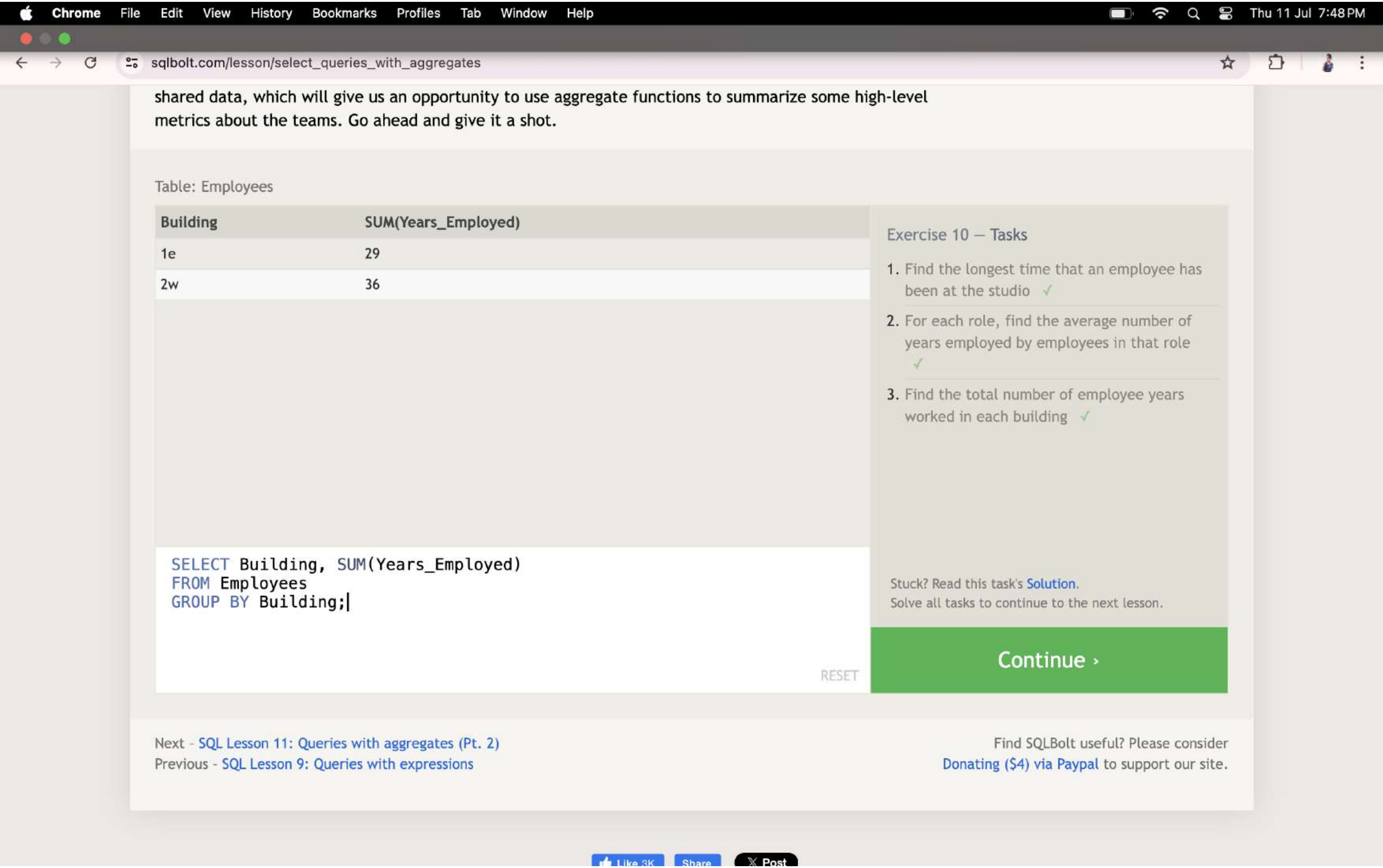


shared data, which will give us an opportunity to use aggregate functions to summarize some high-level metrics about the teams. Go ahead and give it a shot.

Table: Employees

Building	SUM(Years_Employed)
1e	29
2w	36

```
SELECT Building, SUM(Years_Employed)
FROM Employees
GROUP BY Building;
```

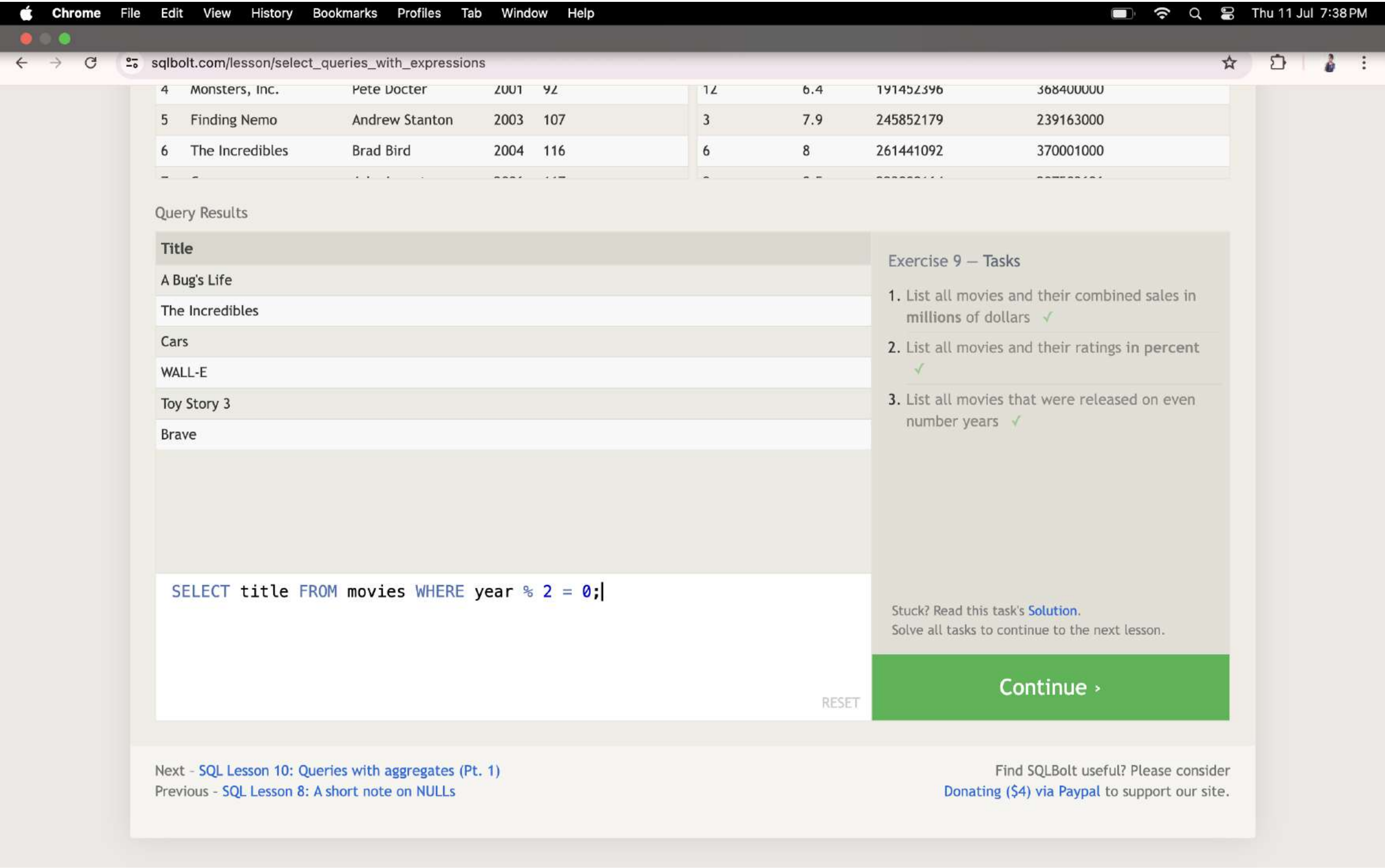
Exercise 10 — Tasks

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.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 11 Jul 7:38 PM
sqlbolt.com/lesson/select_queries_with_expressions

4	Monsters, Inc.	Pete Docter	2001	92	12	6.4	191452396	368400000
5	Finding Nemo	Andrew Stanton	2003	107	3	7.9	245852179	239163000
6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000

Query Results

Title
A Bug's Life
The Incredibles
Cars
WALL-E
Toy Story 3
Brave

```
SELECT title FROM movies WHERE year % 2 = 0;
```

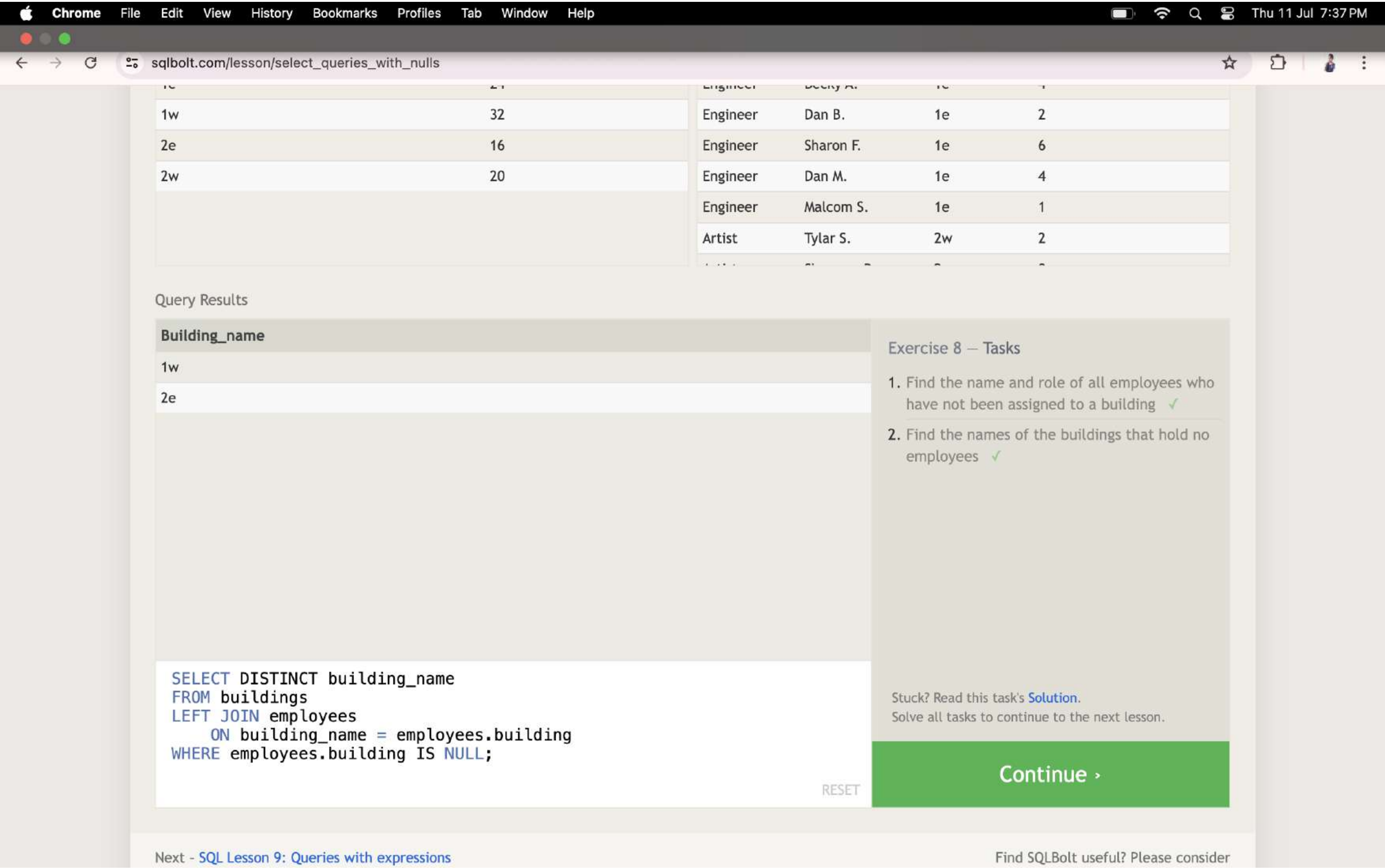
Exercise 9 — Tasks

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.



1w	32
2e	16
2w	20

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

Query Results

Building_name
1w
2e

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

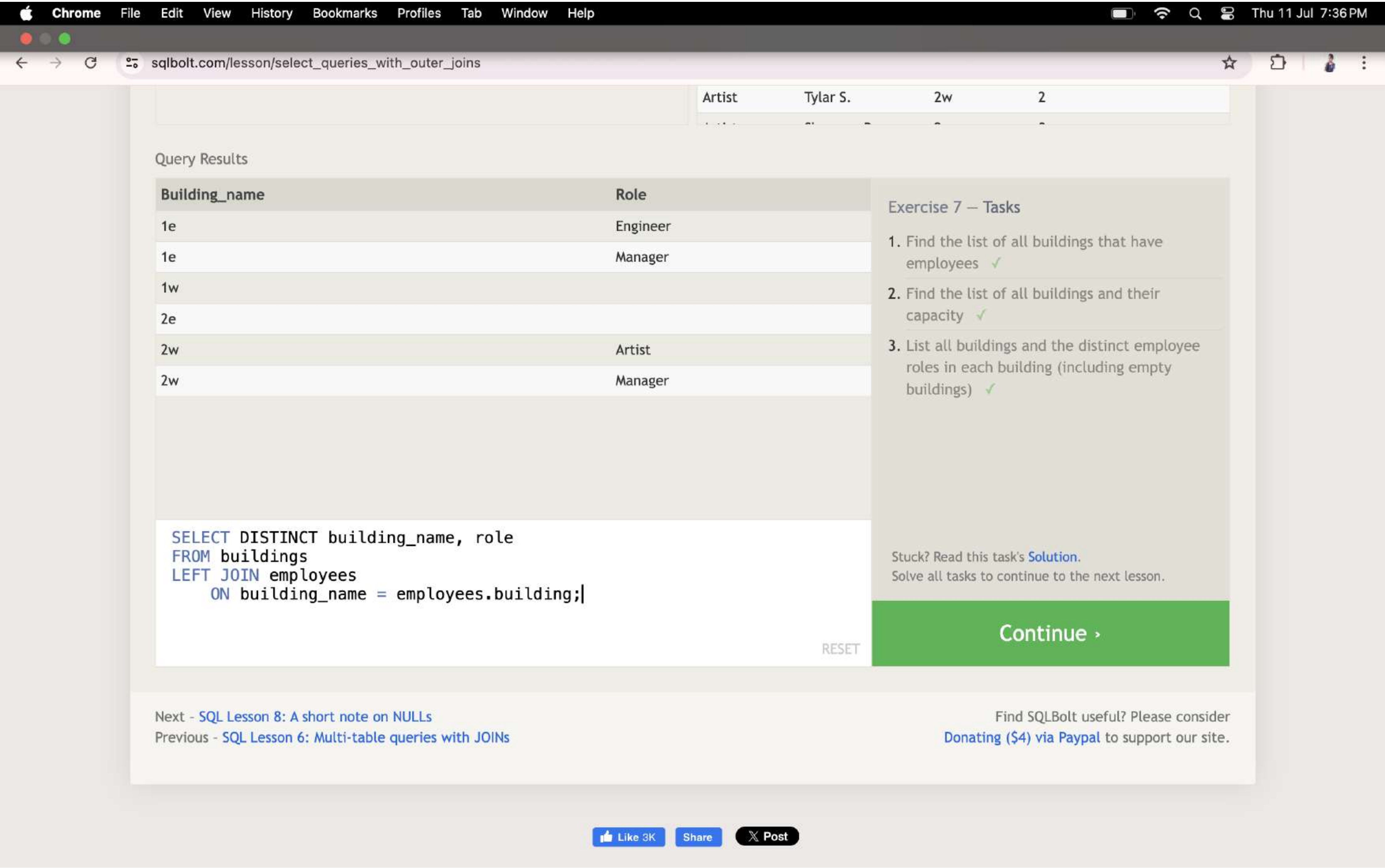
Exercise 8 — Tasks

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

Find SQLBolt useful? Please consider



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 11 Jul 7:36 PM
sqlbolt.com/lesson/select_queries_with_outer_joins

Artist	Tylar S.	2w	2

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 building_name = employees.building;
```

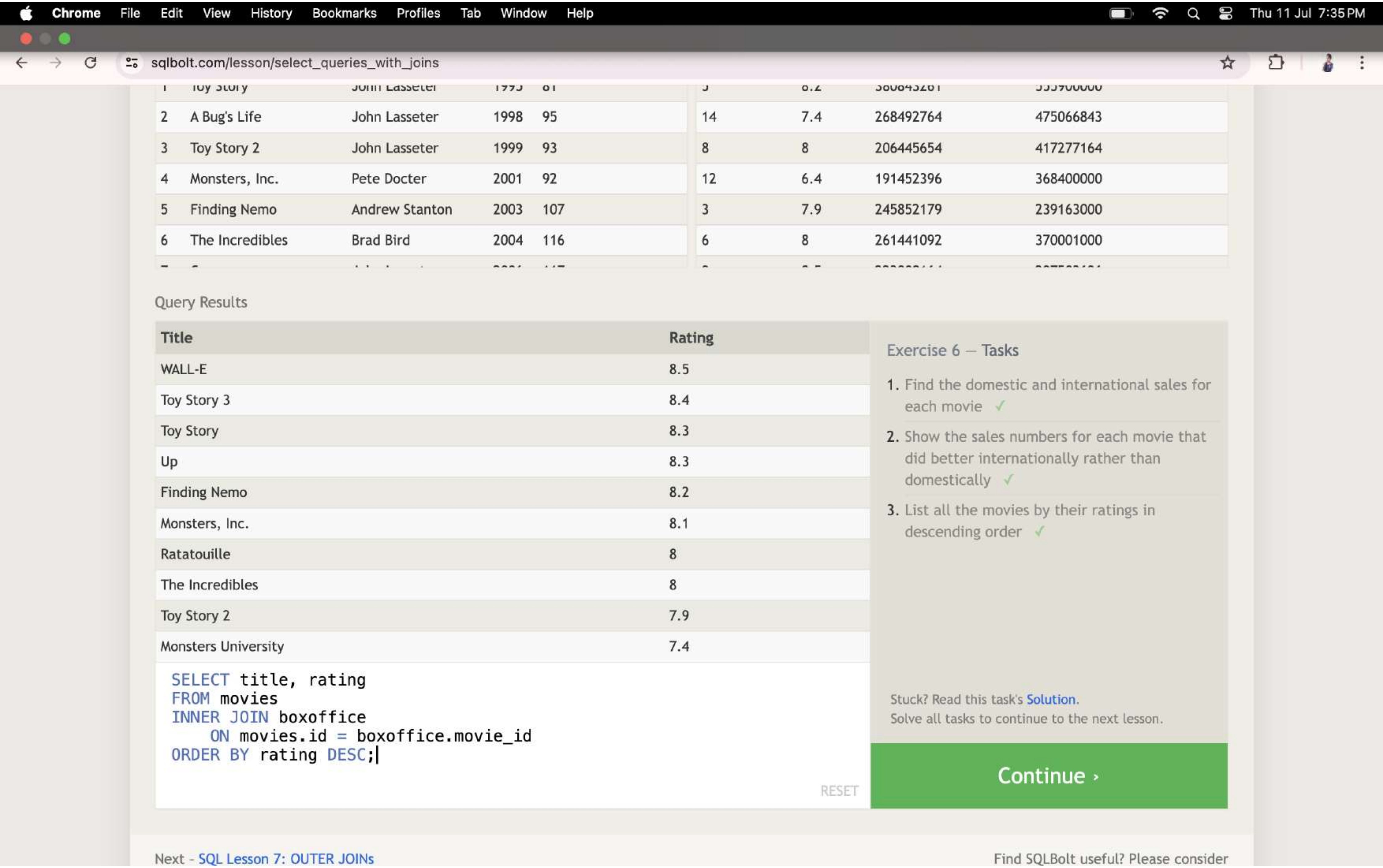
Exercise 7 — Tasks

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.

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.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 11 Jul 7:35 PM
sqlbolt.com/lesson/select_queries_with_joins

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

5	8.2	380843261	555900000
14	7.4	268492764	475066843
8	8	206445654	417277164
12	6.4	191452396	368400000
3	7.9	245852179	239163000
6	8	261441092	370001000

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;
```

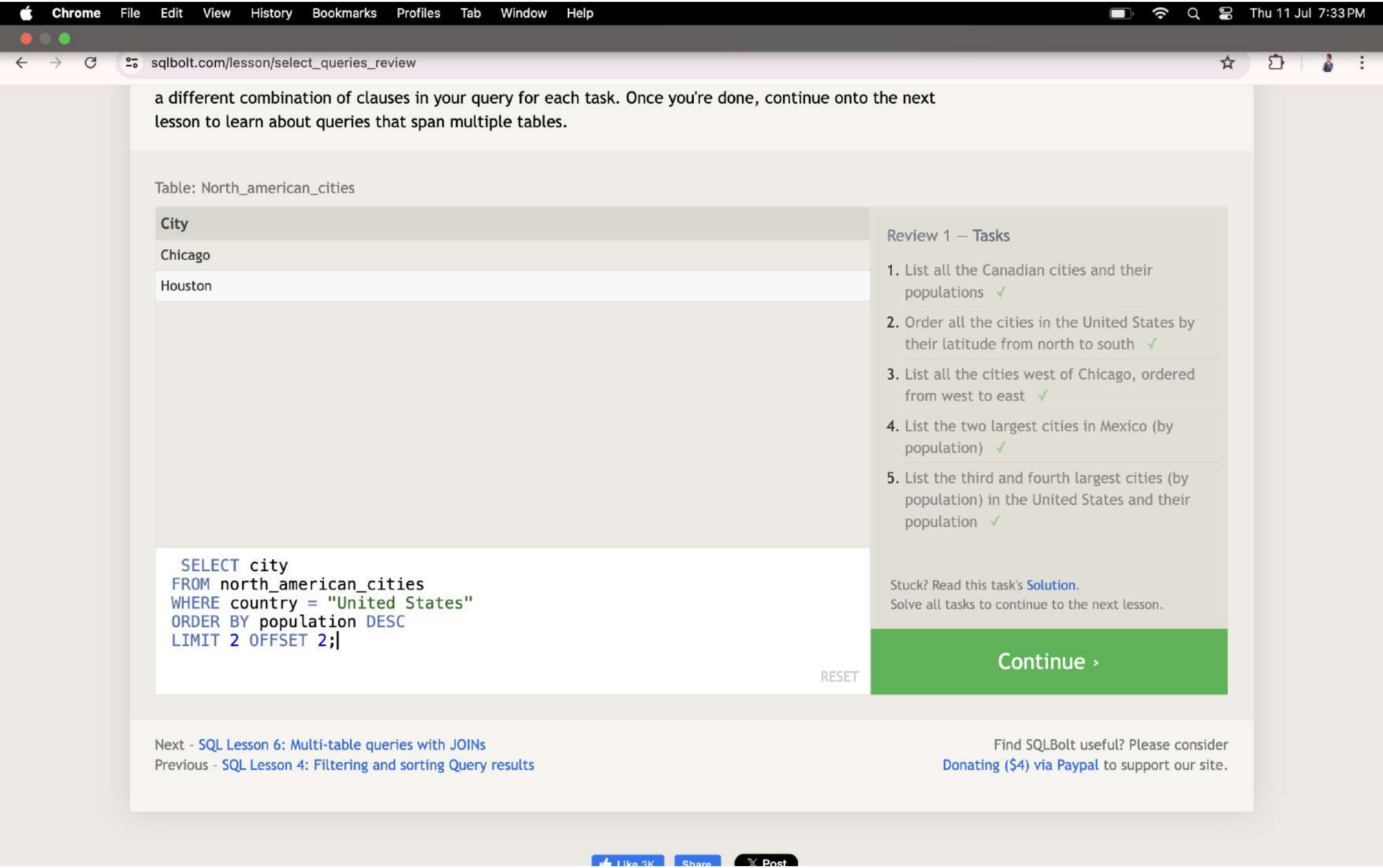
Exercise 6 — Tasks

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.

Next - SQL Lesson 7: OUTER JOINs

Find SQLBolt useful? Please consider


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 11 Jul 7:33 PM
sqlbolt.com/lesson/select_queries_review

a different combination of clauses in your query for each task. Once you're done, continue onto the next lesson to learn about queries that span multiple tables.

Table: North_american_cities

City
Chicago
Houston

```
 SELECT city
FROM north_american_cities
WHERE country = "United States"
ORDER BY population DESC
LIMIT 2 OFFSET 2;
```

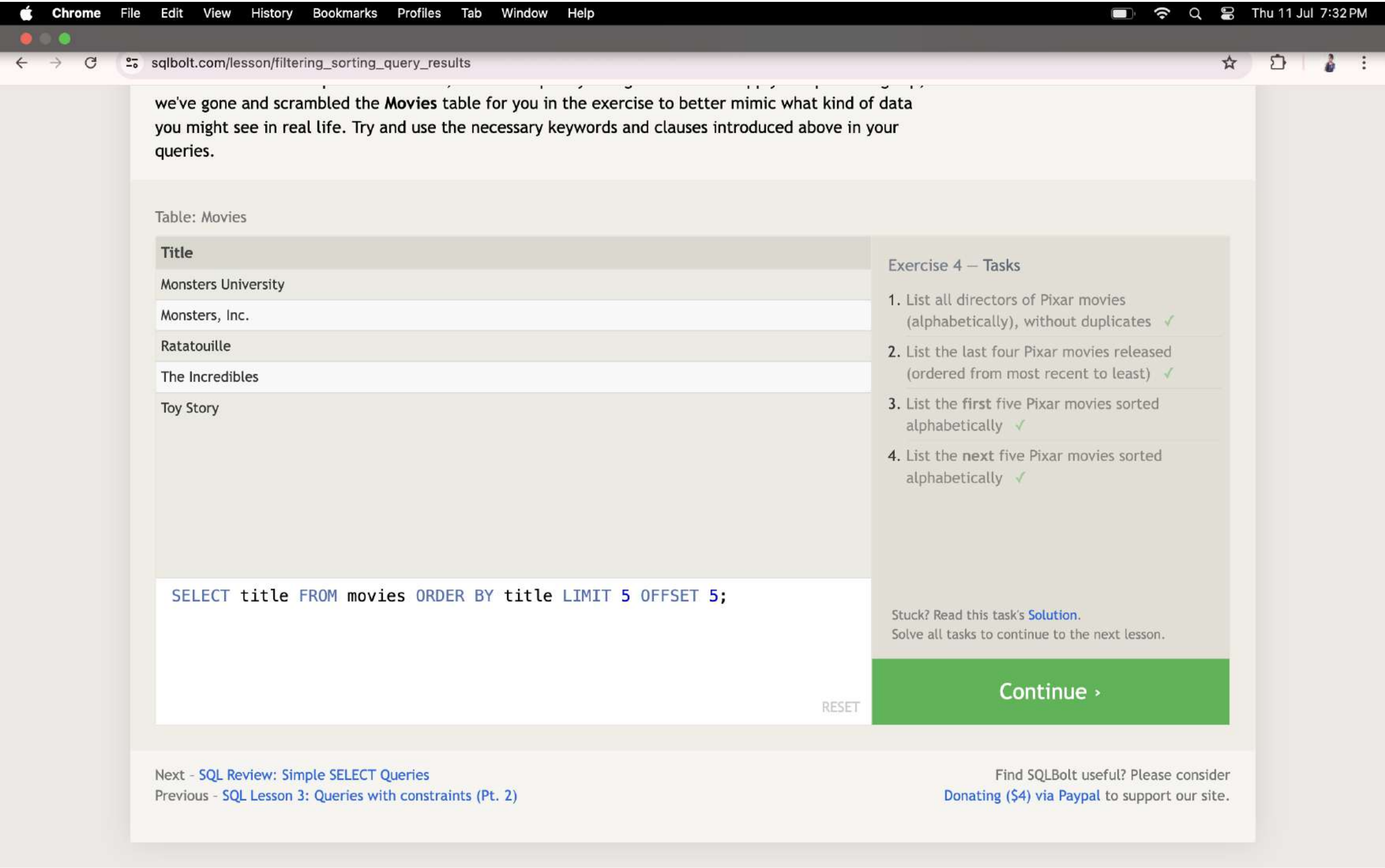
Review 1 — Tasks

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
Previous - SQL Lesson 4: Filtering and sorting Query results

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



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 11 Jul 7:32 PM
sqlbolt.com/lesson/filtering_sorting_query_results

we've gone and scrambled the **Movies** table for you in the exercise to better mimic what kind of data you might see in real life. Try and use the necessary keywords and clauses introduced above in your queries.

Table: Movies

Title
Monsters University
Monsters, Inc.
Ratatouille
The Incredibles
Toy Story

```
SELECT title FROM movies ORDER BY title LIMIT 5 OFFSET 5;
```

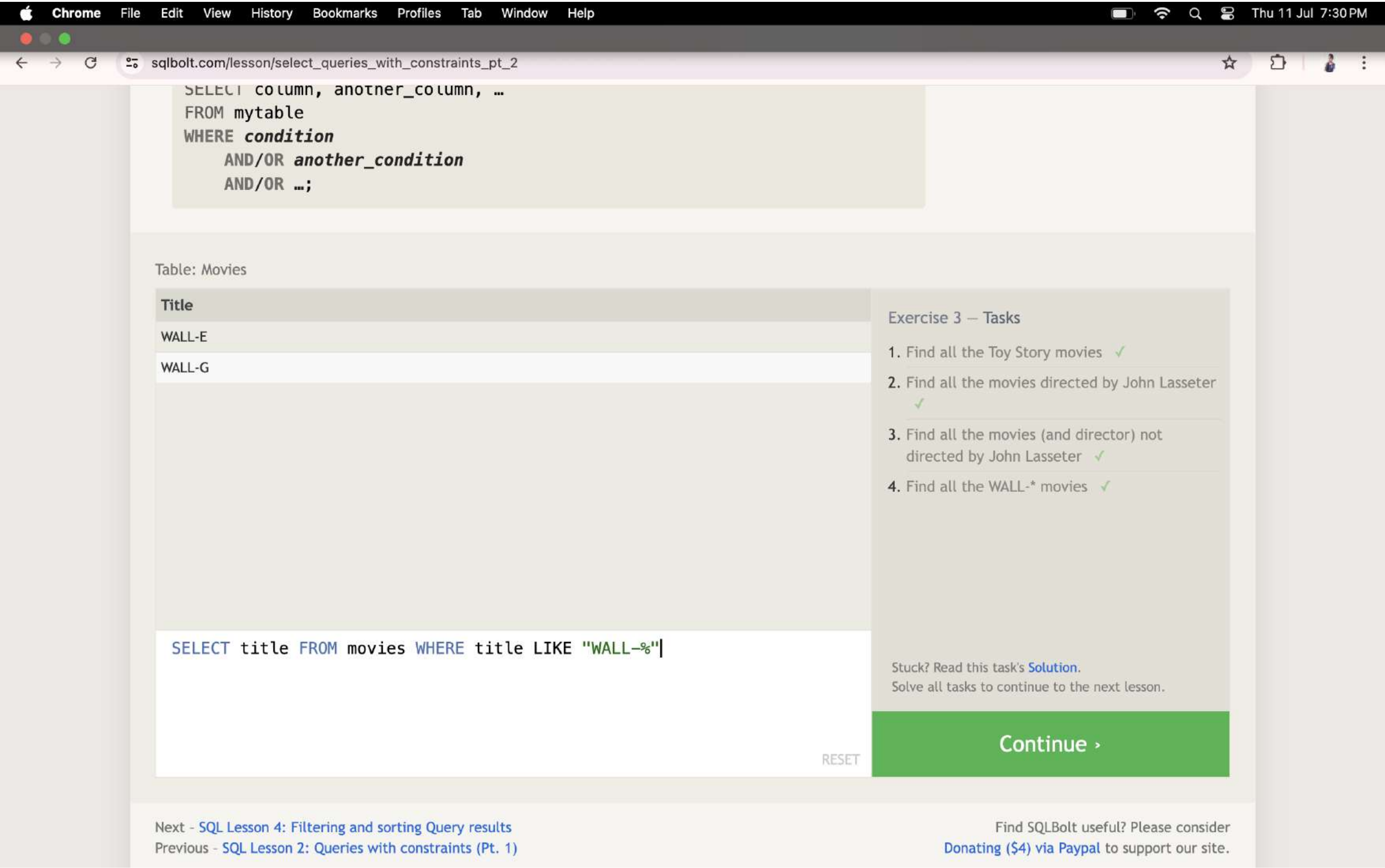
Exercise 4 — Tasks

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

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



```
SELECT column, another_column, …
FROM mytable
WHERE condition
    AND/OR another_condition
    AND/OR …;
```

Table: Movies

Title
WALL-E
WALL-G

```
SELECT title FROM movies WHERE title LIKE "WALL-%"
```

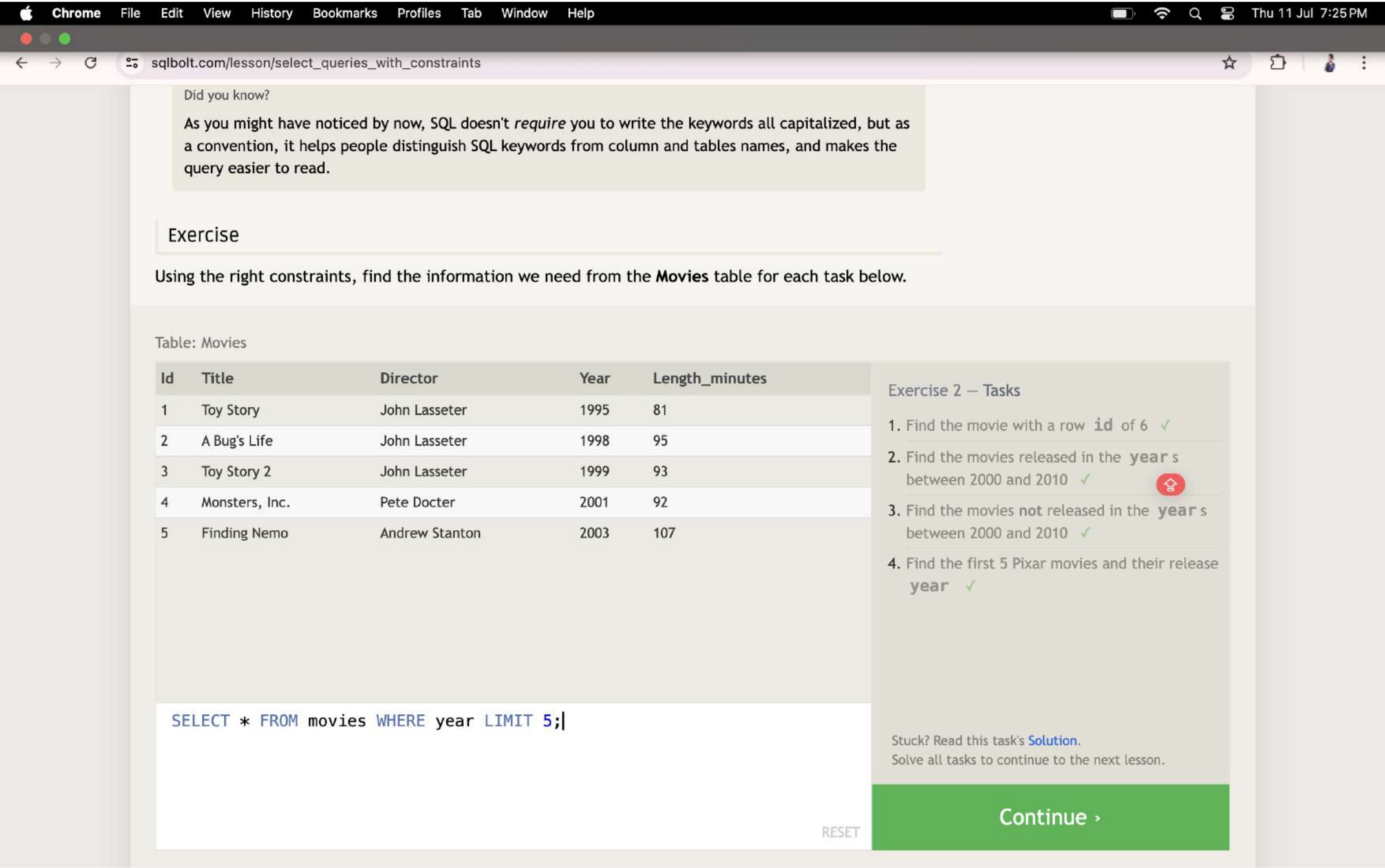
Exercise 3 — Tasks

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.



Did you know?

As you might have noticed by now, SQL doesn't *require* you to write the keywords all capitalized, but as a convention, it helps people distinguish SQL keywords from column and tables names, and makes the query easier to read.

Exercise

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

Table: Movies

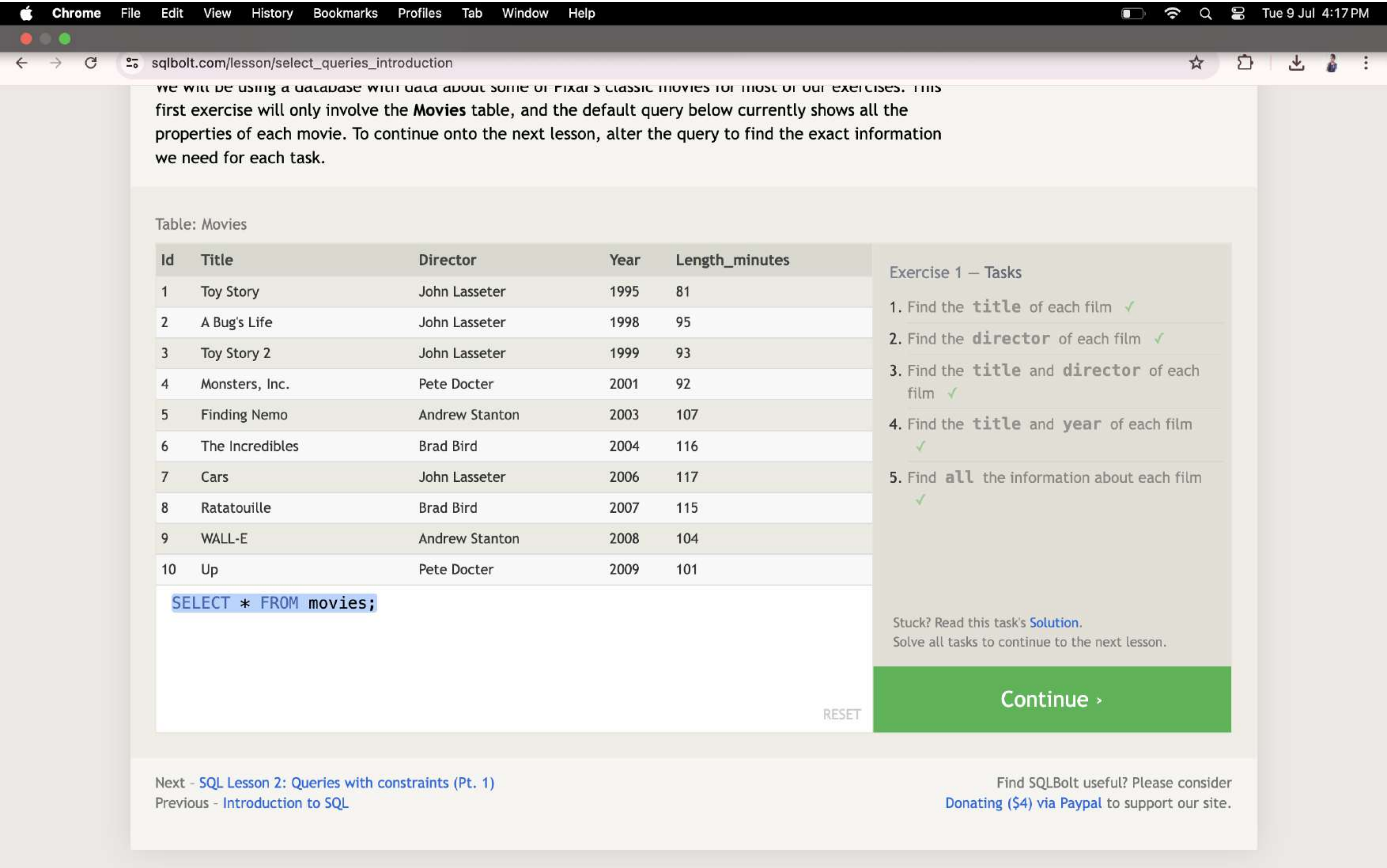
Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107

```
SELECT * FROM movies WHERE year 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.



We will be using a database with data about some of Pixar's classic movies for most of our exercises. This first exercise will only involve the **Movies** table, and the default query below currently shows all the properties of each movie. To continue onto the next lesson, alter the query to find the exact information we need for each task.

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.