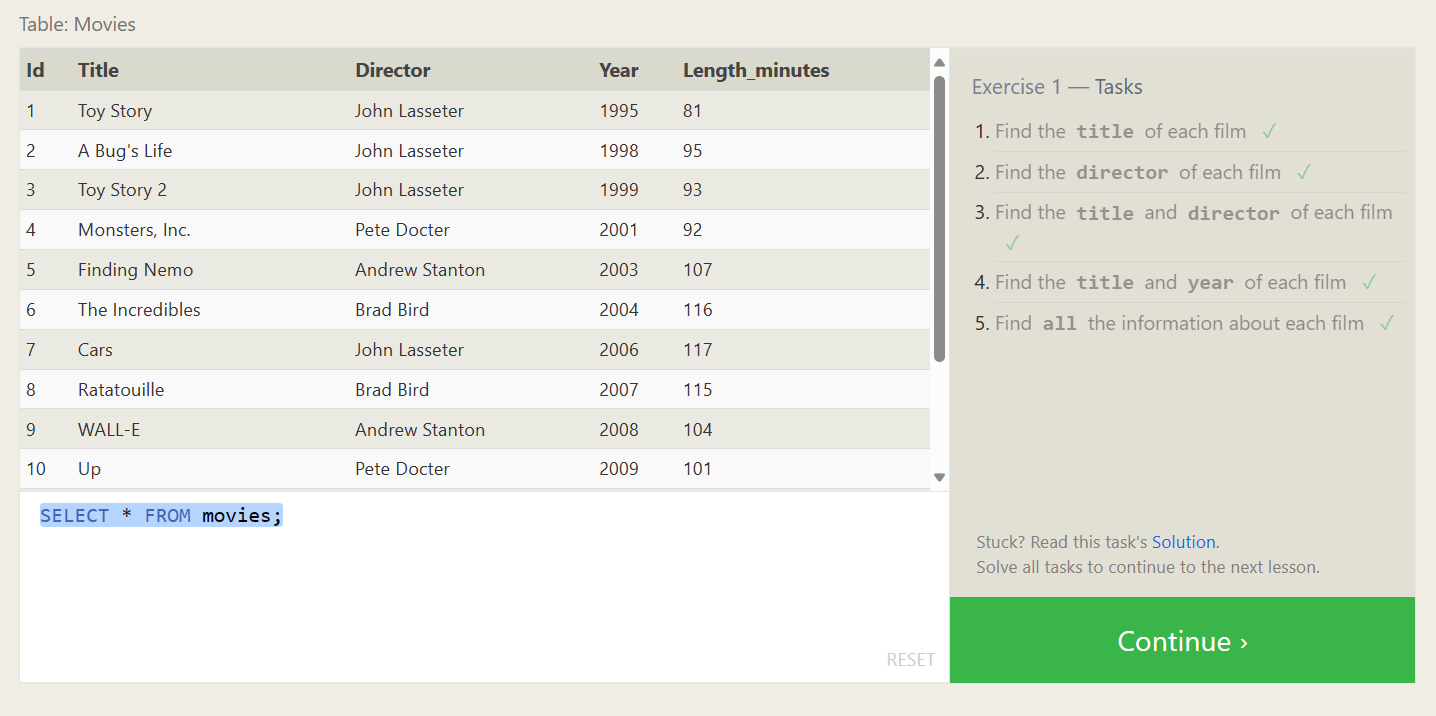
SQL Lesson 1: SELECT queries 101



SELECT title FROM movies;

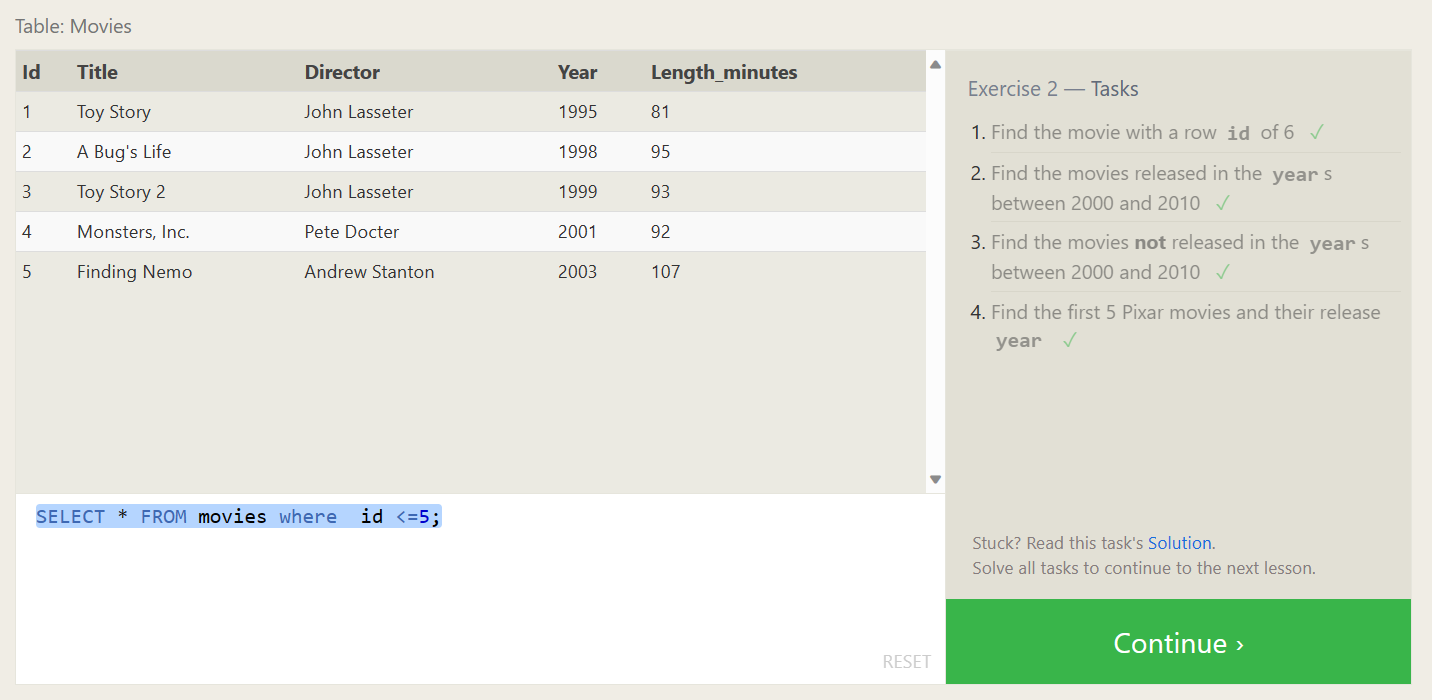
SELECT director FROM movies;

SELECT title, director FROM movies;

SELECT title, year FROM movies;

SELECT \* FROM movies;

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



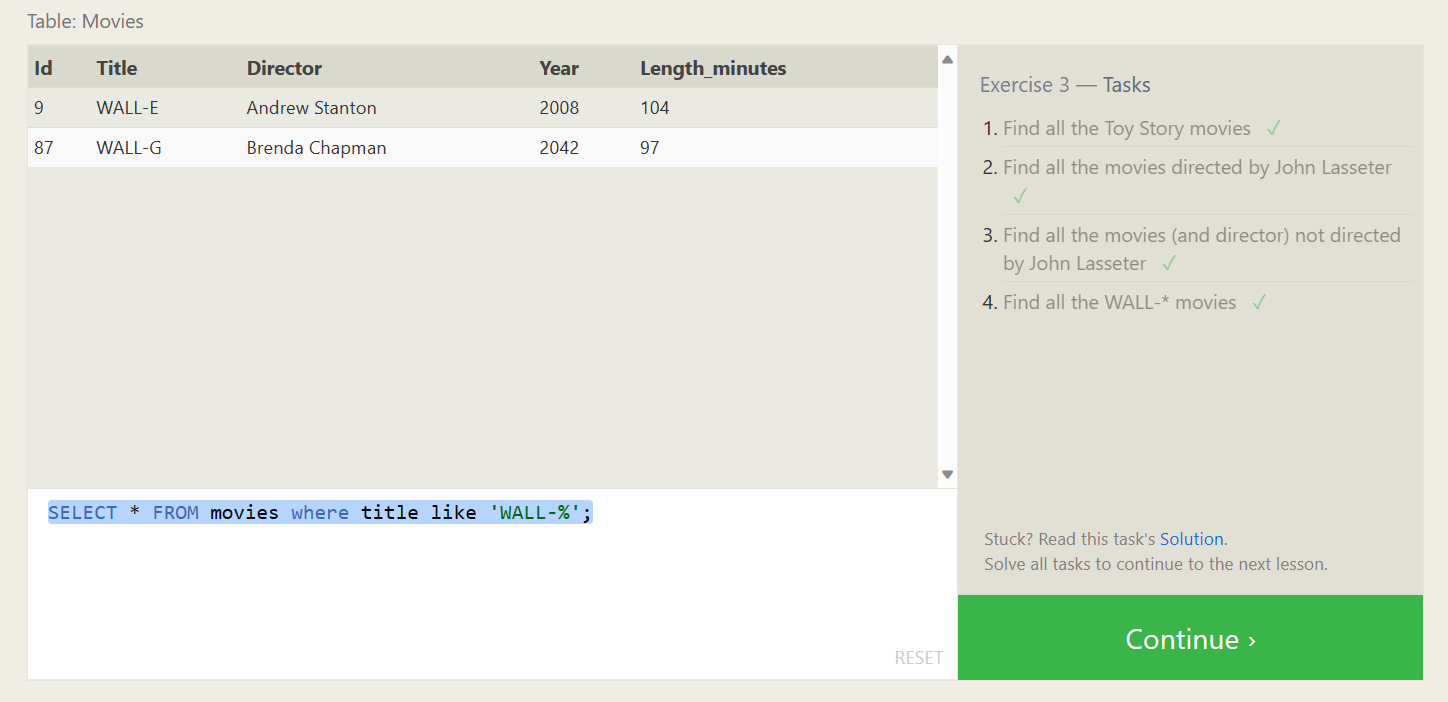
SELECT \* FROM movies where id =6;

SELECT \* FROM movies where year between 2000 and 2010;

SELECT \* FROM movies where year not between 2000 and 2010;

SELECT \* FROM movies where id <=5;

SQL Lesson 3: Queries with constraints (Pt. 2)



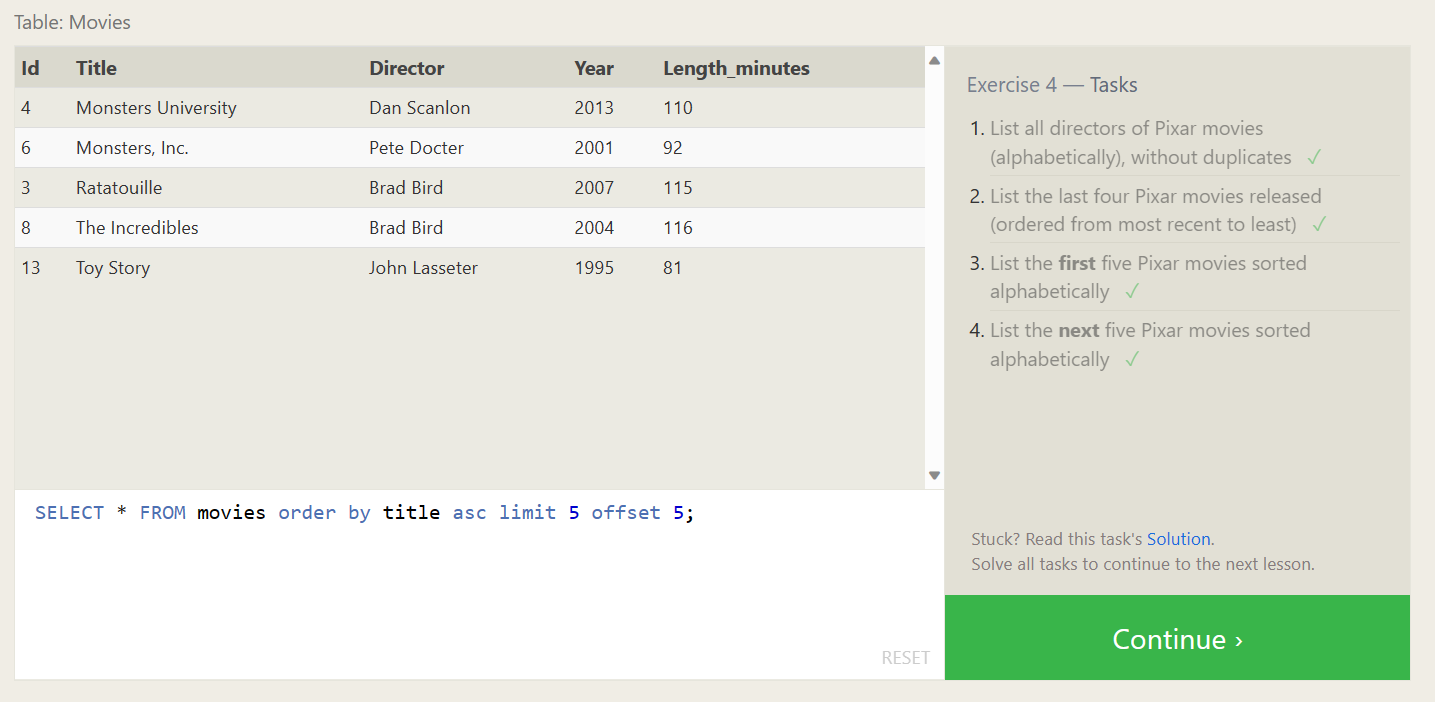
SELECT \* FROM movies where title like 'Toy story%';

SELECT \* FROM movies where director =' John Lasseter';

SELECT \* FROM movies where director != 'John Lasseter';

SELECT \* FROM movies where title like 'WALL-%';

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

****

SELECT distinct director FROM movies order by director;

SELECT \* FROM movies order by year desc limit 4;

SELECT \* FROM movies order by title asc limit 5;

SELECT \* FROM movies order by title asc limit 5 offset 5;

**SQL Review: Simple SELECT Queries**

****

SELECT \* FROM north\_american\_cities where country = 'Canada';

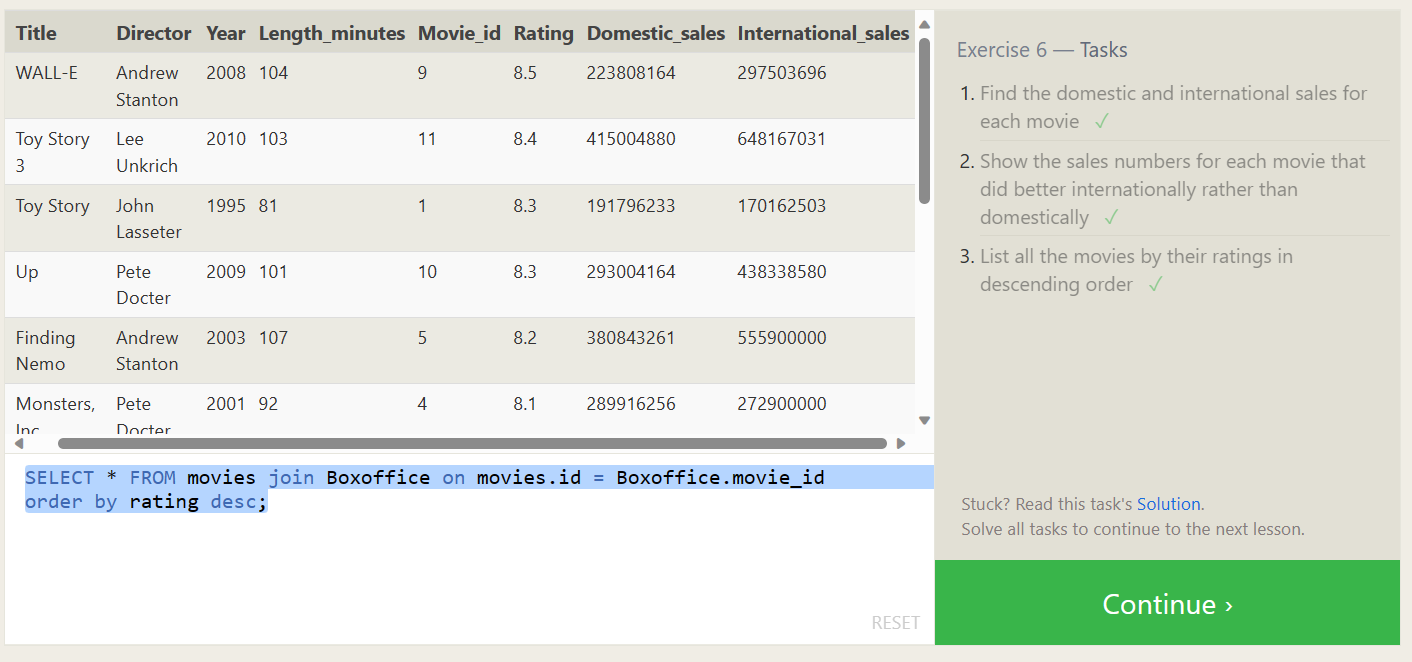
SELECT \* FROM north\_american\_cities where country = 'United States' order by latitude desc;

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;

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

****

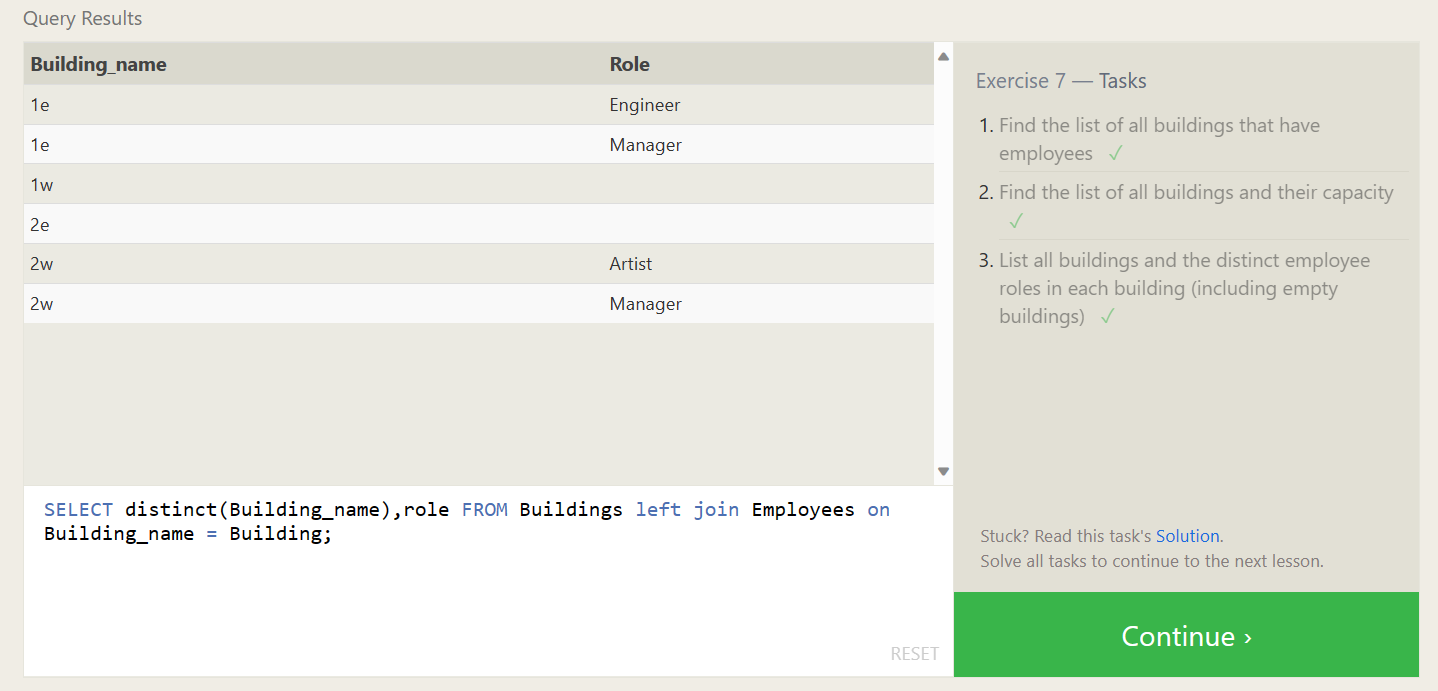
SELECT \* FROM movies join Boxoffice on movies.id=Boxoffice.movie\_id;

SELECT \* FROM movies join Boxoffice on movies.id = Boxoffice.movie\_id where International\_sales > Domestic\_sales;

SELECT \* FROM movies join Boxoffice on movies.id = Boxoffice.movie\_id

order by rating desc;

**SQL Lesson 7: OUTER JOINs**



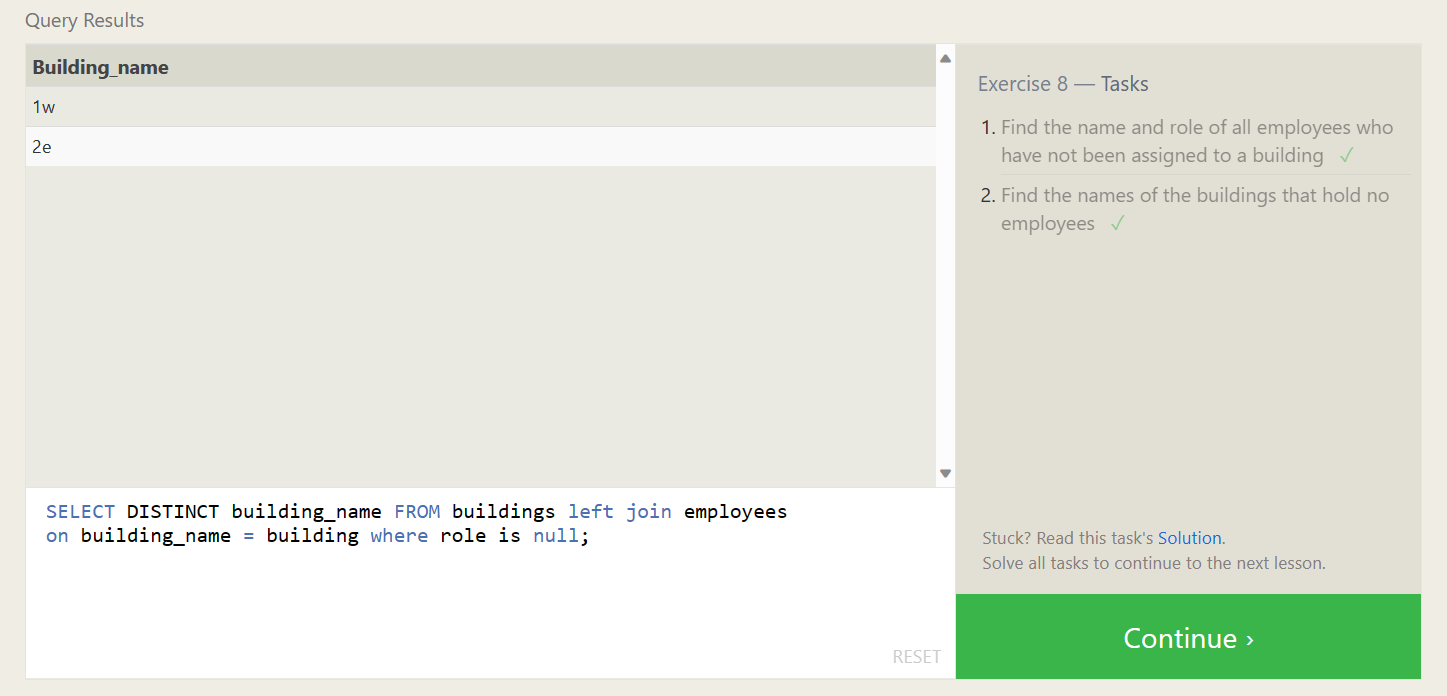
SELECT distinct(Building) FROM employees;

SELECT \* FROM Buildings;

SELECT distinct(Building\_name),role FROM Buildings left join Employees on

Building\_name = Building;

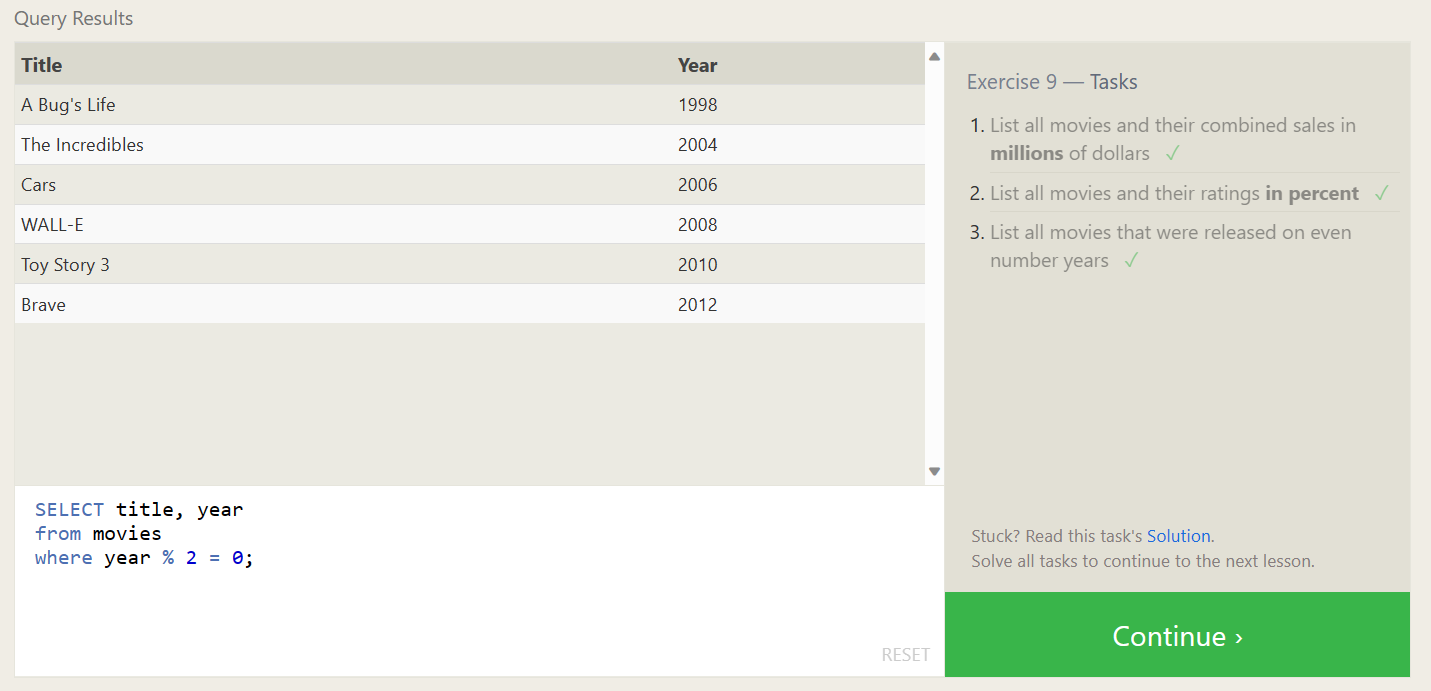
**SQL Lesson 8: A short note on NULLs**



SELECT name, role FROM employees where building is null;

SELECT distinct building\_name FROM buildings left join employee on building\_name = building where role is null;

**SQL Lesson 9: Queries with expressions**

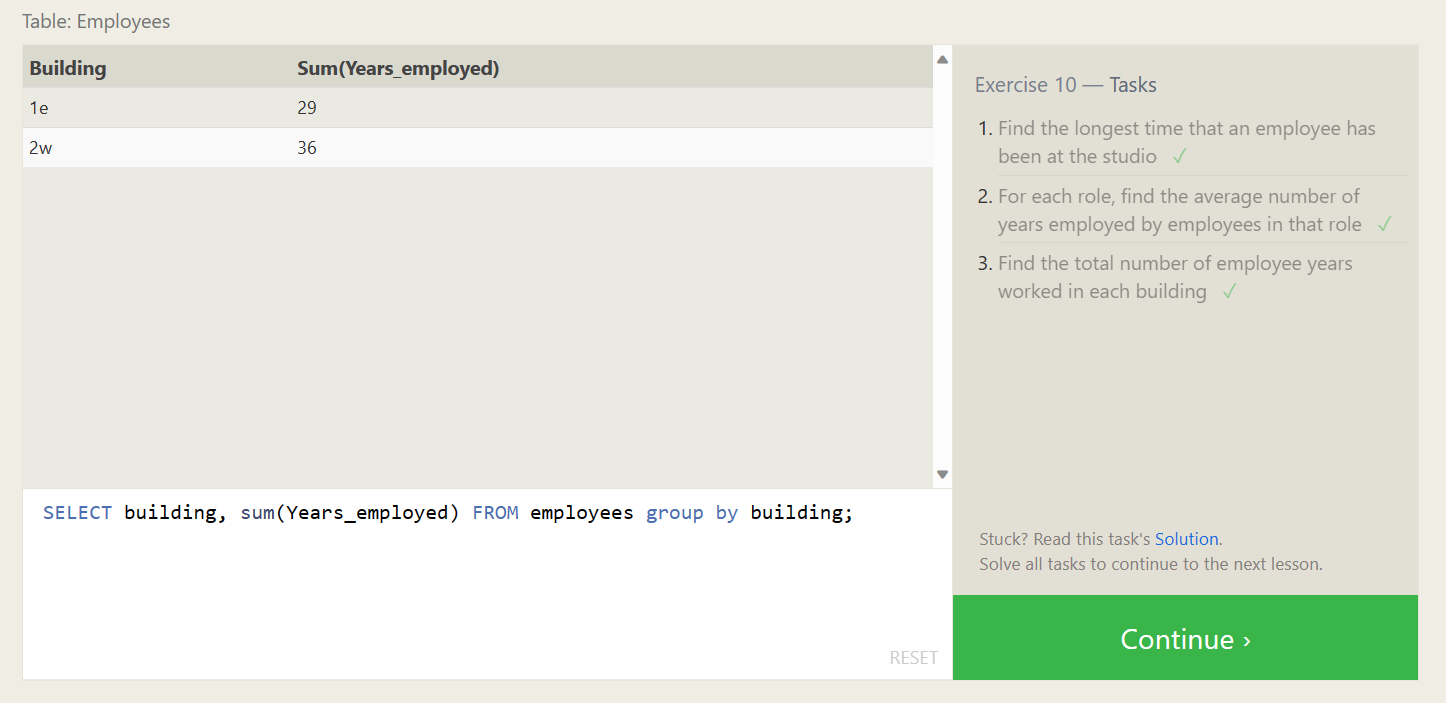
****

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;

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

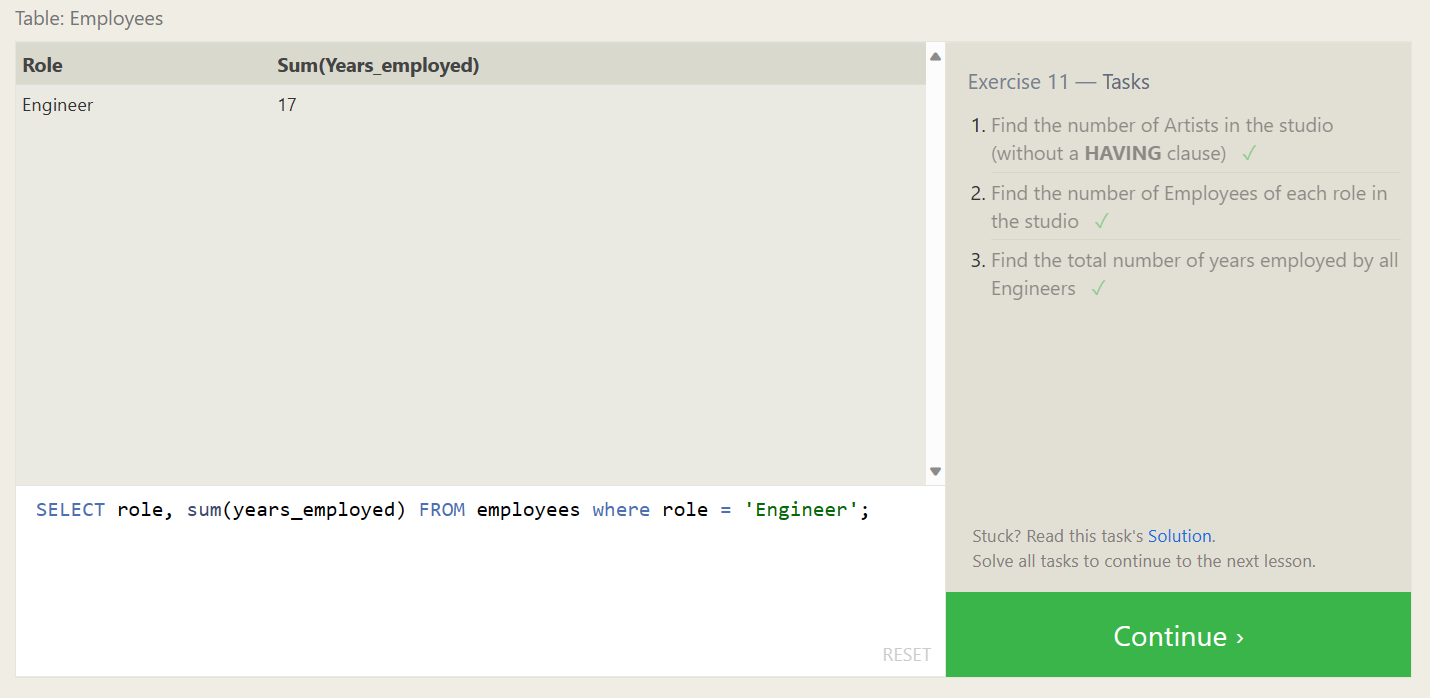


SELECT max(Years\_employed) FROM employees;

SELECT role,avg(Years\_employed) FROM employees group by role;

SELECT building, sum(Years\_employed) FROM employees group by building;

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

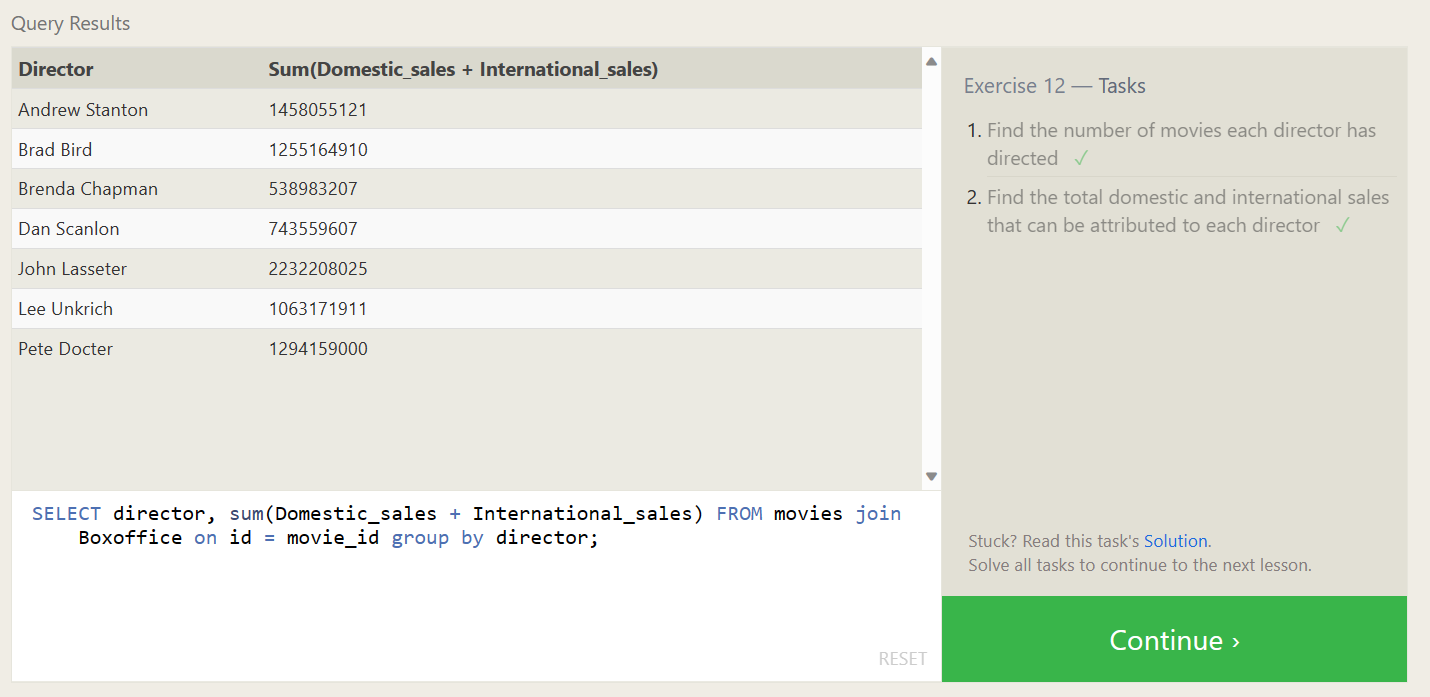
****

SELECT count(\*) FROM employees where role = 'Artist';

SELECT role, count(\*) FROM employees group by role;

SELECT role, sum(years\_employed) FROM employees where role = 'Engineer';

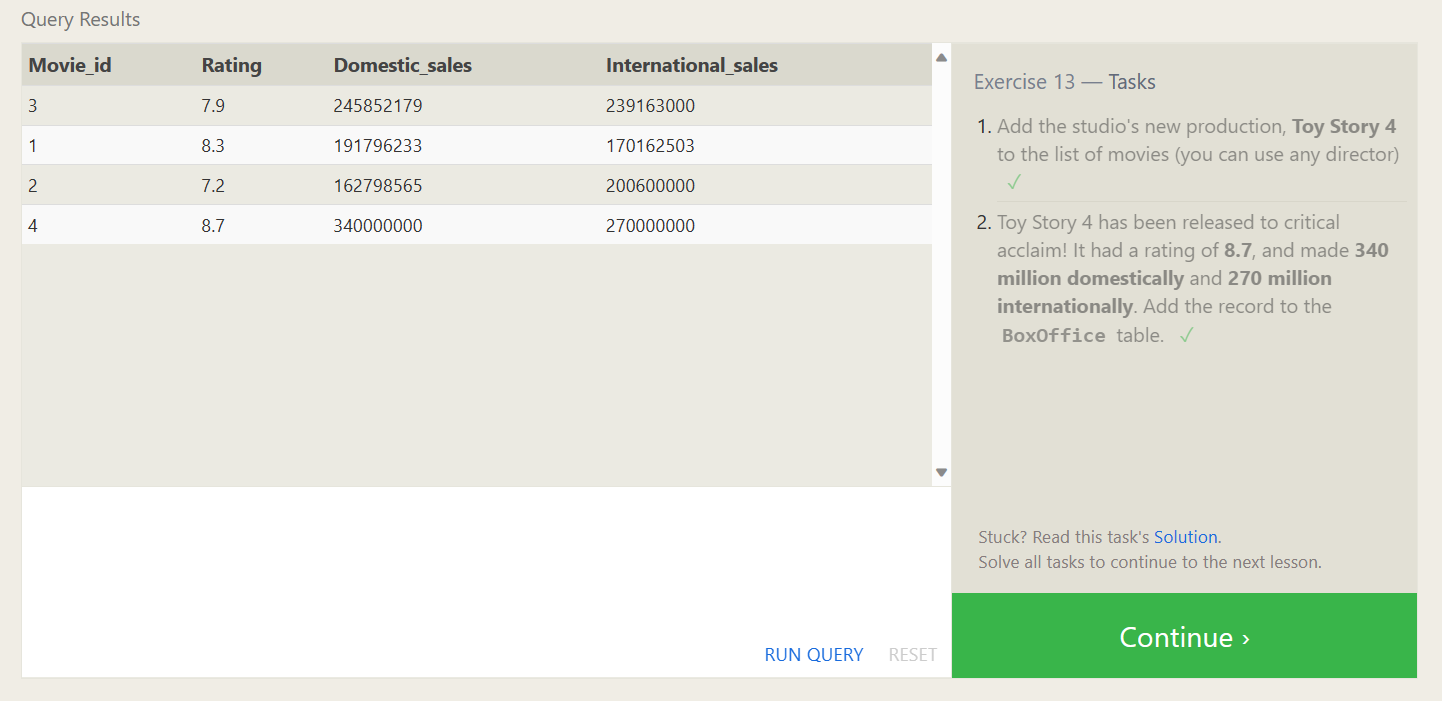
**SQL Lesson 12: Order of execution of a Query**

****

SELECT director, count(\*) FROM movies group by director;

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

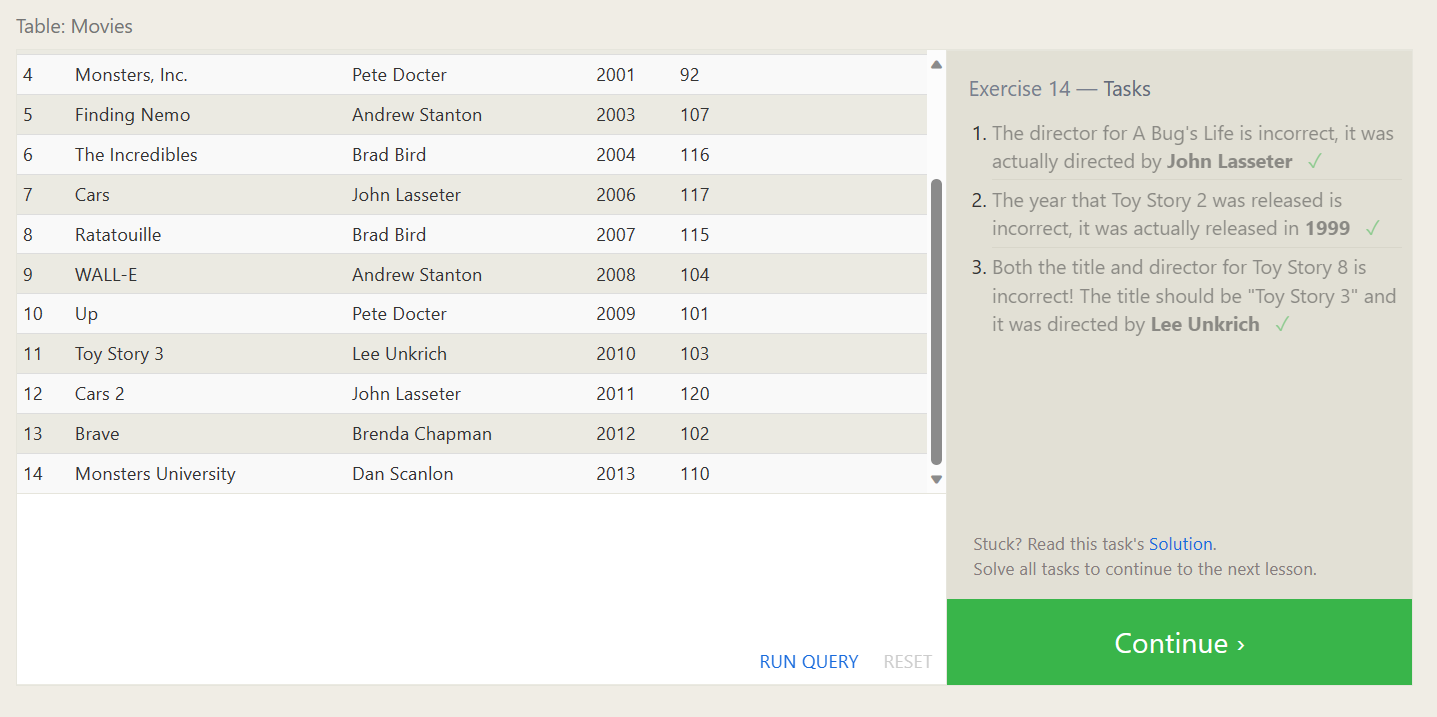
**SQL Lesson 13: Inserting rows**

****

Insert into movies values (4, "Toy Story 4", "El Directore", 2015, 90);

Insert into boxoffice values (4, 8.7, 340000000, 270000000);

**SQL Lesson 14: Updating rows**

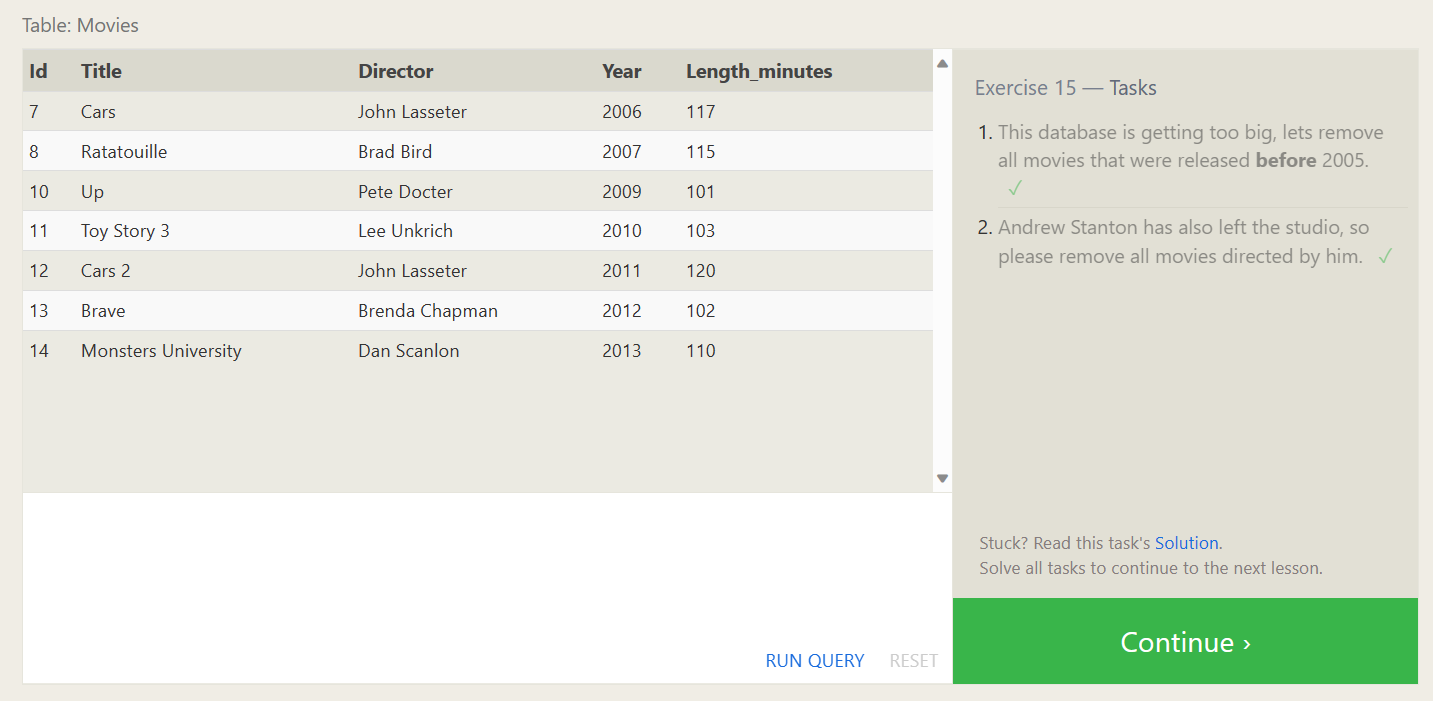


update movies set director = "John Lasseter" where id = 2;

update movies set year = 1999 where id = 3;

update movies set title = 'Toy Story 3', director = 'Lee Unkrich' where id = 11;

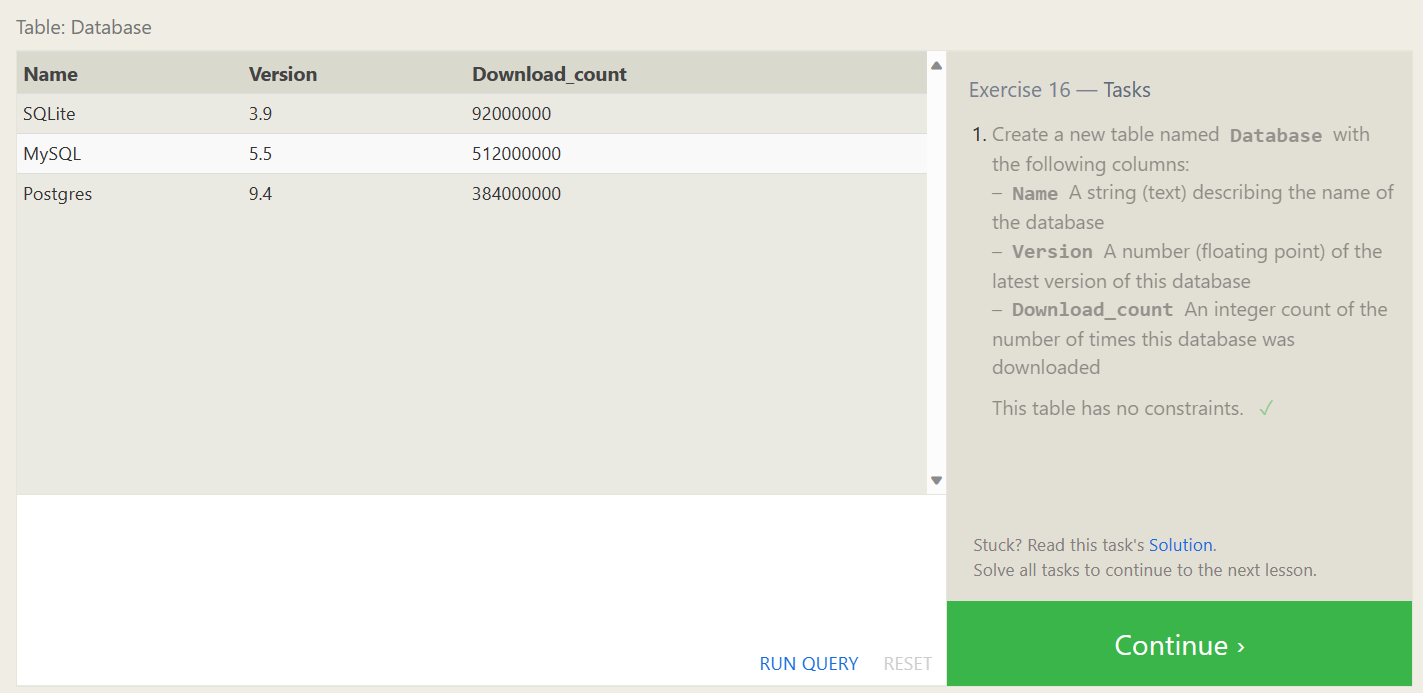
**SQL Lesson 15: Deleting rows**



delete from movies where year <=2005;

delete from movies where director = 'Andrew Stanton';

**SQL Lesson 16: Creating tables**



crea Database (

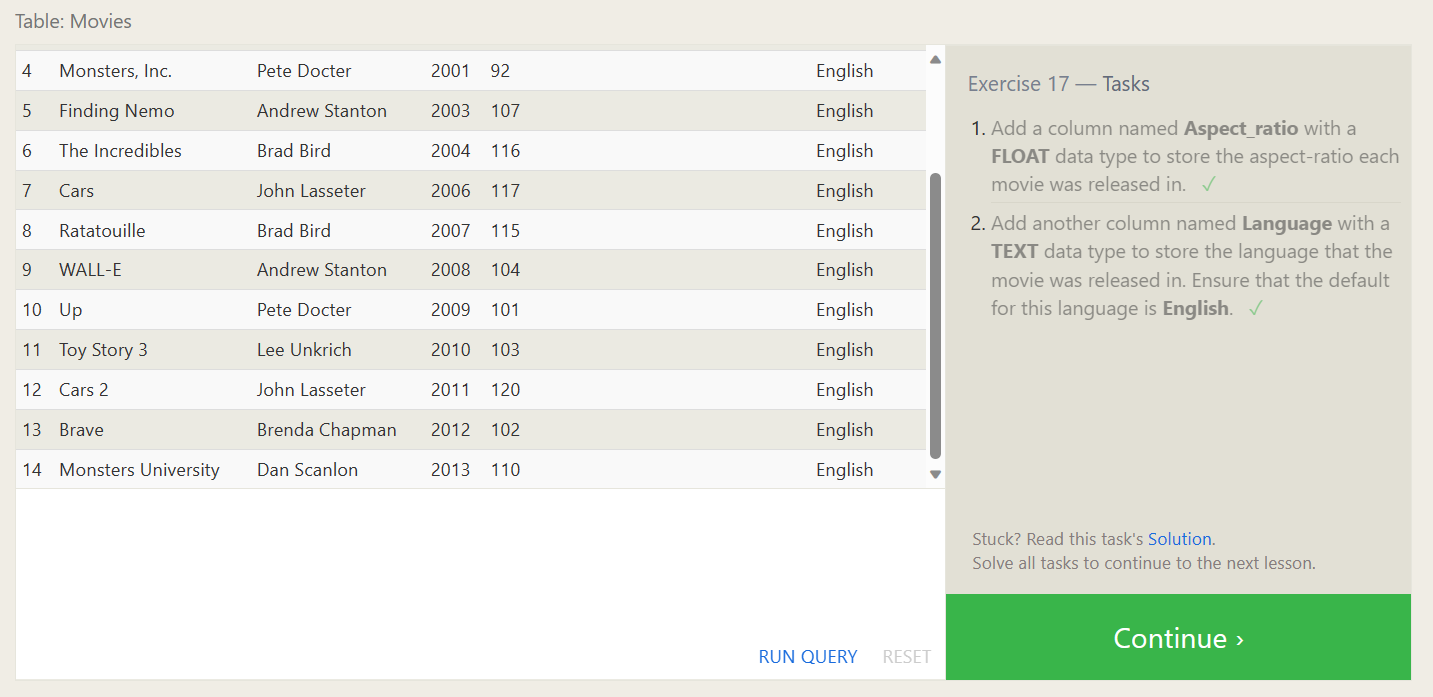
Name text,

Version float,

Download\_count integer

);

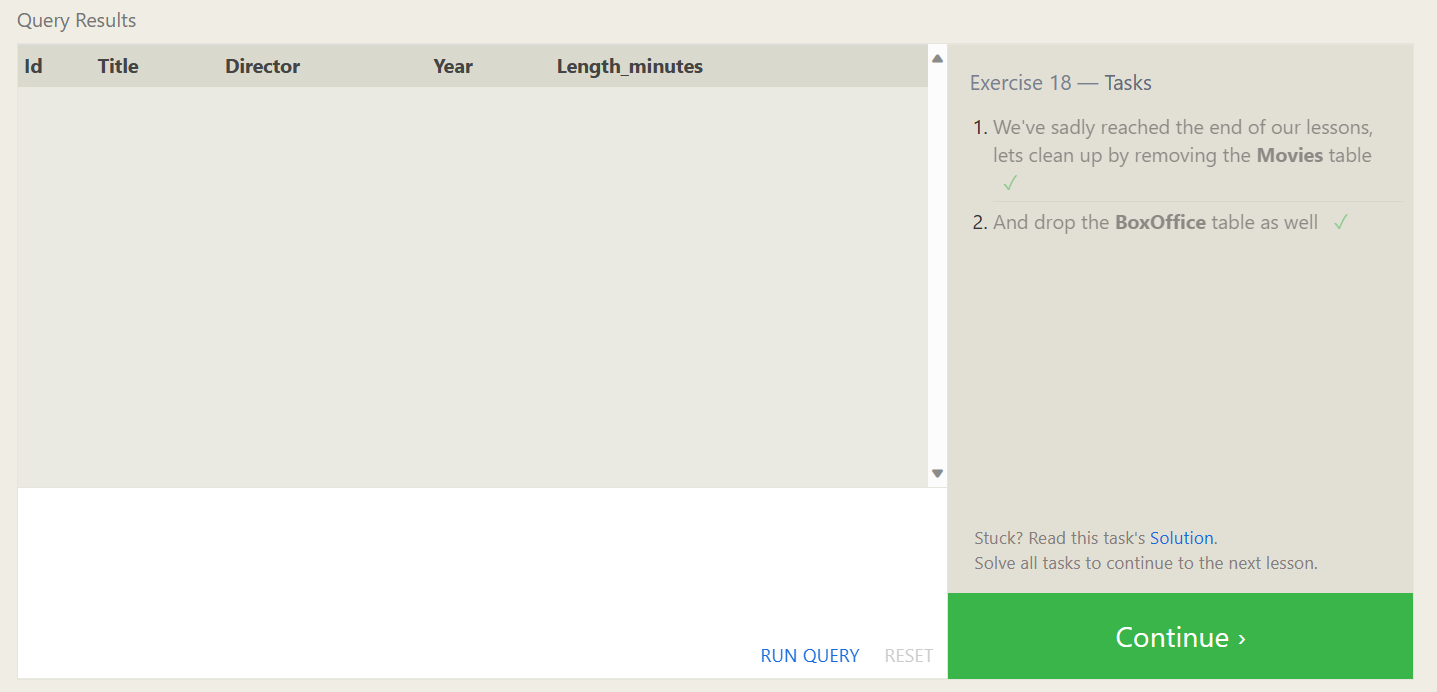
**SQL Lesson 17: Altering tables**



alter table movies add Aspect\_ratio datatype float;

alter table movies add Language datatype text default English;

**SQL Lesson 18: Dropping tables**



drop table if exists movies;

drop table if exists Boxoffice;