MySQL Task

(using sqlbolt.com)

SQL Lesson 1: SELECT queries 101:

Exercise 1 — Tasks

1. Find the title of each film ✓
    - ***SELECT title FROM movies;***
2. Find the director of each film ✓
    - ***SELECT director FROM movies;***
3. Find the title and director of each film ✓
    - ***SELECT title, director FROM movies;***
4. Find the title and year of each film ✓
    - ***SELECT title, year FROM movies;***
5. Find all the information about each film ✓
    - ***SELECT * FROM movies;***

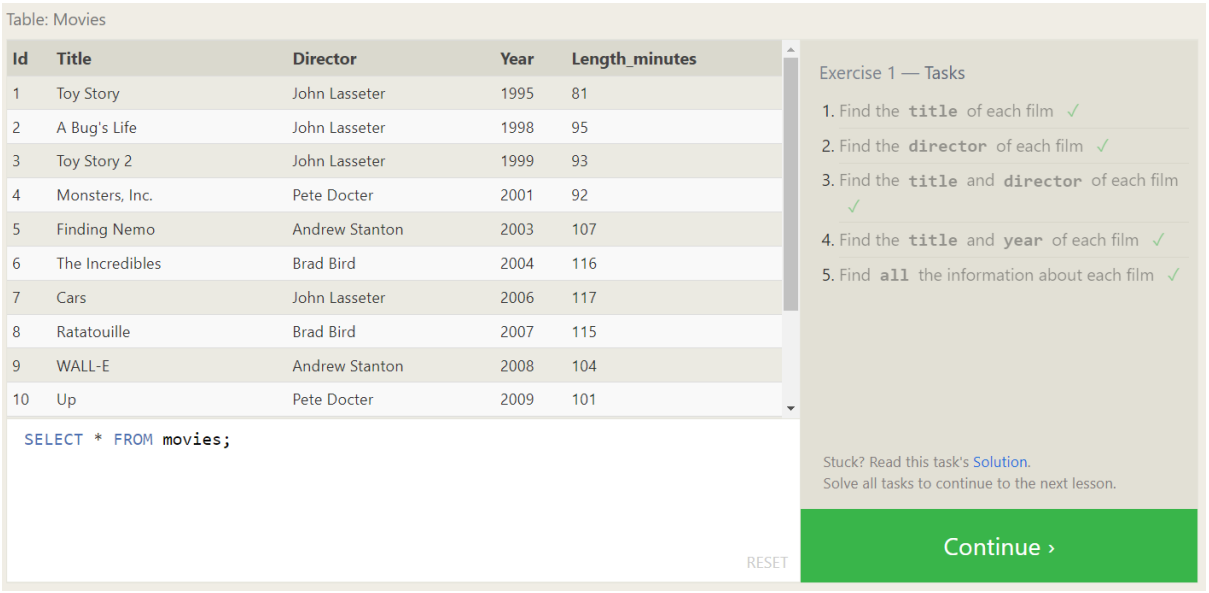


Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101

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

Exercise 2 — Tasks

1. Find the movie with a row **id** of 6 ✓
    - ***SELECT * FROM movies WHERE id=6;***
2. Find the movies released in the **year**s between 2000 and 2010 ✓
    - ***SELECT * FROM movies WHERE year BETWEEN 2000 AND 2010;***
3. Find the movies **not** released in the **year** s between 2000 and 2010 ✓
    - ***SELECT * FROM movies WHERE year NOT BETWEEN 2000 AND 2010;***
4. Find the first 5 Pixar movies and their release ✓
    - ***SELECT * FROM movies LIMIT 5;***

Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101

```
SELECT * FROM movies;
```

RESET

Exercise 2 — Tasks

1. Find the movie with a row `id` of 6 ✓
2. Find the movies released in the `year` s between 2000 and 2010 ✓
3. Find the movies **not** released in the `year` s between 2000 and 2010 ✓
4. Find the first 5 Pixar movies and their release `year` ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

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

Exercise 3 — Tasks

1. Find all the Toy Story movies ✓
    - ***SELECT * FROM movies WHERE title LIKE "%Toy Story%";***
2. Find all the movies directed by John Lasseter ✓
    - ***SELECT * FROM movies WHERE director="John Lasseter";***
3. Find all the movies (and director) not directed by John Lasseter ✓
    - ***SELECT * FROM movies WHERE director!="John Lasseter";***
4. Find all the WALL-* movies ✓
    - ***SELECT * FROM movies WHERE title LIKE "%WALL-%";***

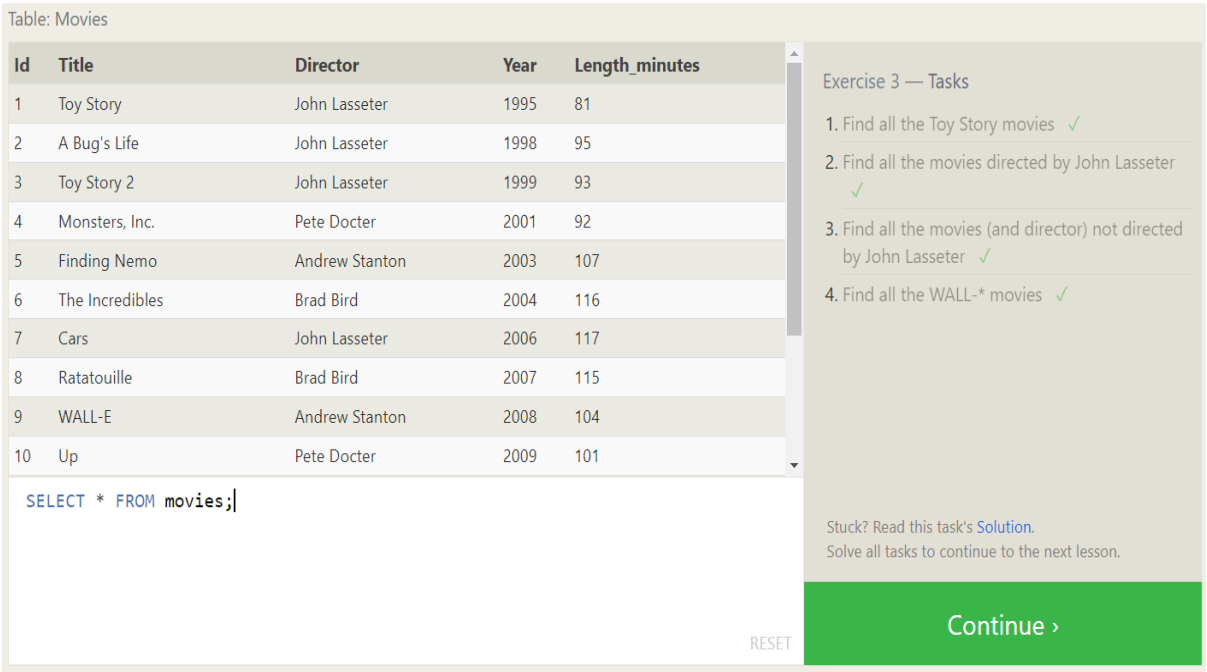


Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101

```
SELECT * FROM movies;
```

RESET

Exercise 3 — Tasks

1. Find all the Toy Story movies ✓
2. Find all the movies directed by John Lasseter ✓
3. Find all the movies (and director) not directed by John Lasseter ✓
4. Find all the WALL-* movies ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

SQL Lesson 4: Filtering and sorting Query results

Exercise 4 — Tasks

1. List all directors of Pixar movies (alphabetically), without duplicates ✓
   - ***SELECT DISTINCT director FROM movies ORDER BY director ASC;***
2. List the last four Pixar movies released (ordered from most recent to least) ✓
   - ***SELECT * FROM movies ORDER BY year DESC LIMIT 4;***
3. List the **first** five Pixar movies sorted alphabetically ✓
   - ***SELECT * FROM movies ORDER BY title ASC LIMIT 5;***
4. List the **next** five Pixar movies sorted alphabetically ✓
   - ***SELECT * FROM movies ORDER BY title ASC LIMIT 5 OFFSET 5;***

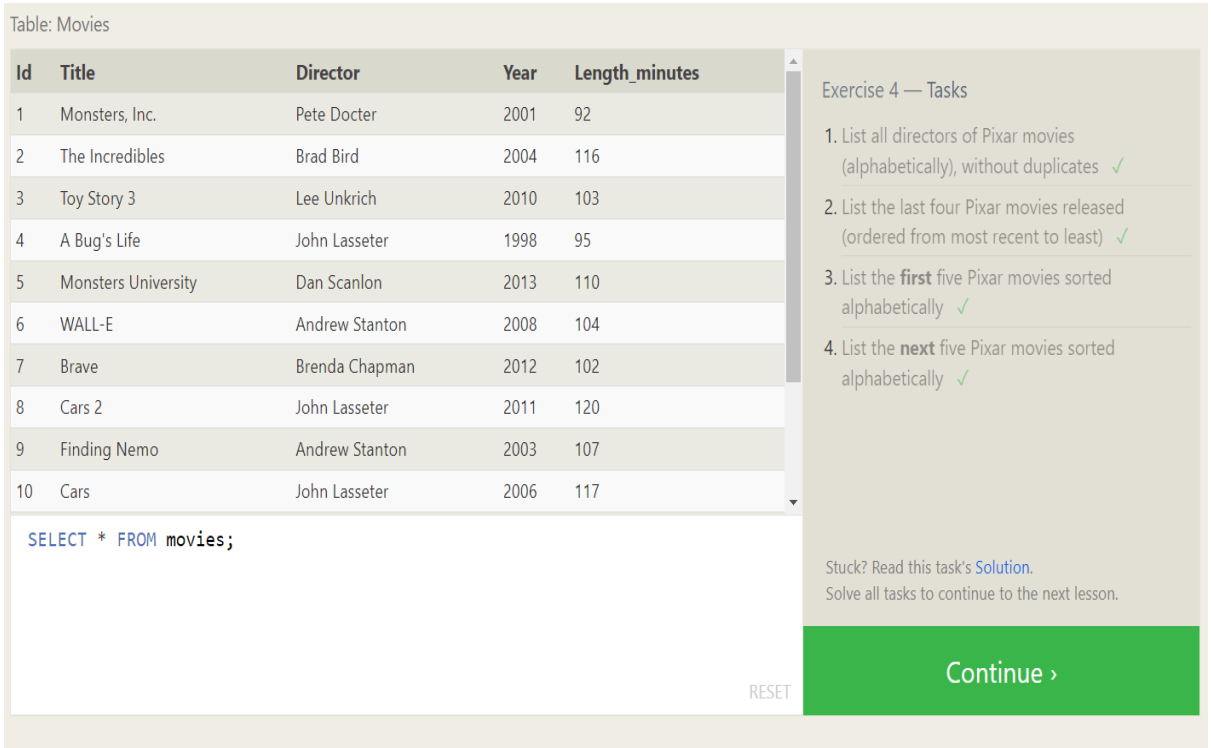


Table: Movies

Id	Title	Director	Year	Length_minutes
1	Monsters, Inc.	Pete Docter	2001	92
2	The Incredibles	Brad Bird	2004	116
3	Toy Story 3	Lee Unkrich	2010	103
4	A Bug's Life	John Lasseter	1998	95
5	Monsters University	Dan Scanlon	2013	110
6	WALL-E	Andrew Stanton	2008	104
7	Brave	Brenda Chapman	2012	102
8	Cars 2	John Lasseter	2011	120
9	Finding Nemo	Andrew Stanton	2003	107
10	Cars	John Lasseter	2006	117

```
SELECT * FROM movies;
```

RESET

Exercise 4 — Tasks

1. List all directors of Pixar movies (alphabetically), without duplicates ✓
2. List the last four Pixar movies released (ordered from most recent to least) ✓
3. List the **first** five Pixar movies sorted alphabetically ✓
4. List the **next** five Pixar movies sorted alphabetically ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

SQL Review: Simple SELECT Queries

Review 1 — Tasks

1. List all the Canadian cities and their populations ✓
   - ***SELECT city, population FROM north_american_cities WHERE country='Canada';***
2. Order all the cities in the United States by their latitude from north to south ✓
   - ***SELECT * FROM north_american_cities WHERE country='United States' ORDER BY latitude DESC;***
3. List all the cities west of Chicago, ordered from west to east ✓
   - ***SELECT city, longitude FROM north_american_cities WHERE longitude < -87.629798 ORDER BY longitude ASC;***
4. List the two largest cities in Mexico (by population) ✓
   - ***SELECT * FROM north_american_cities WHERE country='Mexico' ORDER BY population DESC LIMIT 2;***
5. List the third and fourth largest cities (by population) in the United States and their population ✓
   - ***SELECT * FROM north_american_cities WHERE country='United States' ORDER BY population DESC LIMIT 2 OFFSET 2;***

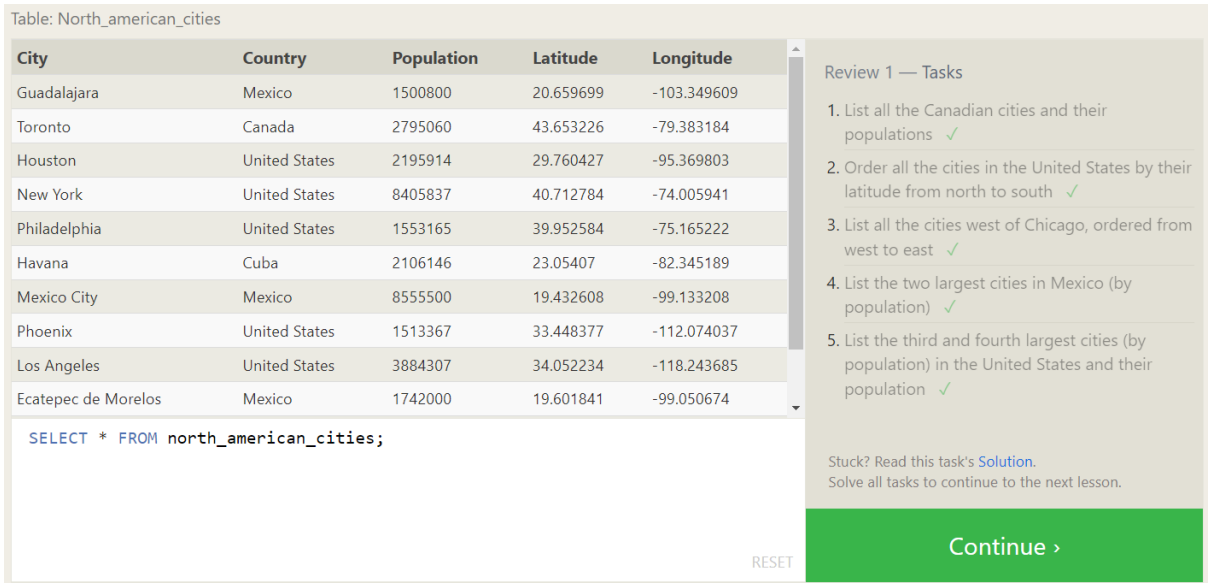


Table: North_american_cities

City	Country	Population	Latitude	Longitude
Guadalajara	Mexico	1500800	20.659699	-103.349609
Toronto	Canada	2795060	43.653226	-79.383184
Houston	United States	2195914	29.760427	-95.369803
New York	United States	8405837	40.712784	-74.005941
Philadelphia	United States	1553165	39.952584	-75.165222
Havana	Cuba	2106146	23.05407	-82.345189
Mexico City	Mexico	8555500	19.432608	-99.133208
Phoenix	United States	1513367	33.448377	-112.074037
Los Angeles	United States	3884307	34.052234	-118.243685
Ecatepec de Morelos	Mexico	1742000	19.601841	-99.050674

```
SELECT * FROM north_american_cities;
```

RESET

Review 1 — Tasks

1. List all the Canadian cities and their populations ✓
2. Order all the cities in the United States by their latitude from north to south ✓
3. List all the cities west of Chicago, ordered from west to east ✓
4. List the two largest cities in Mexico (by population) ✓
5. List the third and fourth largest cities (by population) in the United States and their population ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

SQL Lesson 6: Multi-table queries with JOINs

Exercise 6 — Tasks

1. Find the domestic and international sales for each movie ✓

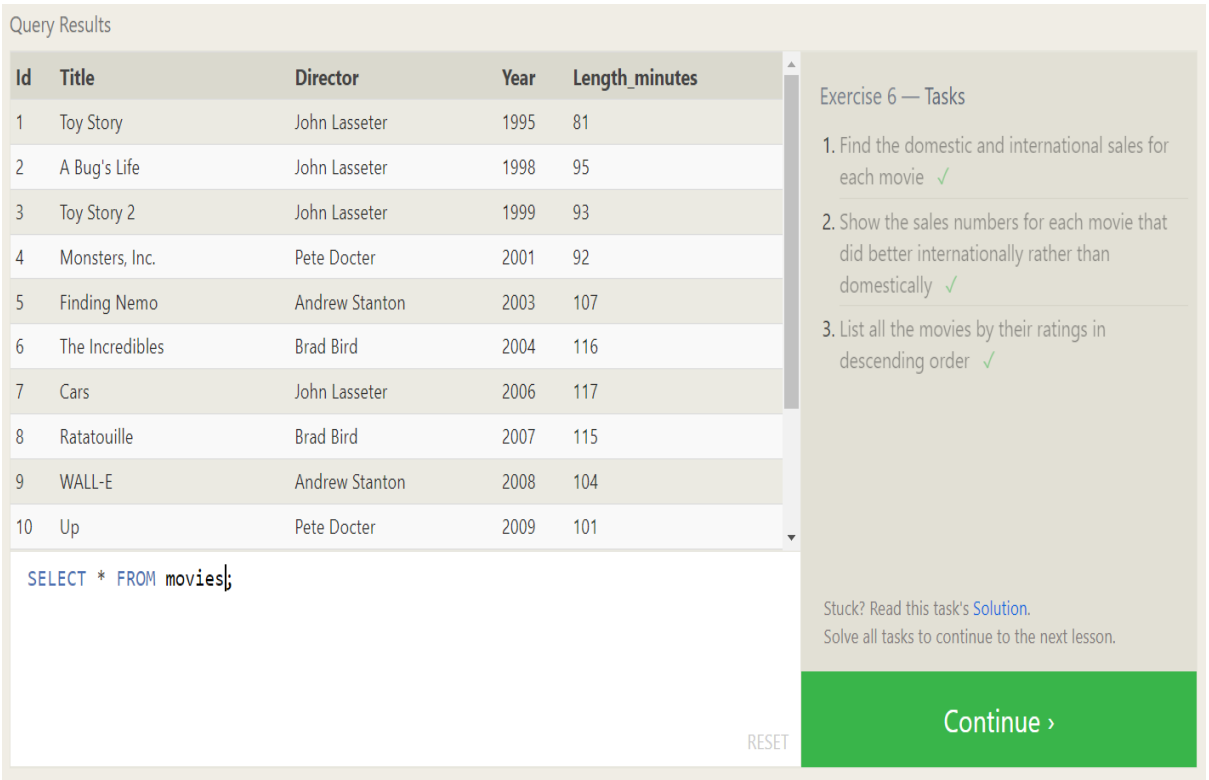
    - ***SELECT title, domestic_sales, international_sales FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id;***

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

    - ***SELECT title, international_sales, domestic_sales FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id WHERE international_sales > domestic_sales;***

3. List all the movies by their ratings in descending order ✓

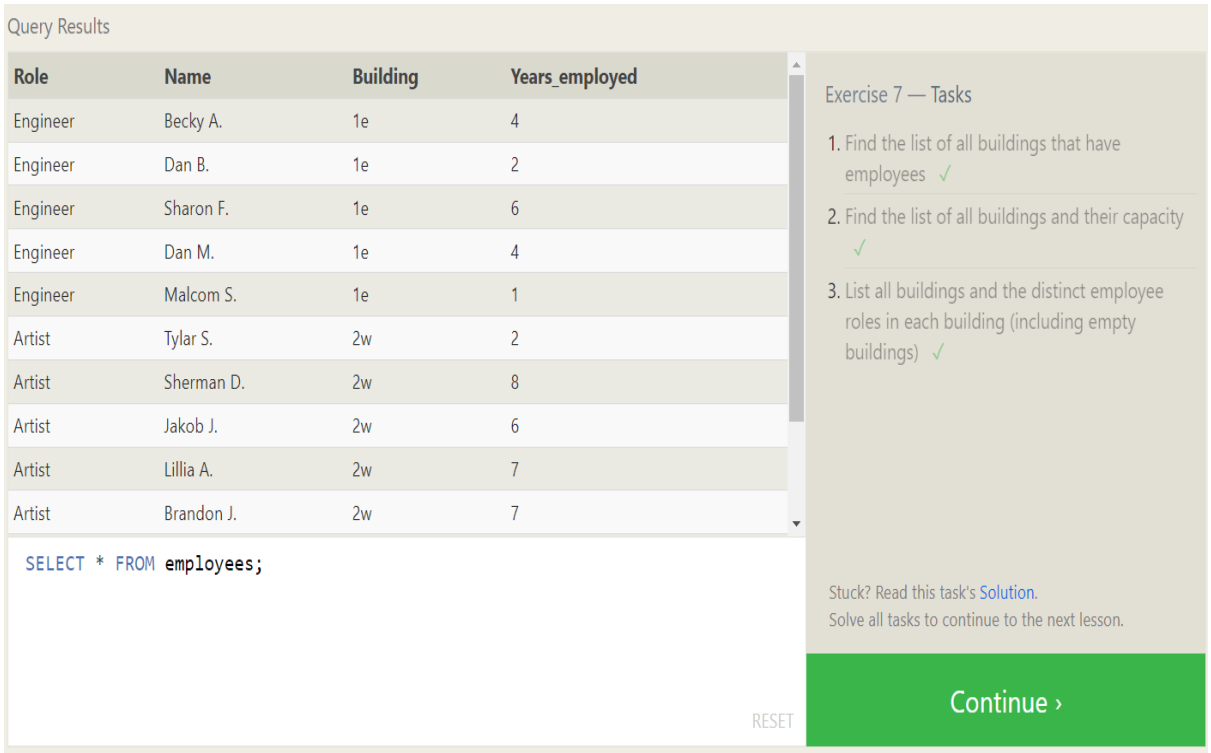
    - ***SELECT * FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id ORDER BY rating DESC;***



SQL Lesson 7: OUTER JOINs

Exercise 7 — Tasks

1. Find the list of all buildings that have employees ✓
   - ***SELECT DISTINCT building FROM employees;***
2. Find the list of all buildings and their capacity ✓
   - ***SELECT * FROM buildings;***
3. List all buildings and the distinct employee roles in each building (including empty buildings) ✓
   - ***SELECT DISTINCT building_name, role FROM buildings LEFT JOIN employees ON building_name = building;***



Query Results

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2
Artist	Sherman D.	2w	8
Artist	Jakob J.	2w	6
Artist	Lillia A.	2w	7
Artist	Brandon J.	2w	7

```
SELECT * FROM employees;
```

RESET

Exercise 7 — Tasks

1. Find the list of all buildings that have employees ✓
2. Find the list of all buildings and their capacity ✓
3. List all buildings and the distinct employee roles in each building (including empty buildings) ✓

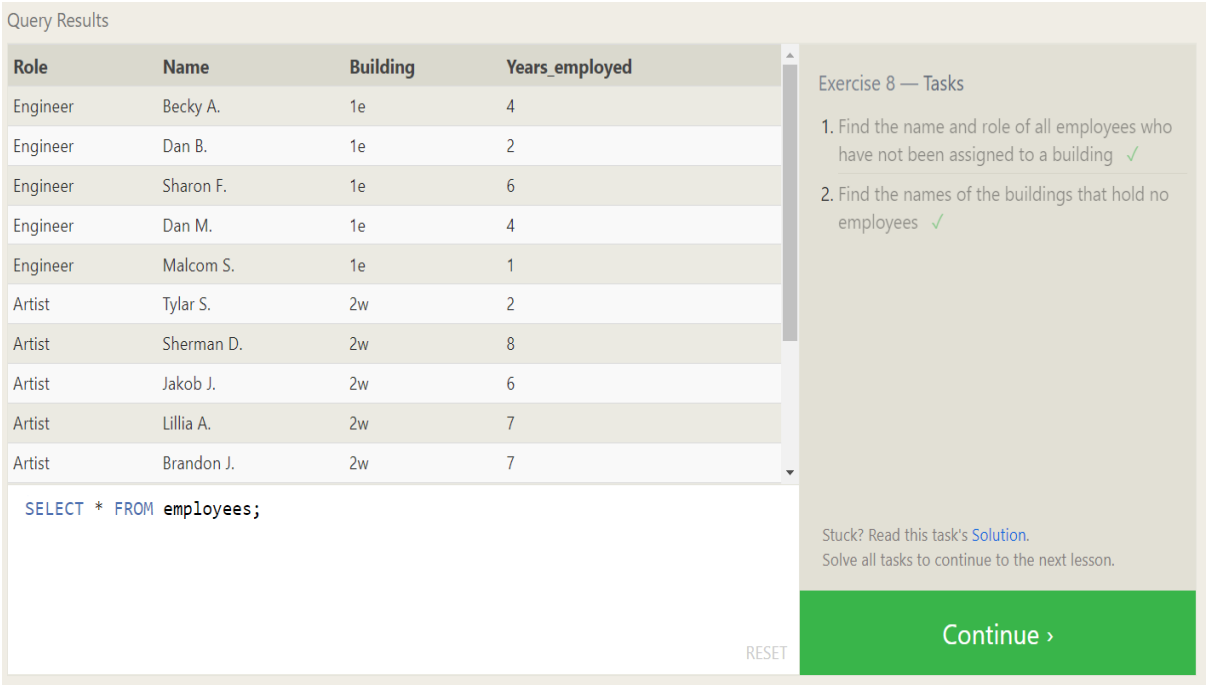
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

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 ✓
    - ***SELECT name, role FROM employees WHERE building IS NULL;***
2. Find the names of the buildings that hold no employees ✓
    - ***SELECT building_name, name FROM buildings LEFT JOIN employees ON building_name = building WHERE name IS NULL;***



SQL Lesson 9: Queries with expressions

Exercise 9 — Tasks

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

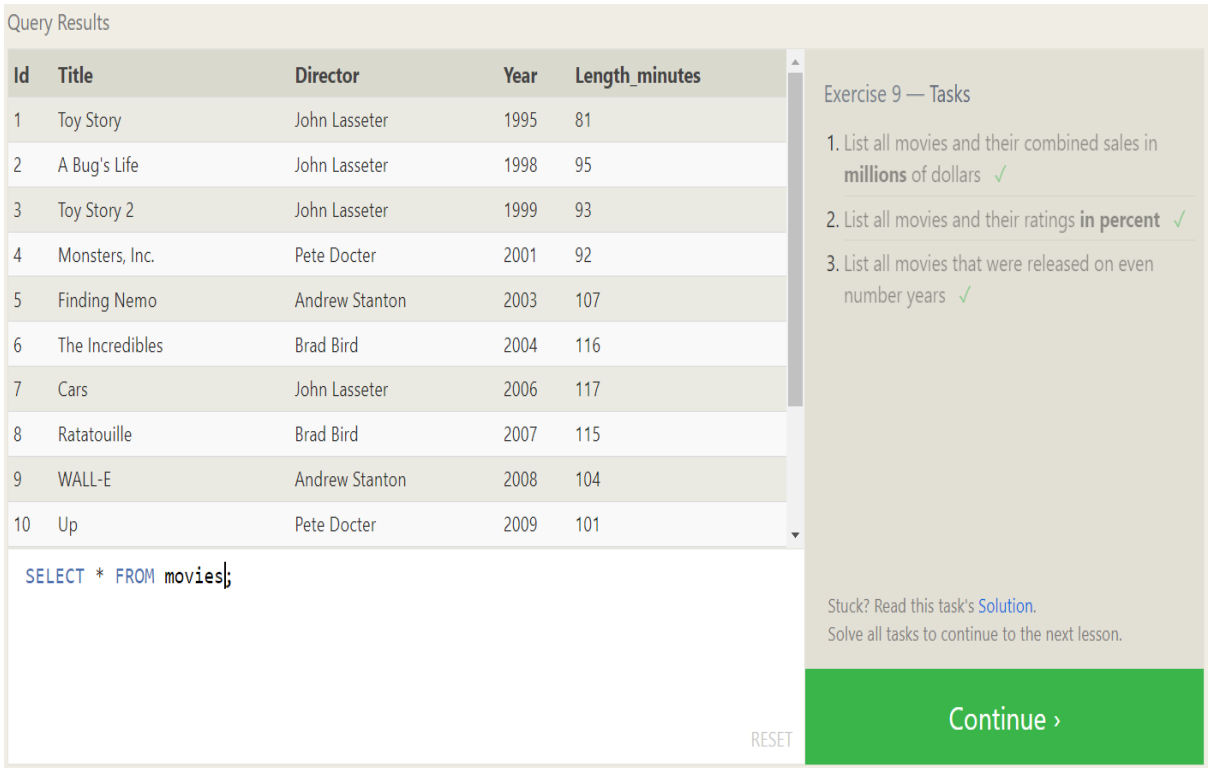
    - ***SELECT title, (domestic_sales + international_sales) / 1000000 AS combined_sales_in_millions FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id;***

2. List all movies and their ratings **in percent** ✓

    - ***SELECT title, (rating)*10 AS ratings_in_percent FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id;***

3. List all movies that were released on even number years ✓

    - ***SELECT * FROM movies WHERE year%2=0;***



Query Results

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101

```
SELECT * FROM movies;
```

RESET

Exercise 9 — Tasks

1. List all movies and their combined sales in **millions** of dollars ✓
2. List all movies and their ratings **in percent** ✓
3. List all movies that were released on even number years ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

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

Exercise 10 — Tasks

1. Find the longest time that an employee has been at the studio ✓
   - ***SELECT name, MAX(years_employed) FROM employees;***
2. For each role, find the average number of years employed by employees in that role ✓
   - ***SELECT role, AVG(years_employed) FROM employees GROUP BY role;***
3. Find the total number of employee years worked in each building ✓
   - ***SELECT building, SUM(years_employed) FROM employees GROUP BY building;***

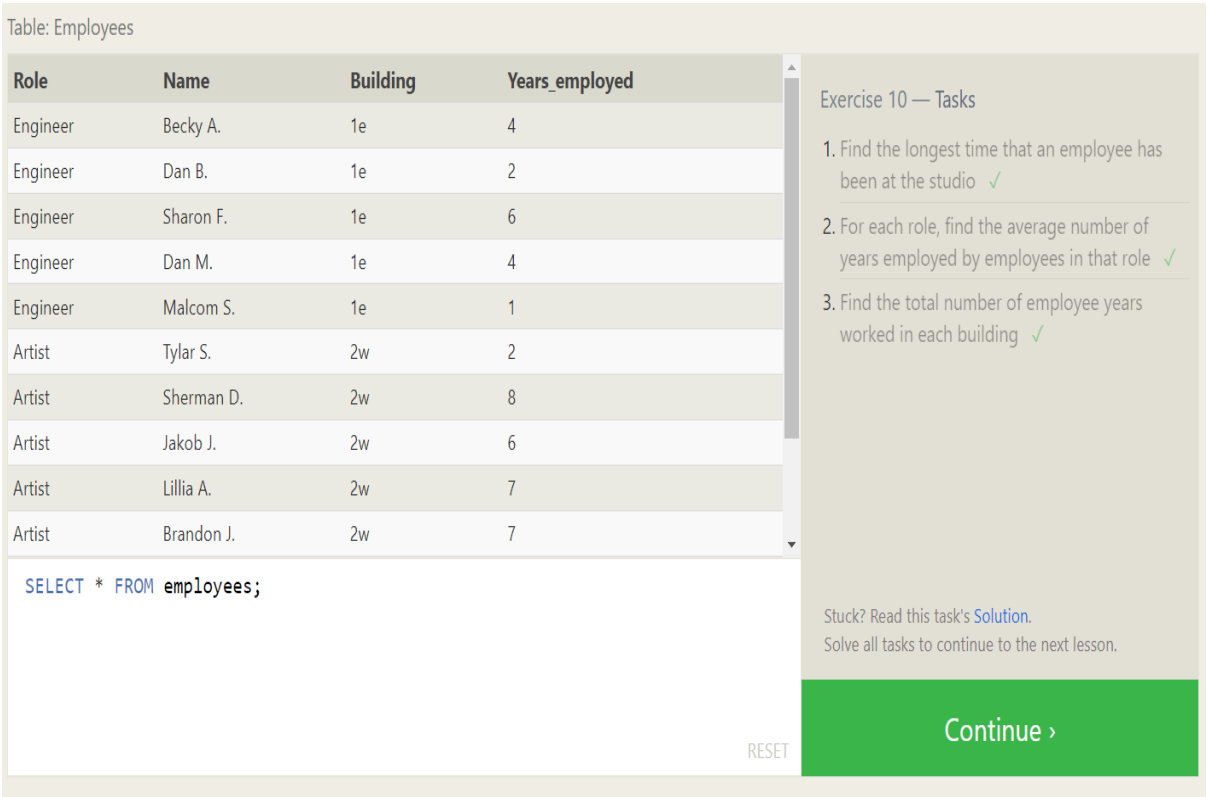


Table: Employees

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2
Artist	Sherman D.	2w	8
Artist	Jakob J.	2w	6
Artist	Lillia A.	2w	7
Artist	Brandon J.	2w	7

```
SELECT * FROM employees;
```

RESET

Exercise 10 — Tasks

1. Find the longest time that an employee has been at the studio ✓
2. For each role, find the average number of years employed by employees in that role ✓
3. Find the total number of employee years worked in each building ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

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

Exercise 11 — Tasks

1. Find the number of Artists in the studio (without a **HAVING** clause) ✓
   - ***SELECT role, COUNT(role) AS number_of_artists FROM employees WHERE role='Artist';***
2. Find the number of Employees of each role in the studio ✓
   - ***SELECT role, COUNT(role) AS number_of_employees FROM employees GROUP BY role;***
3. Find the total number of years employed by all Engineers ✓
   - ***SELECT role, SUM(years_employed) AS total_number_of_years_employed FROM employees WHERE role='Engineer';***

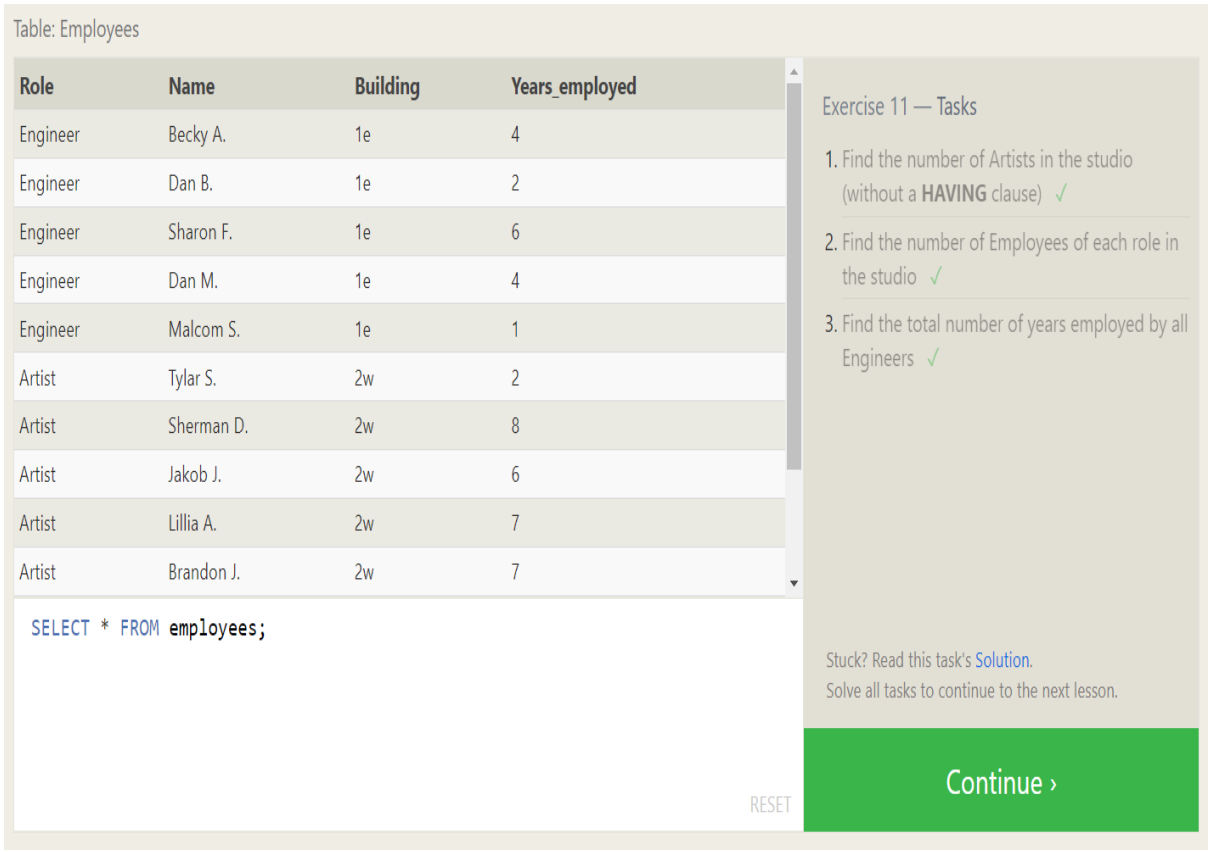


Table: Employees

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2
Artist	Sherman D.	2w	8
Artist	Jakob J.	2w	6
Artist	Lillia A.	2w	7
Artist	Brandon J.	2w	7

```
SELECT * FROM employees;
```

RESET

Exercise 11 — Tasks

1. Find the number of Artists in the studio (without a **HAVING** clause) ✓
2. Find the number of Employees of each role in the studio ✓
3. Find the total number of years employed by all Engineers ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

SQL Lesson 12: Order of execution of a Query

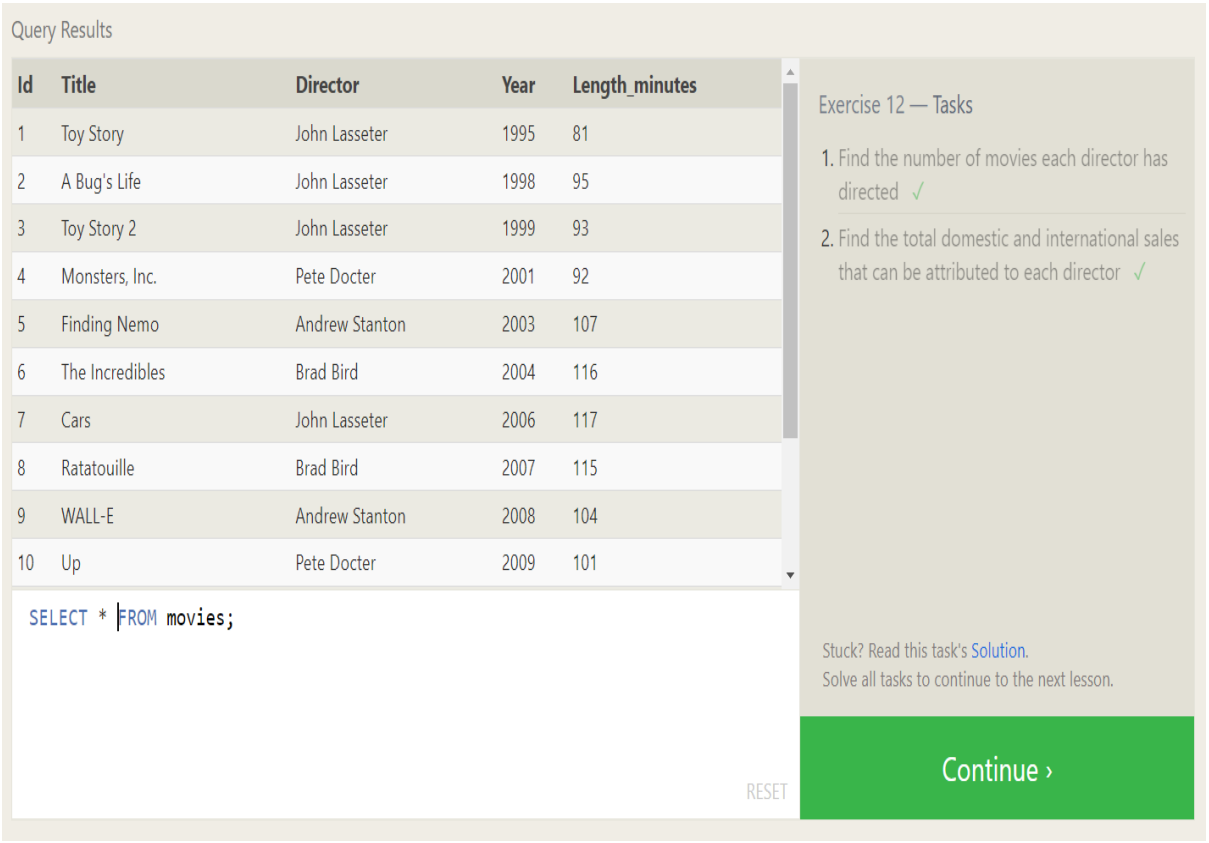
Exercise 12 — Tasks

1. Find the number of movies each director has directed ✓

    - ***SELECT director, count(director) AS number_of_movies_directed FROM movies GROUP BY director;***

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

    - ***SELECT director, SUM(domestic_sales) + SUM(international_sales) AS total_sales FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id GROUP BY director;***



Query Results

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101

```
SELECT * FROM movies;
```

RESET

Exercise 12 — Tasks

1. Find the number of movies each director has directed ✓
2. Find the total domestic and international sales that can be attributed to each director ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

SQL Lesson 13: Inserting rows

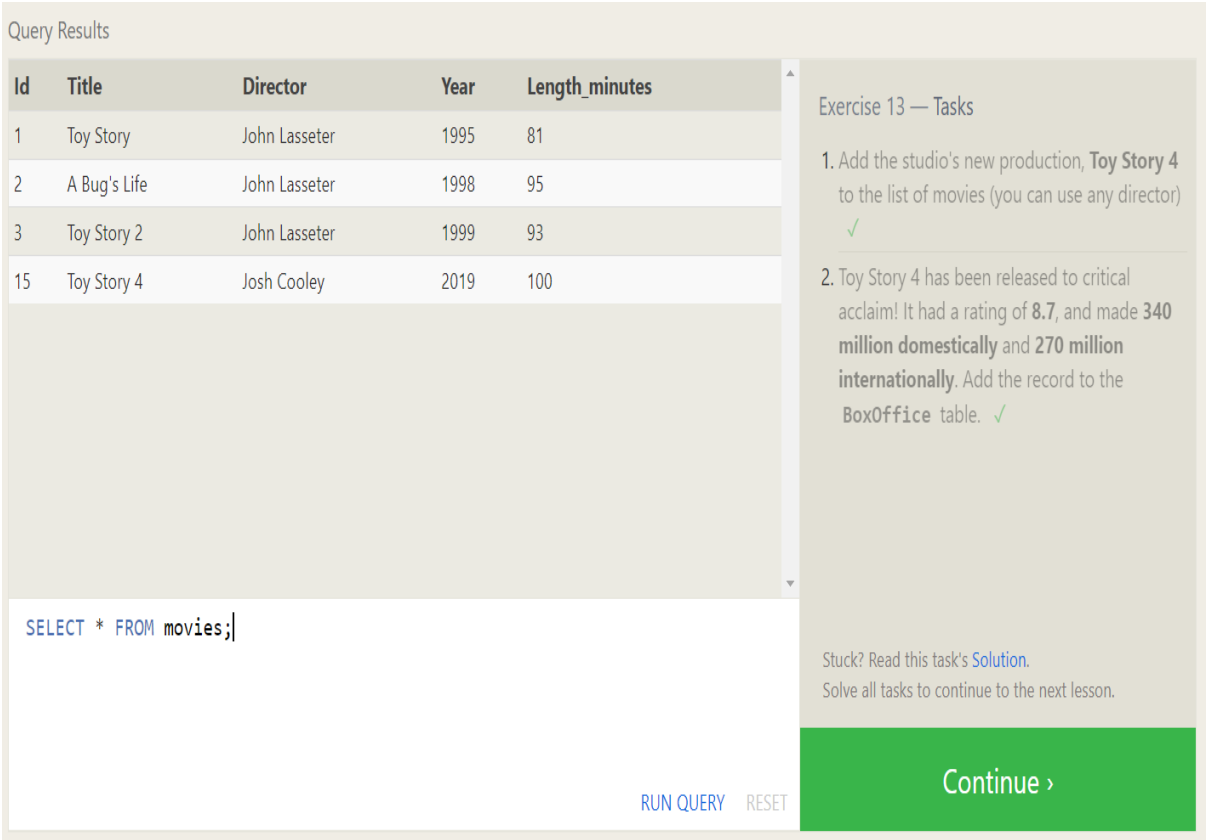
Exercise 13 — Tasks

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

   - ***INSERT INTO movies (id, title, director, year, length_minutes) VALUES (15, "Toy Story 4", "Josh Cooley", 2019, 100);***

2. 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 ✓

   - ***INSERT INTO boxoffice (movie_id, rating, domestic_sales, international_sales) VALUES (15, 8.7, 340000000, 270000000);***



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** ✓
   - ***UPDATE movies SET director="John Lasseter" WHERE title="A Bug's Life";***
2. The year that Toy Story 2 was released is incorrect, it was actually released in **1999** ✓
   - ***UPDATE movies SET year=1999 WHERE title="Toy Story 2";***
3. 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** ✓
   - ***UPDATE movies SET title="Toy Story 3", director="Lee Unkrich" WHERE title="Toy Story 8";***

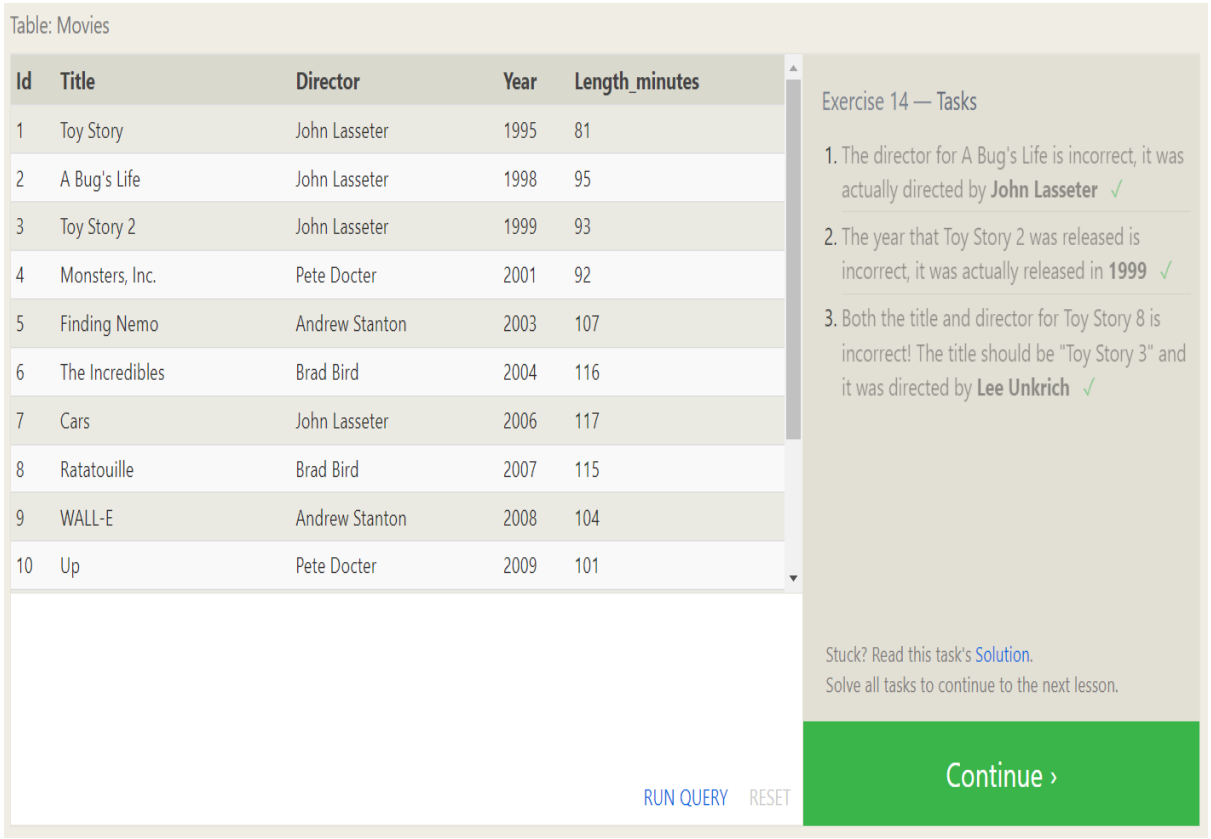


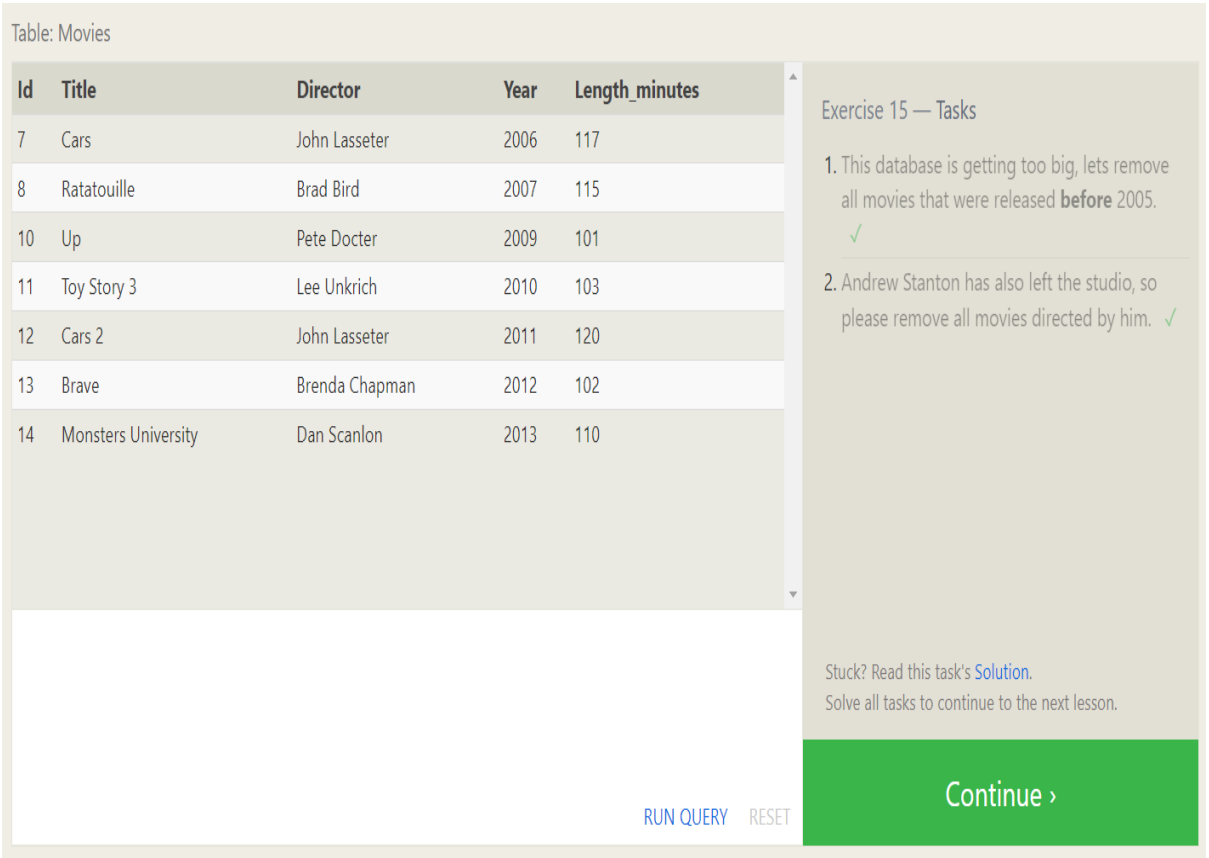
Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101

SQL Lesson 15: Deleting rows

Exercise 15 — Tasks

1. This database is getting too big, lets remove all movies that were released **before** 2005. ✓
   - ***DELETE FROM movies WHERE year < 2005;***
2. Andrew Stanton has also left the studio, so please remove all movies directed by him ✓
   - ***DELETE FROM movies WHERE director="Andrew Stanton";***



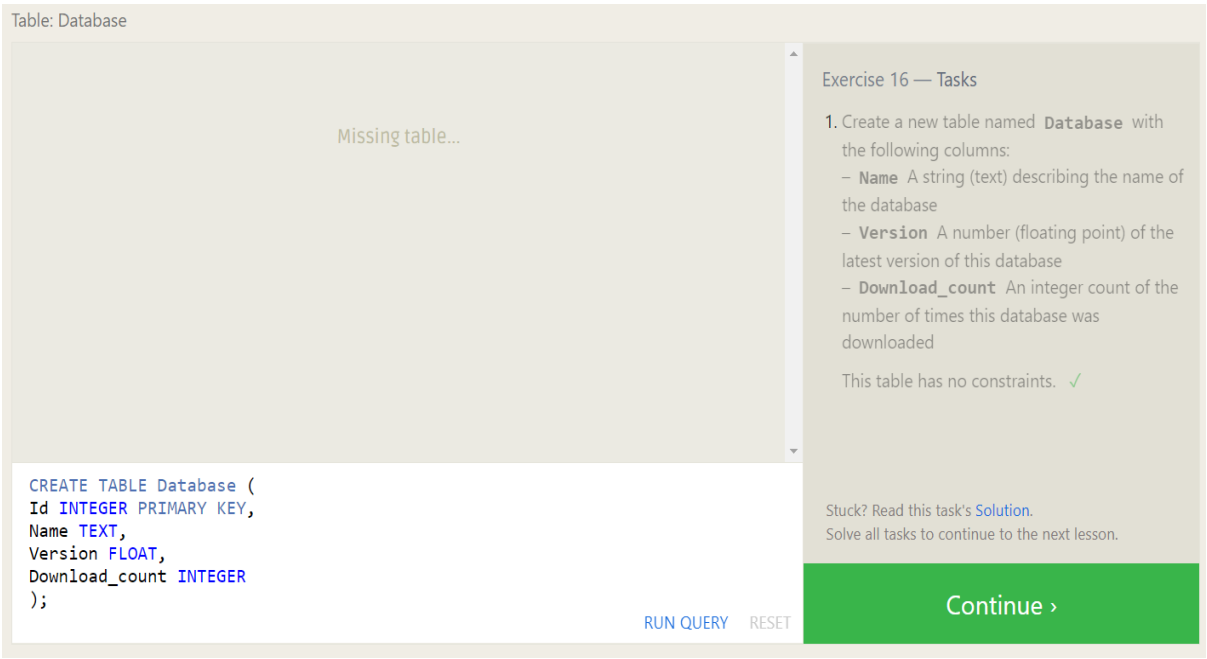
SQL Lesson 16: Creating tables

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. ✓

   - ***CREATE TABLE Database (***
   - ***Id INTEGER PRIMARY KEY,***
   - ***Name TEXT,***
   - ***Version FLOAT,***
   - ***Download_count INTEGER***
   - ***);***



SQL Lesson 17: Altering tables

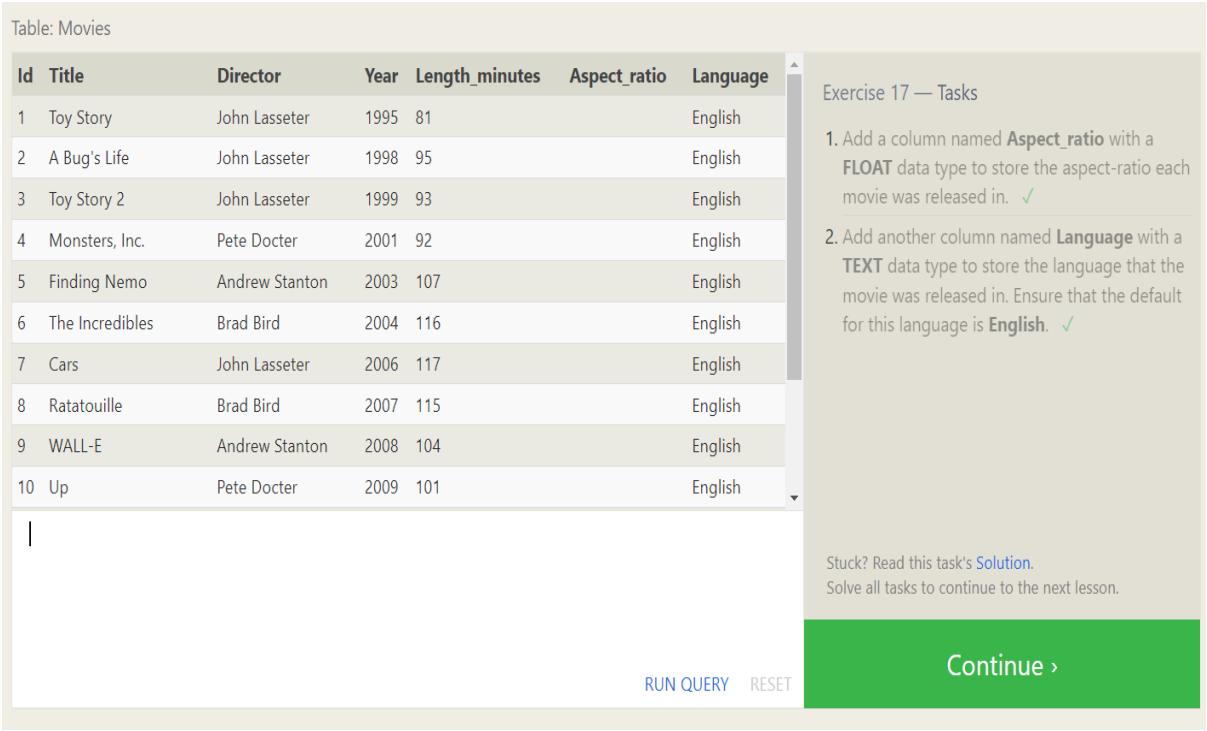
Exercise 17 — Tasks

1. Add a column named **Aspect_ratio** with a **FLOAT** data type to store the aspect-ratio each movie was released in. ✓

   - ***ALTER TABLE movies ADD Aspect_ratio FLOAT;***

2. 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** ✓

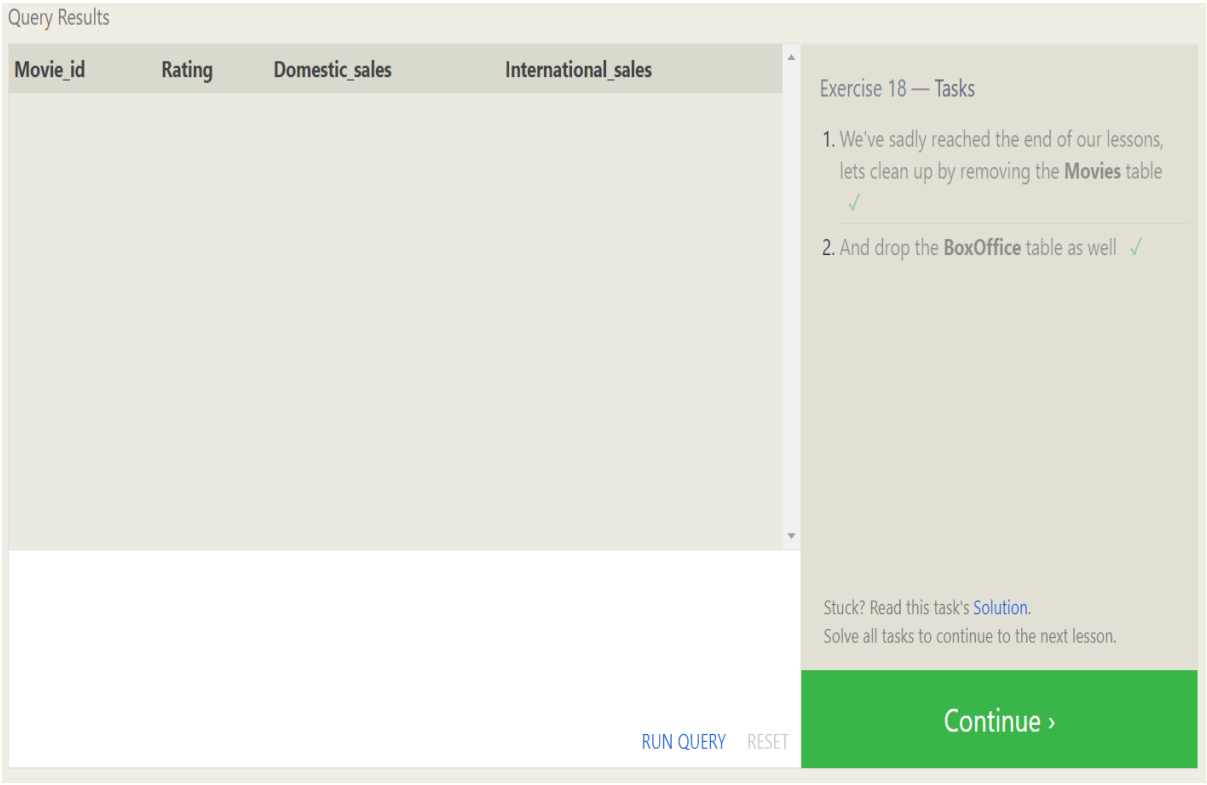
   - ***ALTER TABLE movies ADD Language TEXT DEFAULT English;***



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. ✓
   - ***DROP TABLE movies;***
2. And drop the **BoxOffice** table as well ✓
   - ***DROP TABLE boxoffice;***



SQL Lesson X: To infinity and beyond!

You've finished the tutorial!