SQL Lesson 1: SELECT queries 101

Exercise 1 — Tasks

1. Find the title of each film ✓

ANS: SELECT Title FROM movies;

1. Find the director of each film ✓

ANS : SELECT Director FROM movies ;

1. Find the title and director of each film ✓

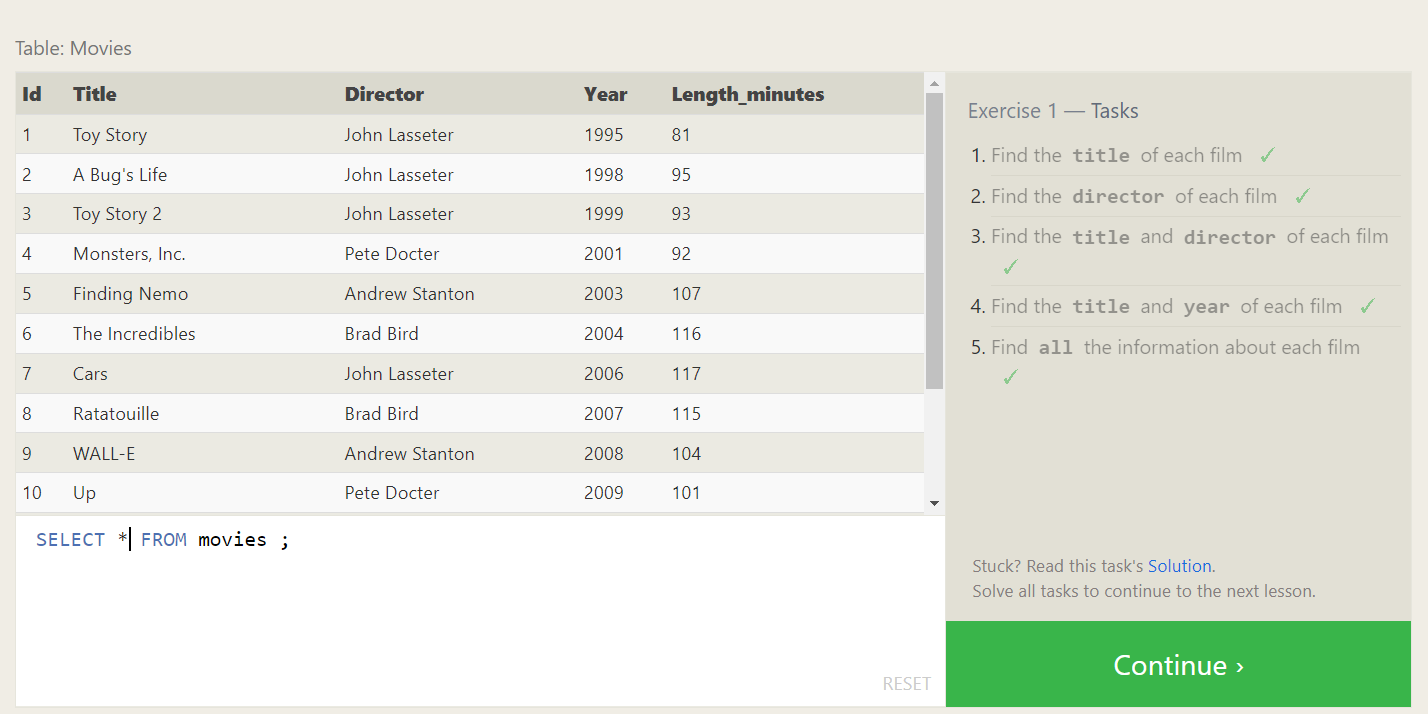
ANS : SELECT Title, Director FROM movies ;

1. Find the title and year of each film ✓

ANS : SELECT Title, Year FROM movies ;

1. Find all the information about each film ✓

ANS : SELECT \* FROM movies ;



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

Exercise 2 — Tasks

1. Find the movie with a row id of 6 ✓

ANS: SELECT \* from Movies where id = 6;

1. Find the movies released in the years between 2000 and 2010 ✓

ANS: SELECT \* from Movies where Year <= 2010 and Year >= 2000;

1. Find the movies not released in the years between 2000 and 2010 ✓

ANS: SELECT \* from Movies where Year > 2010 or Year < 2000;

1. Find the first 5 Pixar movies and their release year ✓

ANS: SELECT YEAR from Movies where Id >= 1 and Id<=5 ;



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

Exercise 3 — Tasks

1. Find all the Toy Story movies ✓

ANS: SELECT \* FROM movies where title like 'toy%'

1. Find all the movies directed by John Lasseter ✓

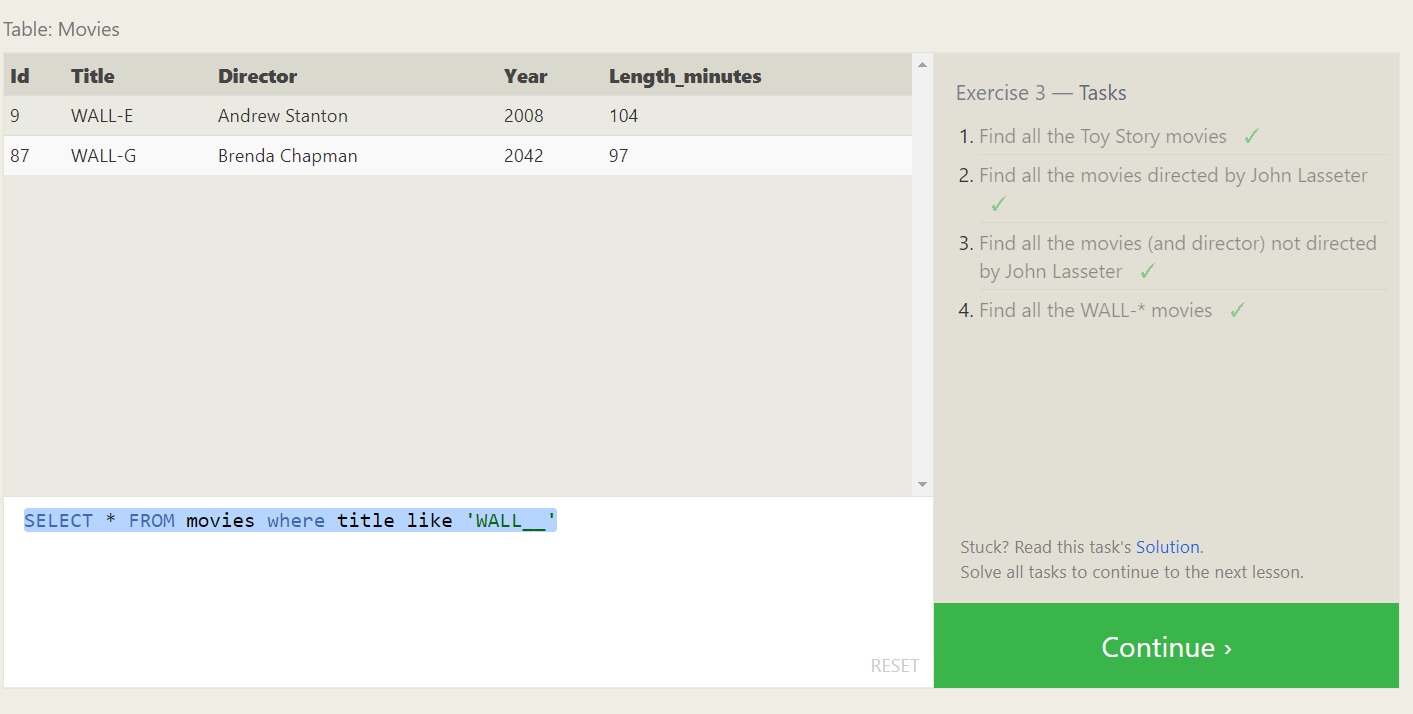
ANS: SELECT \* FROM movies where director like 'john%'

1. Find all the movies (and director) not directed by John Lasseter ✓

ANS: SELECT \* FROM movies where director not like 'john%'

1. Find all the WALL-\* movies ✓

ANS: SELECT \* FROM movies where title like 'WALL\_\_'



SQL Lesson 4: Filtering and sorting Query results

Exercise 4 — Tasks

1. List all directors of Pixar movies (alphabetically), without duplicates ✓

ANS: SELECT distinct director FROM movies order by director

1. List the last four Pixar movies released (ordered from most recent to least) ✓

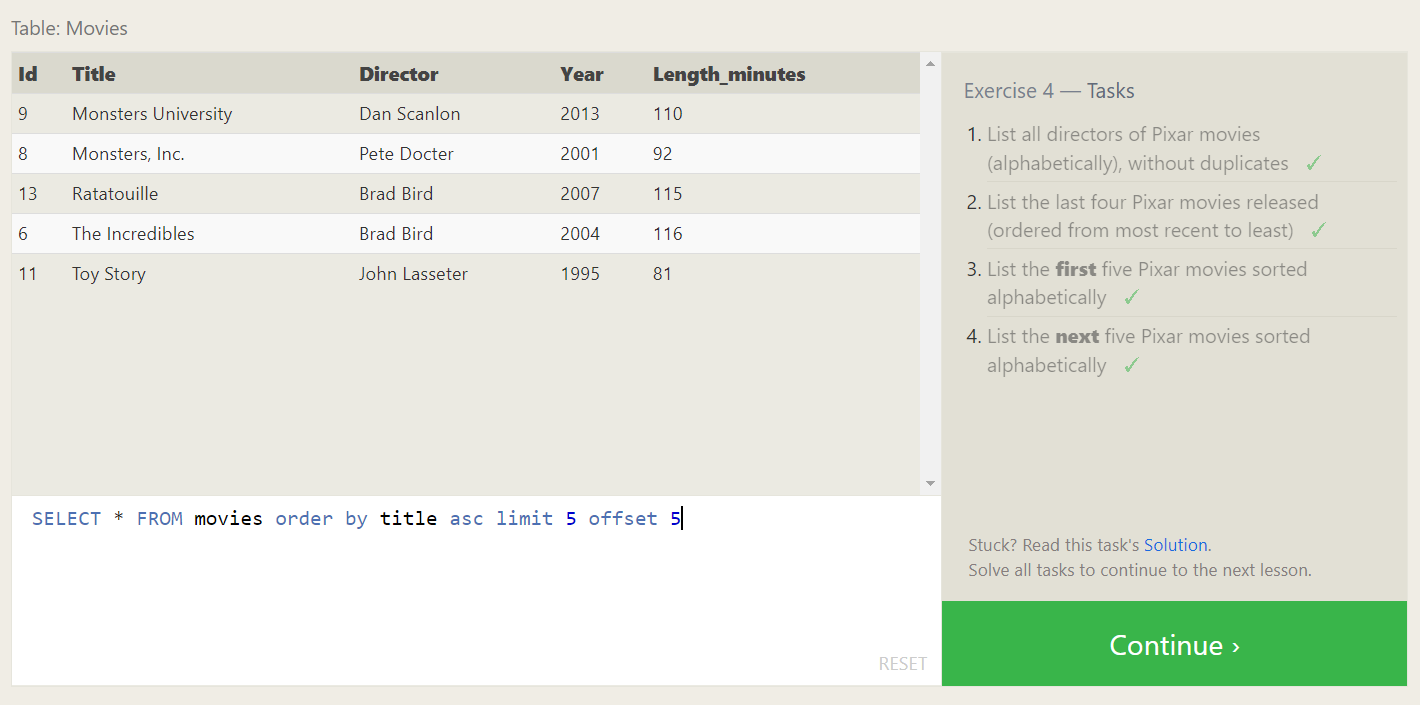
ANS: SELECT \* FROM movies order by year desc limit 4

1. List the first five Pixar movies sorted alphabetically ✓

ANS: SELECT \* FROM movies order by title asc limit 5

1. List the next five Pixar movies sorted alphabetically ✓

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



**SQL Review: Simple SELECT Queries**

Review 1 — Tasks

1. List all the Canadian cities and their populations

ANS: SELECT city,population FROM north\_american\_cities where country = 'Canada';

1. Order all the cities in the United States by their latitude from north to south

ANS: select \* from North\_american\_cities where country = 'United States' order by latitude desc

1. List all the cities west of Chicago, ordered from west to east

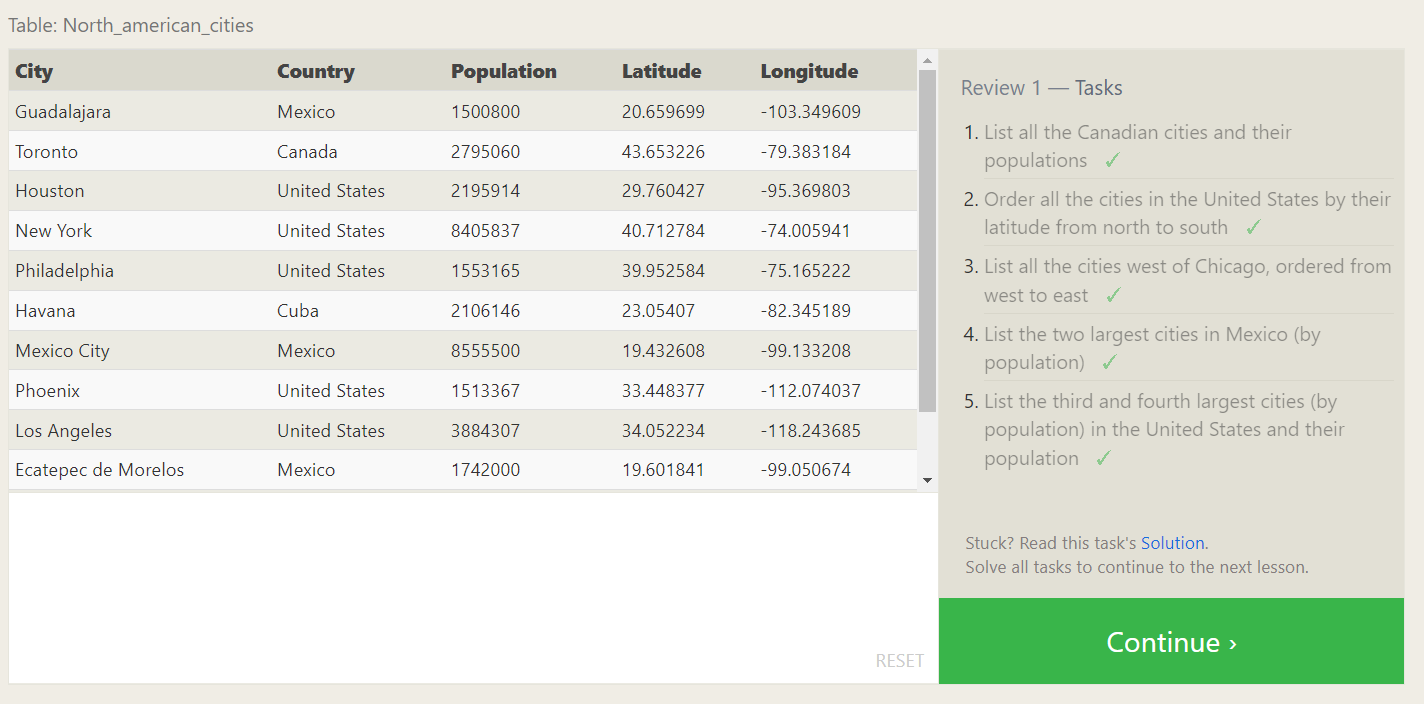
ANS : select \* from North\_american\_cities where longitude <-87.629798 order by longitude asc

1. List the two largest cities in Mexico (by population)

ANS: select \* from North\_american\_cities where country = 'Mexico' order by population desc limit 2

1. List the third and fourth largest cities (by population) in the United States and their population

ANS: 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**

Exercise 6 — Tasks

1. Find the domestic and international sales for each movie

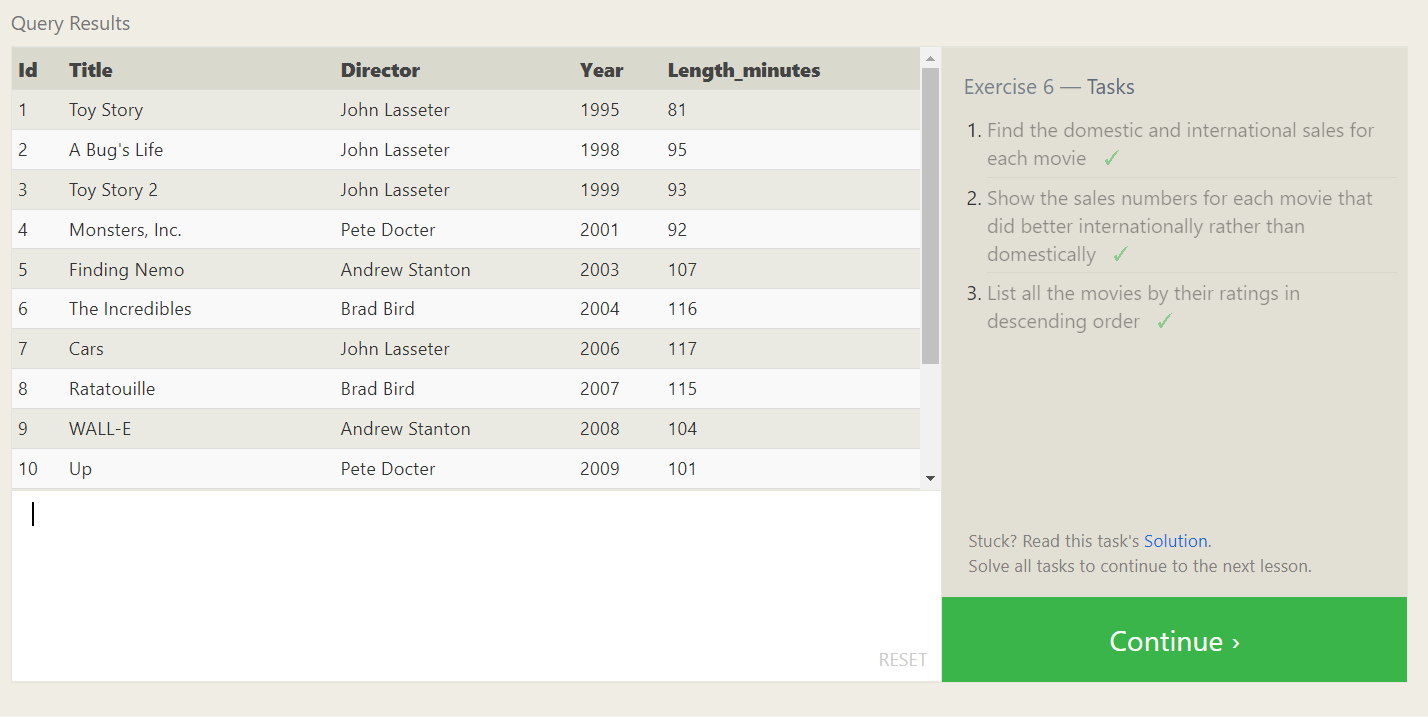
ANS: SELECT \* FROM movies m join boxoffice b on m.id = b.movie\_id;

1. Show the sales numbers for each movie that did better internationally rather than domestically

ANS: SELECT \* FROM movies m join boxoffice b on m.id = b.movie\_id where b.international\_sales > b. domestic\_sales;

1. List all the movies by their ratings in descending order

ANS: SELECT \* FROM movies m join boxoffice b on m.id = b.movie\_id order by b.Rating desc



**SQL Lesson 7: OUTER JOINs**

1. Find the list of all buildings that have employees

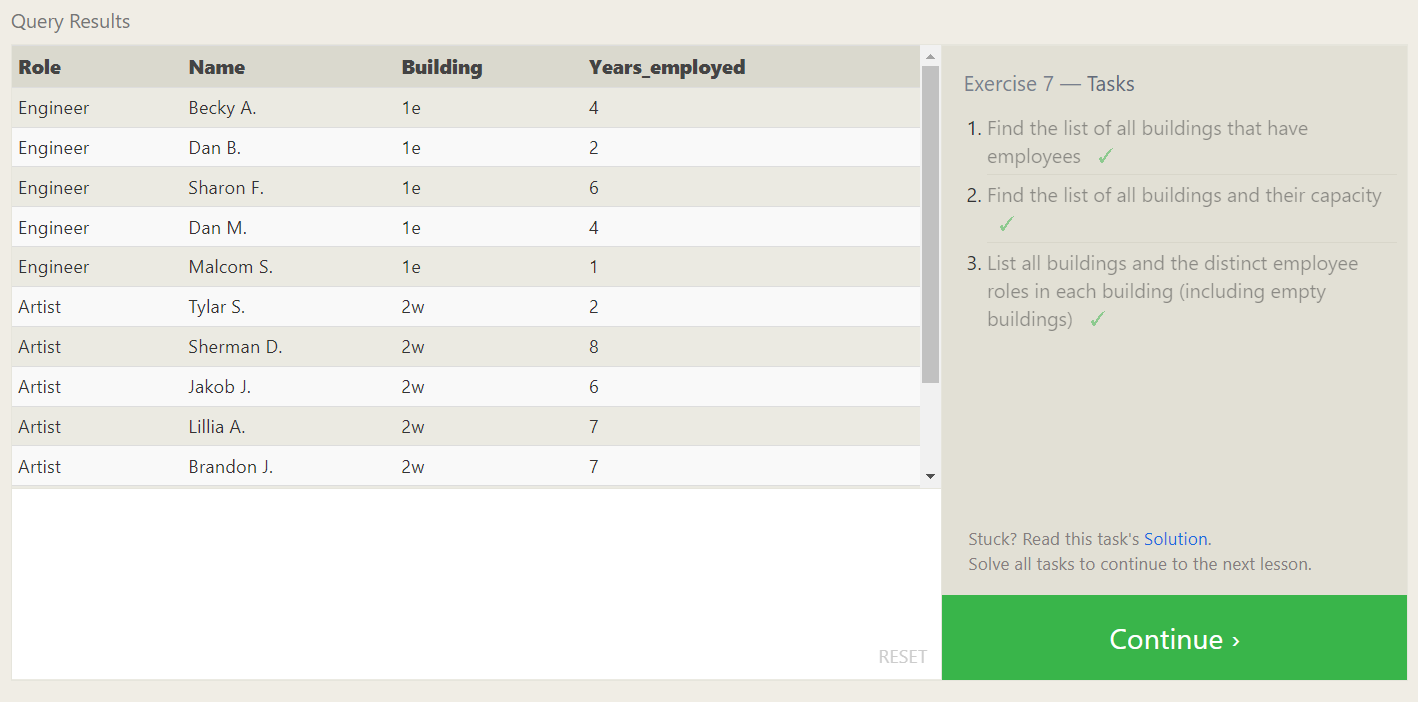
ANS: select distinct building from employees

1. Find the list of all buildings and their capacity

ANS: select \* from buildings

1. List all buildings and the distinct employee roles in each building (including empty buildings)

ANS: select distinct building\_name,role from buildings left join employees on building\_name = building



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

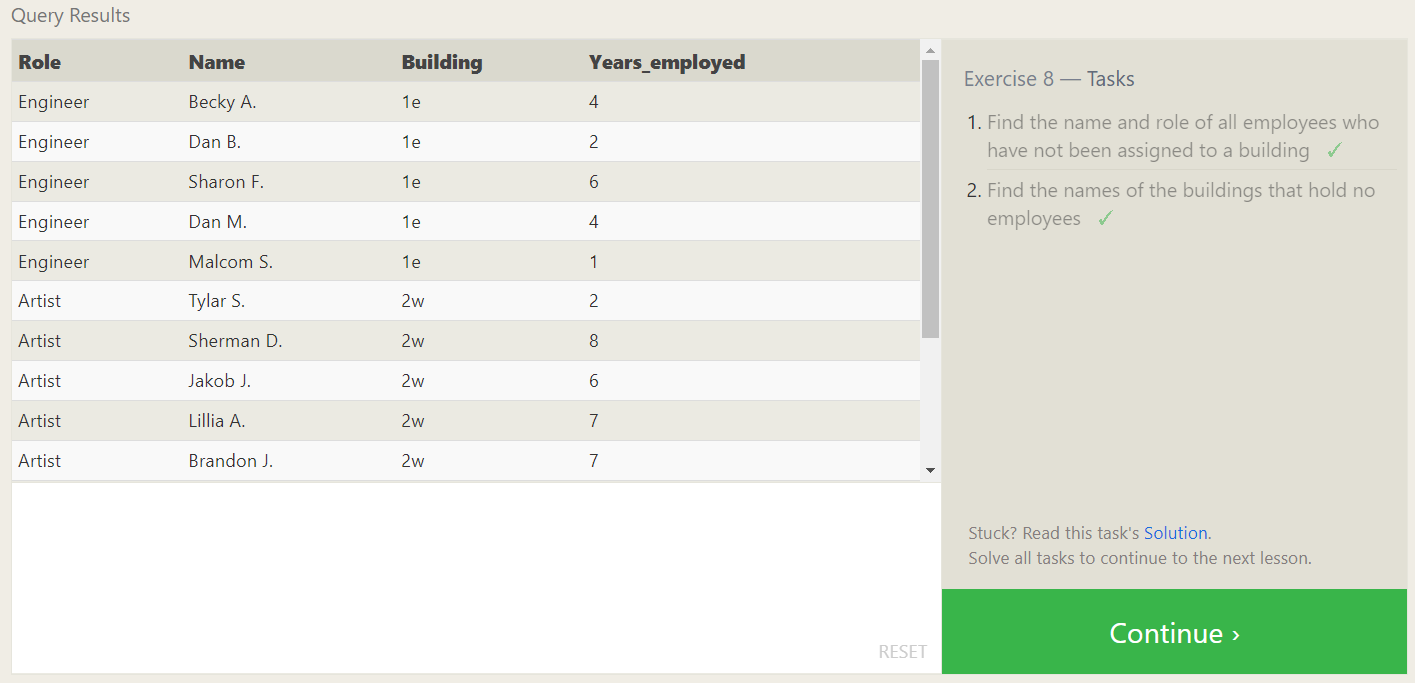
Exercise 8 — Tasks

1. Find the name and role of all employees who have not been assigned to a building

ANS: SELECT \* FROM employees where building is null

1. Find the names of the buildings that hold no employees

ANS: SELECT \* FROM buildings left join employees on building\_name = building where building is null



**SQL Lesson 9: Queries with expressions**

Exercise 9 — Tasks

1. List all movies and their combined sales in **millions** of dollars

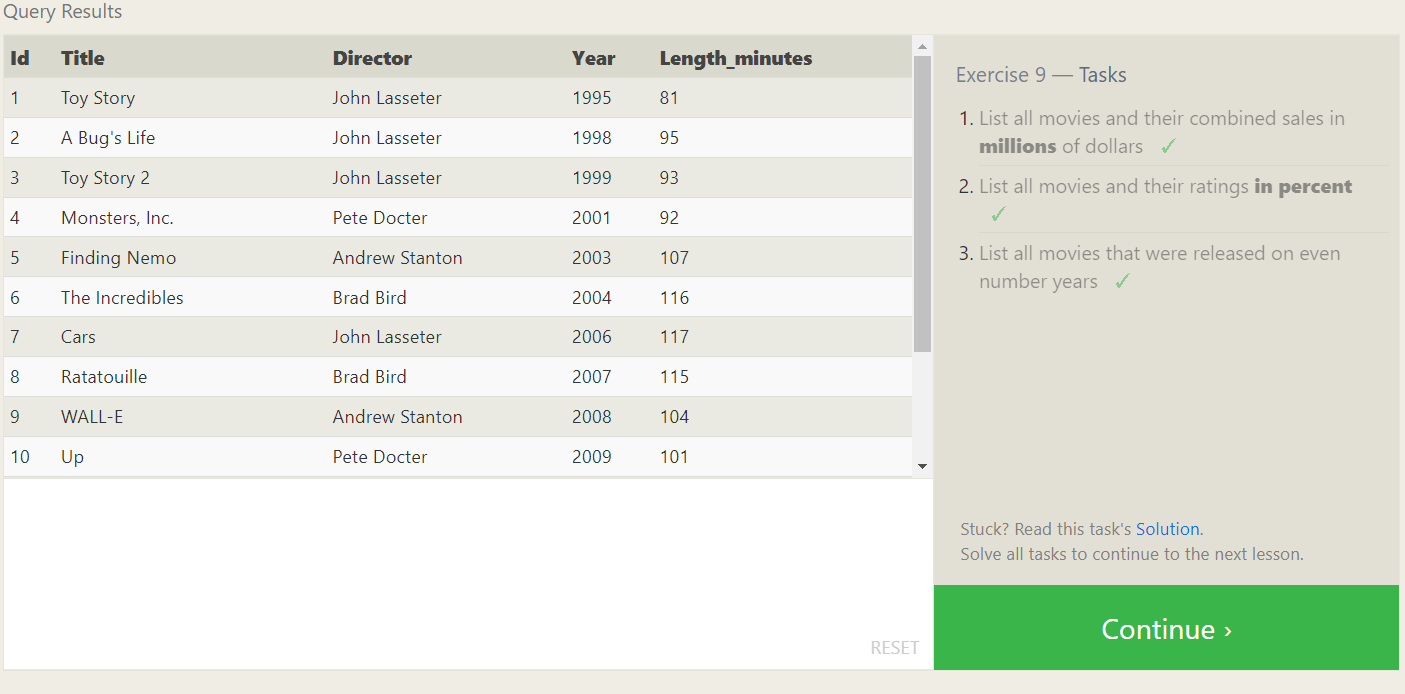
ANS: SELECT title, (domestic\_sales + international\_sales) / 1000000 as sales from movies join boxoffice on movies.id = boxoffice.movie\_id

1. List all movies and their ratings **in percent**

**ANS: select distinct title,(rating\*10) as rating from movies m join boxoffice b on m.id = b.movie\_id**

1. List all movies that were released on even number years

ANS: select \* from movies where year%2=0



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

Exercise 10 — Tasks

1. Find the longest time that an employee has been at the studio

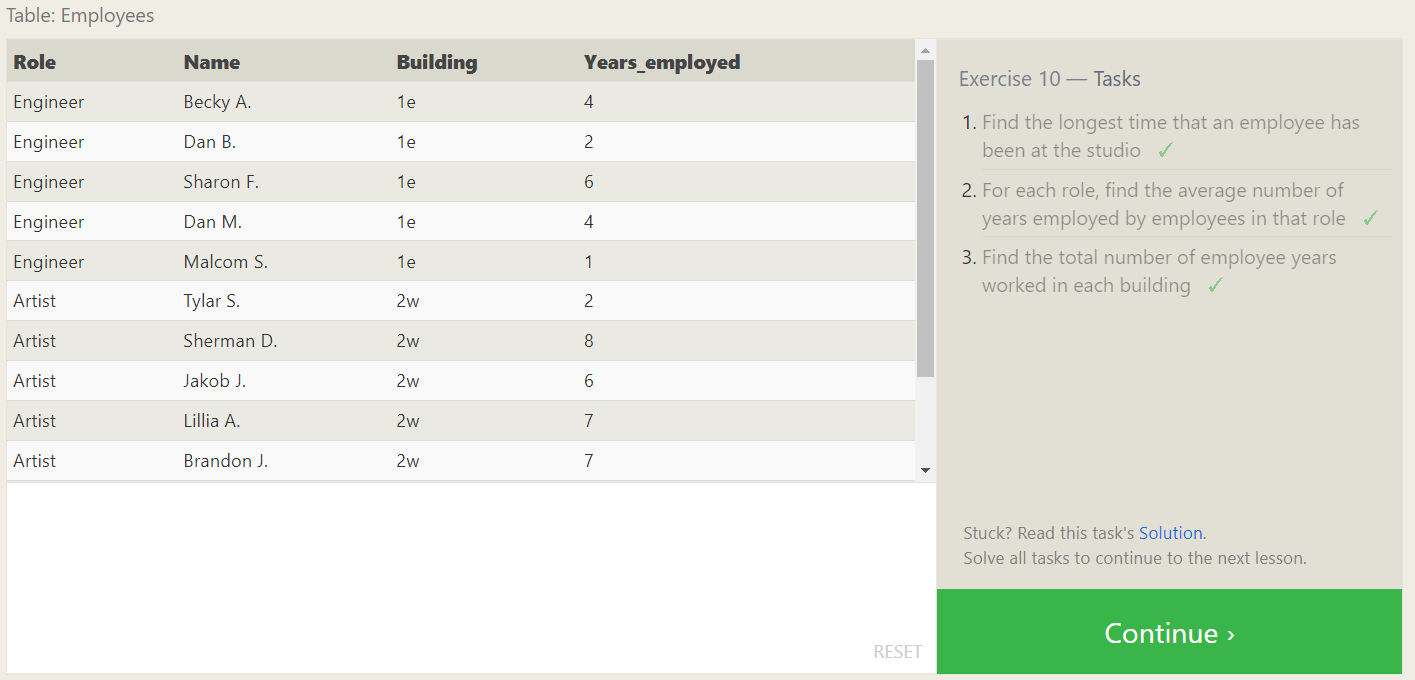
ANS: SELECT max(years\_employed) FROM employees

1. For each role, find the average number of years employed by employees in that role

ANS: select role , avg(years\_employed) from employees group by role

1. Find the total number of employee years worked in each building

ANS: select building, sum(years\_employed) from employees group by building



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

Exercise 11 — Tasks

1. Find the number of Artists in the studio (without a **HAVING** clause)

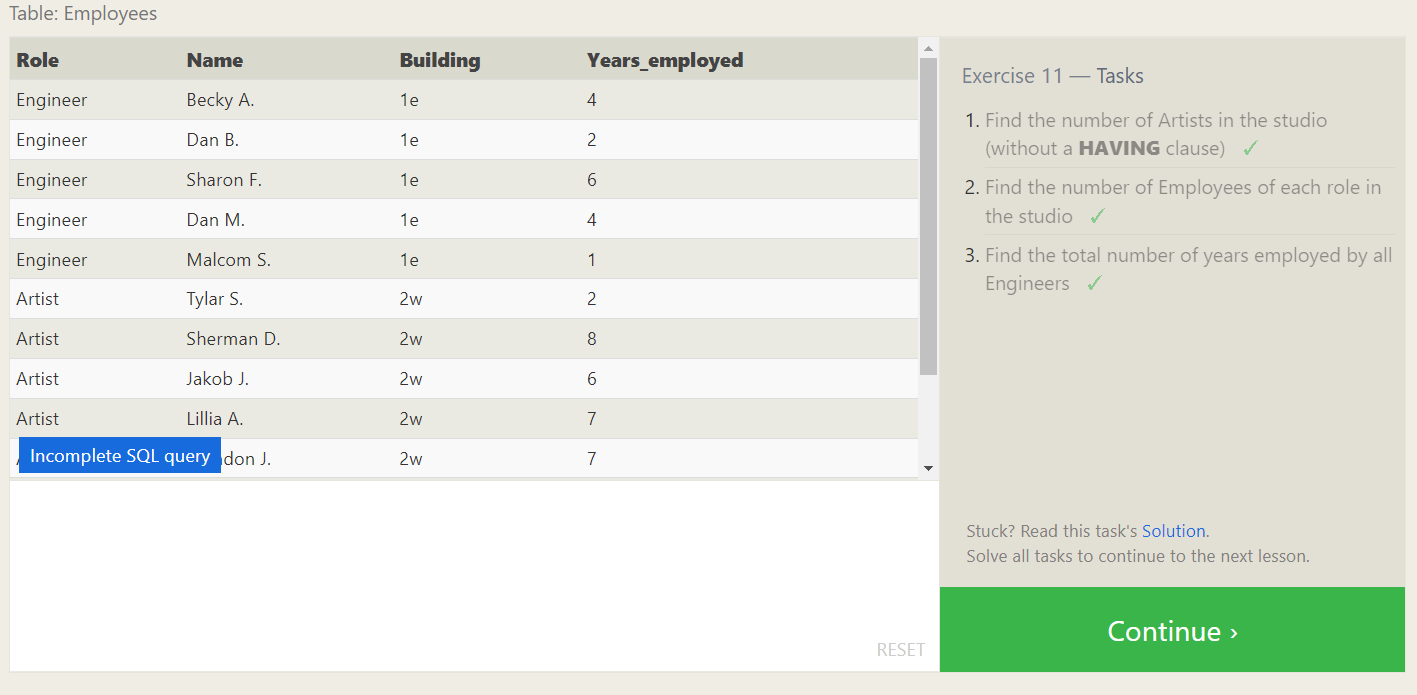
ANS: SELECT count(role) FROM employees where role = "Artist"

1. Find the number of Employees of each role in the studio

ANS: SELECT role,count(role) FROM employees group by role

1. Find the total number of years employed by all Engineers

ANS: SELECT role,sum(years\_employed) from employees group by role having = "Engineer"



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

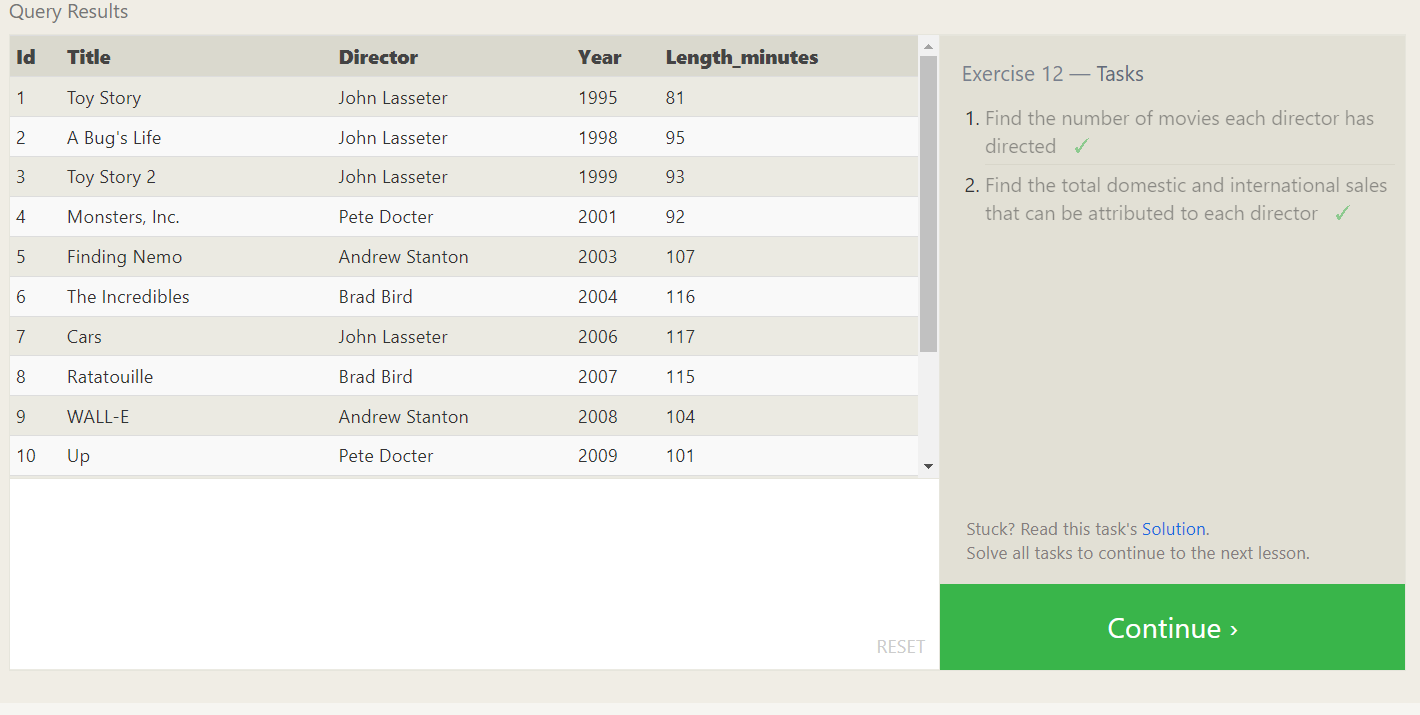
Exercise 12 — Tasks

1. Find the number of movies each director has directed

ANS: SELECT director,count(title) FROM movies group by director;

1. Find the total domestic and international sales that can be attributed to each director

ANS: SELECT director,sum(domestic\_sales + international\_sales) FROM movies m join boxoffice b on m.id = b.movie\_id group by director;



SQL Lesson 13: Inserting rows

Exercise 13 — Tasks

1. Add the studio's new production, Toy Story 4 to the list of movies (you can use any director) ✓

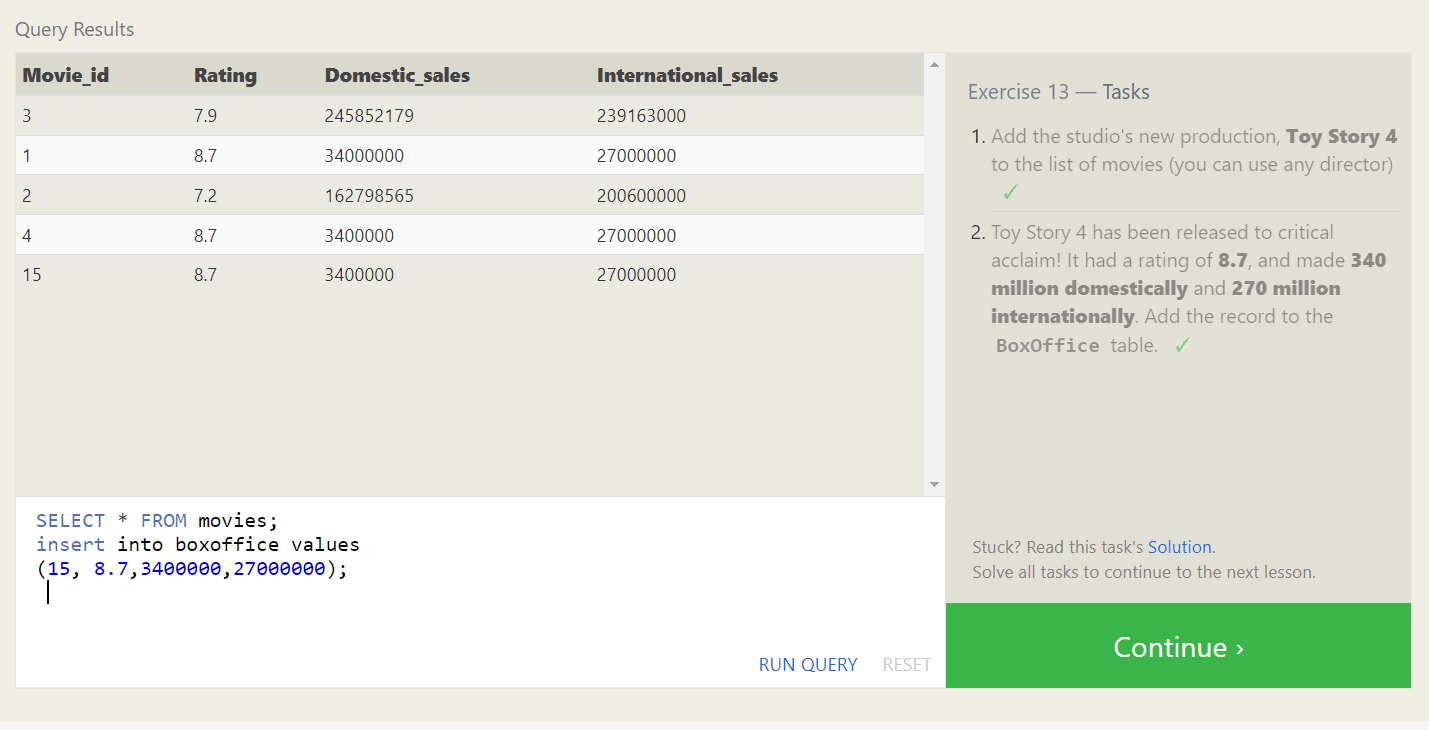
ANS: insert into movies values

(15, ‘Toy Story 4’, ‘John Lasseter’, 2012,115);

1. Toy Story 4 has been released to critical acclaim! It had a rating of 8.7, and made 340 million domestically and 270 million internationally. Add the record to the BoxOffice table. ✓

ANS: insert into boxoffice values

(15, 8.7,3400000,27000000);



SQL Lesson 14: Updating rows

Exercise 14 — Tasks

1. The director for A Bug's Life is incorrect, it was actually directed by John Lasseter ✓

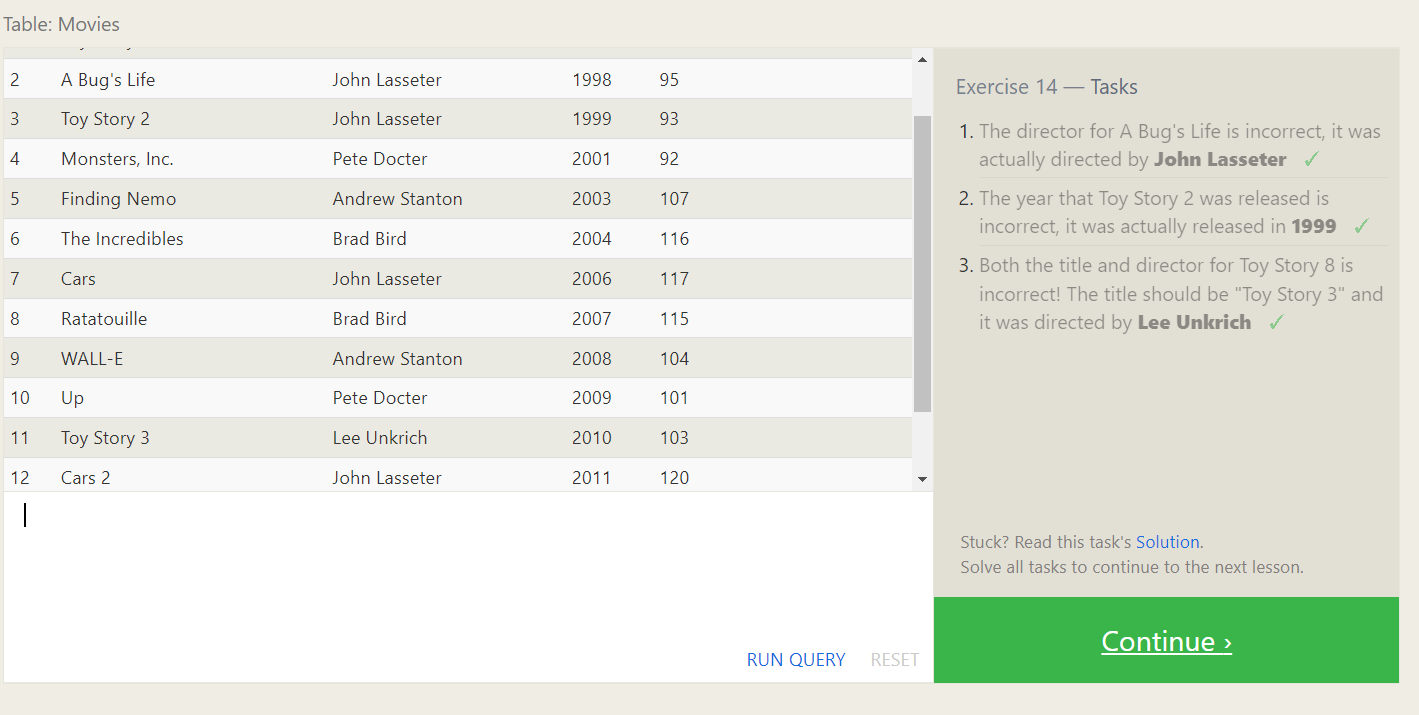
ANS: update movies set director = 'John Lasseter' where id = 2;

1. The year that Toy Story 2 was released is incorrect, it was actually released in 1999 ✓

ANS: update movies set year = 1999 where id = 3;

1. Both the title and director for Toy Story 8 is incorrect! The title should be "Toy Story 3" and it was directed by Lee Unkrich ✓

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



SQL Lesson 15: Deleting rows

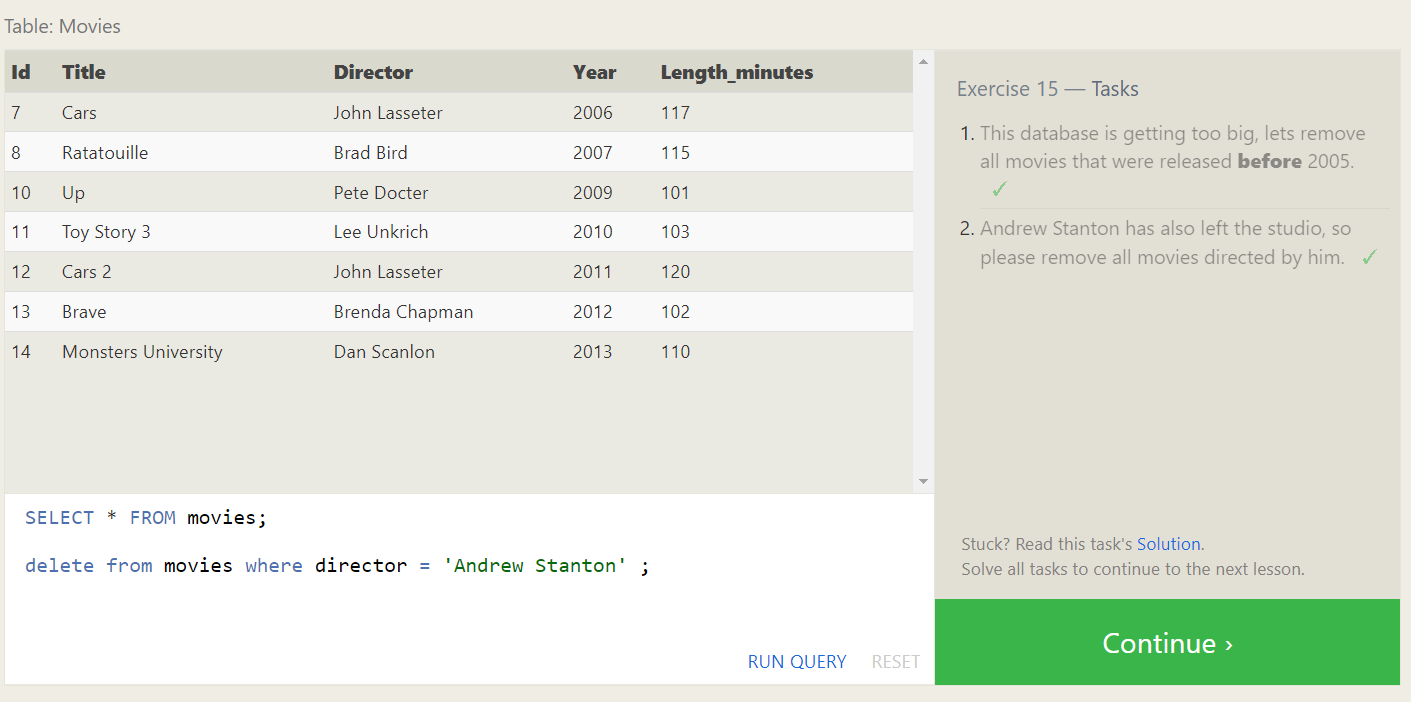
Exercise 15 — Tasks

1. This database is getting too big, lets remove all movies that were released before 2005. ✓

ANS: delete from movies where year < 2005;

1. Andrew Stanton has also left the studio, so please remove all movies directed by him. ✓

ANS: delete from movies where director = 'Andrew Stanton';



SQL Lesson 16: Creating tables

Exercise 16 — Tasks

1. Create a new table named Database with the following columns:

– Name A string (text) describing the name of the database

– Version A number (floating point) of the latest version of this database

– Download\_count An integer count of the number of times this database was downloaded

This table has no constraints. ✓

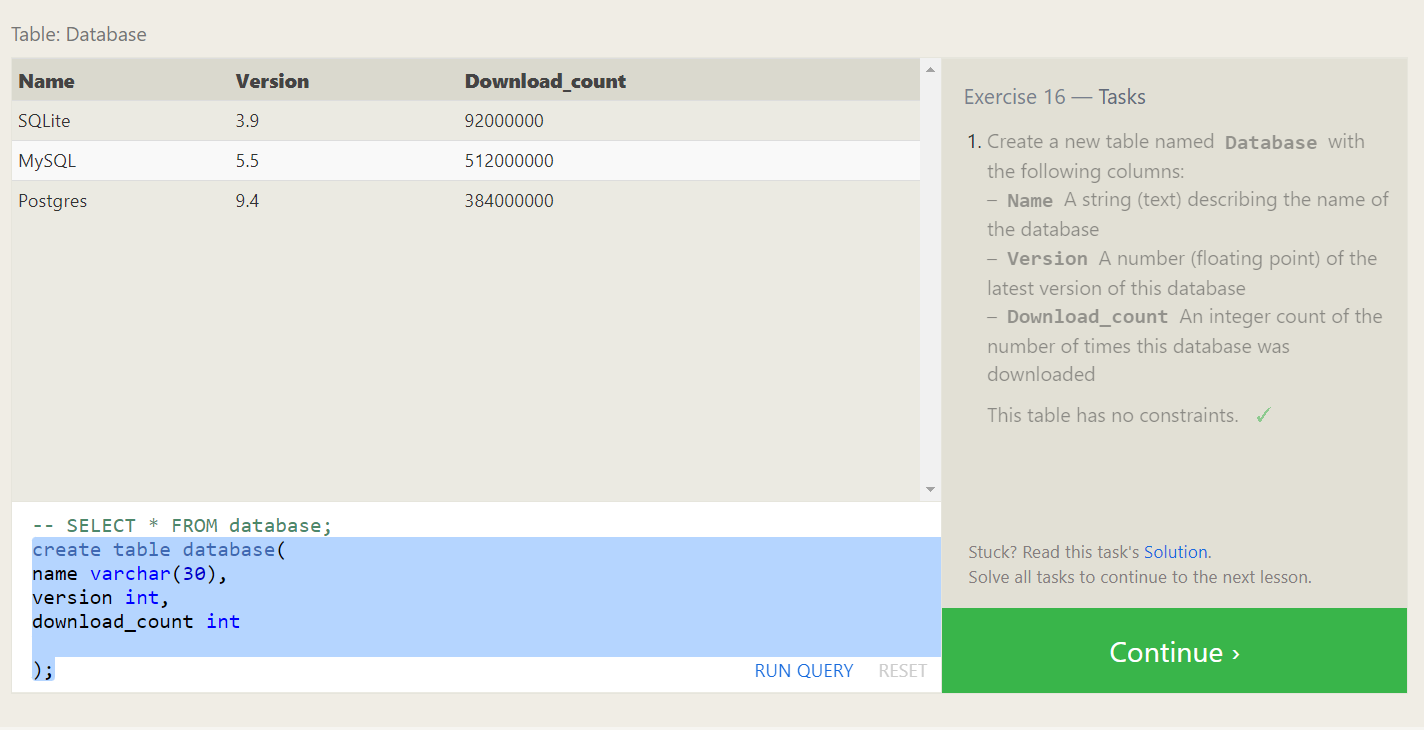
ANS: create table database(

name varchar(30),

version int,

download\_count int

);



SQL Lesson 17: Altering tables

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. ✓

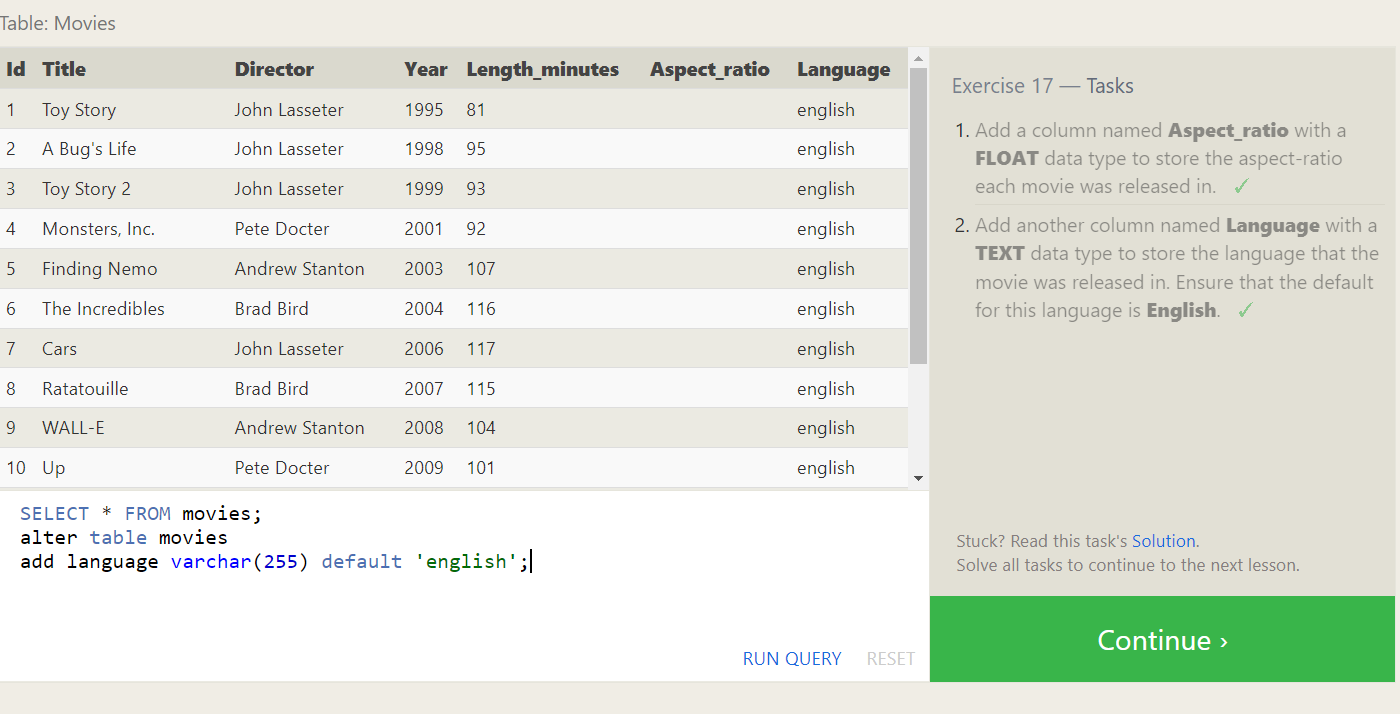
ANS: alter table movies

add aspect\_ratio float;

1. 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. ✓

ANS: alter table movies

add language varchar(255) default 'english';



SQL Lesson 18: Dropping tables

Exercise 18 — Tasks

1. We've sadly reached the end of our lessons, lets clean up by removing the Movies table ✓

ANS: drop table movies;

1. And drop the BoxOffice table as well ✓

ANS: SELECT \* FROM Boxoffice;

drop table Boxoffice;

