SQL Lesson 1: SELECT queries 101

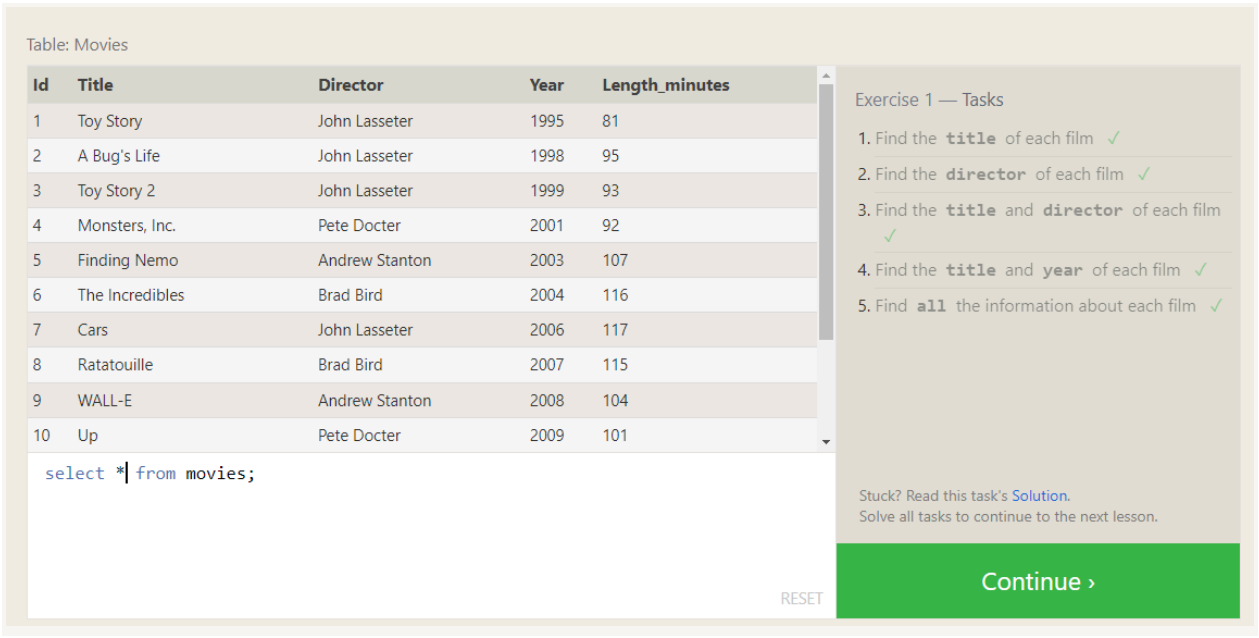


Table: Movies

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 1 — Tasks

1. Find the `title` of each film ✓
2. Find the `director` of each film ✓
3. Find the `title` and `director` of each film ✓
4. Find the `title` and `year` of each film ✓
5. Find `all` the information about each film ✓

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

Continue ›

Answers:

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: Queries with constraints (Pt. 1)

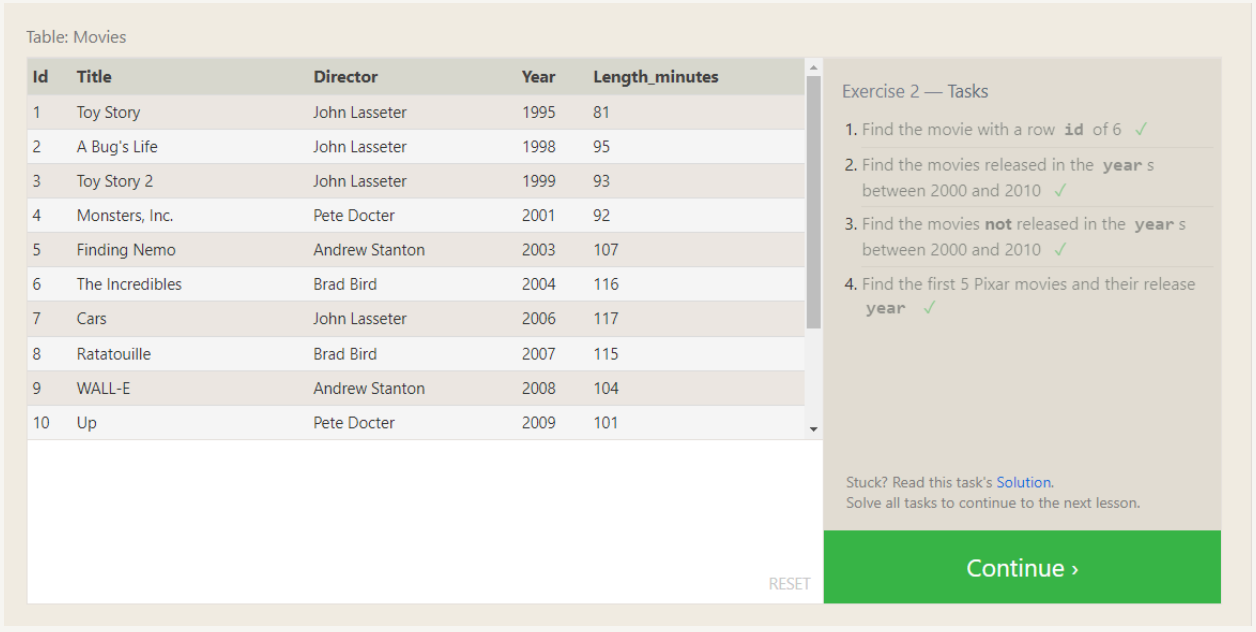


Table: Movies

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

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 ›

Answers:

1.select * from movies where id=6;
2.select * from movies where year between 2000 and 2010;
3.select * from movies where year not between 2000 and 2010;
4.select * from movies where year limit 5;

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

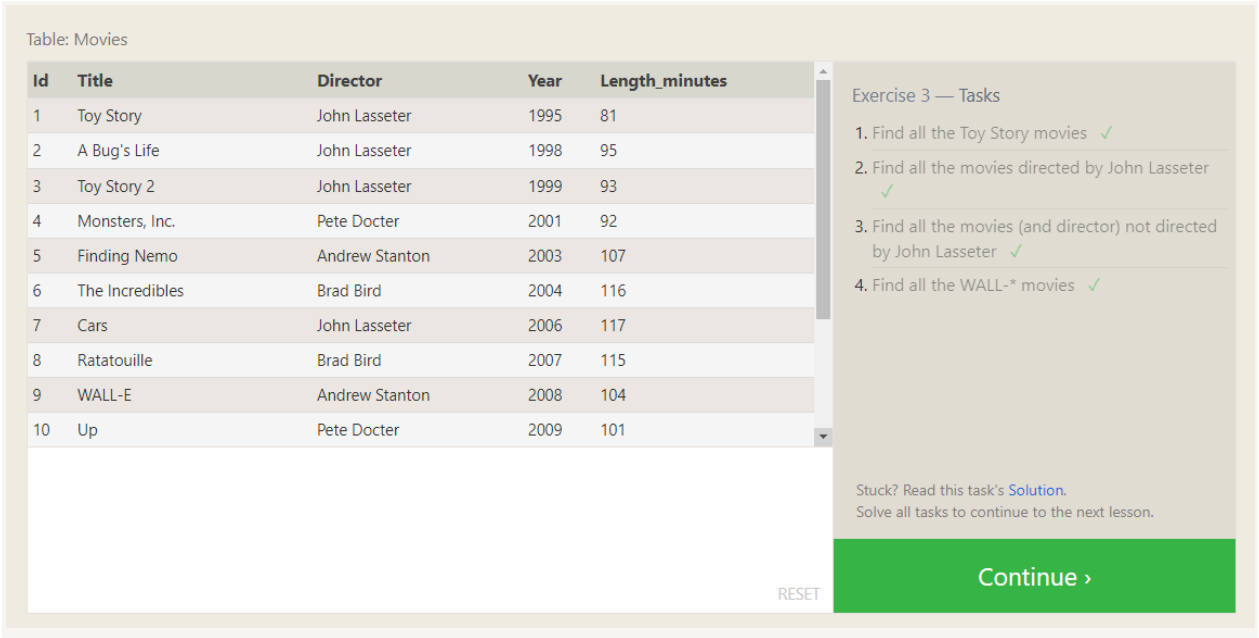


Table: Movies

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

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 ›

Answers:

1.select title from movies where title like "Toy Story%";
2.select title from movies where director="John Lasseter";
3.select * from movies where director !="John Lasseter";
4.select title from movies where title like "WALL%";

SQL Lesson 4: Filtering and sorting Query results

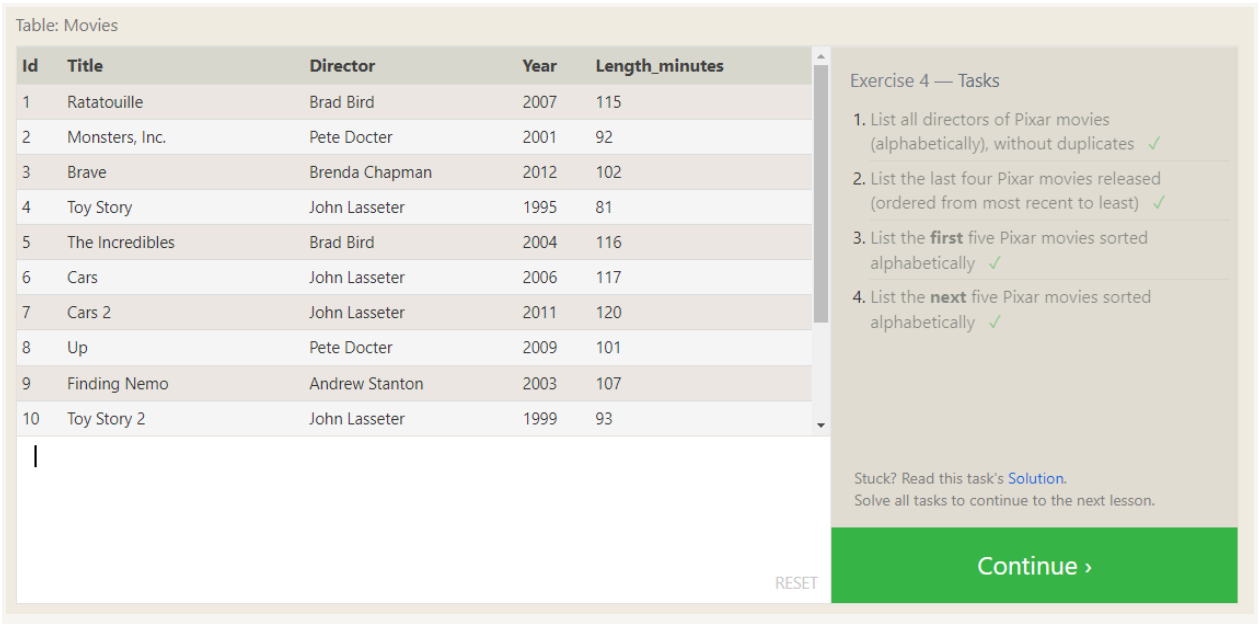


Table: Movies

Id	Title	Director	Year	Length_minutes
1	Ratatouille	Brad Bird	2007	115
2	Monsters, Inc.	Pete Docter	2001	92
3	Brave	Brenda Chapman	2012	102
4	Toy Story	John Lasseter	1995	81
5	The Incredibles	Brad Bird	2004	116
6	Cars	John Lasseter	2006	117
7	Cars 2	John Lasseter	2011	120
8	Up	Pete Docter	2009	101
9	Finding Nemo	Andrew Stanton	2003	107
10	Toy Story 2	John Lasseter	1999	93

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 ›

Answers:

1.SELECT distinct director FROM movies order by director;
2.SELECT title,year 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 Review: Simple SELECT Queries

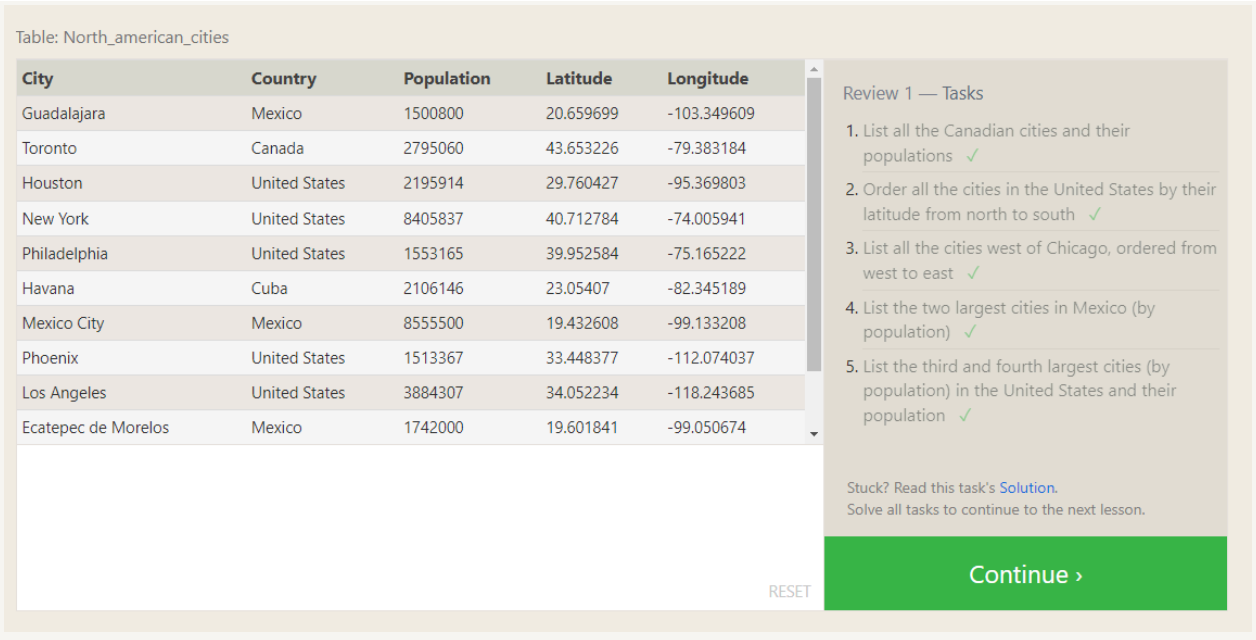


Table: North_american_cities

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

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 ✓

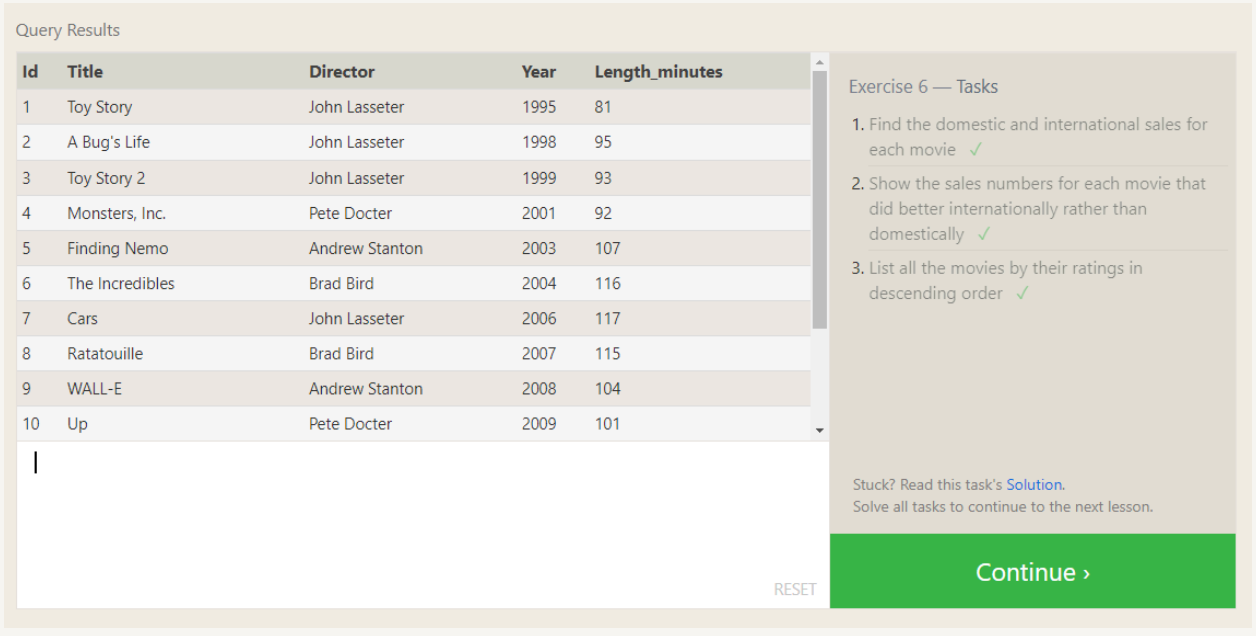
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

Answers:

1.SELECT city,population FROM north_american_cities where country="Canada";
2.SELECT city FROM north_american_cities WHERE country = "United States" ORDER BY latitude DESC;
3.SELECT city FROM north_american_cities WHERE longitude < -87.629798 ORDER BY longitude;
4.SELECT city FROM north_american_cities WHERE longitude < -87.629798 ORDER BY longitude;
5.SELECT city FROM north_american_cities WHERE longitude < -87.629798 ORDER BY longitude;

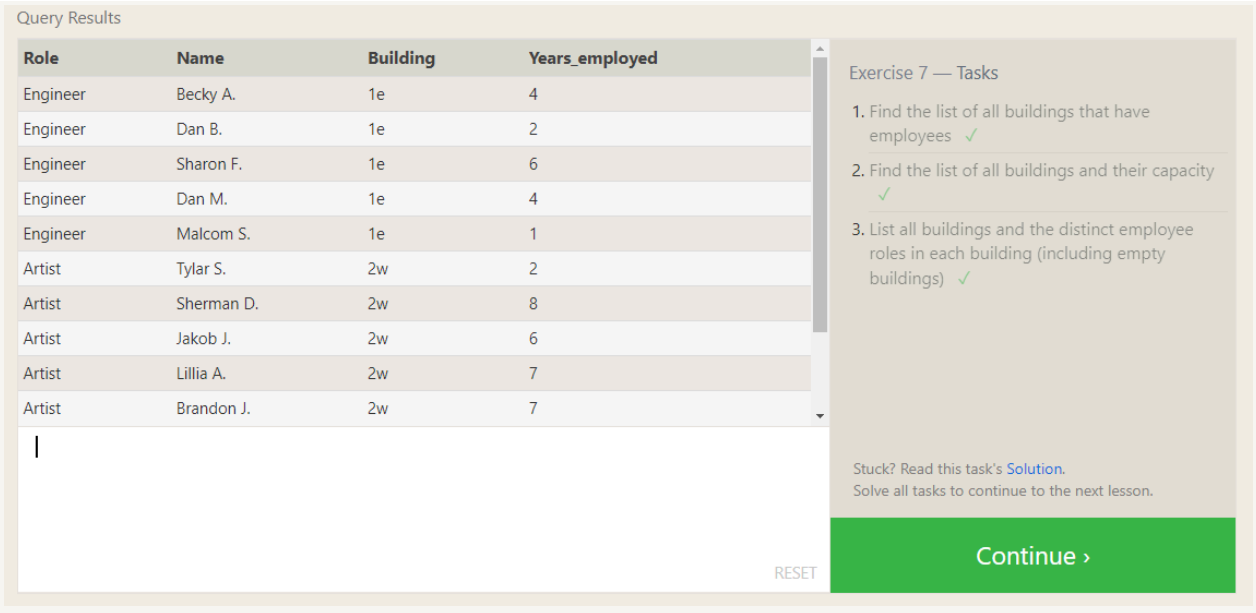
SQL Lesson 6: Multi-table queries with JOINs



Answers:

1.SELECT title, domestic_sales, international_sales FROM movies JOIN boxoffice ON movies.id = boxoffice.movie_id;
2.SELECT title, domestic_sales, international_sales FROM movies JOIN boxoffice ON movies.id = boxoffice.movie_id WHERE international_sales > domestic_sales;
3.SELECT title, rating FROM movies JOIN boxoffice ON movies.id = boxoffice.movie_id ORDER BY rating DESC;

SQL Lesson 7: OUTER JOINs



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

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) ✓

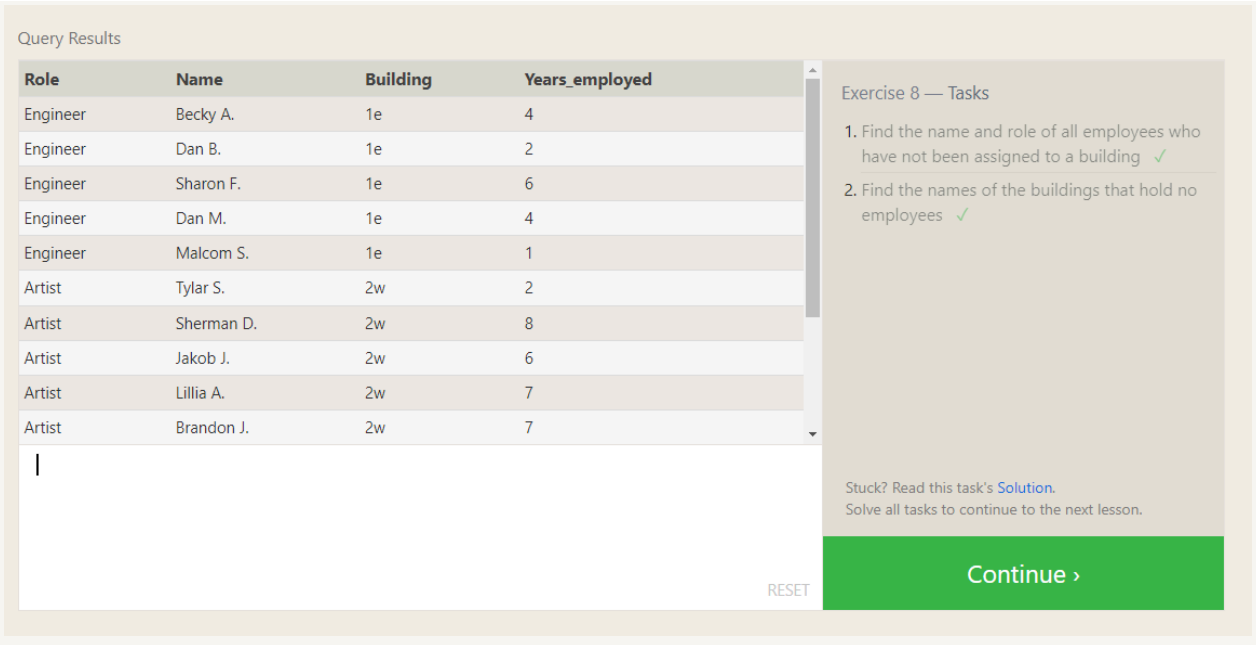
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

Answers:

1.SELECT DISTINCT building FROM employees;
2.SELECT * FROM buildings;
3.SELECT DISTINCT building_name, role FROM buildings LEFT JOIN employees ON building_name = building;

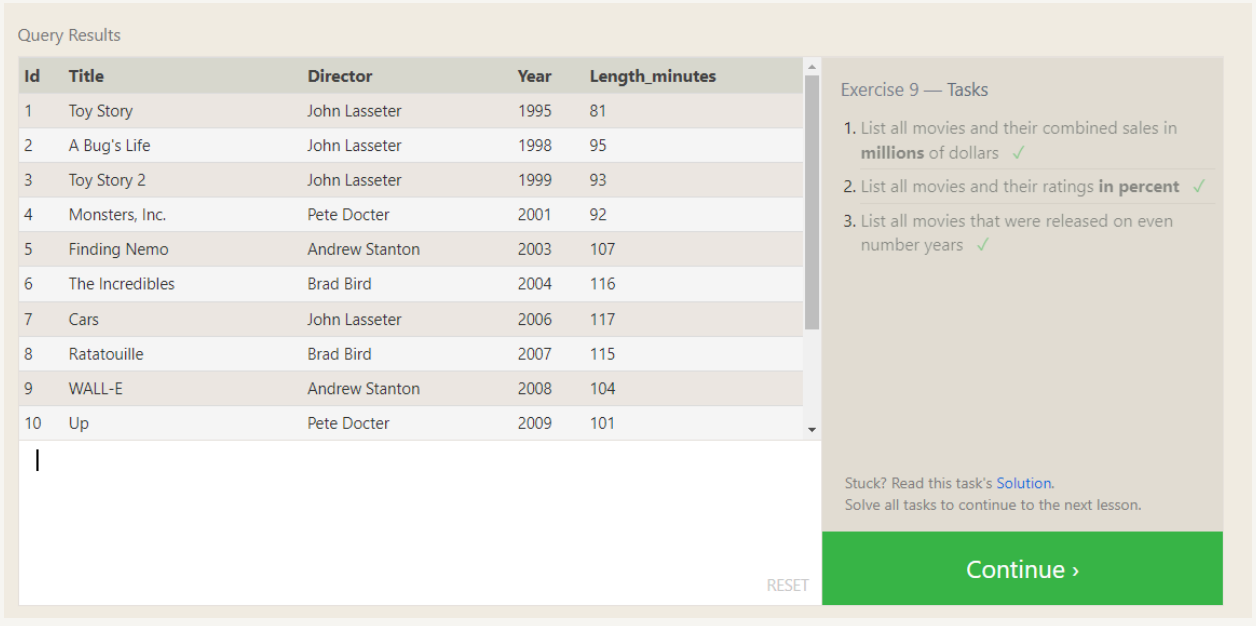
SQL Lesson 8: A short note on NULLs



Answers:

1.SELECT name, role FROM employees WHERE building IS NULL;
2.SELECT DISTINCT building_name FROM buildings LEFT JOIN employees ON building_name = building WHERE role IS NULL;

SQL Lesson 9: Queries with expressions



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

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 ›

Answers:

1.SELECT title, (domestic_sales + international_sales) / 1000000 AS gross_sales_million FROM movies JOIN boxoffice ON movies.id = boxoffice.movie_id;
2.SELECT title, rating * 10 AS rating_percent FROM movies JOIN boxoffice ON movies.id = boxoffice.movie_id;
3.SELECT title, year FROM movies WHERE year % 2 = 0;

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

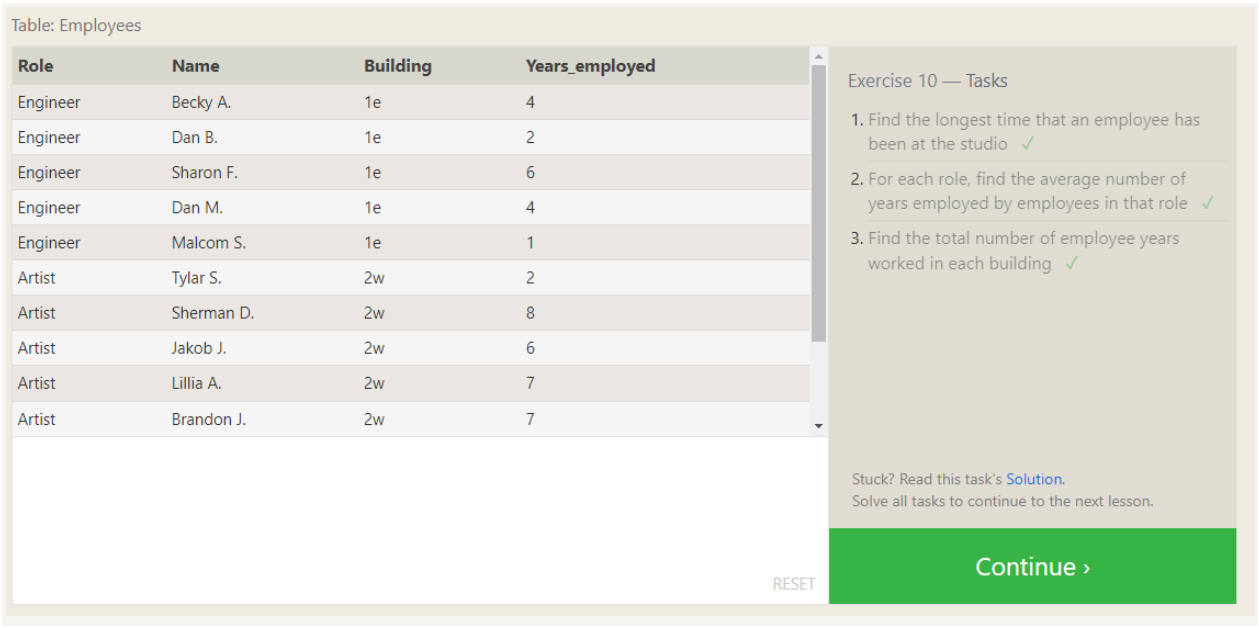


Table: Employees

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

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 ›

Answers:

1.SELECT MAX(years_employed) as Max_years_employed FROM employees;
2.SELECT role, AVG(years_employed) as Average_years_employed FROM employees GROUP BY role;
3.SELECT building, SUM(years_employed) as Total_years_employed FROM employees GROUP BY building;

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

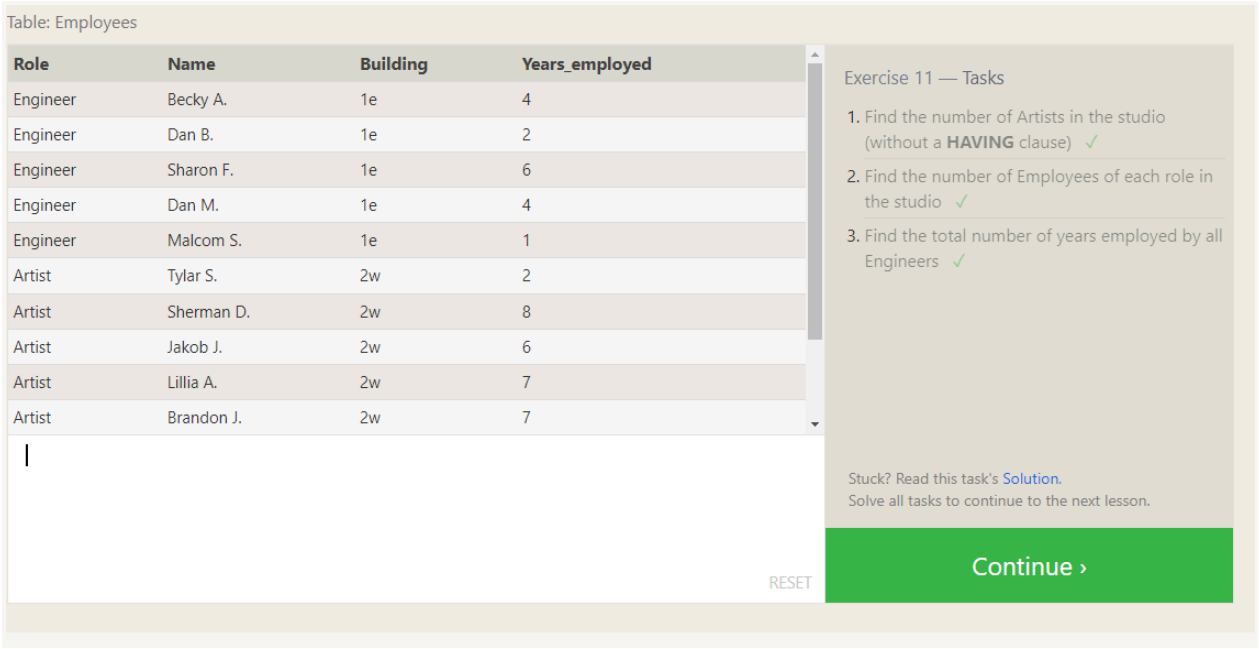


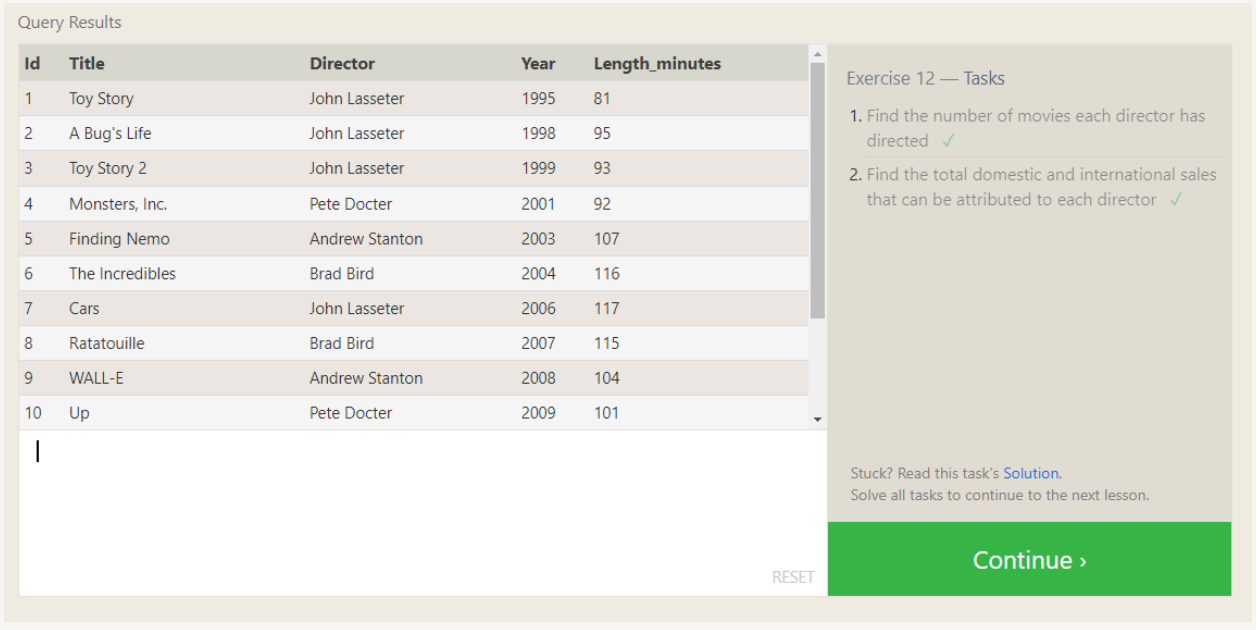
Table: Employees

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

Answers:

1.SELECT role, COUNT(*) as Number_of_artists FROM employees WHERE role = "Artist";
2.SELECT role, COUNT(*) FROM employees GROUP BY role;
3.SELECT role, SUM(years_employed) FROM employees GROUP BY role HAVING role = "Engineer";

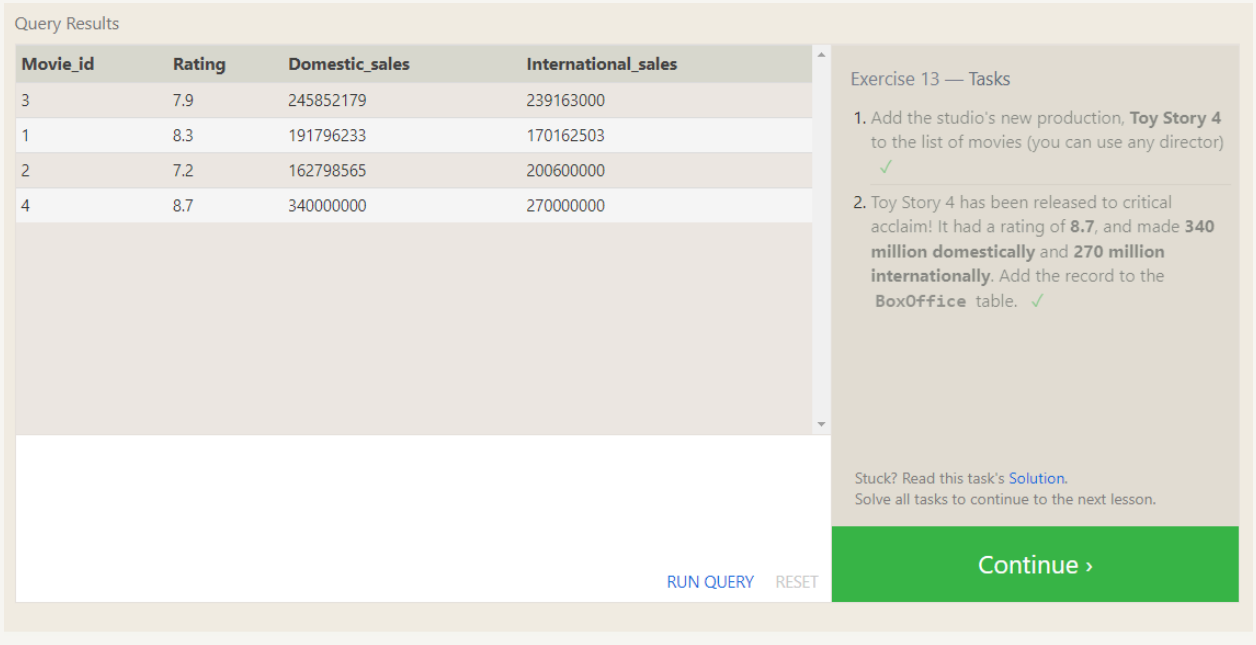
SQL Lesson 12: Order of execution of a Query



Answers:

1.SELECT director, COUNT(id) as Num_movies_directed FROM movies GROUP BY director;
2.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;
3.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;

SQL Lesson 13: Inserting rows



Answers:

1.INSERT INTO movies VALUES (4, "Toy Story 4", "El Directore", 2015, 90);
2.INSERT INTO boxoffice VALUES (4, 8.7, 340000000, 270000000);

SQL Lesson 14: Updating rows

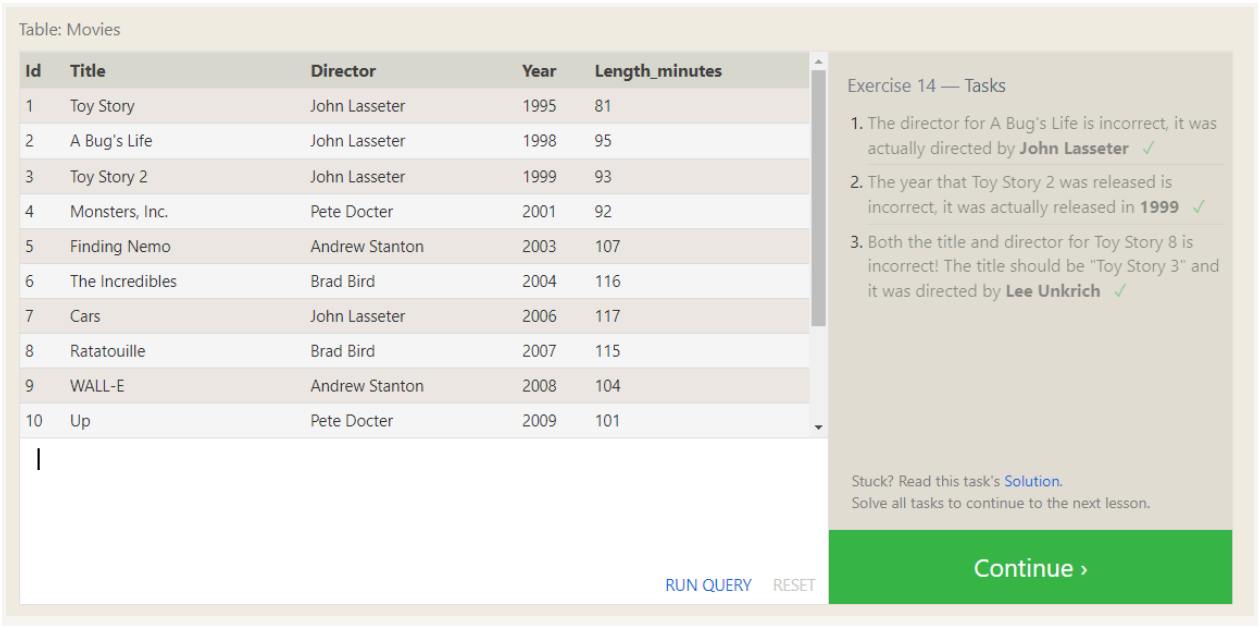


Table: Movies

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

Answers:

1.UPDATE movies SET director = "John Lasseter" WHERE id = 2;
2.UPDATE movies SET year = 1999 WHERE id = 3;
3.UPDATE movies SET title = "Toy Story 3", director = "Lee Unkrich" WHERE id = 11;

SQL Lesson 15: Deleting rows

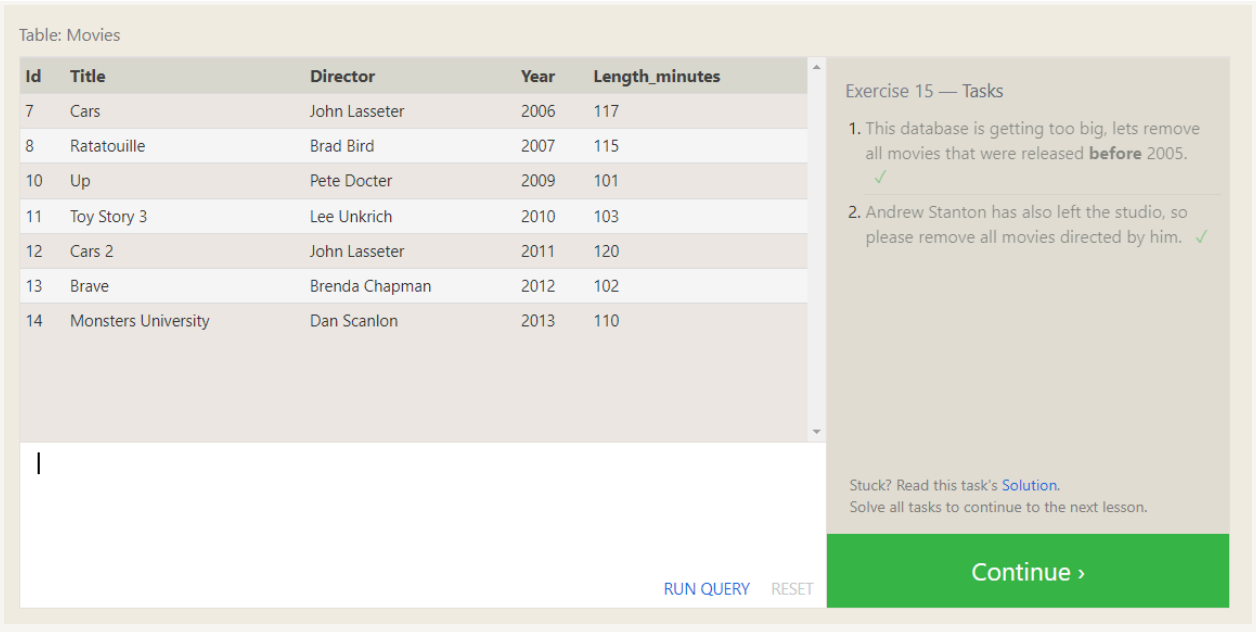


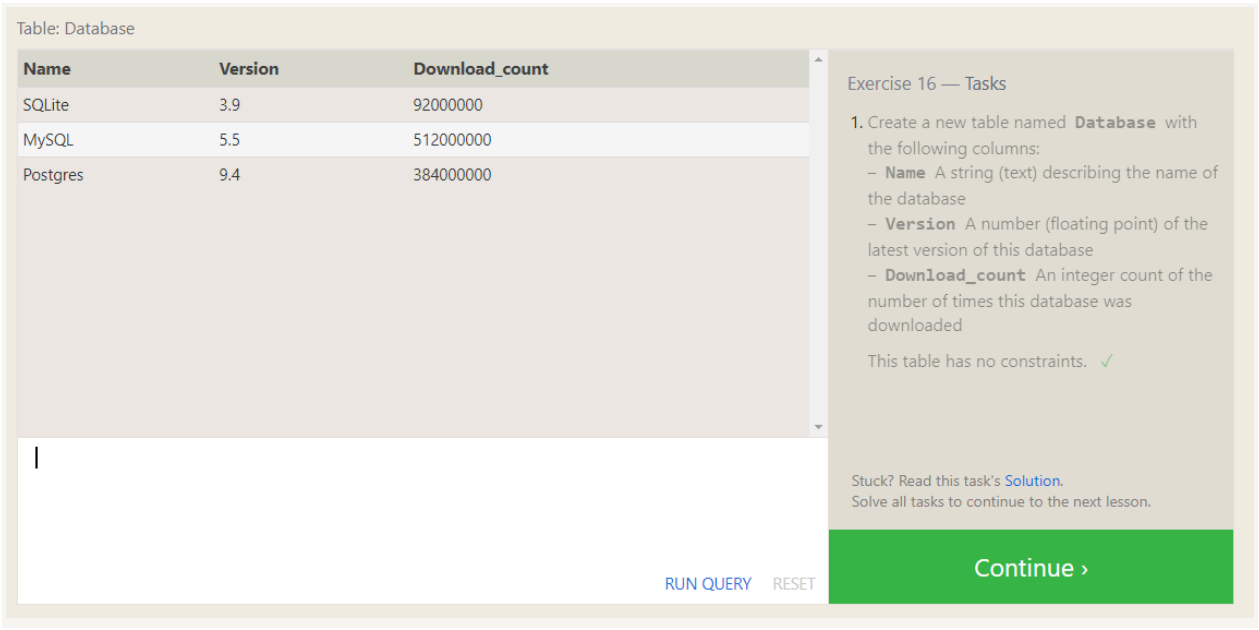
Table: Movies

Id	Title	Director	Year	Length_minutes
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
10	Up	Pete Docter	2009	101
11	Toy Story 3	Lee Unkrich	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

Answers:

1.DELETE FROM movies where year < 2005;
2.DELETE FROM movies where director = "Andrew Stanton";

SQL Lesson 16: Creating tables



Answers:

```
CREATE TABLE Database (
    Name TEXT,
    Version FLOAT,
    Download_count INTEGER
);
```

SQL Lesson 17: Altering tables

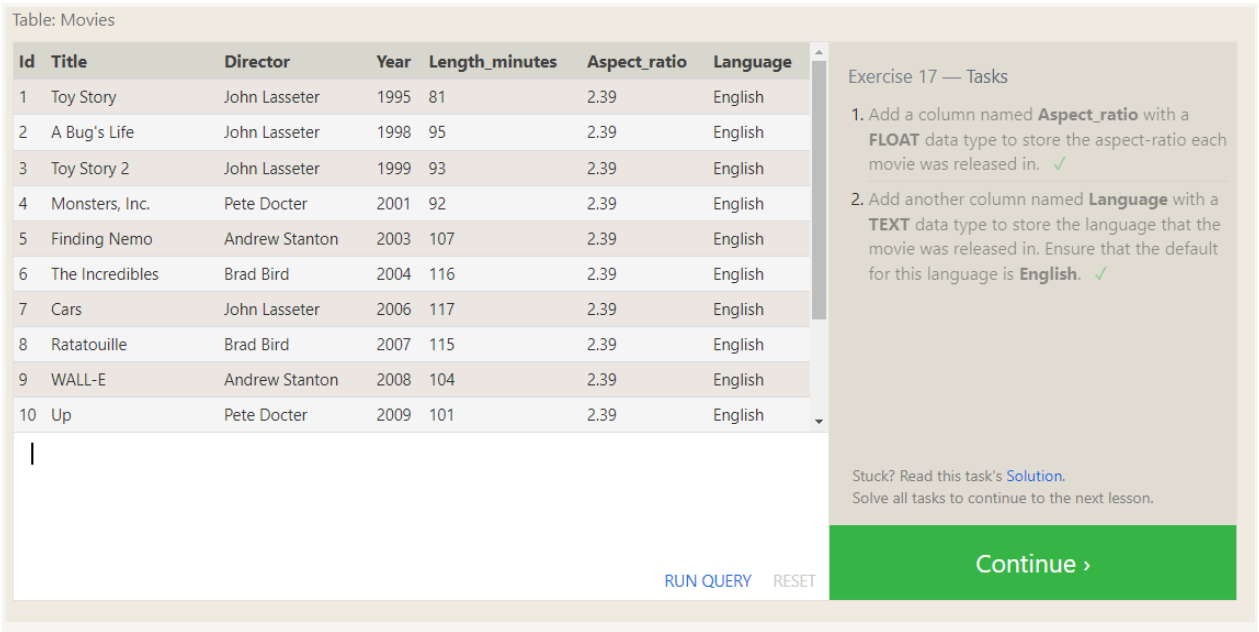


Table: Movies

Id	Title	Director	Year	Length_minutes	Aspect_ratio	Language
1	Toy Story	John Lasseter	1995	81	2.39	English
2	A Bug's Life	John Lasseter	1998	95	2.39	English
3	Toy Story 2	John Lasseter	1999	93	2.39	English
4	Monsters, Inc.	Pete Docter	2001	92	2.39	English
5	Finding Nemo	Andrew Stanton	2003	107	2.39	English
6	The Incredibles	Brad Bird	2004	116	2.39	English
7	Cars	John Lasseter	2006	117	2.39	English
8	Ratatouille	Brad Bird	2007	115	2.39	English
9	WALL-E	Andrew Stanton	2008	104	2.39	English
10	Up	Pete Docter	2009	101	2.39	English

RUN QUERY RESET

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. ✓
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**. ✓

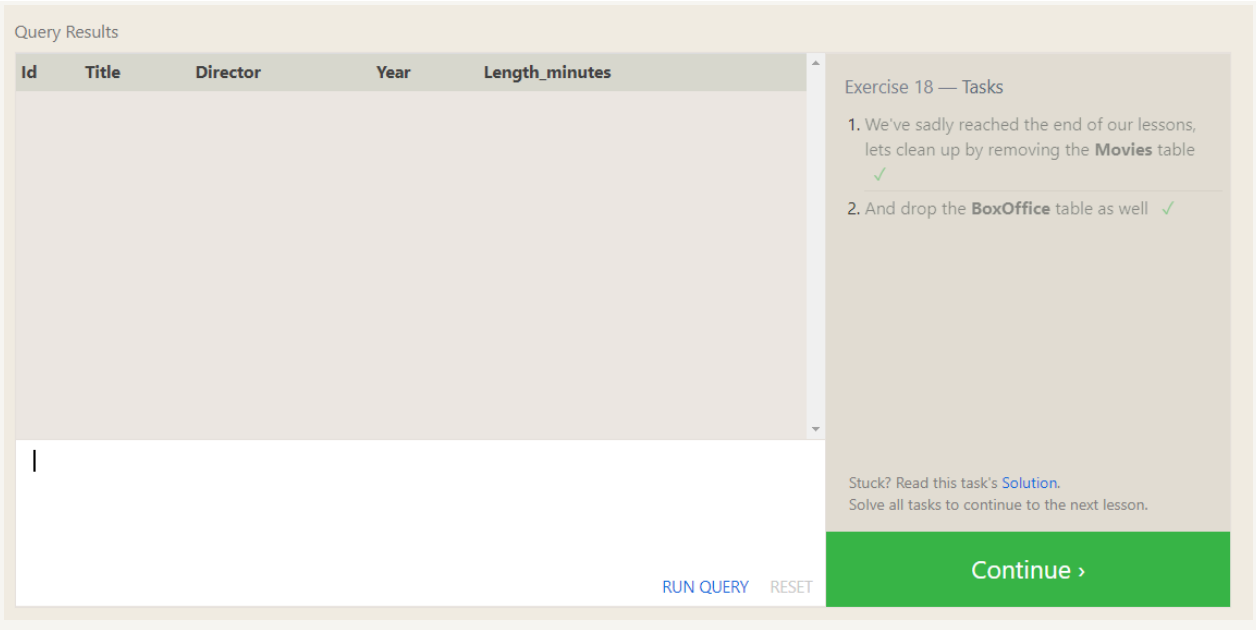
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

Answers:

1.ALTER TABLE Movies
  ADD COLUMN Aspect_ratio FLOAT DEFAULT 2.39;
2.ALTER TABLE Movies
  ADD COLUMN Language TEXT DEFAULT "English";

SQL Lesson 18: Dropping tables



Answers:

1.DROP TABLE Movies;
2.DROP TABLE BoxOffice;