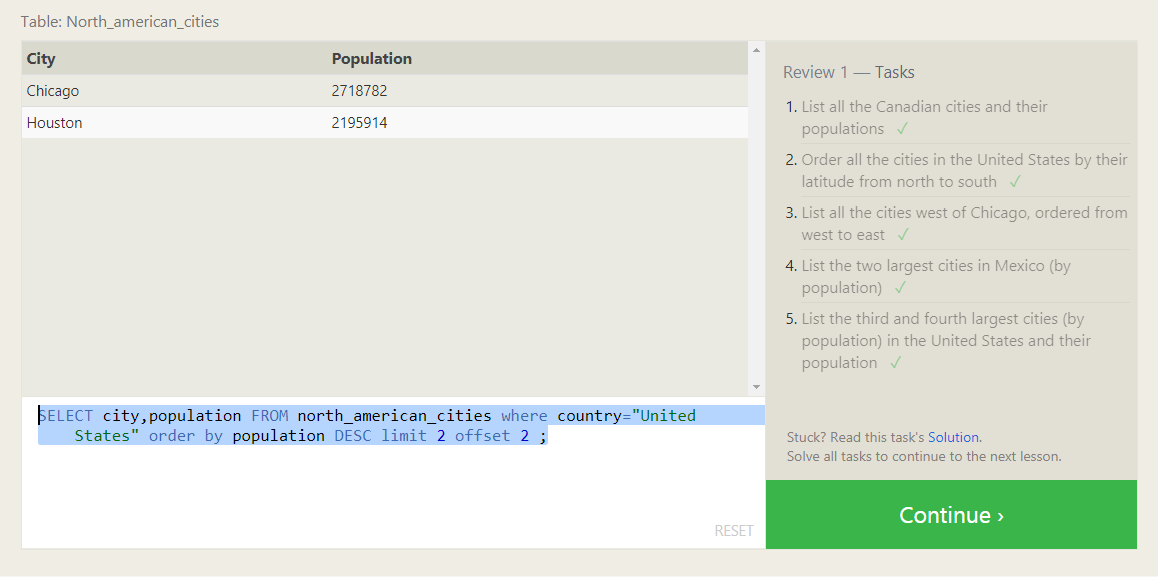
or

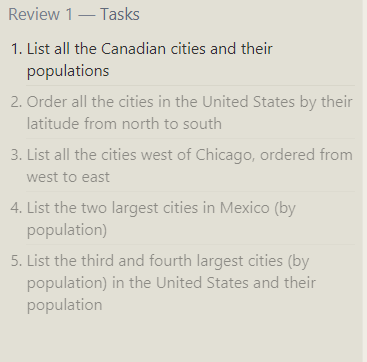
SELECT title FROM movies ORDER BY year DESC limit 4;

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 Lesson 5:



Que:

Queries:

1.SELECT city,population FROM north\_american\_cities where country="Canada";

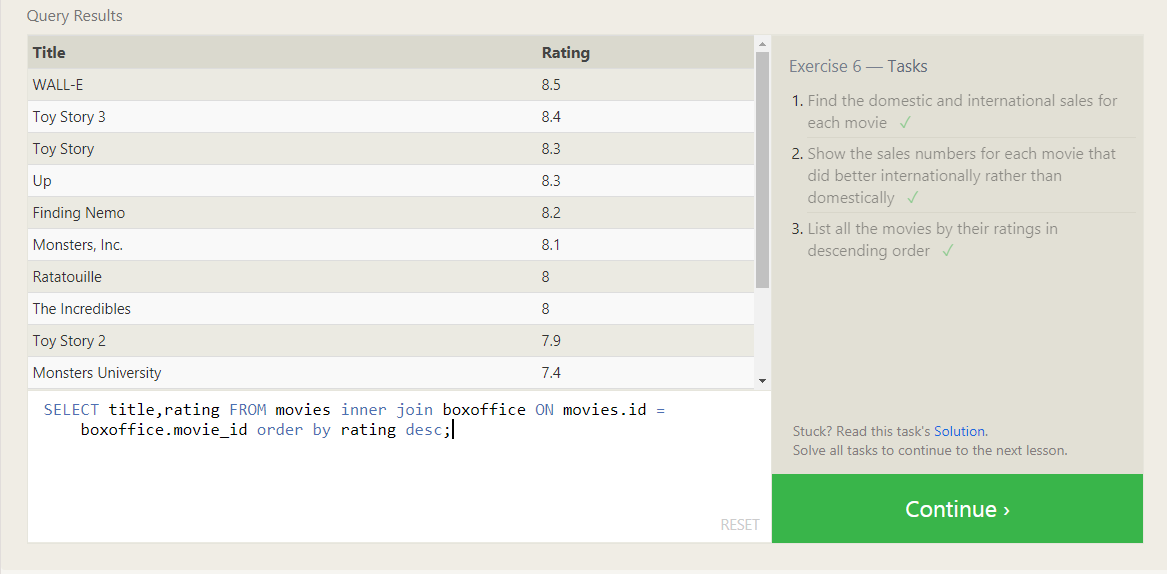
2.SELECT city,latitude FROM north\_american\_cities where country="United States" order by latitude DESC;

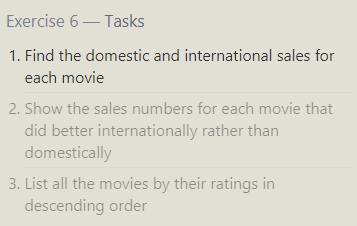
3.SELECT \* FROM north\_american\_cities where longitude<(-87.629798) order by longitude;

4.SELECT city,population FROM north\_american\_cities where country="Mexico" order by population DESC limit 2 ;

5.SELECT city,population FROM north\_american\_cities where country="United States" order by population DESC limit 2 offset 2

SQL Lesson 6:



Que:

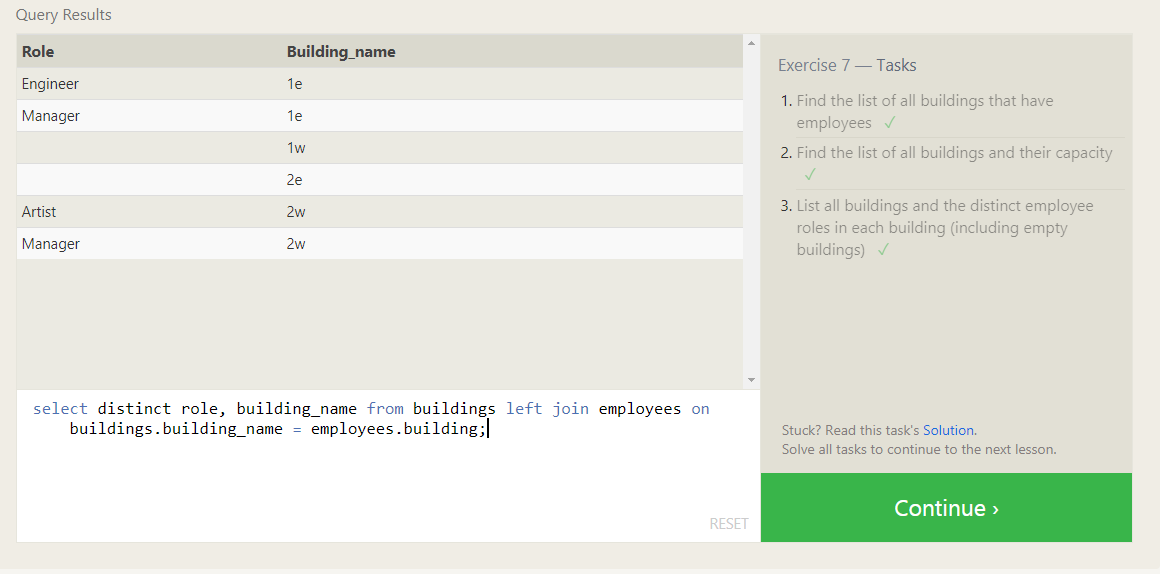
Queries:

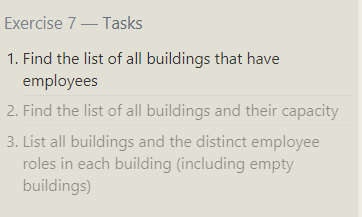
1.SELECT title,domestic\_sales,international\_sales FROM movies left join boxoffice ON movies.id = boxoffice.movie\_id;

2.SELECT title,international\_sales,domestic\_sales FROM movies inner join boxoffice ON movies.id = boxoffice.movie\_id where (international\_sales>domestic\_sales);

3.SELECT title,rating FROM movies inner join boxoffice ON movies.id = boxoffice.movie\_id order by rating desc;

SQL Lesson 7:



Que:

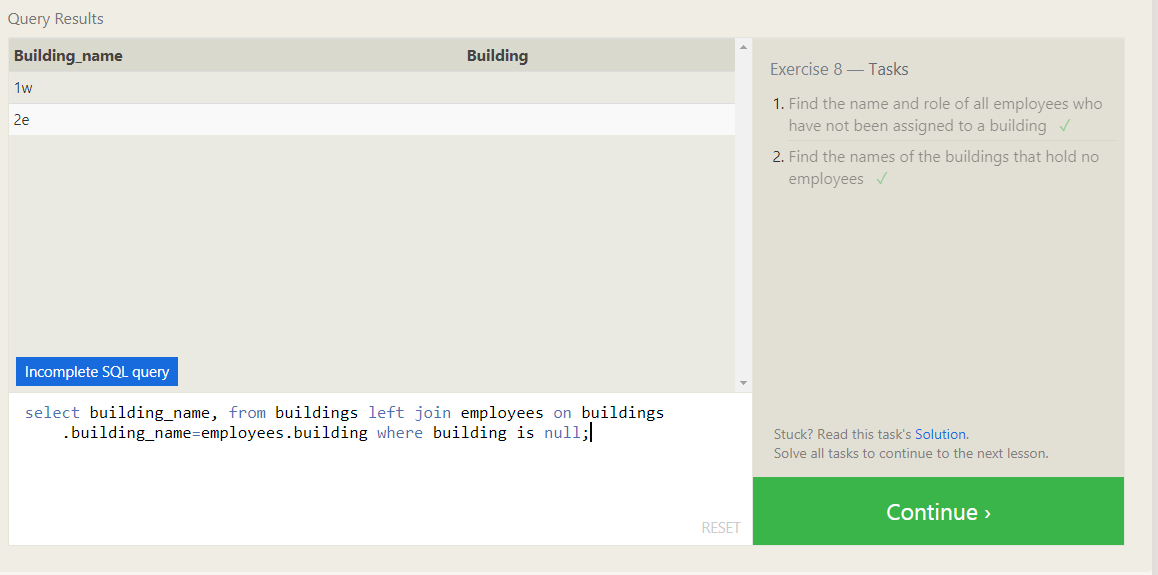
Queries:

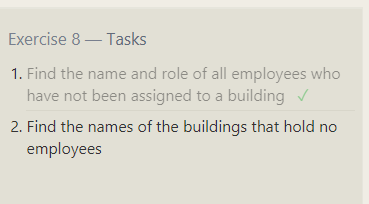
1.select distinct building\_name from buildings left join employees on buildings.building\_name = employees.building where building\_name=building;

2.select distinct building\_name,capacity from buildings ;

3.select distinct role, building\_name from buildings left join employees on buildings.building\_name = employees.building;

SQL Lesson 8:



Que:

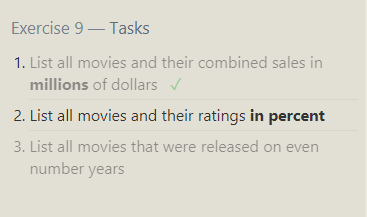
Queries:

1.select role,name from employees where building IS NULL;

2select building\_name, from buildings left join employees on buildings.building\_name=employees.building where building is null;

SQL Lesson 9:



Que:

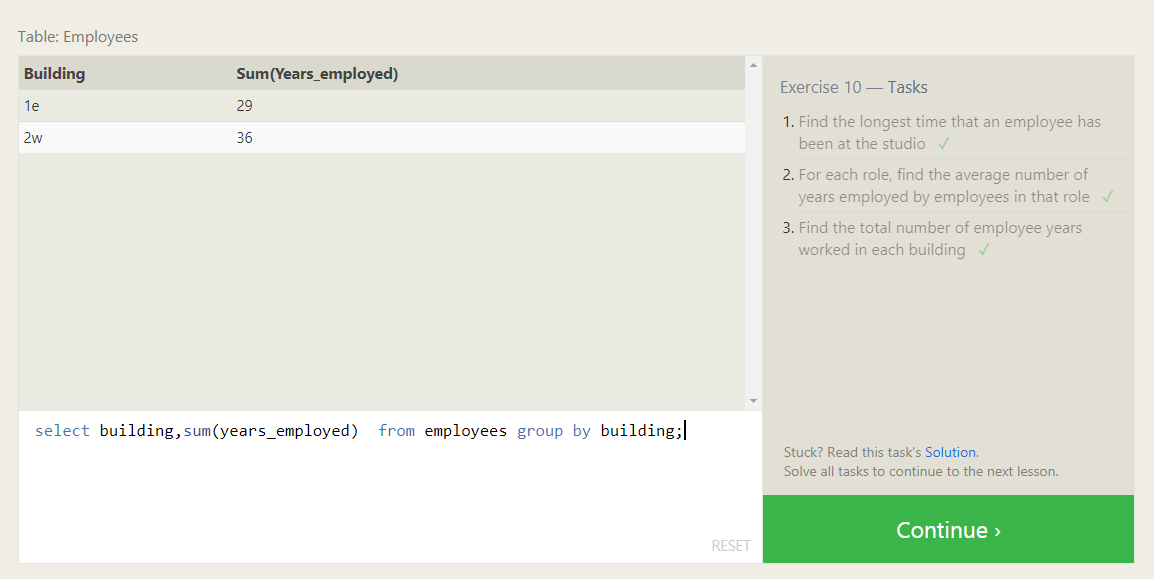
Queries:

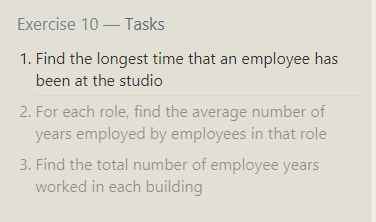
SELECT title, ((domestic\_sales+international\_sales)/1000000) as combined\_sales FROM movies left join boxoffice ON movies.id = boxoffice.movie\_id;

2. SELECT title, (rating\*10) as combined\_sales FROM movies left join boxoffice ON movies.id = boxoffice.movie\_id;

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

SQL Lesson 10:



Que:

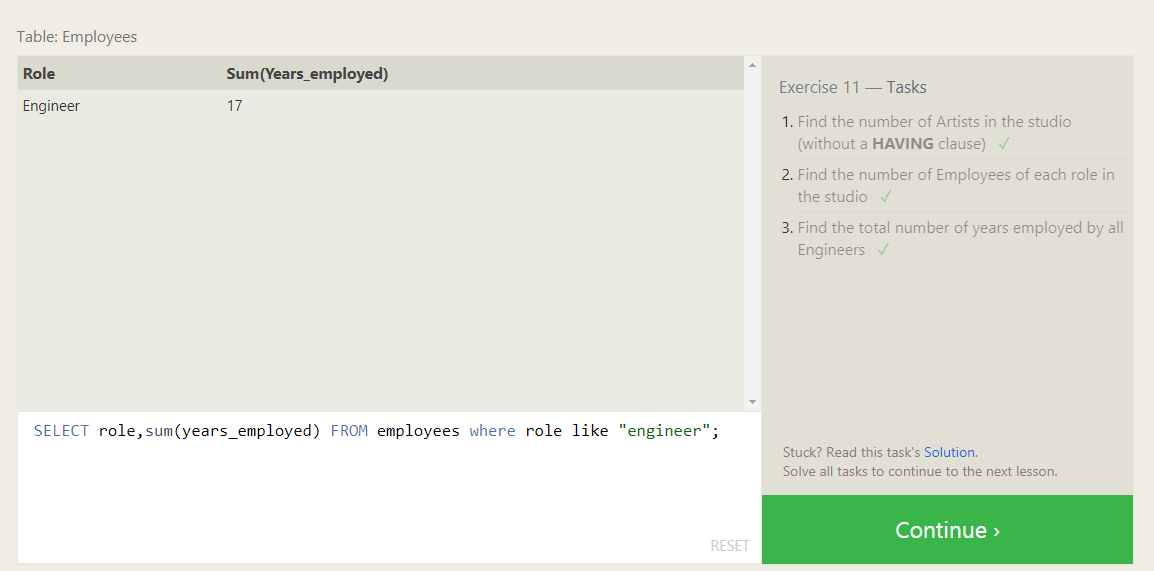
Queries:

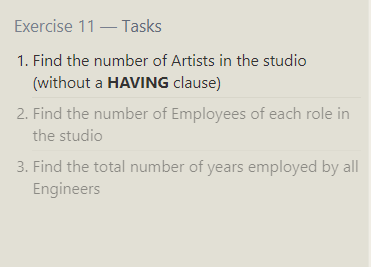
1. select name, max(years\_employed) as longest\_time from employees;

2. select role, avg(years\_employed) from employees group by role;

3. select building,sum(years\_employed) from employees group by building;

SQL Lesson 11:



Que:

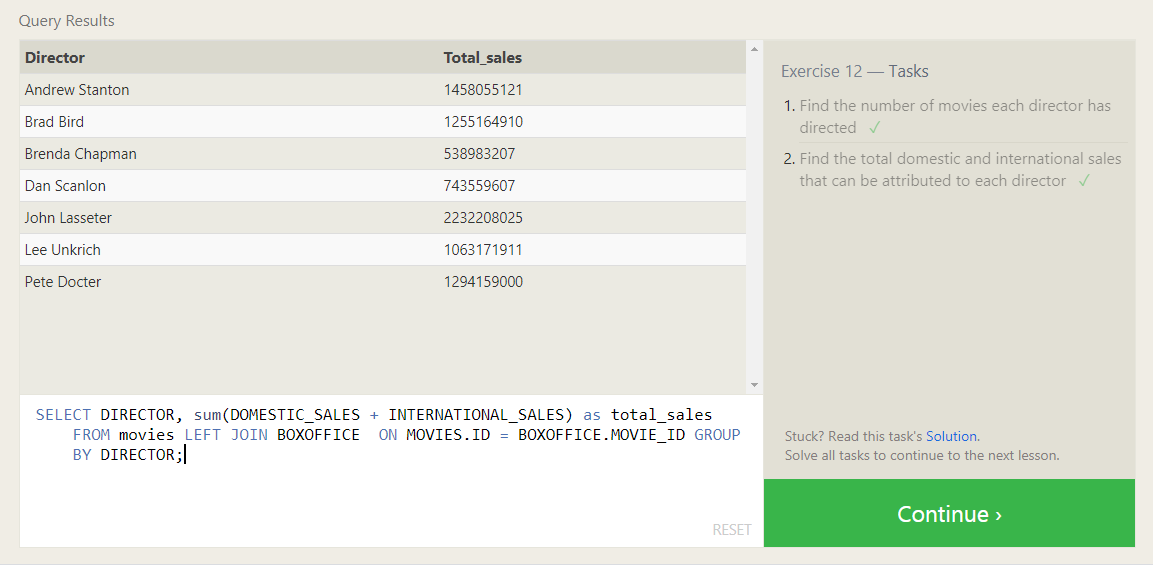
Queries:

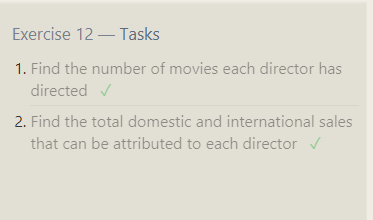
1. SELECT count(role) FROM employees where role like "artist";

2. SELECT count(name),role FROM employees group by role;

3. SELECT role,sum(years\_employed) FROM employees where role like "engineer";

SQL Lesson 12:



Que:

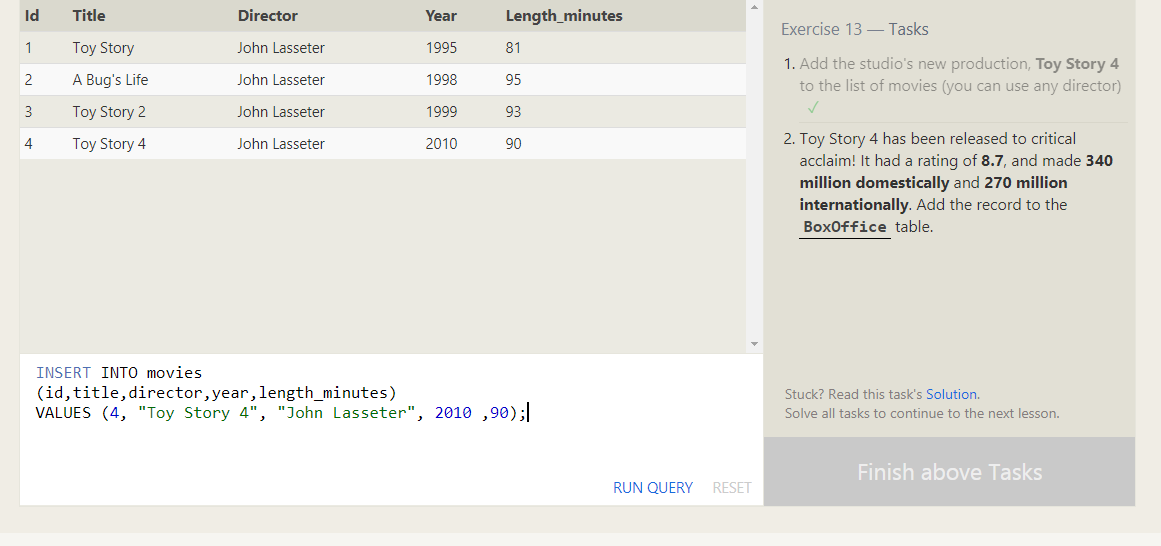
Queries:

1. SELECT DIRECTOR,COUNT(TITLE) FROM movies GROUP BY DIRECTOR;

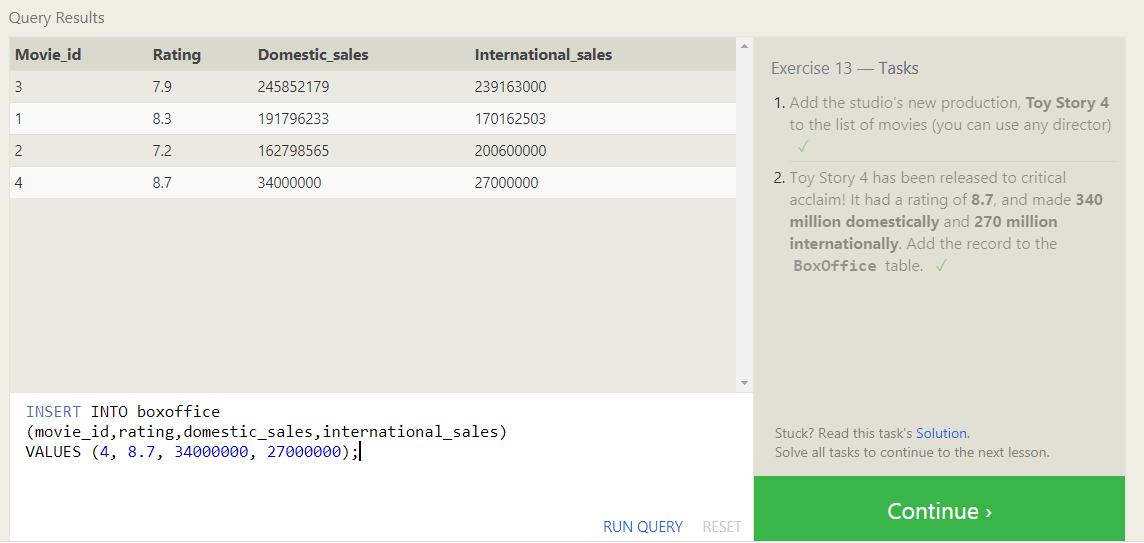
2. SELECT DIRECTOR, sum(DOMESTIC\_SALES + INTERNATIONAL\_SALES) as total\_sales FROM movies LEFT JOIN BOXOFFICE ON MOVIES.ID = BOXOFFICE.MOVIE\_ID GROUP BY DIRECTOR;

SQL Lesson 13:

Que:1



Que:2



SQL Lesson:14