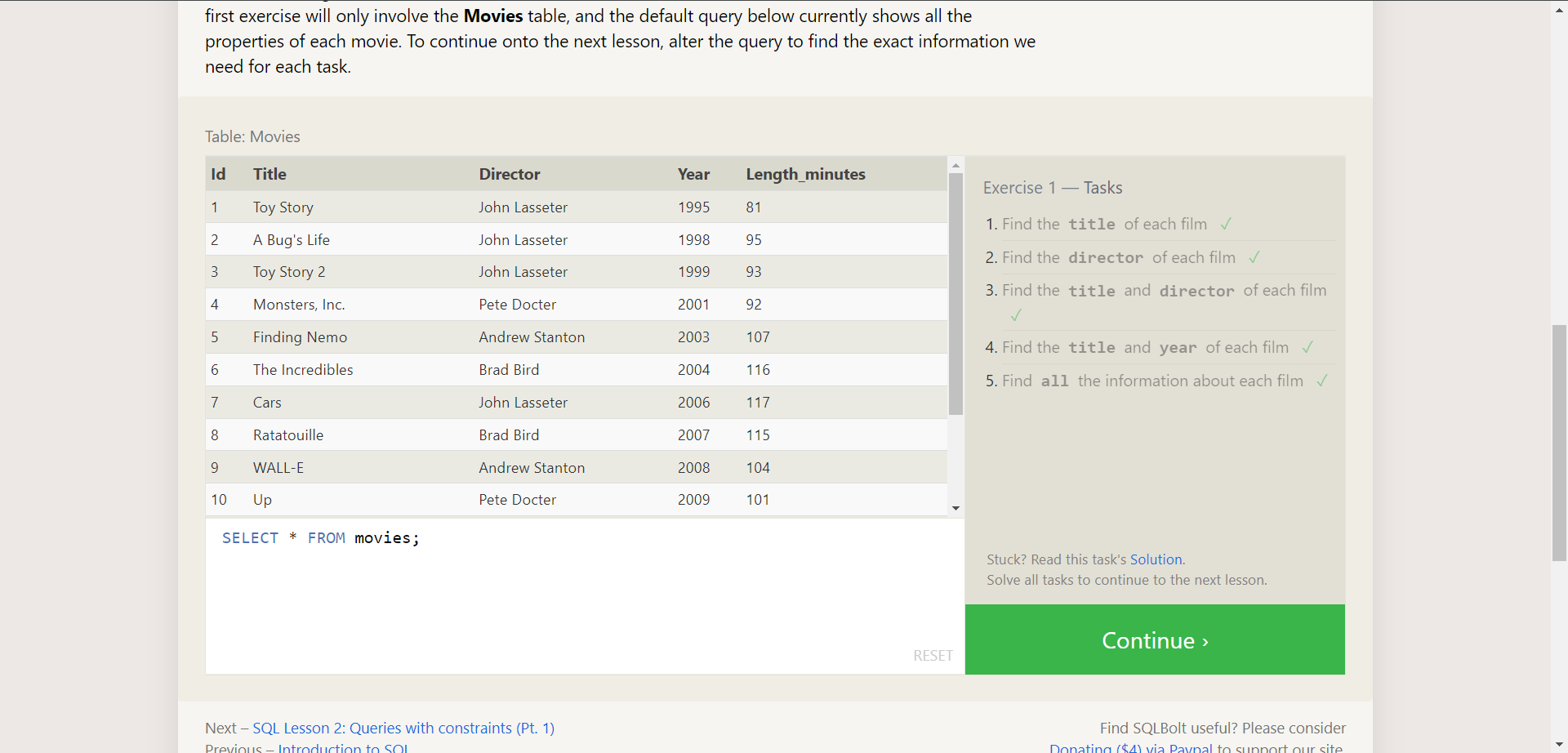
**Question 1** : Select queries 101  


**Answer :**  SELECT title FROM movies;

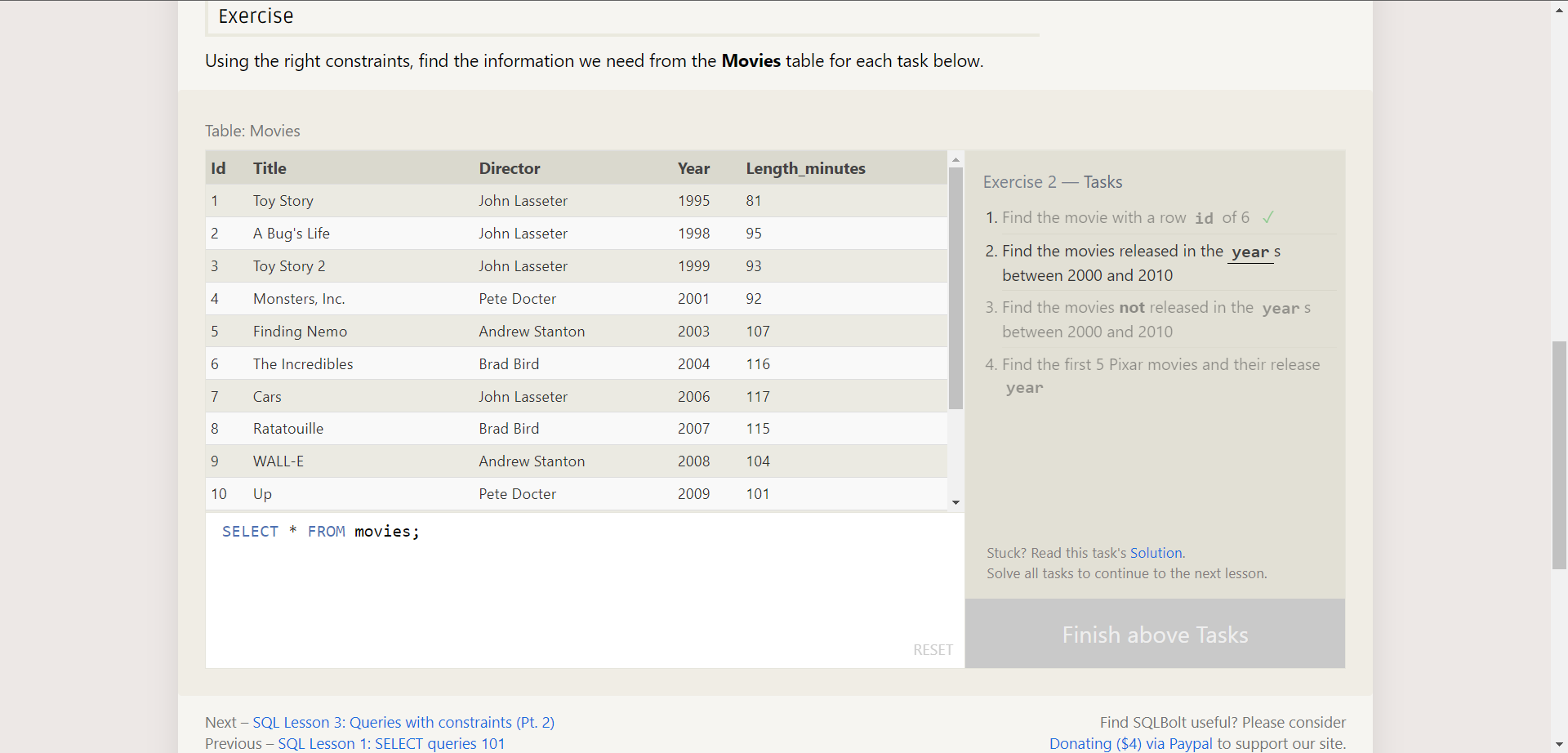
SELECT director FROM movies;

SELECT title,director FROM movies;

SELECT title,year FROM movies;

SELECT \* FROM movies;

-------------------------------------------------------------------------------------------------------------------------------

**Question 2** : Queries with constraints(pt-1)

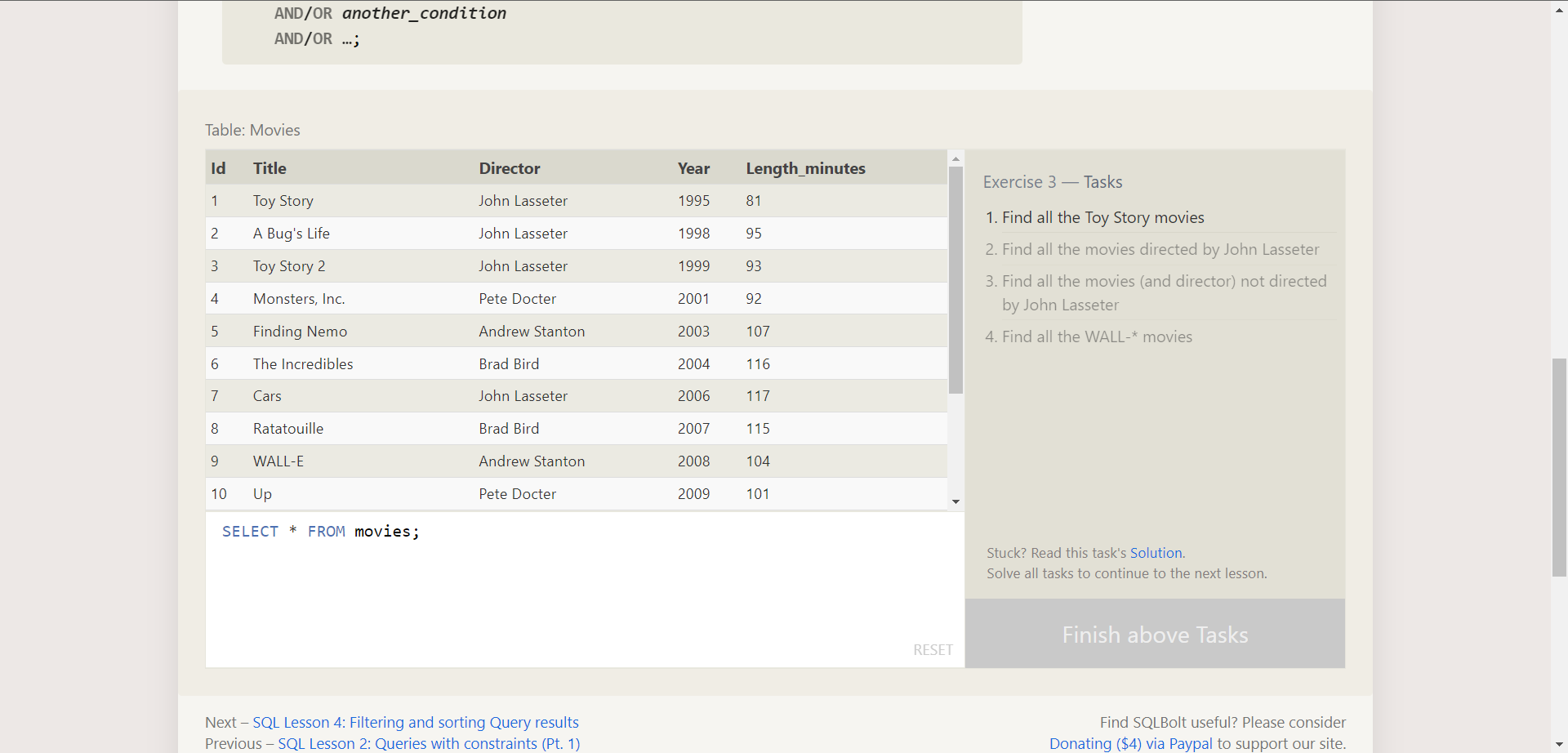
**Answer :** SELECT title FROM movies where id=6;

SELECT title FROM movies where year between 2000 and 2010;

SELECT title, year FROM movies where year < 2000 or year > 2010;

SELECT title, year FROM movies where year <= 2003;

-------------------------------------------------------------------------------------------------------------------------------

**Question 3** : Queries with constraints(pt-2)

**Answer :** SELECT title, director FROM movies where title like "Toy Story%";

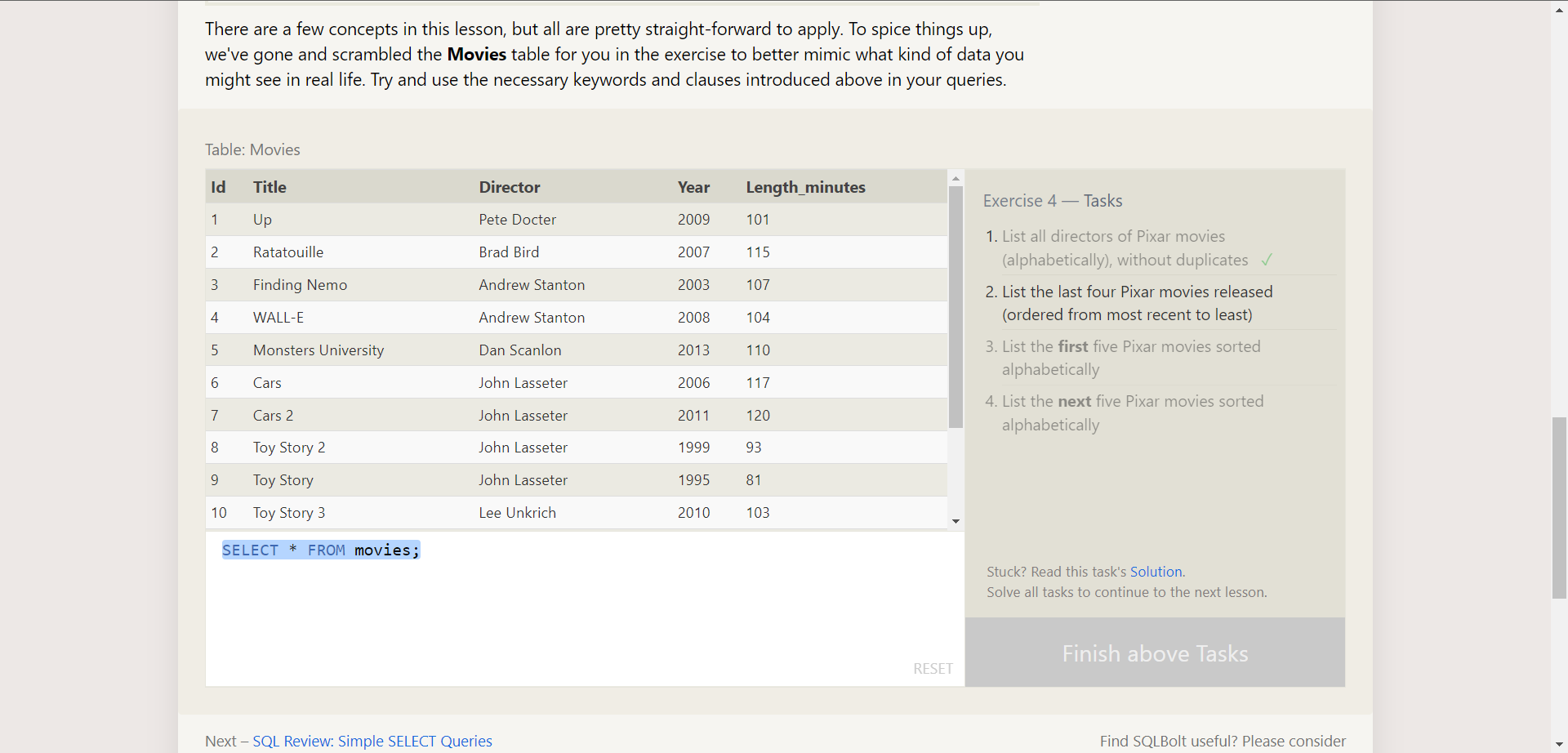
SELECT title FROM movies where director="John Lasseter";

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

SELECT \* FROM movies where title LIKE "WALL-\_";

-------------------------------------------------------------------------------------------------------------------------------

**Question 4** : Filtering & sorting query



**Answer :** SELECT distinct director FROM movies order by director asc ;

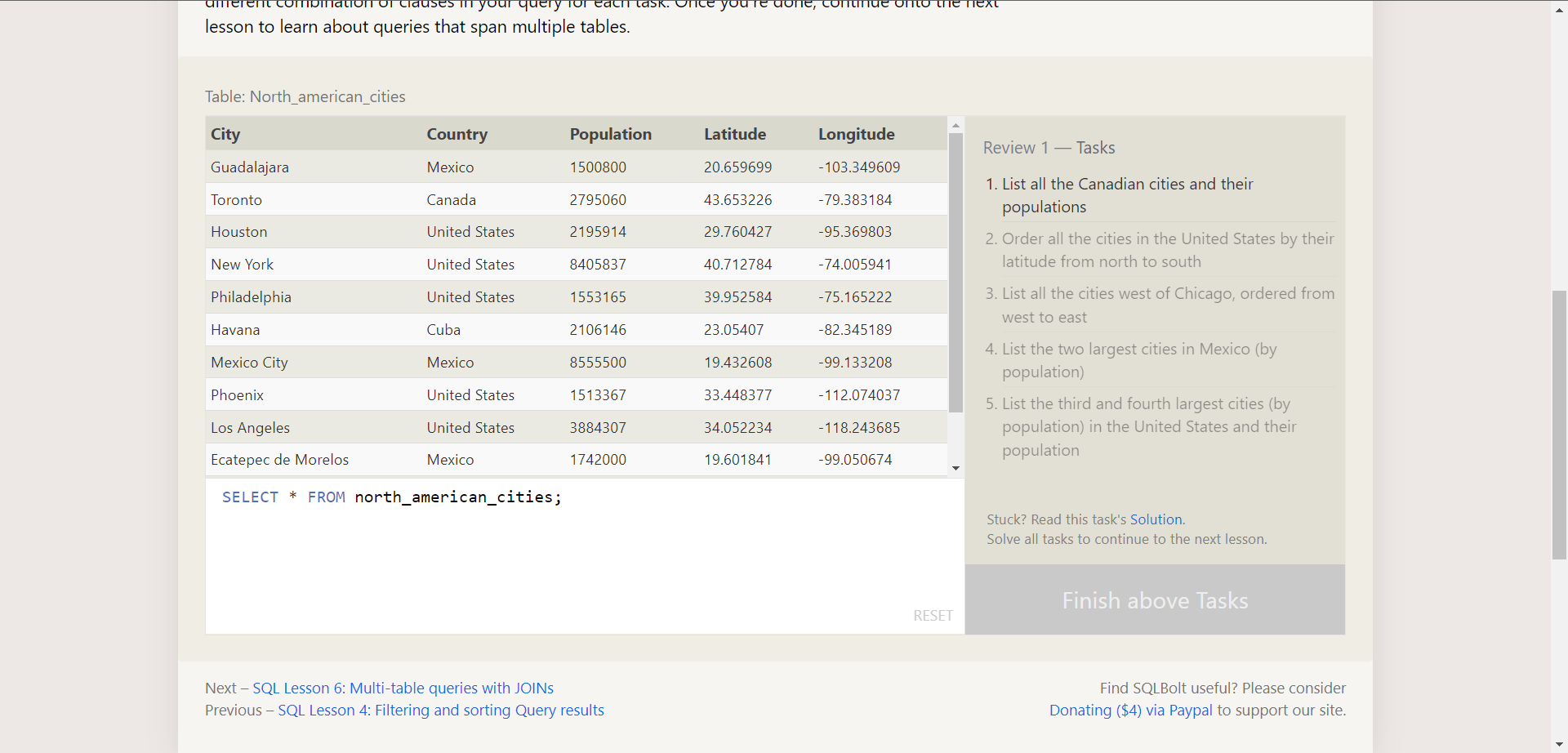
SELECT title,year FROM movies order by year desc limit 4;

SELECT title FROM movies order by title asc limit 5 ;

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

-------------------------------------------------------------------------------------------------------------------------------

**Question 5** : simple select queries



**Answer :** SELECT city, population FROM north\_american\_cities

where country = "Canada";

SELECT city, latitude FROM north\_american\_cities

where country = "United States"

Order by latitude desc;

SELECT city, latitude FROM north\_american\_cities

where longitude < -87.629798

Order by longitude asc;

SELECT city, population FROM north\_american\_cities

where country LIKE "Mexico"

order by population desc limit 2;

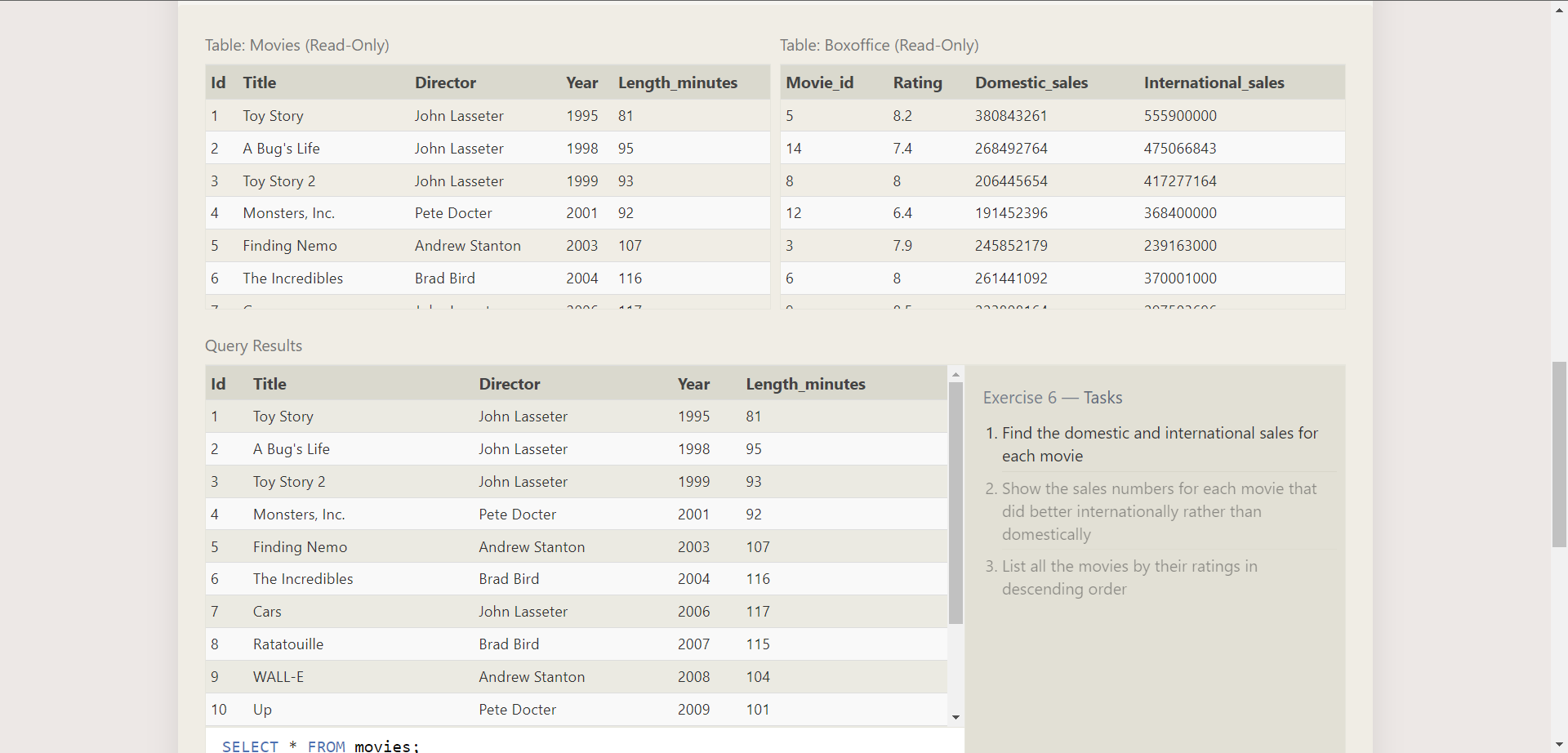
SELECT city, population FROM north\_american\_cities

where country like "United States"

order by population desc limit 2 offset 2;

-------------------------------------------------------------------------------------------------------------------------------

**Question 6** : Multi -table queries with joins



**Answer :** 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;

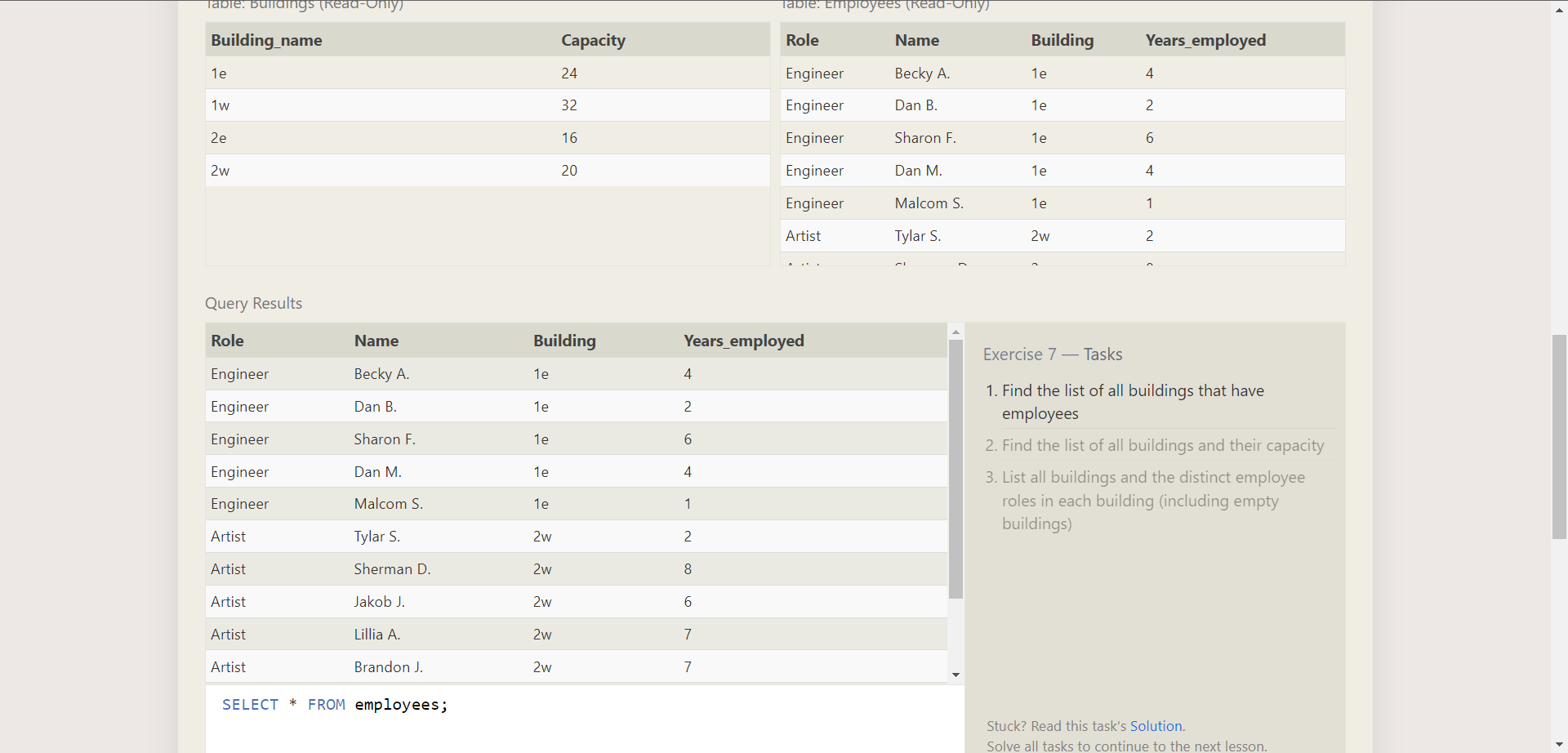
SELECT title, rating FROM movies join boxoffice

on movies.id = boxoffice.movie\_id

order by rating desc;

-------------------------------------------------------------------------------------------------------------------------------

**Question 7** : Outer joins



**Answer :** SELECT distinct building FROM employees;

SELECT \* FROM buildings;

SELECT distinct building\_name, role FROM buildings left join employees

on building\_name = building;

-------------------------------------------------------------------------------------------------------------------------------

**Question 8** : A short notes on null

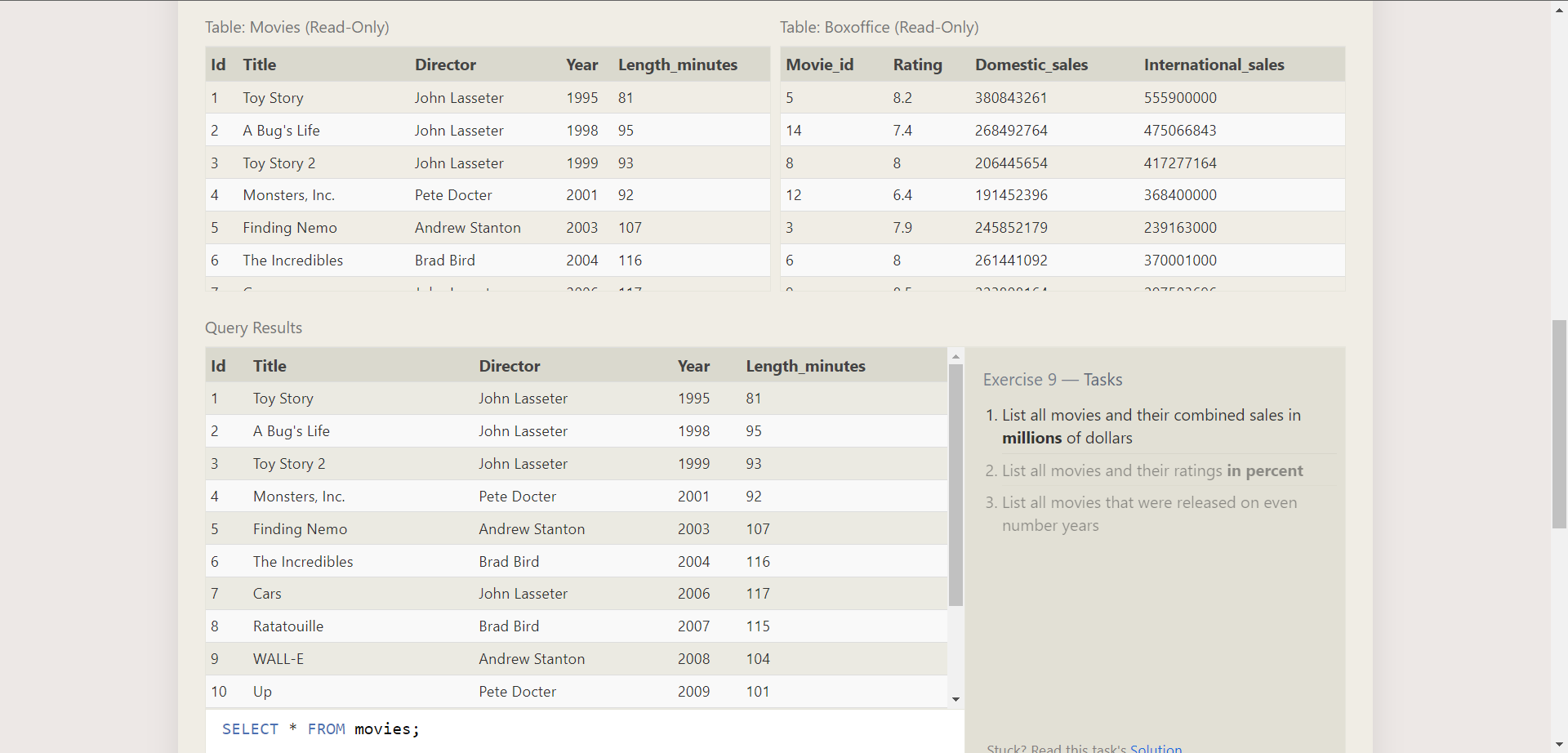
**Answer :** SELECT name, role FROM employees

where building IS null;

SELECT DISTINCT building\_name FROM buildings

left join employees on building\_name = building where role IS null;

-------------------------------------------------------------------------------------------------------------------------------

**Question 9** : Queries with expressions

**Answer :** SELECT title, (domestic\_sales + international\_sales) / 1000000

as gross\_sales\_millions FROM movies

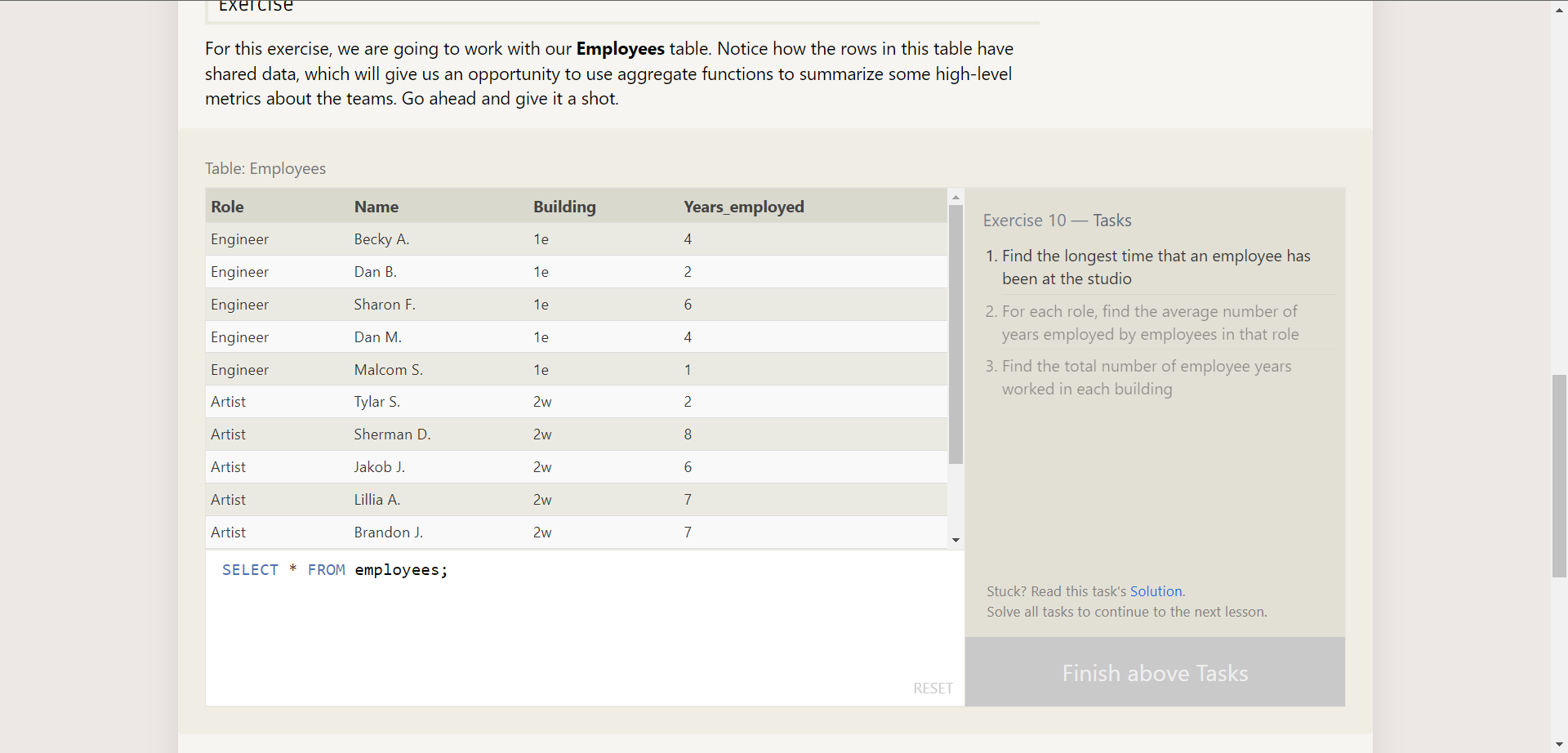
join boxoffice on movies.id = boxoffice.movie\_id;

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;

-------------------------------------------------------------------------------------------------------------------------------

**Question 10** : Queries with aggregates(pt-1)

**Answer :** SELECT max(years\_employed) as Max\_years\_employed

FROM employees;

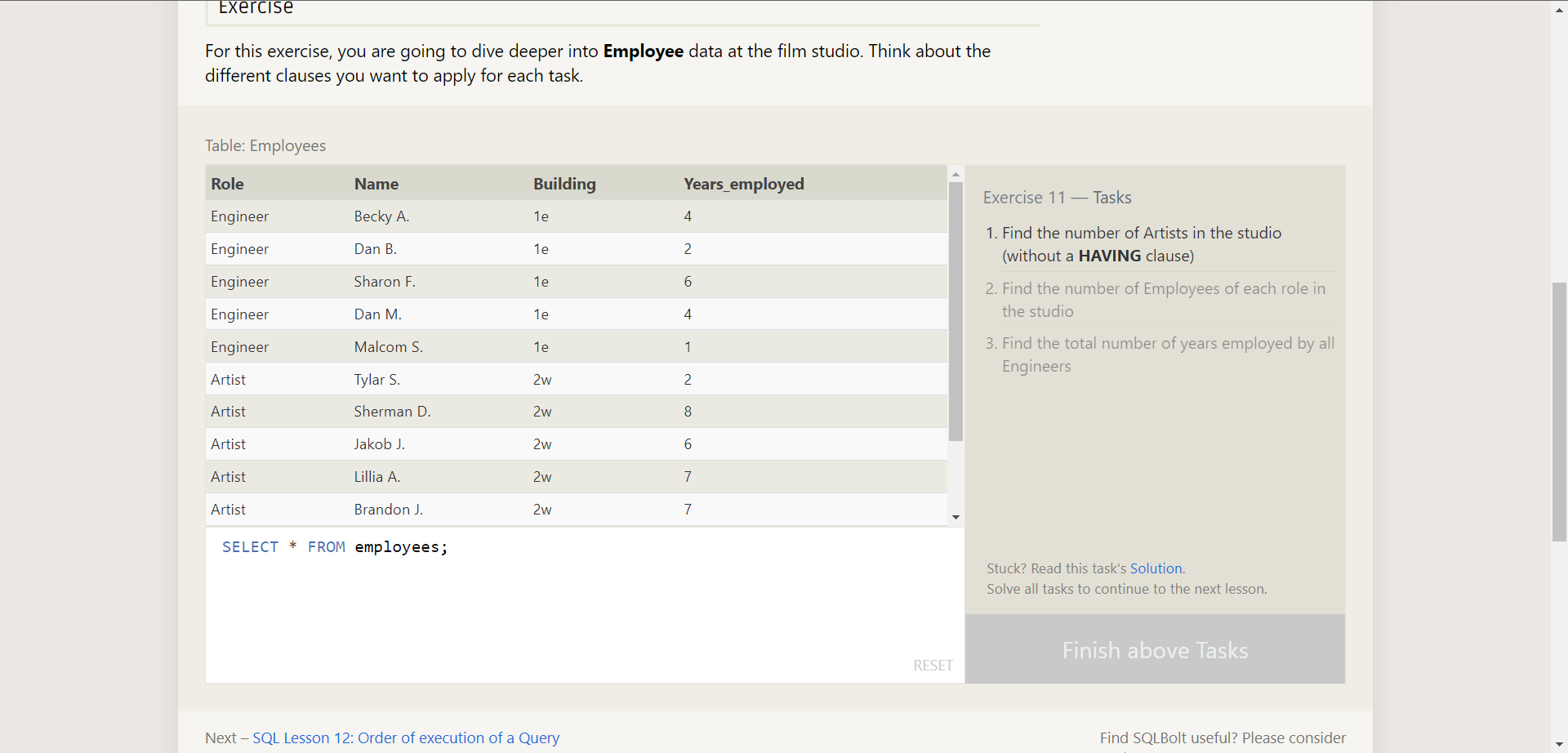
SELECT role, avg(years\_employed) as Average\_years\_employed

FROM employees group by role;

SELECT building, sum(years\_employed) as Total\_years\_employed

FROM employees group by building;

-------------------------------------------------------------------------------------------------------------------------------

**Question 11** : Queries with aggregates(pt-2)

**Answer :** SELECT role, count(\*) as Number\_of\_artists FROM employees

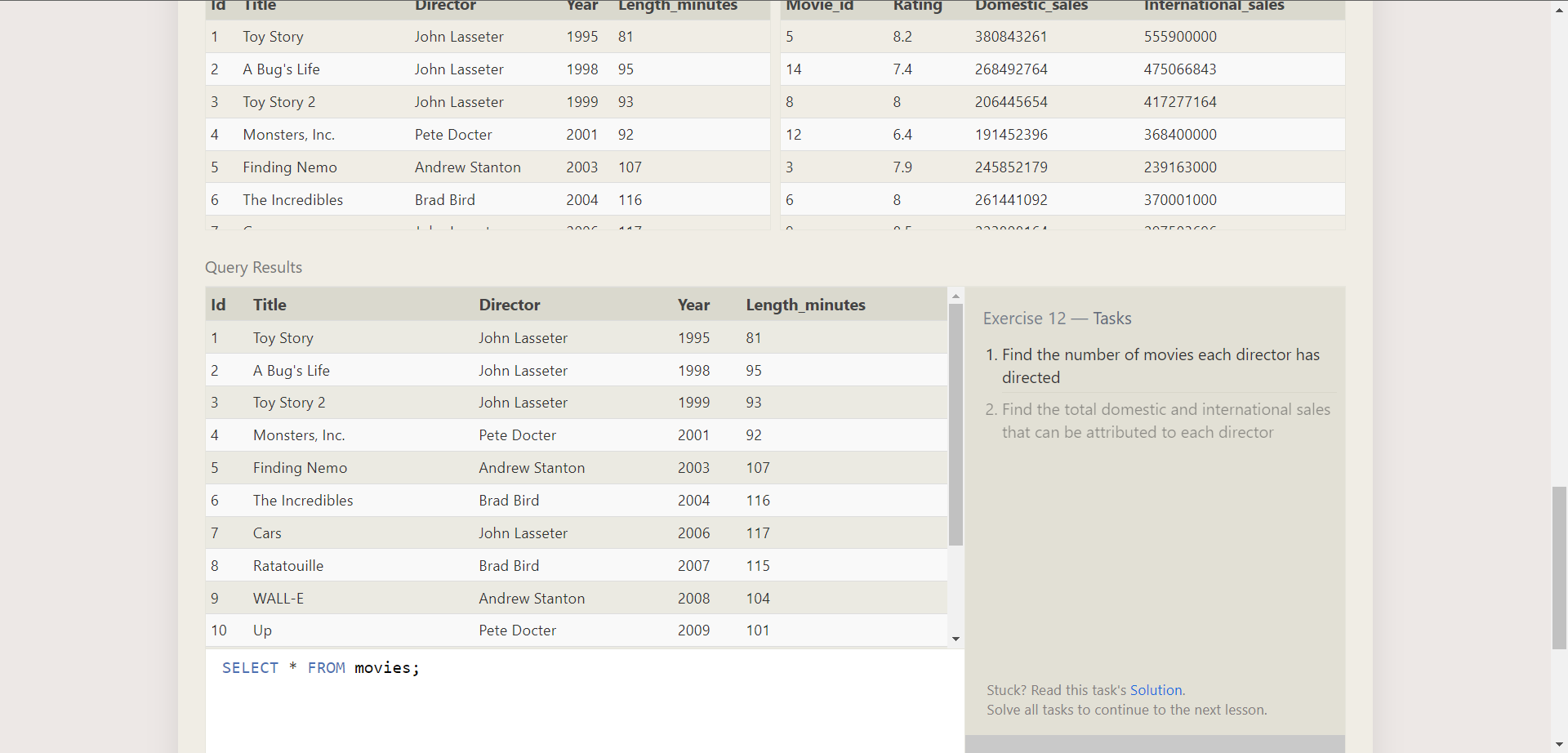
where role = "Artist";

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

SELECT role, sum(years\_employed) FROM employees

group by role having role = "Engineer";

-------------------------------------------------------------------------------------------------------------------------------

**Question 12** : Order of execution of a query

**Answer :** SELECT director, COUNT(id) as Num\_movies\_directed

FROM movies group by director;

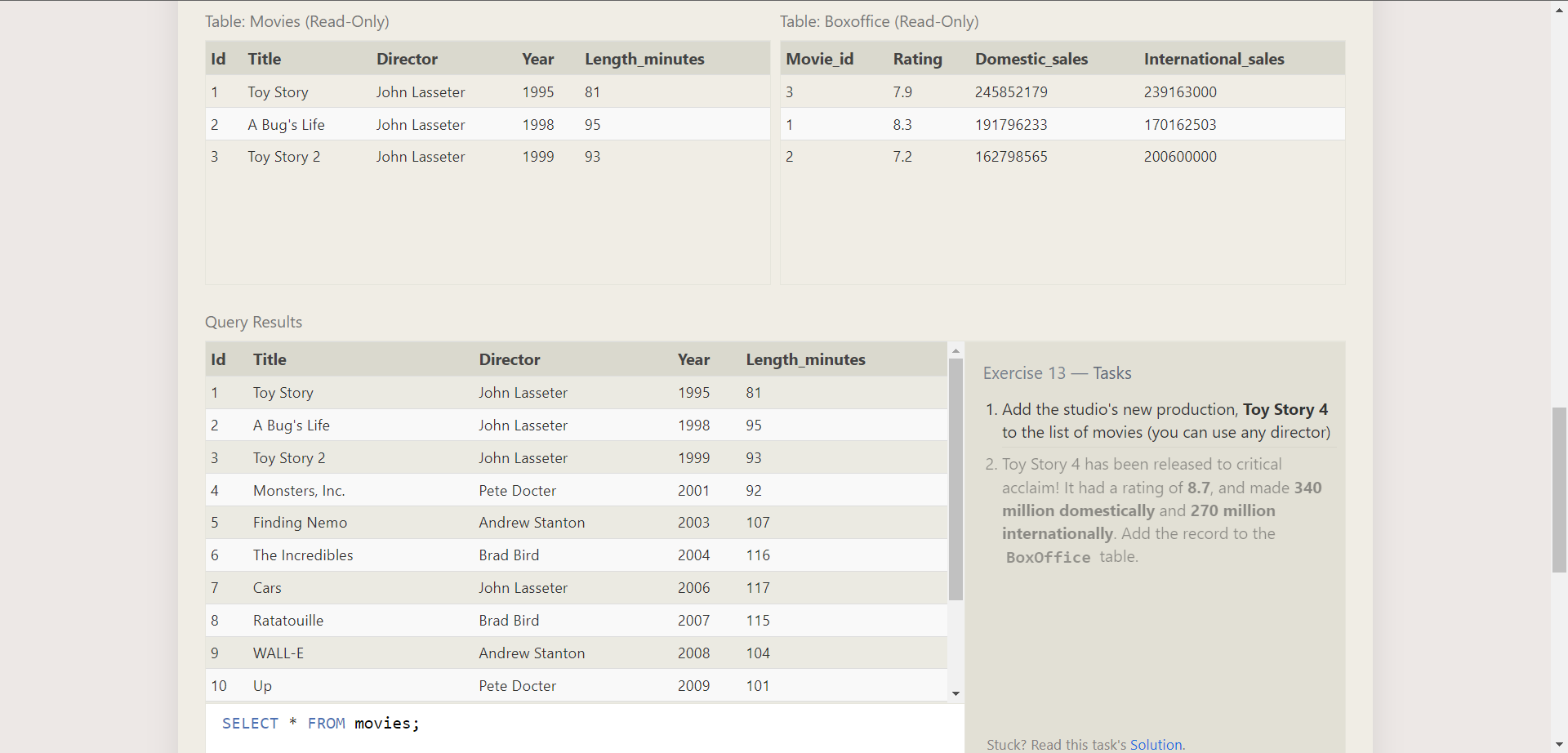
SELECT director, sum(domestic\_sales + international\_sales) as

Cumulative\_sales\_from\_all\_movies FROM movies

inner join boxoffice on

movies.id = boxoffice.movie\_id group by director;

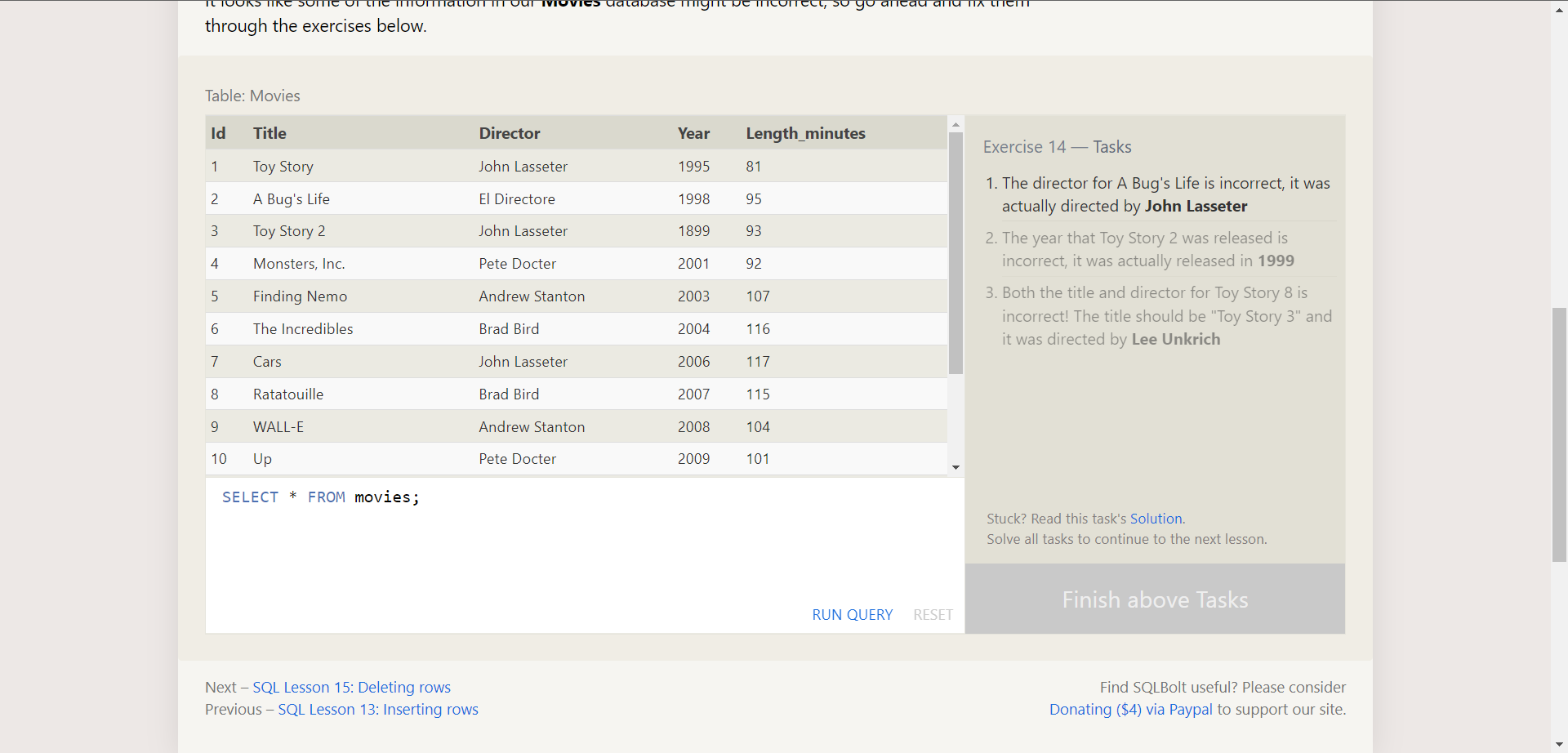
-------------------------------------------------------------------------------------------------------------------------------

**Question 13** : Inserting rows

**Answer :** INSERT INTO movies values (4, "Toy Story 4", "El Directore", 2015, 90);

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

-------------------------------------------------------------------------------------------------------------------------------

**Question 14** : Updating rows

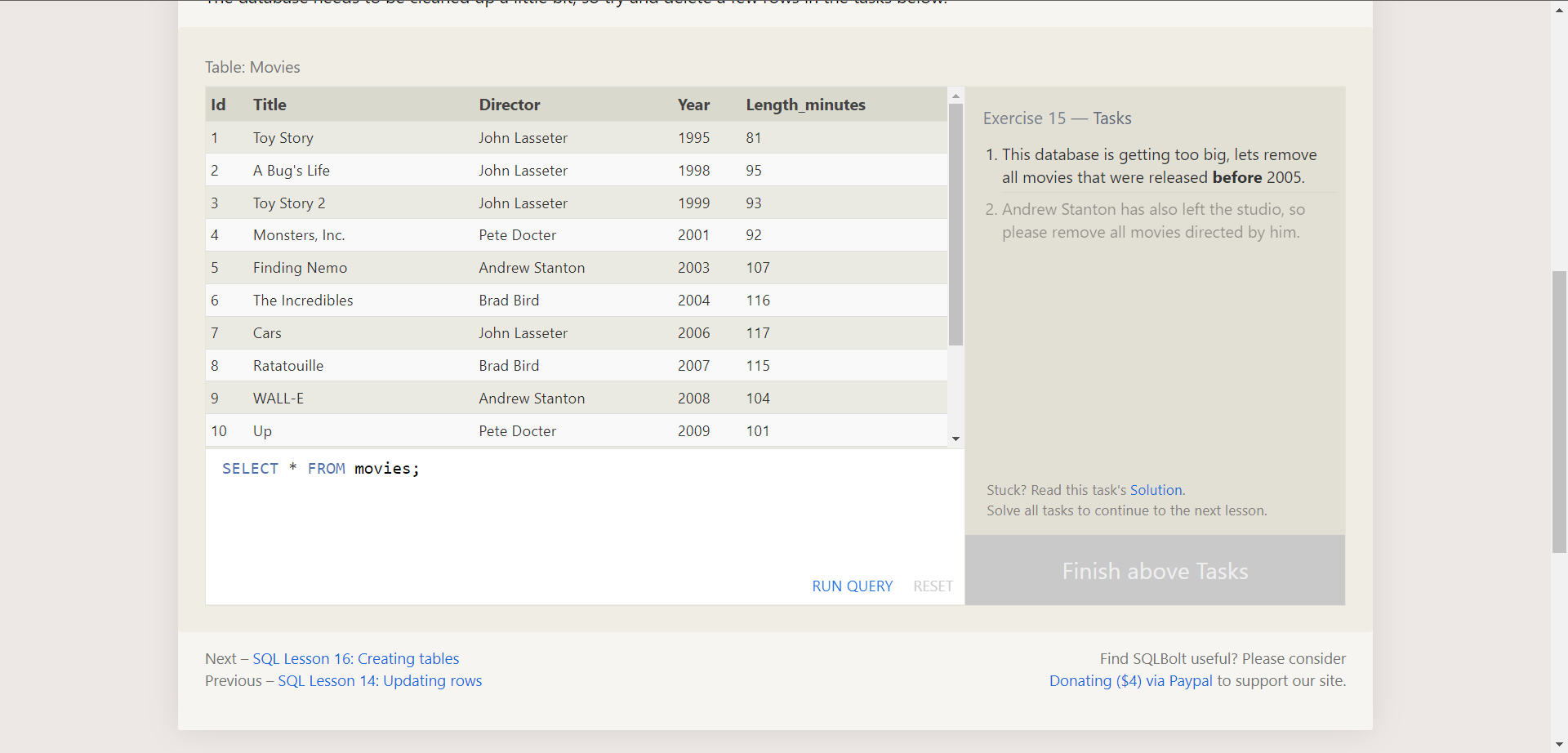
**Answer :** UPDATE movies set director = "John Lasseter" where id = 2;

UPDATE movies set year = 1999 where id = 3;

UPDATE movies set year = 1999 title = "Toy Story 3",

director = "Lee Unkrich" where id = 11;

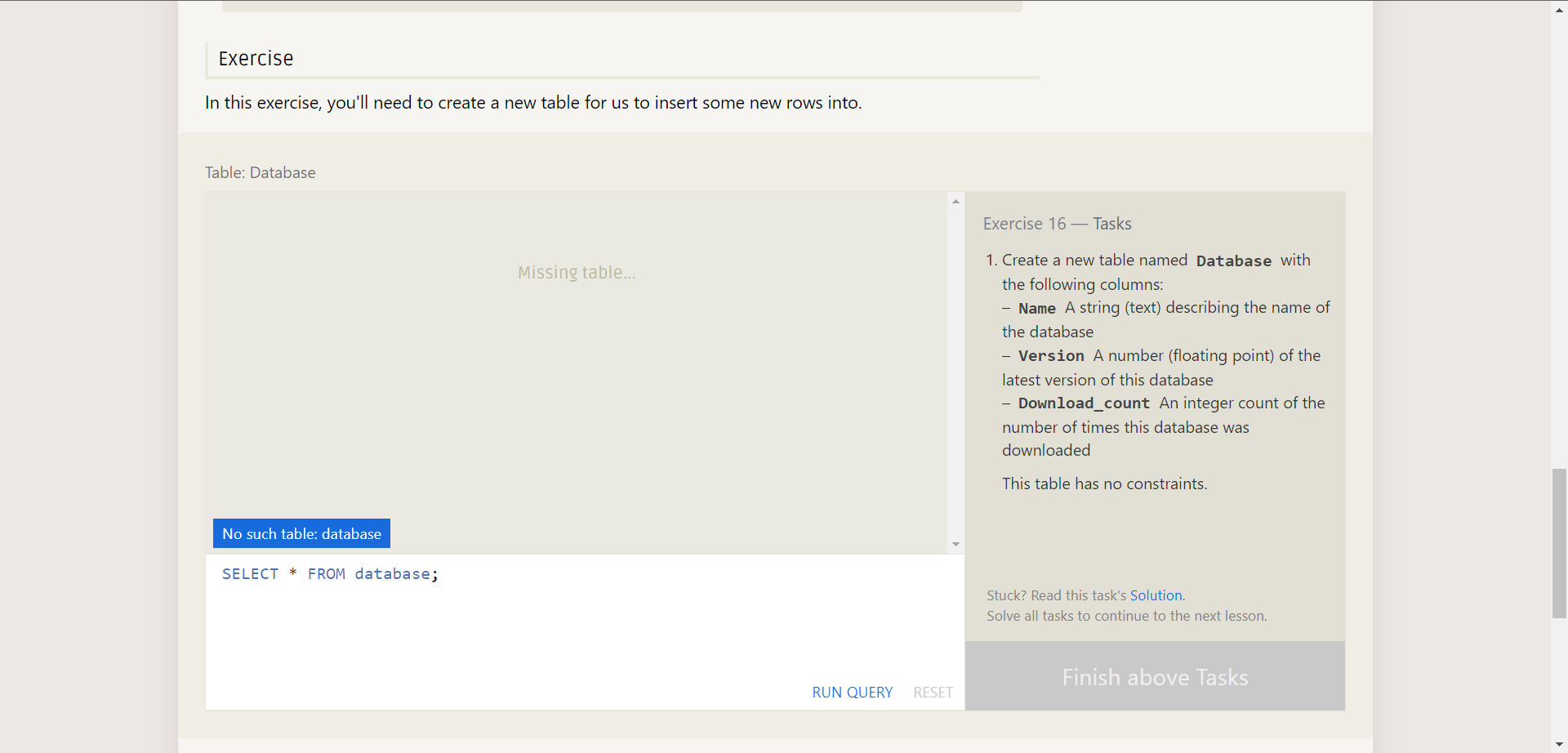
-------------------------------------------------------------------------------------------------------------------------------

**Question 15** : Deleting rows

**Answer :** DELETE FROM movies where year < 2005;

DELETE FROM movies where director = "Andrew Stanton";

-------------------------------------------------------------------------------------------------------------------------------

**Question 16** : Creating tables

**Answer :** CREATE TABLE Database (

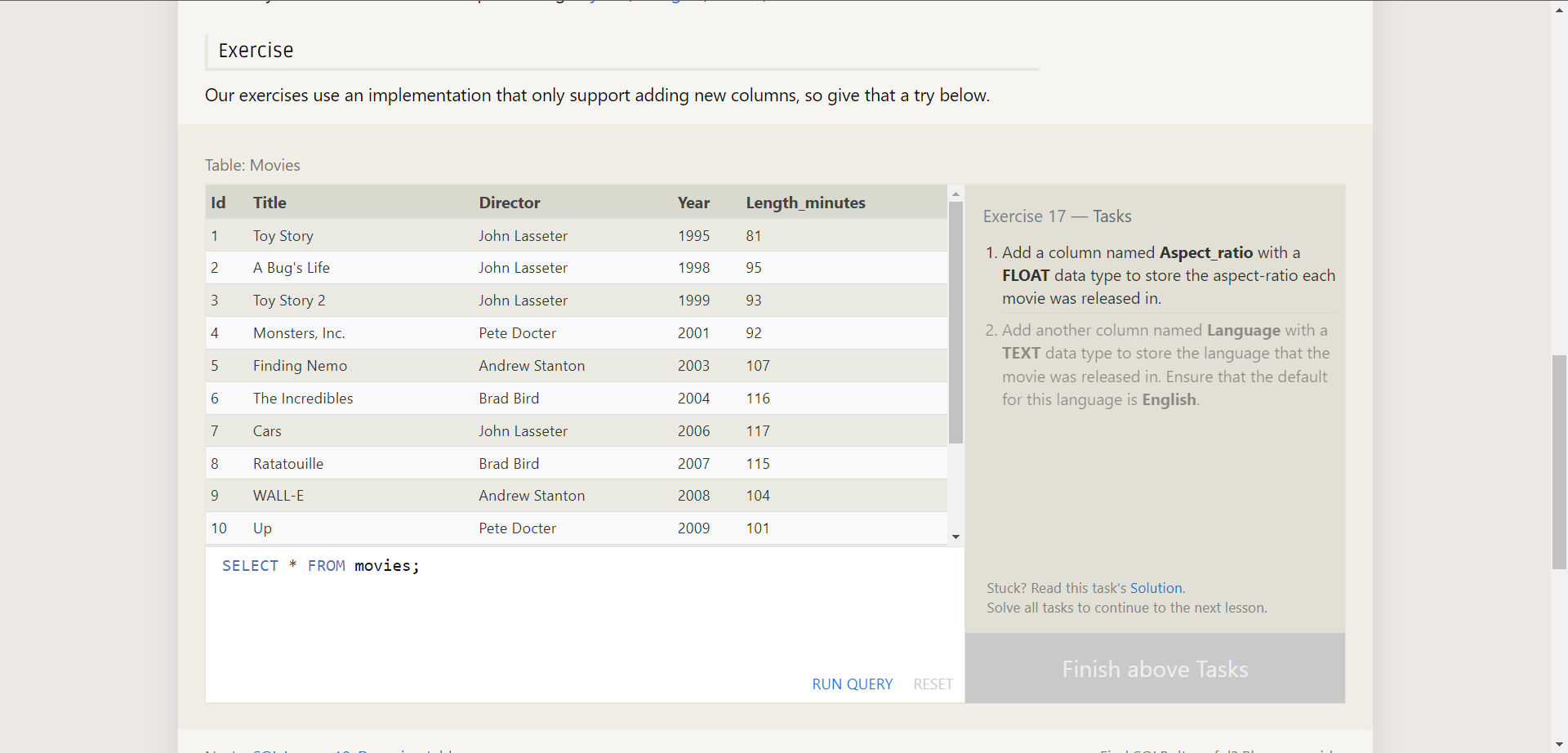
Name TEXT,

Version FLOAT,

Download\_count INTEGER

);

-------------------------------------------------------------------------------------------------------------------------------

**Question 17** : Altering tables

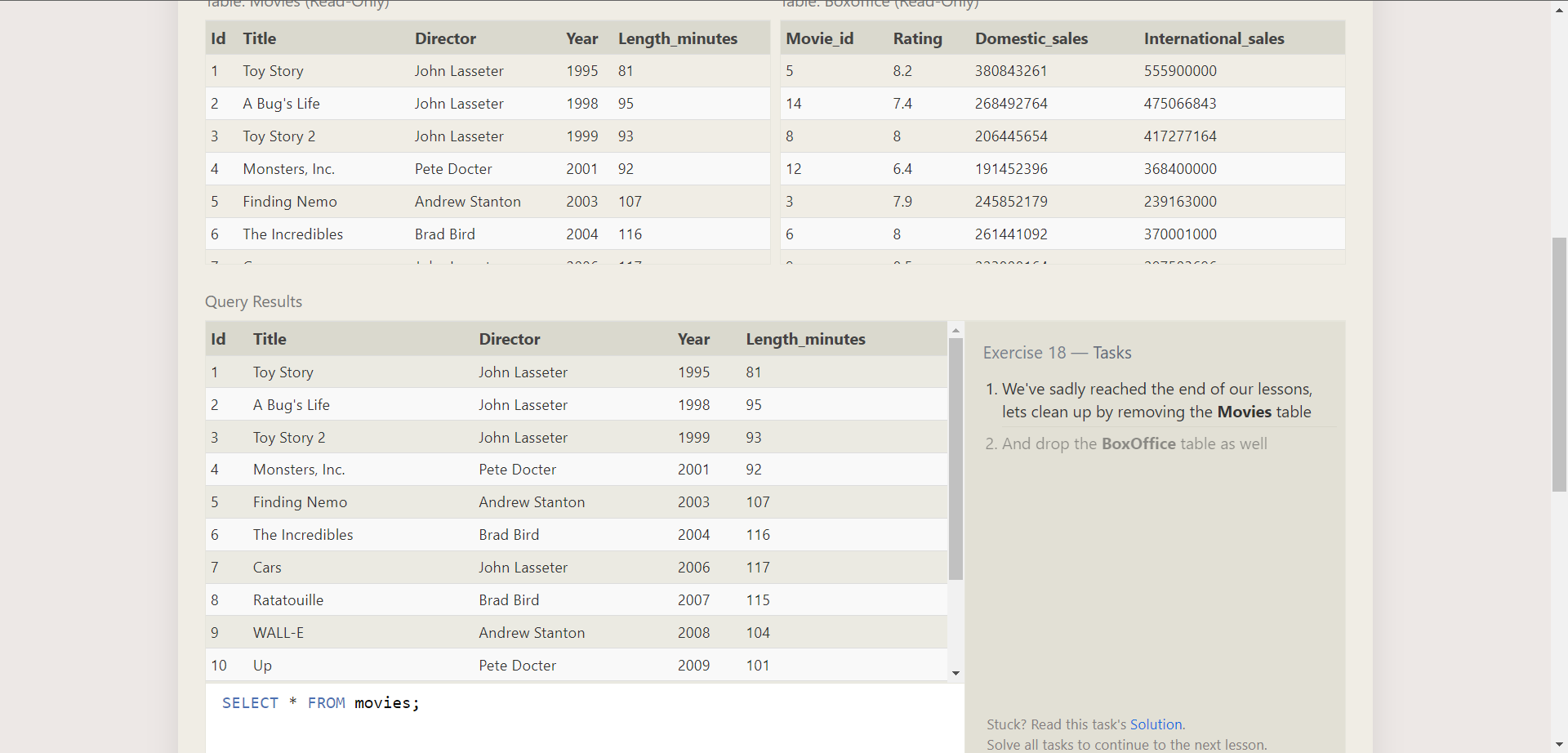
**Answer :** ALTER TABLE Movies ADD COLUMN Aspect\_ratio

FLOAT DEFAULT 2.39;

ALTER TABLE Movies ADD COLUMN Language

TEXT DEFAULT "English";

-------------------------------------------------------------------------------------------------------------------------------

**Question 18** : Dropping tables

**Answer :** DROP TABLE Movies;

DROP TABLE BoxOffice;

-------------------------------------------------------------------------------------------------------------------------------