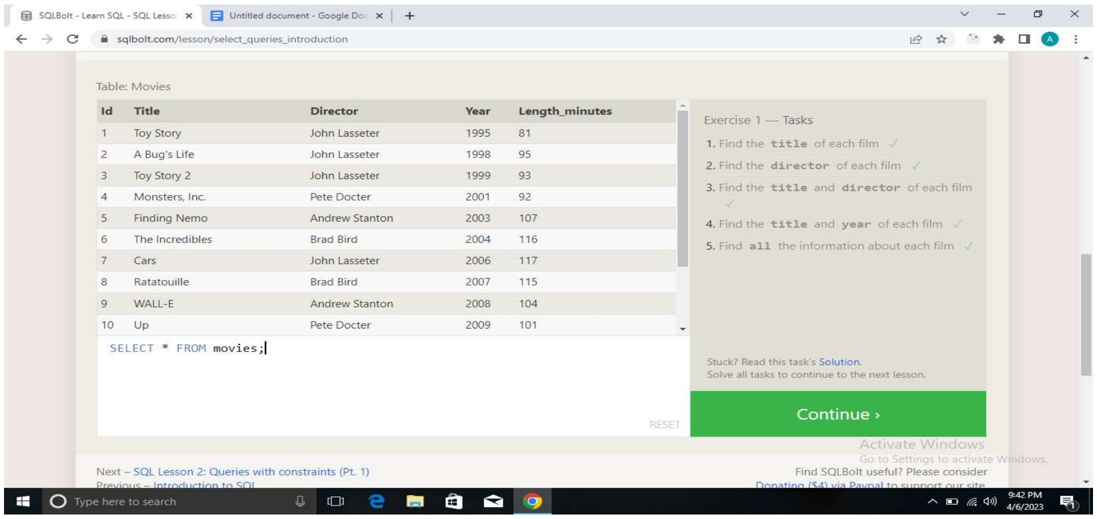
MYSQL TASK 2 SQLBOLT

Lesson 1

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;



SQLBolt - Learn SQL - SQL Lesso | Untitled document - Google Doc

sqlbolt.com/lesson/select_queries_introduction

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;
```

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 ✓

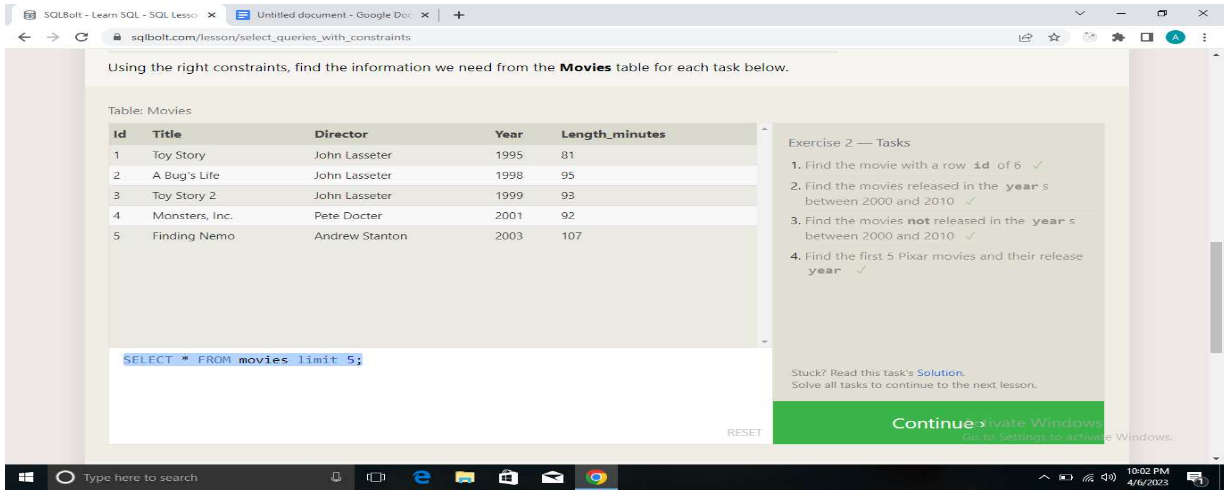
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Next – SQL Lesson 2: Queries with constraints (Pt. 1)
Previous – Introduction to SQL

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Lesson 2

1. SELECT * FROM movies WHERE id=6;
2. SELECT * FROM movies WHERE year between 2000 and 2010;
3. SELECT * FROM movies WHERE not year between 2000 and 2010;
4. SELECT * FROM movies limit 5;



SQLBolt - Learn SQL - SQL Lesso | Untitled document - Google Doc

sqlbolt.com/lesson/select_queries_with_constraints

Using the right constraints, find the information we need from the **Movies** table for each task below.

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

```
SELECT * FROM movies limit 5;
```

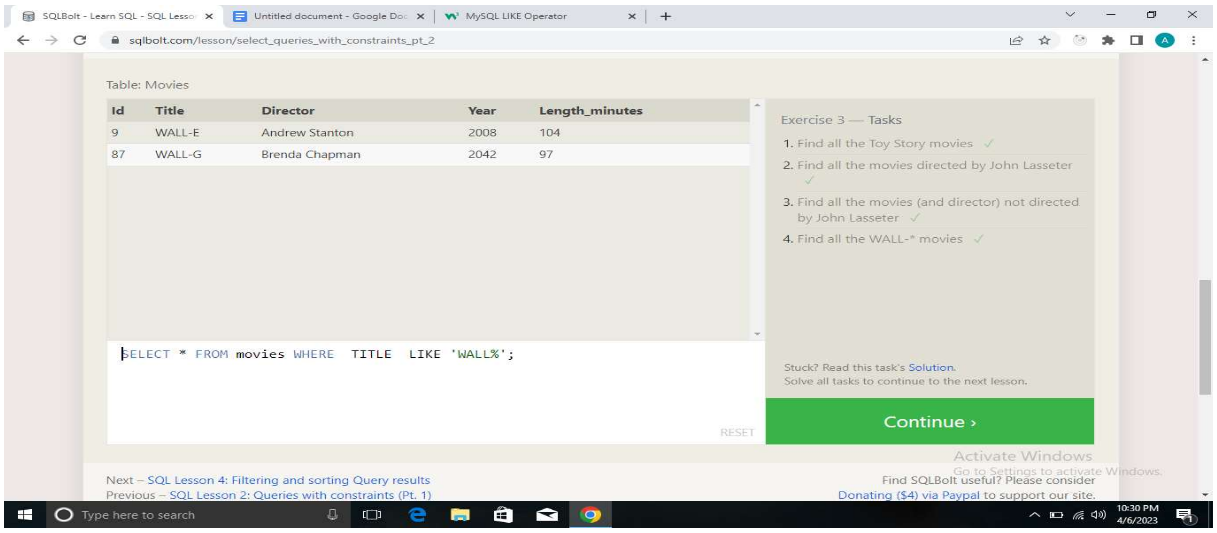
Exercise 2 — Tasks

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.

Lesson 3

1. SELECT * FROM movies WHERE TITLE LIKE 'TOY%';
2. SELECT * FROM movies WHERE DIRECTOR LIKE 'jOHN %';
3. SELECT * FROM movies WHERE NOT DIRECTOR LIKE'jOHN %';
4. SELECT * FROM movies WHERE TITLE LIKE 'WALL%';



sqlbolt.com/lesson/select_queries_with_constraints_pt_2

Table: Movies

Id	Title	Director	Year	Length_minutes
9	WALL-E	Andrew Stanton	2008	104
87	WALL-G	Brenda Chapman	2042	97

```
SELECT * FROM movies WHERE  TITLE  LIKE 'WALL%';
```

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 ✓

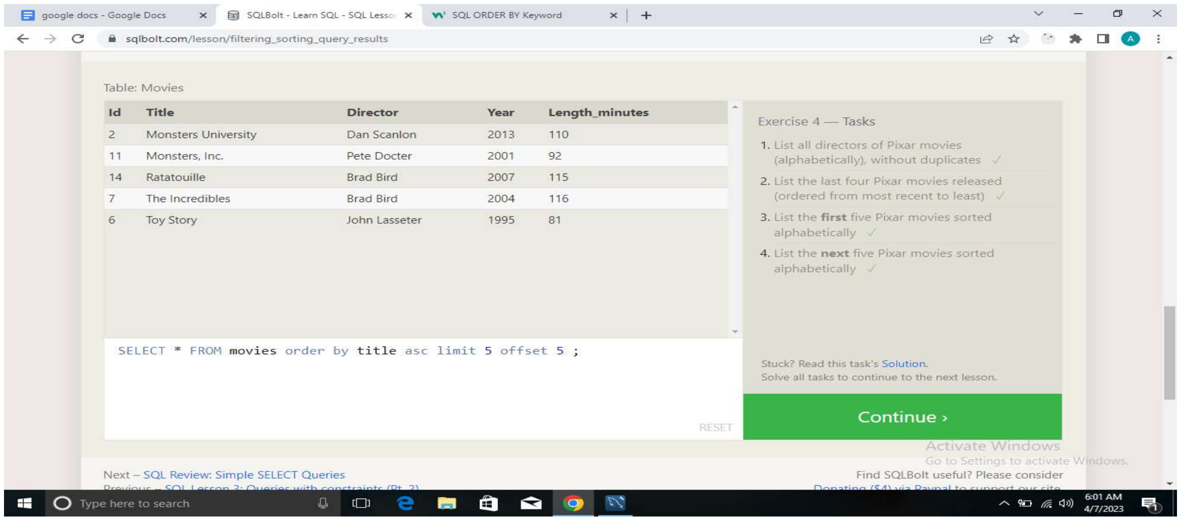
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Next – SQL Lesson 4: Filtering and sorting Query results
Previous – SQL Lesson 2: Queries with constraints (Pt. 1)

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Lesson 4

1. SELECT * FROM movies group by director;
2. SELECT * FROM movies order by year desc limit 4 ;
3. SELECT * FROM movies order by title asc limit 5 offset 0 ;
4. SELECT * FROM movies order by title asc limit 5 offset 5 ;



sqlbolt.com/lesson/filtering_sorting_query_results

Table: Movies

Id	Title	Director	Year	Length_minutes
2	Monsters University	Dan Scanlon	2013	110
11	Monsters, Inc.	Pete Docter	2001	92
14	Ratatouille	Brad Bird	2007	115
7	The Incredibles	Brad Bird	2004	116
6	Toy Story	John Lasseter	1995	81

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

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.

Next – SQL Review: Simple SELECT Queries

Find SQLBolt useful? Please consider

Lesson 5

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

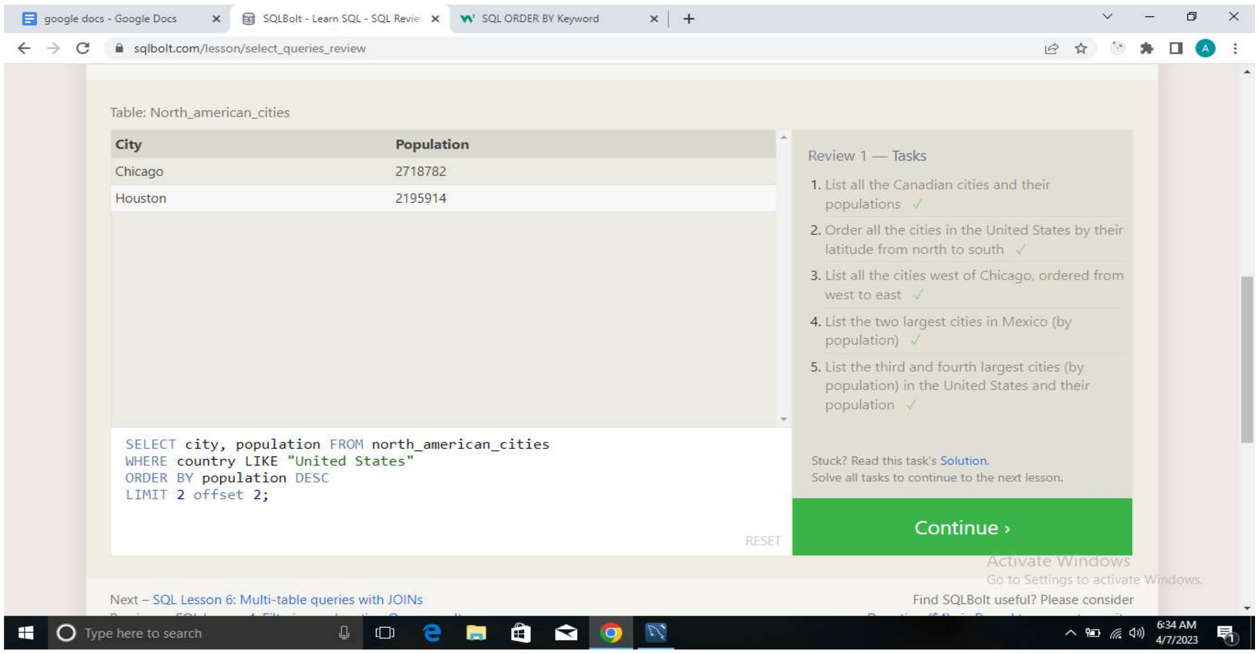

google docs - Google Docs
SQLBolt - Learn SQL - SQL Review
SQL ORDER BY Keyword
sqlbolt.com/lesson/select_queries_review

Table: North_american_cities

City	Population
Chicago	2718782
Houston	2195914

```
SELECT city, population FROM north_american_cities
WHERE country LIKE "United States"
ORDER BY population DESC
LIMIT 2 offset 2;
```

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 ✓

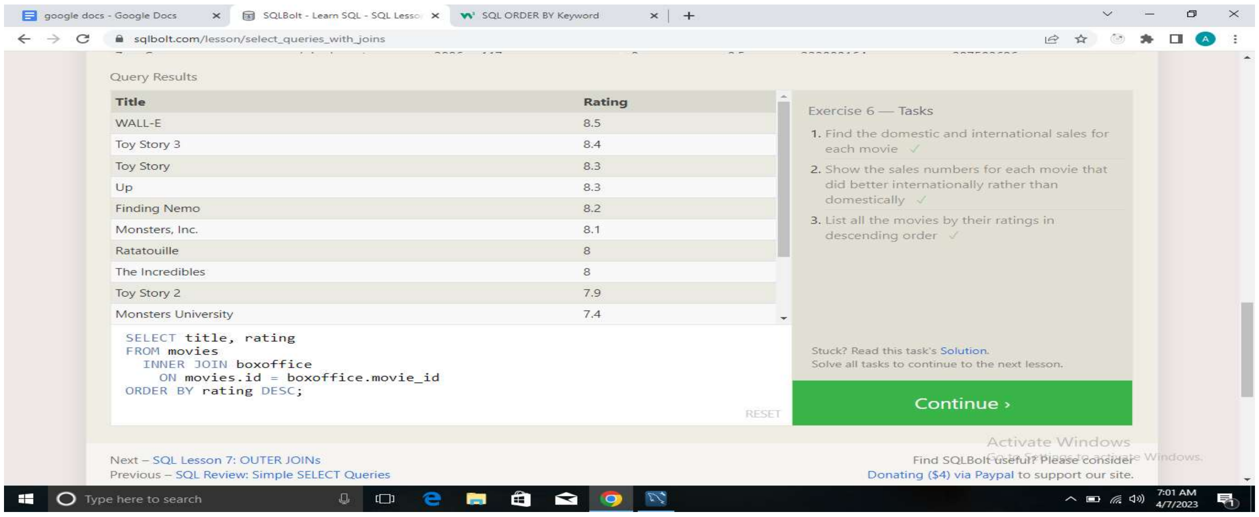
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Next – SQL Lesson 6: Multi-table queries with JOINs

Find SQLBolt useful? Please consider

Lesson 6

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


google docs - Google Docs | SQLBolt - Learn SQL - SQL Lesso | SQL ORDER BY Keyword

sqlbolt.com/lesson/select_queries_with_joins

Query Results

Title	Rating
WALL-E	8.5
Toy Story 3	8.4
Toy Story	8.3
Up	8.3
Finding Nemo	8.2
Monsters, Inc.	8.1
Ratatouille	8
The Incredibles	8
Toy Story 2	7.9
Monsters University	7.4

```
SELECT title, rating
FROM movies
  INNER JOIN boxoffice
    ON movies.id = boxoffice.movie_id
ORDER BY rating DESC;
```

Exercise 6 — Tasks

1. Find the domestic and international sales for each movie ✓
2. Show the sales numbers for each movie that did better internationally rather than domestically ✓
3. List all the movies by their ratings in descending order ✓

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

Continue ›

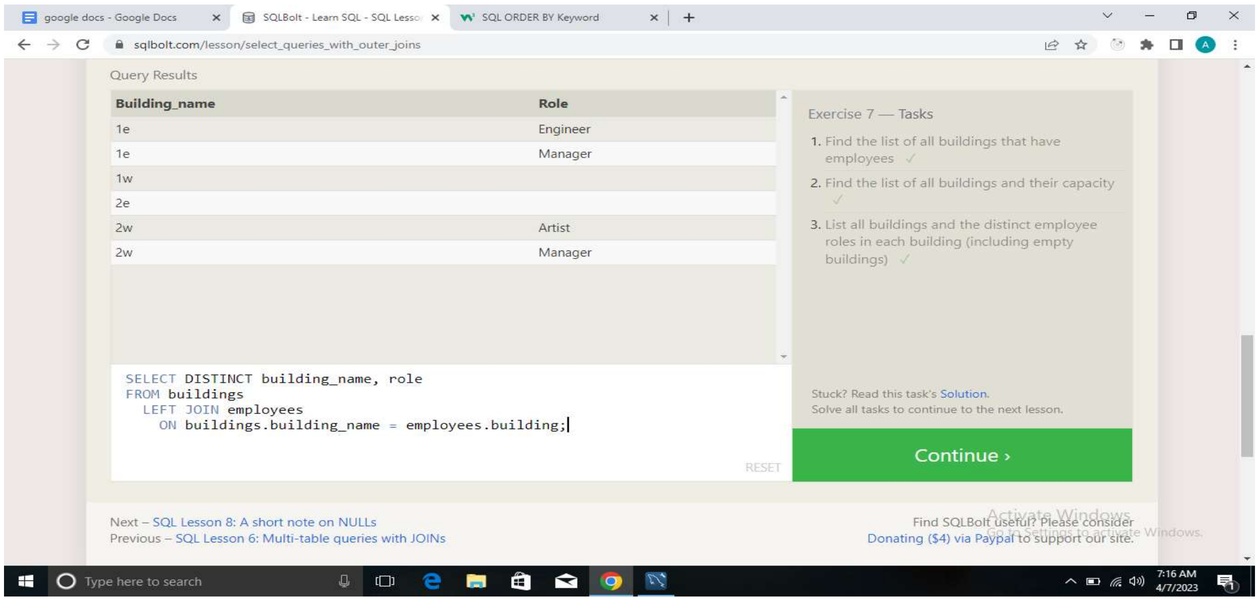
Next – SQL Lesson 7: OUTER JOINs
Previous – SQL Review: Simple SELECT Queries

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Type here to search 7:01 AM 4/7/2023

Lesson 7

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


google docs - Google Docs | SQLBolt - Learn SQL - SQL Lesso | SQL ORDER BY Keyword

sqlbolt.com/lesson/select_queries_with_outer_joins

Query Results

Building_name	Role
1e	Engineer
1e	Manager
1w	
2e	
2w	Artist
2w	Manager

```
SELECT DISTINCT building_name, role
FROM buildings
  LEFT JOIN employees
    ON buildings.building_name = employees.building;
```

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 ›

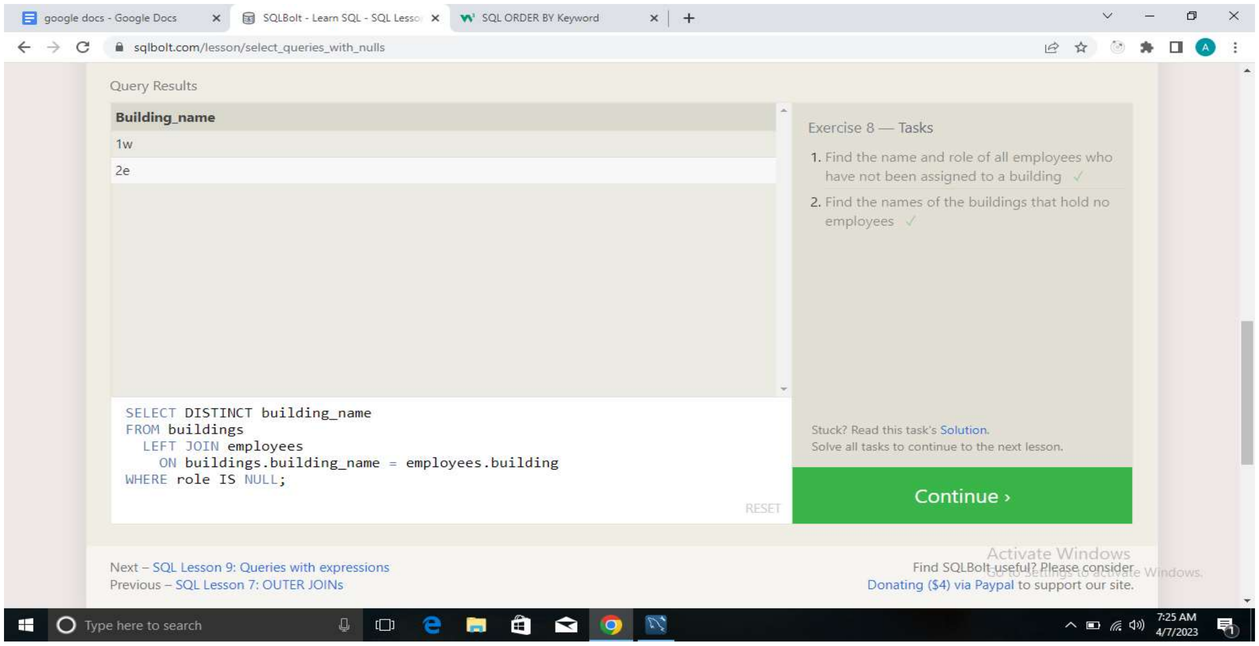
Next – SQL Lesson 8: A short note on NULLs
Previous – SQL Lesson 6: Multi-table queries with JOINs

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Type here to search 7:16 AM 4/7/2023

Lesson 8

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



sqlbolt.com/lesson/select_queries_with_nulls

Query Results

Building_name
1w
2e

```
SELECT DISTINCT building_name
FROM buildings
  LEFT JOIN employees
    ON buildings.building_name = employees.building
WHERE role IS NULL;
```

Exercise 8 — Tasks

1. Find the name and role of all employees who have not been assigned to a building ✓
2. Find the names of the buildings that hold no employees ✓

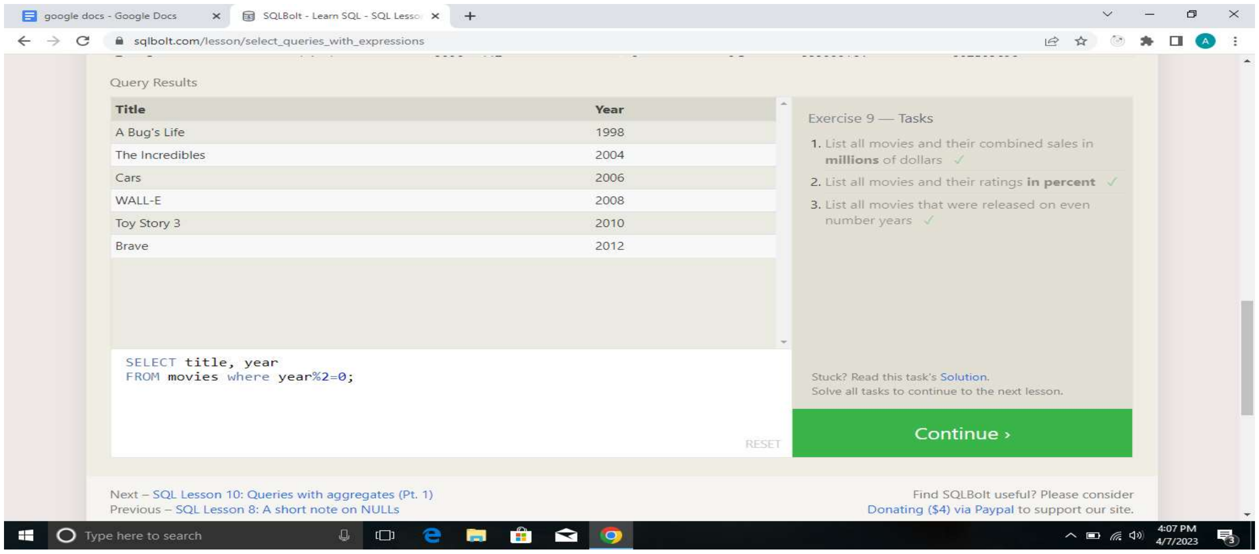
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Next – SQL Lesson 9: Queries with expressions
Previous – SQL Lesson 7: OUTER JOINs

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Lesson 9

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



sqlbolt.com/lesson/select_queries_with_expressions

Query Results

Title	Year
A Bug's Life	1998
The Incredibles	2004
Cars	2006
WALL-E	2008
Toy Story 3	2010
Brave	2012

```
SELECT title, year
FROM movies where year%2=0;
```

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 ✓

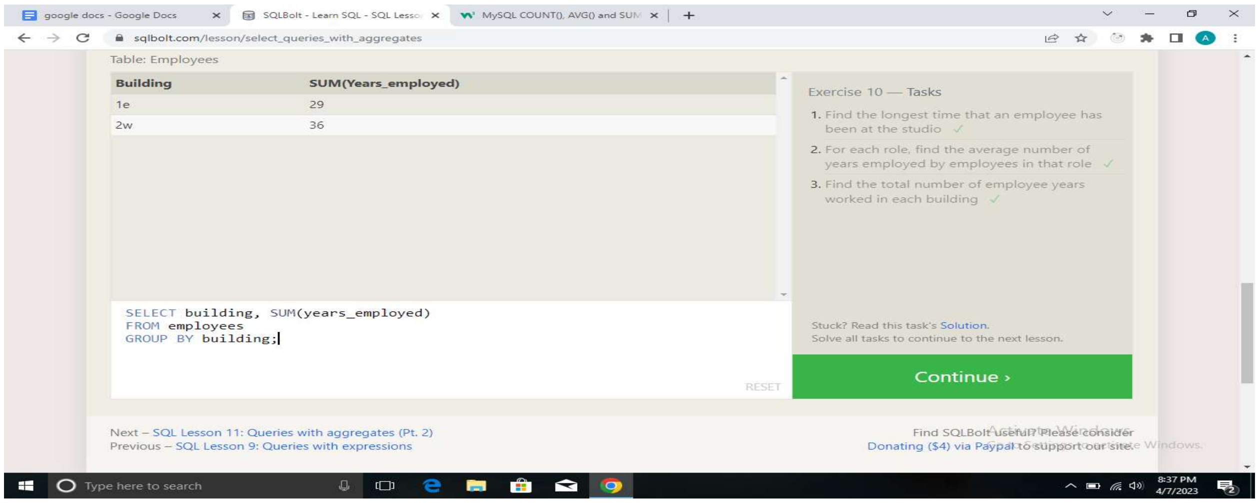
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Next – SQL Lesson 10: Queries with aggregates (Pt. 1)
Previous – SQL Lesson 8: A short note on NULLs

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Lesson 10

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



sqlbolt.com/lesson/select_queries_with_aggregates

Table: Employees

Building	SUM(Years_employed)
1e	29
2w	36

```
SELECT building, SUM(years_employed)
FROM employees
GROUP BY building;
```

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 ✓

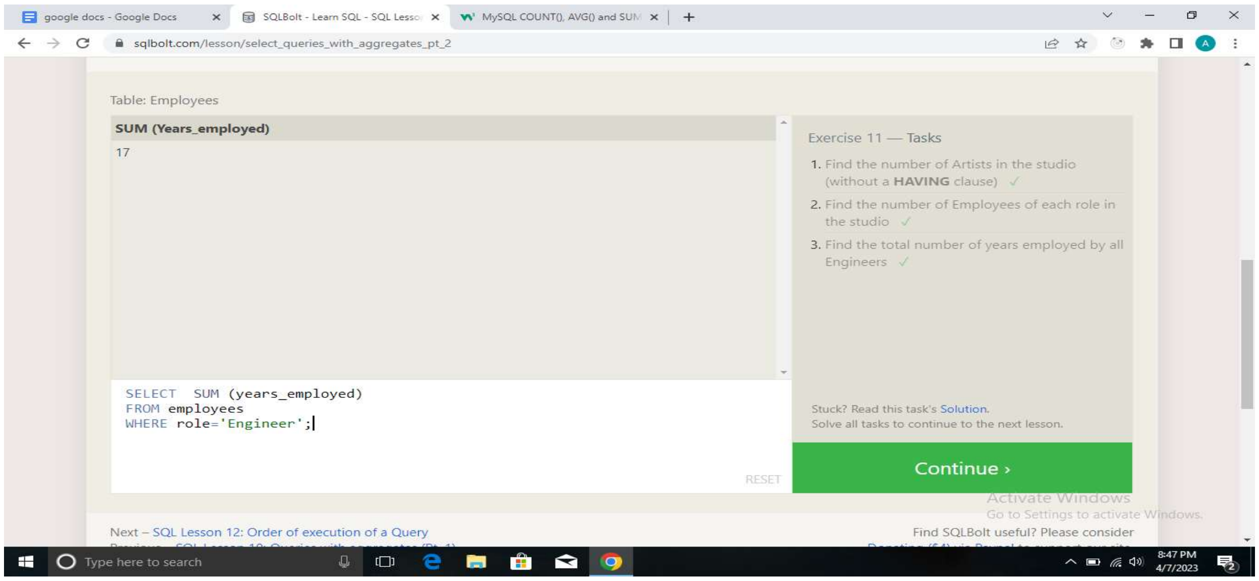
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Next – SQL Lesson 11: Queries with aggregates (Pt. 2)
Previous – SQL Lesson 9: Queries with expressions

Find SQLBolt useful? Please consider
Donating ($4) via Paypal to support our site.

Lesson 11

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



sqlbolt.com/lesson/select_queries_with_aggregates_pt_2

Table: Employees

SUM (Years_employed)
17

```
SELECT  SUM (years_employed)
FROM employees
WHERE role='Engineer';
```

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 ✓

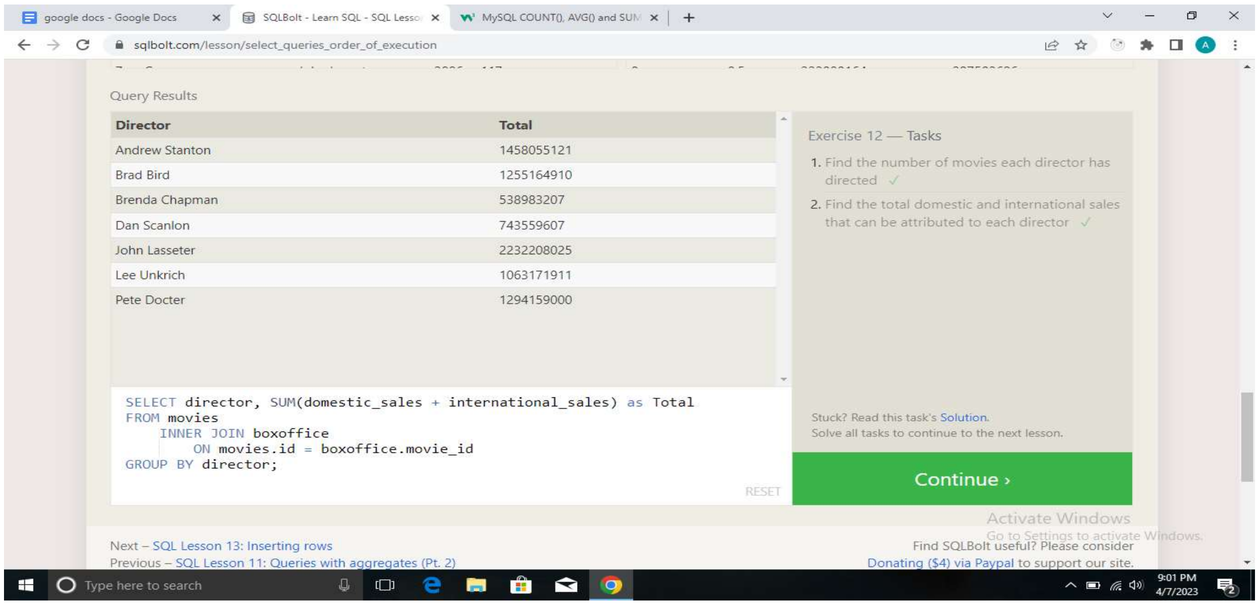
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Next – SQL Lesson 12: Order of execution of a Query

Find SQLBolt useful? Please consider

Lesson 12

1. SELECT director, SUM(id) FROM movies GROUP BY director;
2. SELECT director, SUM(domestic_sales + international_sales) as Total
   FROM movies   INNER JOIN boxoffice   ON movies.id = boxoffice.movie_id
   GROUP BY director;


google docs - Google Docs
SQLBolt - Learn SQL - SQL Lesso
MySQL COUNT(), AVG() and SUM
sqlbolt.com/lesson/select_queries_order_of_execution
Type here to search
9:01 PM
4/7/2023

Query Results

Director	Total
Andrew Stanton	1458055121
Brad Bird	1255164910
Brenda Chapman	538983207
Dan Scanlon	743559607
John Lasseter	2232208025
Lee Unkrich	1063171911
Pete Docter	1294159000

```
SELECT director, SUM(domestic_sales + international_sales) as Total
FROM movies
    INNER JOIN boxoffice
        ON movies.id = boxoffice.movie_id
GROUP BY director;
```

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.

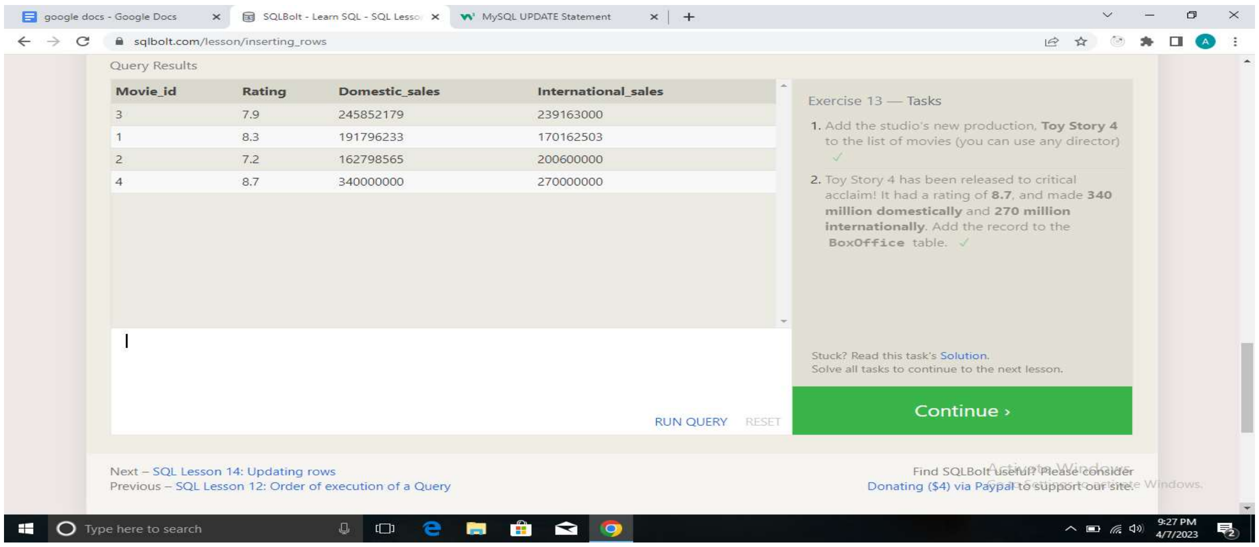
Continue ›

Next – SQL Lesson 13: Inserting rows
Previous – SQL Lesson 11: Queries with aggregates (Pt. 2)

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Lesson 13

1. INSERT INTO movies VALUES (4, "Toy Story 4", "Brad Bird", 2023, 97);
2. INSERT INTO boxoffice VALUES (4, 8.7, 340000000, 270000000);


google docs - Google Docs
SQLBolt - Learn SQL - SQL Lesso
MySQL UPDATE Statement
sqlbolt.com/lesson/inserting_rows
Type here to search
9:27 PM
4/7/2023

Query Results

Movie_id	Rating	Domestic_sales	International_sales
3	7.9	245852179	239163000
1	8.3	191796233	170162503
2	7.2	162798565	200600000
4	8.7	340000000	270000000

RUN QUERY RESET

Exercise 13 — Tasks

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

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

Continue ›

Next – SQL Lesson 14: Updating rows
Previous – SQL Lesson 12: Order of execution of a Query

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Lesson 14

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;

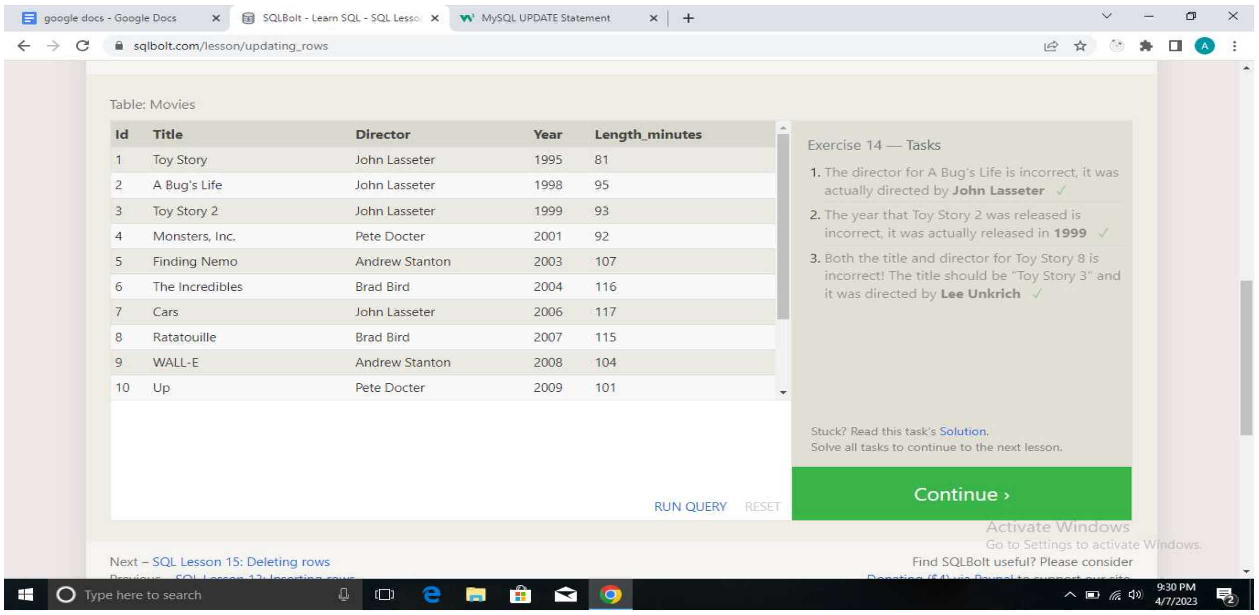

google docs - Google Docs
SQLBolt - Learn SQL - SQL Lesso
MySQL UPDATE Statement
sqlbolt.com/lesson/updating_rows
RUN QUERY
RESET
Continue ›
Activate Windows
Go to Settings to activate Windows.
Type here to search
9:30 PM
4/7/2023

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

Exercise 14 — Tasks

1. The director for A Bug's Life is incorrect, it was actually directed by **John Lasseter** ✓
2. The year that Toy Story 2 was released is incorrect, it was actually released in **1999** ✓
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** ✓

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

Next – SQL Lesson 15: Deleting rows

Find SQLBolt useful? Please consider

Lesson 15

1. DELETE FROM Movies WHERE year < 2005;
2. DELETE FROM Movies WHERE Director ="Andrew Stanton";

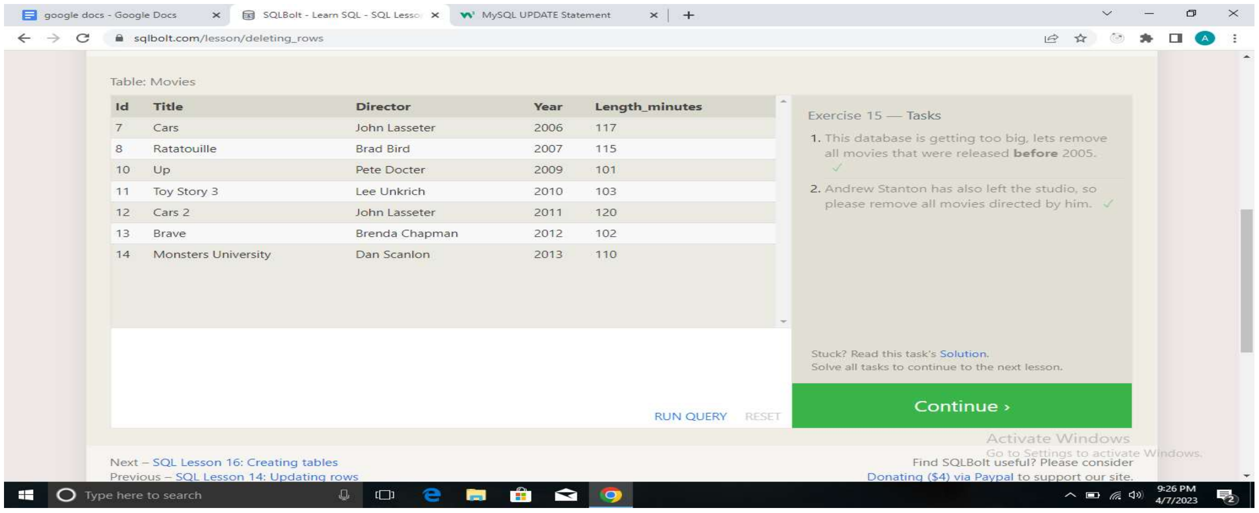

google docs - Google Docs
SQLBolt - Learn SQL - SQL Lesso
MySQL UPDATE Statement
sqlbolt.com/lesson/deleting_rows
RUN QUERY
RESET
Continue ›
Activate Windows
Go to Settings to activate Windows.
Type here to search
9:26 PM
4/7/2023

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

Exercise 15 — Tasks

1. This database is getting too big, lets remove all movies that were released **before** 2005. ✓
2. Andrew Stanton has also left the studio, so please remove all movies directed by him. ✓

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

Next – SQL Lesson 16: Creating tables
Previous – SQL Lesson 14: Updating rows

Find SQLBolt useful? Please consider
Donating ($4) via Paypal to support our site.

Lesson 16

1. CREATE TABLE Database (
   Name varchar(255),
   Version int,
   Download_count varchar(255));

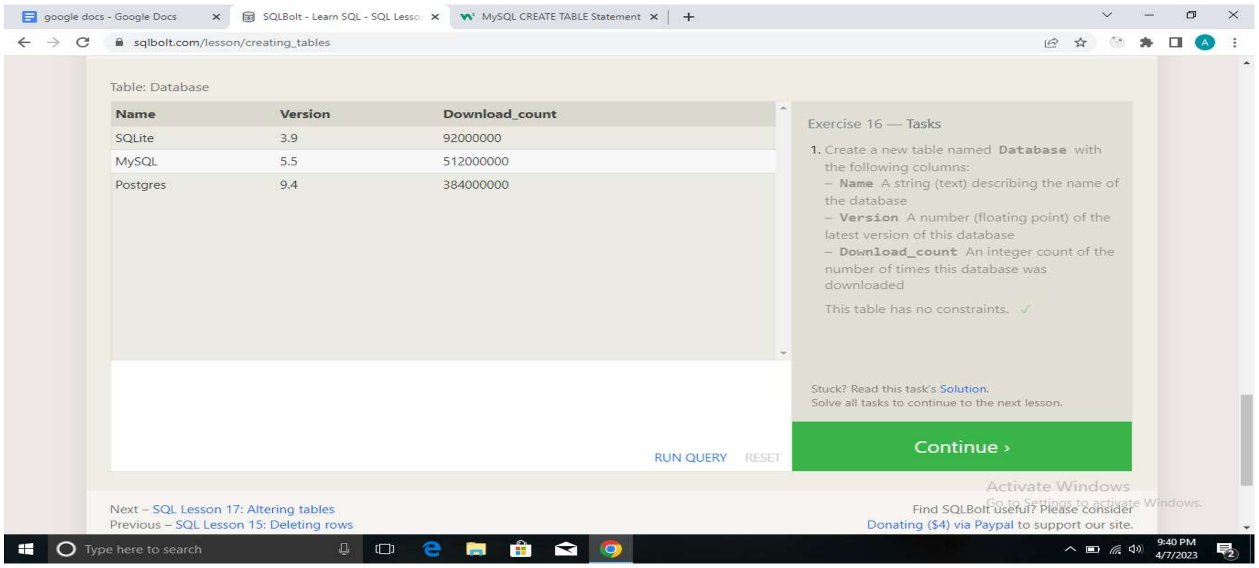

google docs - Google Docs
SQLBolt - Learn SQL - SQL Lesso
MySQL CREATE TABLE Statement
sqlbolt.com/lesson/creating_tables
RUN QUERY
RESET
Continue ›
Type here to search
9:40 PM
4/7/2023

Table: Database

Name	Version	Download_count
SQLite	3.9	92000000
MySQL	5.5	512000000
Postgres	9.4	384000000

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. ✓

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

Activate Windows
Go to Settings to activate Windows.

Next – SQL Lesson 17: Altering tables
Previous – SQL Lesson 15: Deleting rows

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Lesson 17

1. ALTER TABLE Movies ADD COLUMN Aspect_ratio float default 4.8;
2. ALTER TABLE Movies ADD COLUMN Language varchar(255) default “English”;

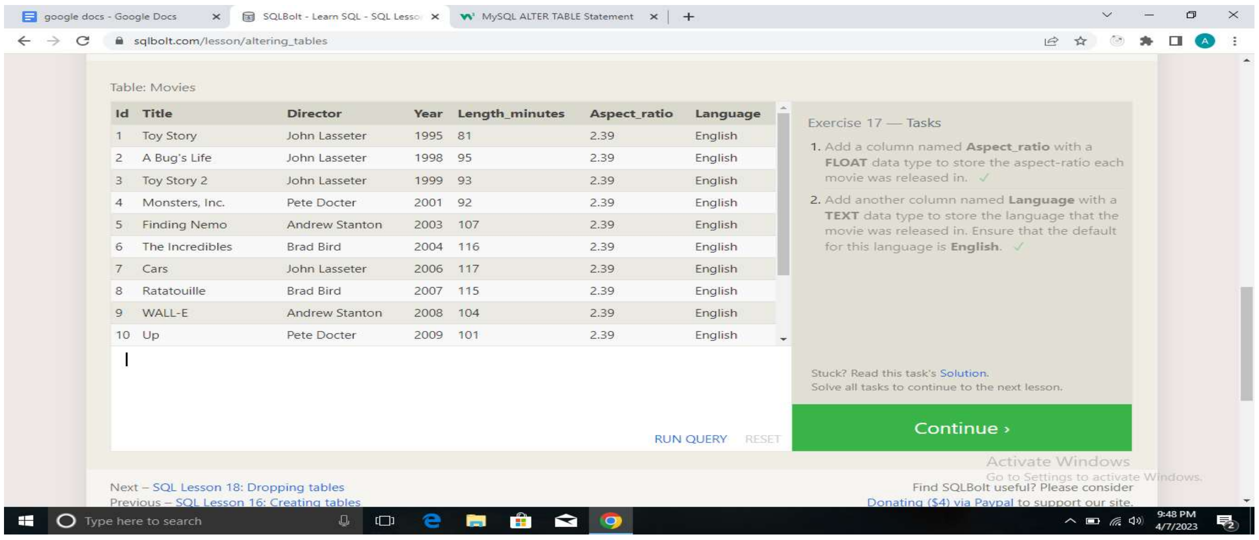

google docs - Google Docs
SQLBolt - Learn SQL - SQL Lesso
MySQL ALTER TABLE Statement
sqlbolt.com/lesson/altering_tables
RUN QUERY
RESET
Continue ›
Type here to search
9:48 PM
4/7/2023

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

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.

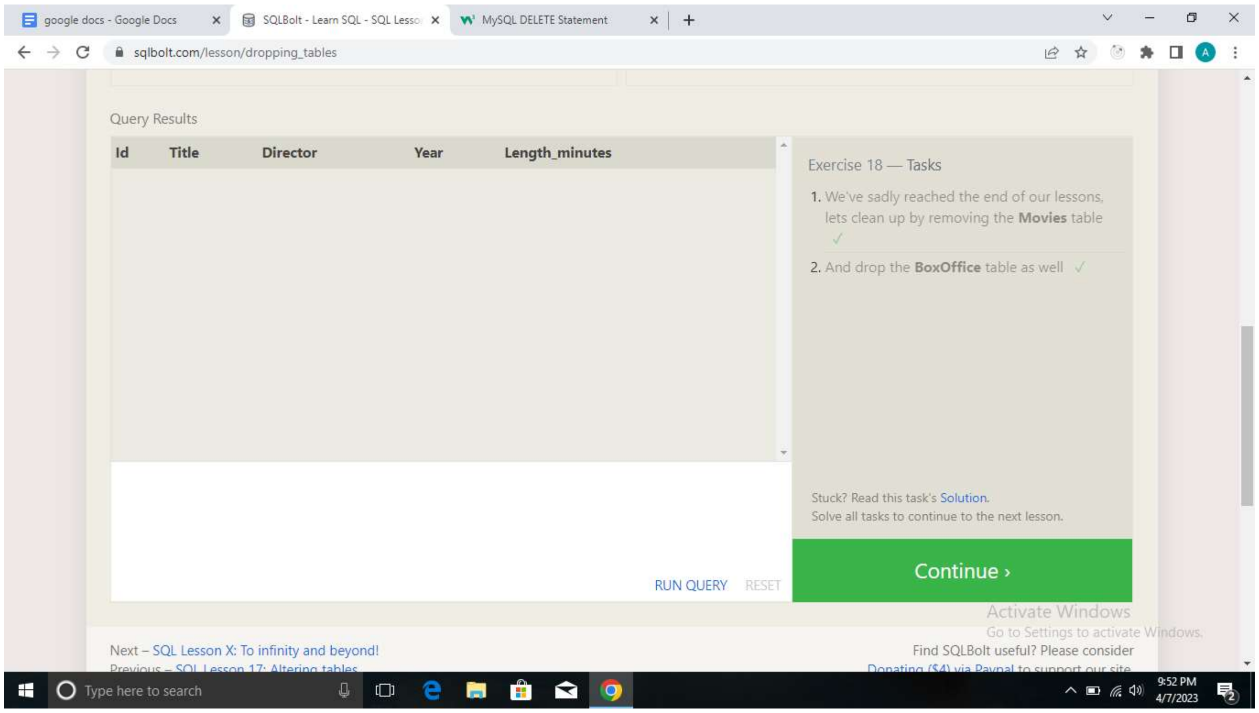
Activate Windows
Go to Settings to activate Windows.

Next – SQL Lesson 18: Dropping tables
Previous – SQL Lesson 16: Creating tables

Find SQLBolt useful? Please consider Donating ($4) via Paypal to support our site.

Lesson 18

1. DROP TABLE Movies;
2. DROP TABLE Boxoffice;


google docs - Google Docs
SQLBolt - Learn SQL - SQL Lesso
MySQL DELETE Statement
sqlbolt.com/lesson/dropping_tables
Query Results

Id	Title	Director	Year	Length_minutes

Exercise 18 — Tasks

1. We've sadly reached the end of our lessons, lets clean up by removing the **Movies** table ✓
2. And drop the **BoxOffice** table as well ✓

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

RUN QUERY
RESET
Continue ›
Activate Windows
Go to Settings to activate Windows.

Next – SQL Lesson X: To infinity and beyond!
Previous – SQL Lesson 17: Altering tables

Find SQLBolt useful? Please consider
Donating ($4) via Paypal to support our site.