Lesson 1

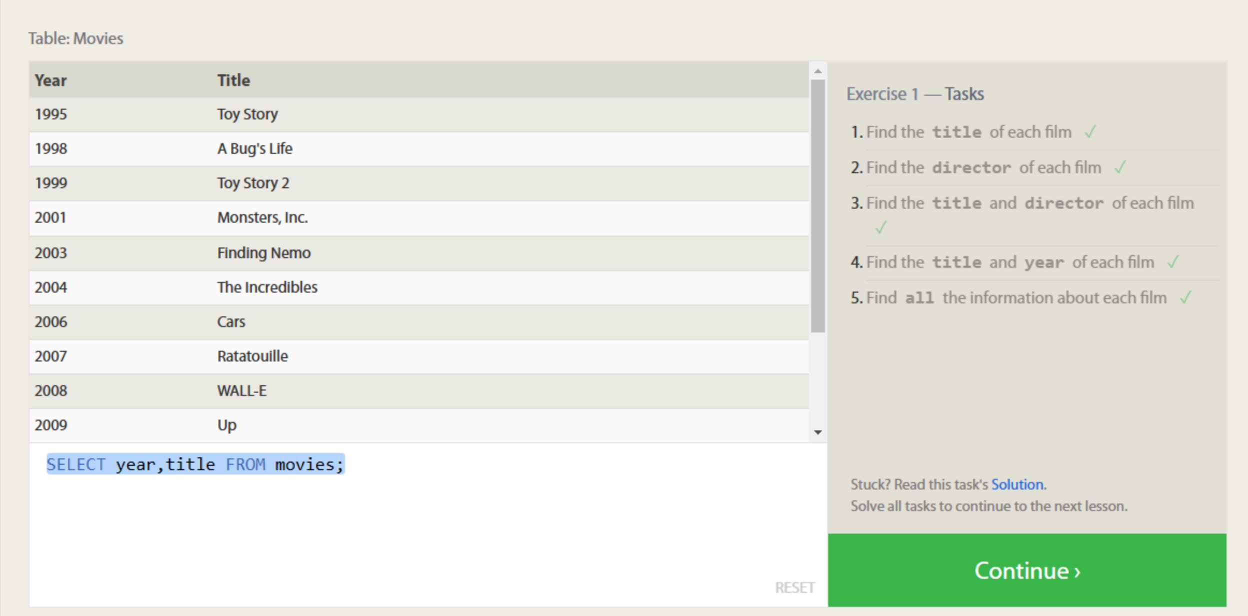


Table: Movies

Year	Title
1995	Toy Story
1998	A Bug's Life
1999	Toy Story 2
2001	Monsters, Inc.
2003	Finding Nemo
2004	The Incredibles
2006	Cars
2007	Ratatouille
2008	WALL-E
2009	Up

```
SELECT year,title 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.

1. SELECT title FROM movies;
2. SELECT director FROM movies;
3. SELECT director,title FROM movies;
4. SELECT year,title FROM movies;
5 SELECT * FROM movies;

Lesson 2

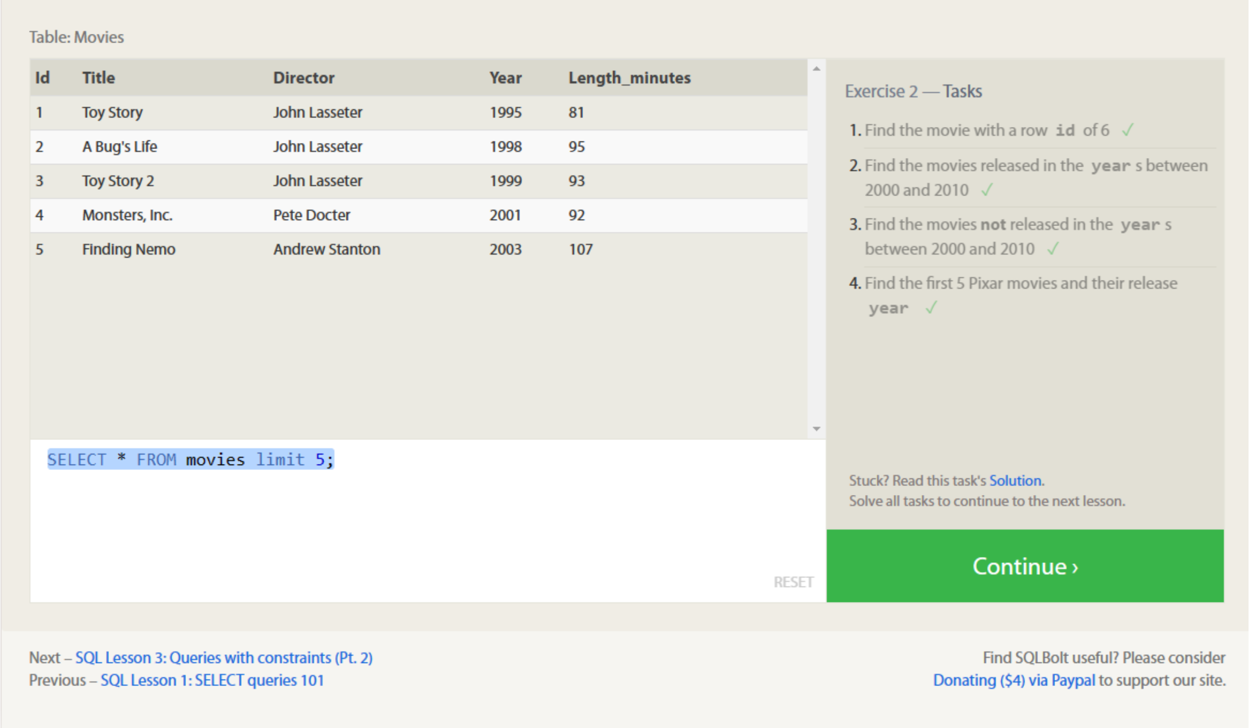


Table: Movies

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

```
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.

Next – SQL Lesson 3: Queries with constraints (Pt. 2)
Previous – SQL Lesson 1: SELECT queries 101

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

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;

Lesson 3

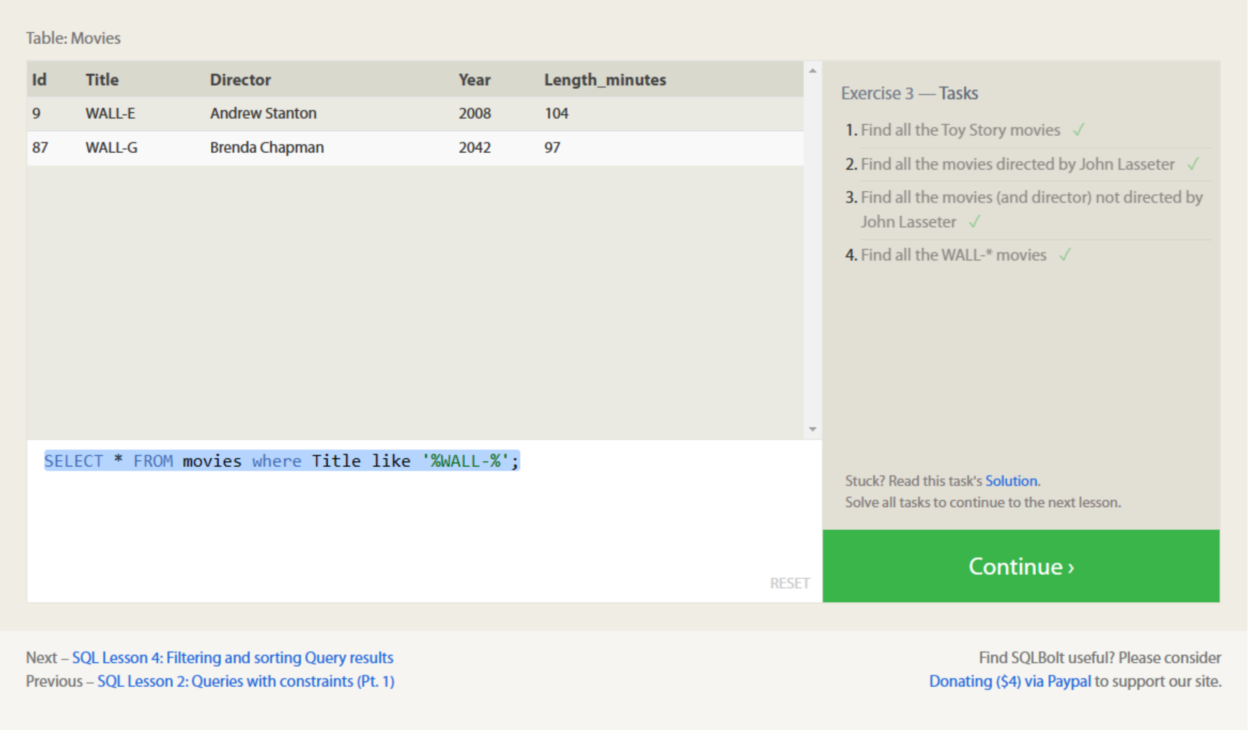


Table: Movies

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.

1. SELECT * FROM movies where Title like '%Toy%';
2. SELECT * FROM movies where Director like '%John Lasseter%';
3. SELECT * FROM movies where not Director like '%John Lasseter%';
4. SELECT * FROM movies where Title like '%WALL-%';

Lesson 4

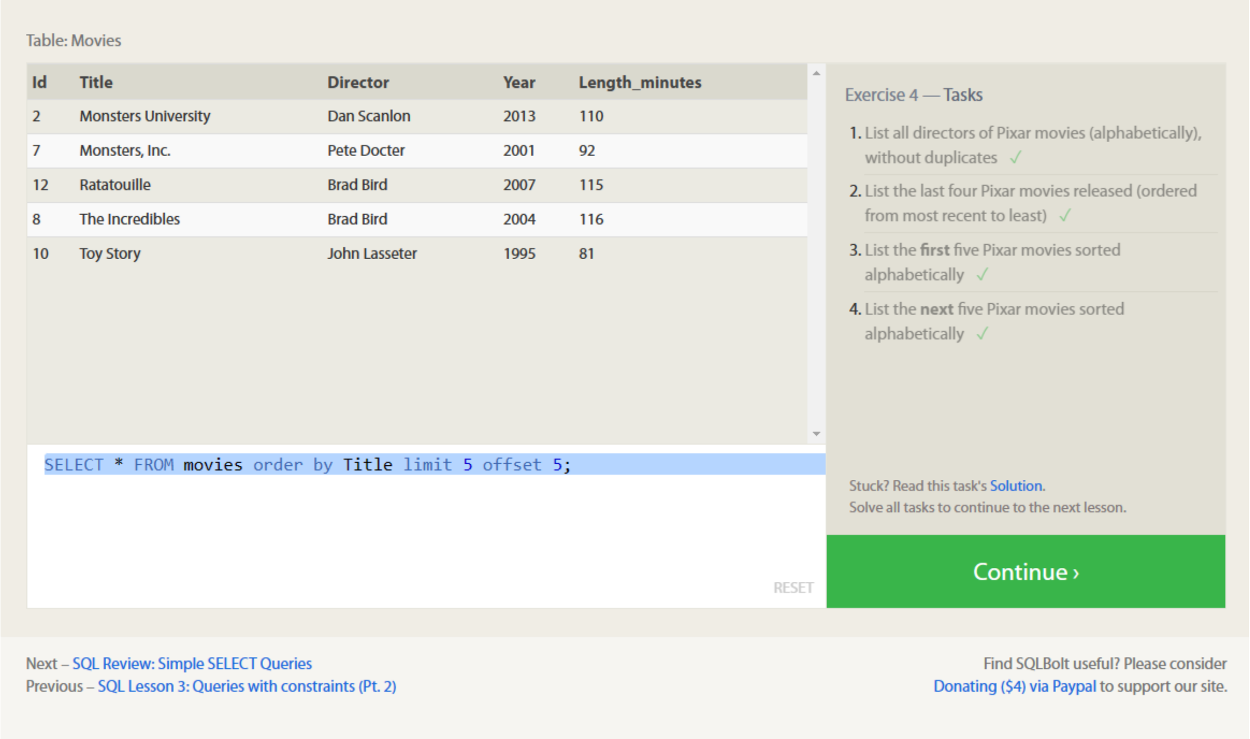


Table: Movies

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

```
SELECT * 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.

1. SELECT DISTINCT director FROM movies order by director;
2. SELECT * FROM movies order by year desc limit 4;
3. SELECT * FROM movies order by Title limit 5;
4. SELECT * FROM movies order by Title limit 5 offset 5;

Lesson 5

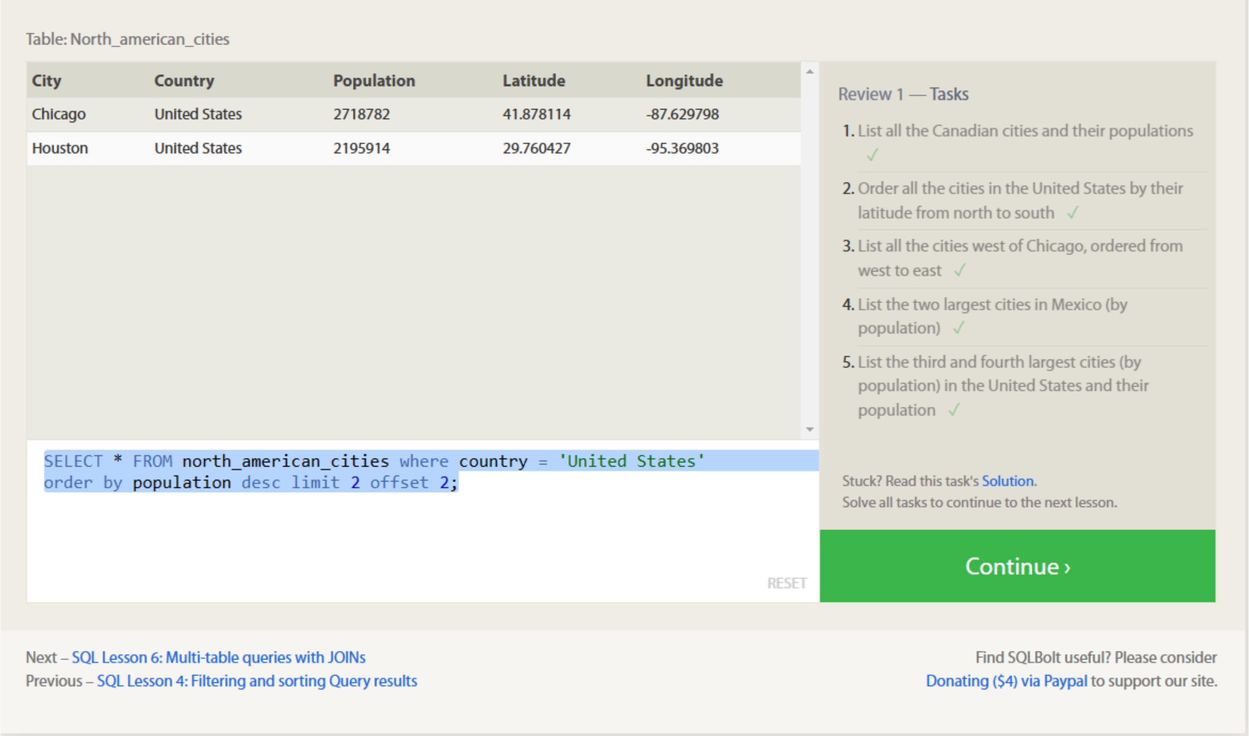


Table: North_american_cities

City	Country	Population	Latitude	Longitude
Chicago	United States	2718782	41.878114	-87.629798
Houston	United States	2195914	29.760427	-95.369803

```
SELECT * 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 ✓

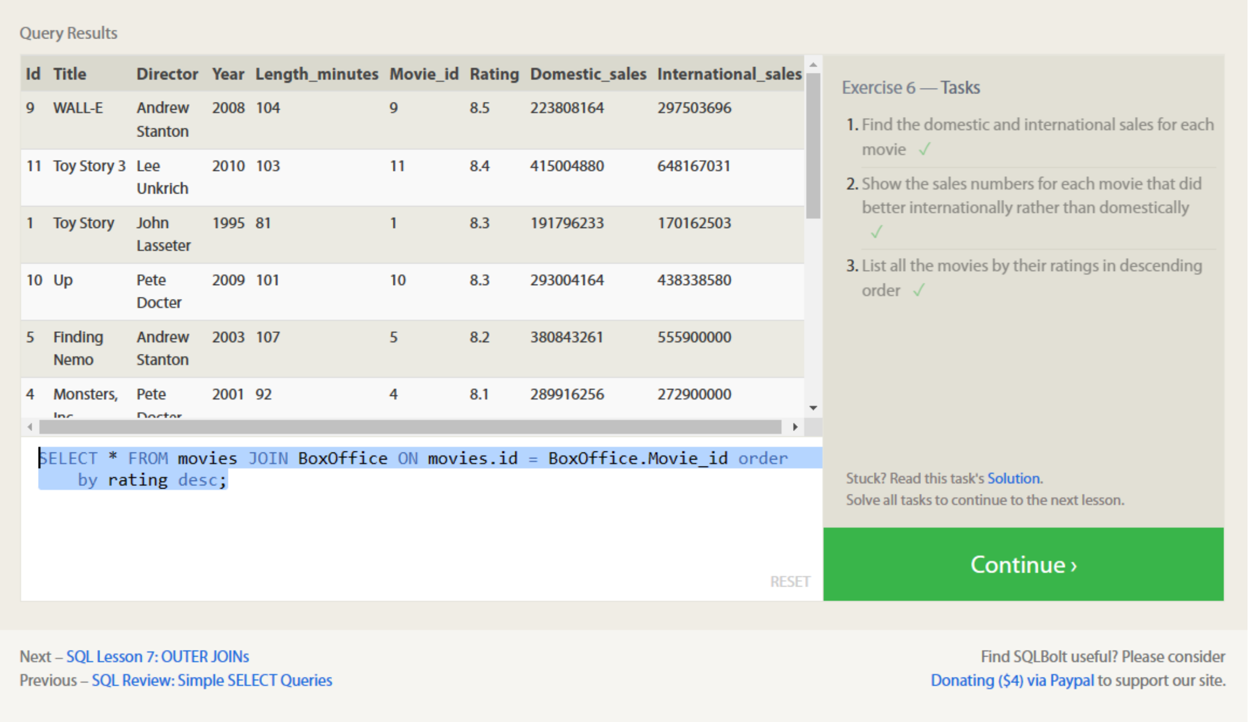
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

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.

1. SELECT * FROM north_american_cities WHERE country LIKE 'Canada';
2. SELECT * FROM north_american_cities WHERE country LIKE 'United States' order by latitude desc;
3. SELECT * FROM north_american_cities where longitude < -87.629798 order by longitude;
4. SELECT * FROM north_american_cities where country = 'Mexico' order by population desc limit 2;
5. SELECT * FROM north_american_cities where country = 'United States'
order by population desc limit 2 offset 2;

Lesson 6



Query Results

Id	Title	Director	Year	Length_minutes	Movie_id	Rating	Domestic_sales	International_sales
9	WALL-E	Andrew Stanton	2008	104	9	8.5	223808164	297503696
11	Toy Story 3	Lee Unkrich	2010	103	11	8.4	415004880	648167031
1	Toy Story	John Lasseter	1995	81	1	8.3	191796233	170162503
10	Up	Pete Docter	2009	101	10	8.3	293004164	438338580
5	Finding Nemo	Andrew Stanton	2003	107	5	8.2	380843261	555900000
4	Monsters, Inc	Pete Docter	2001	92	4	8.1	289916256	272900000

```
SELECT * FROM movies 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
Previous – SQL Review: Simple SELECT Queries

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

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 * FROM movies JOIN BoxOffice ON movies.id = BoxOffice.Movie_id order by rating desc;

Lesson 7

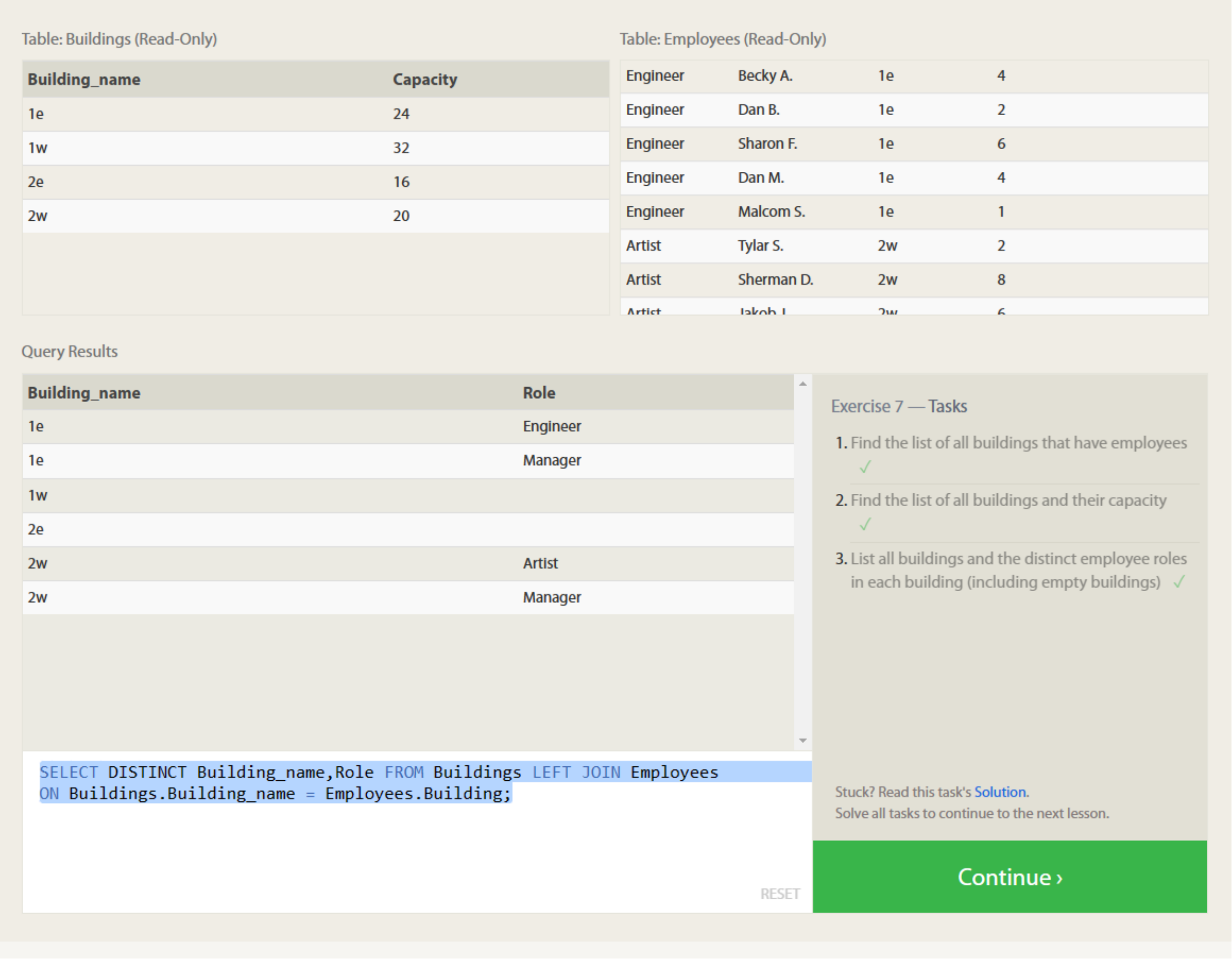


Table: Buildings (Read-Only)

Building_name	Capacity
1e	24
1w	32
2e	16
2w	20

Table: Employees (Read-Only)

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

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.

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;

Lesson 8

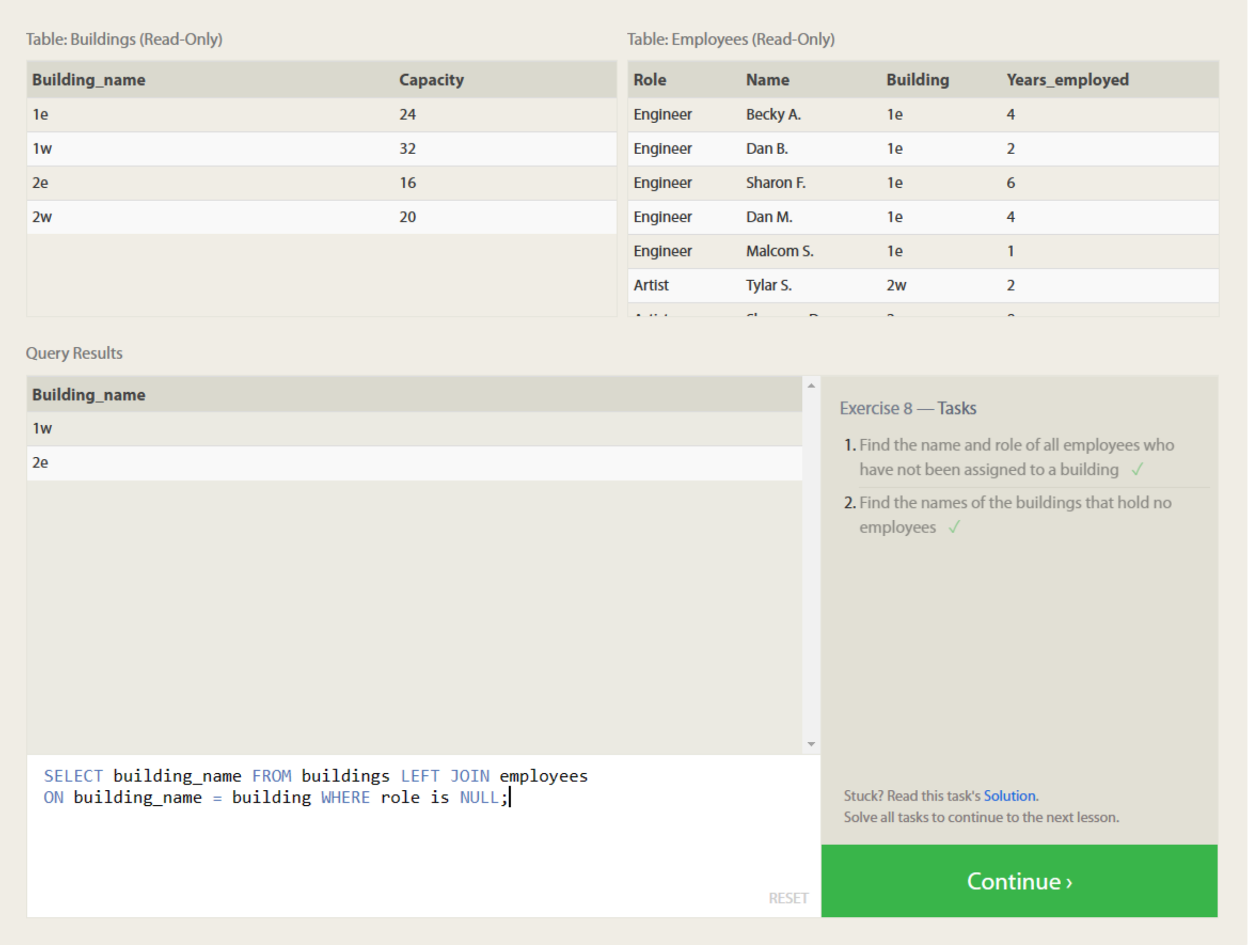


Table: Buildings (Read-Only)

Building_name	Capacity
1e	24
1w	32
2e	16
2w	20

Table: Employees (Read-Only)

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

Query Results

Building_name
1w
2e

```
SELECT building_name FROM buildings LEFT JOIN employees
ON building_name = 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.

1. SELECT * FROM employees where Building is NULL;
2. SELECT building_name FROM buildings LEFT JOIN employees
   ON building_name = building WHERE role is NULL;

Lesson 9

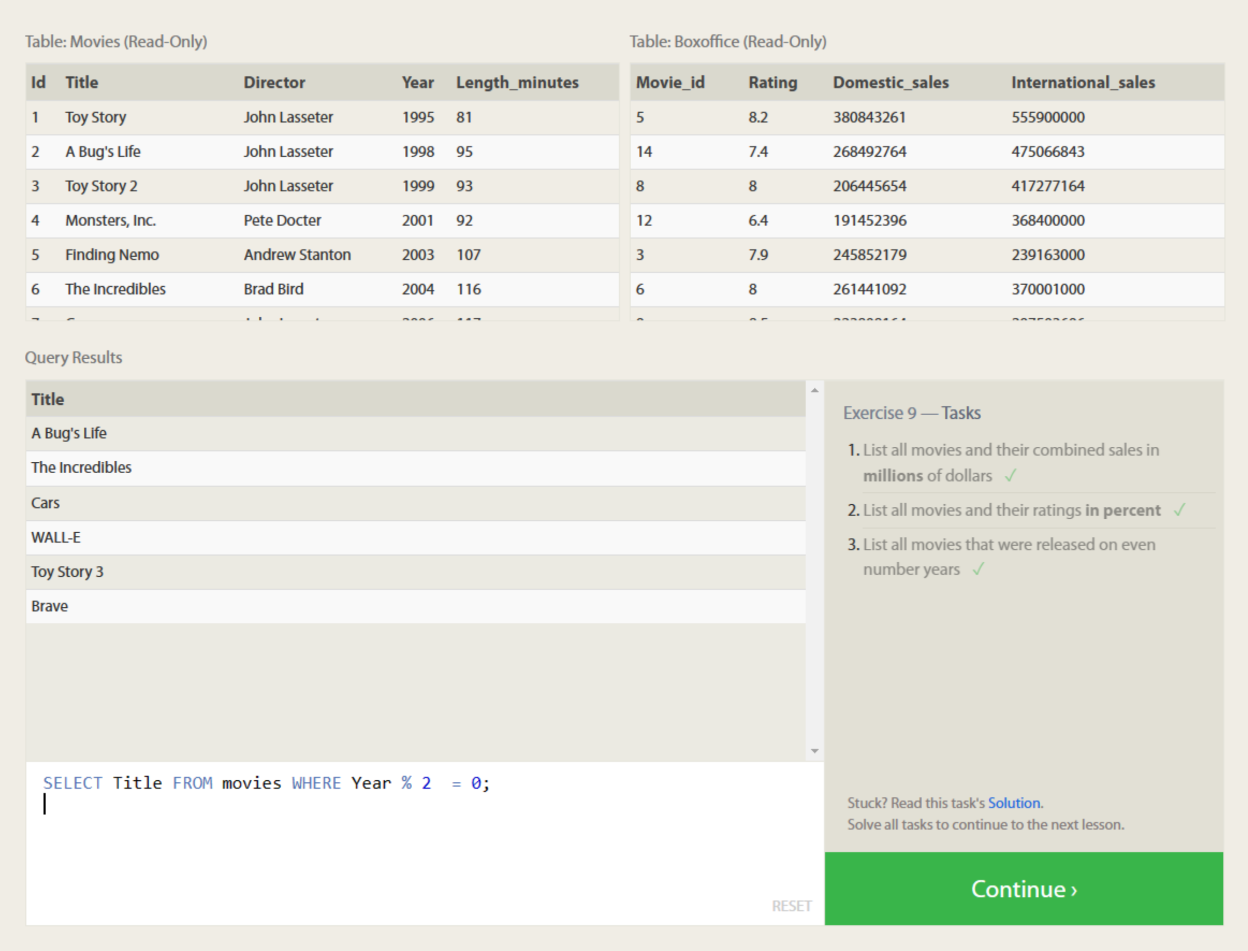


Table: Movies (Read-Only)

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

Table: Boxoffice (Read-Only)

Movie_id	Rating	Domestic_sales	International_sales
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
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.

1. SELECT Title,(Domestic_sales + International_sales)/ 1000000 as Total_Collection 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 FROM movies WHERE Year % 2 = 0;

Lesson 10

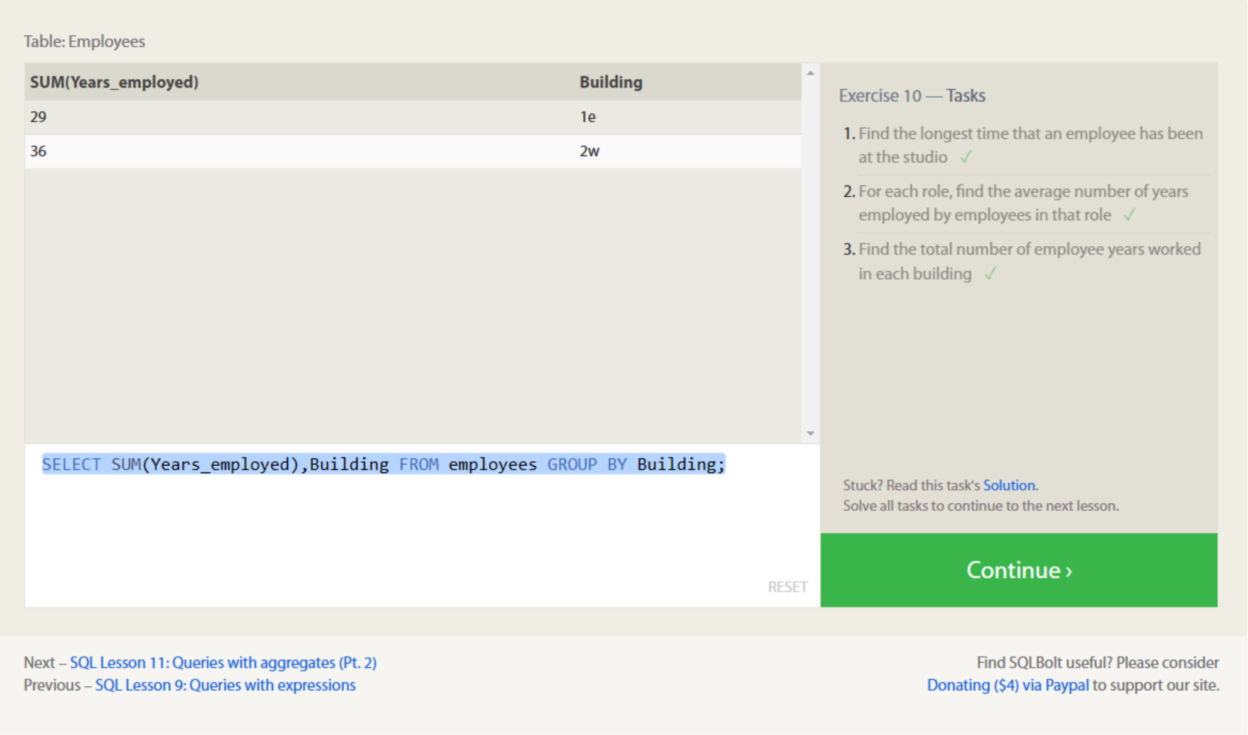


Table: Employees

SUM(Years_employed)	Building
29	1e
36	2w

```
SELECT SUM(Years_employed),Building 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.

1. SELECT MAX(Years_employed) FROM employees;
2. SELECT AVG(Years_Employed), Role FROM Employees GROUP BY Role;
3. SELECT SUM(Years_employed),Building FROM employees GROUP BY Building;

Lesson 11

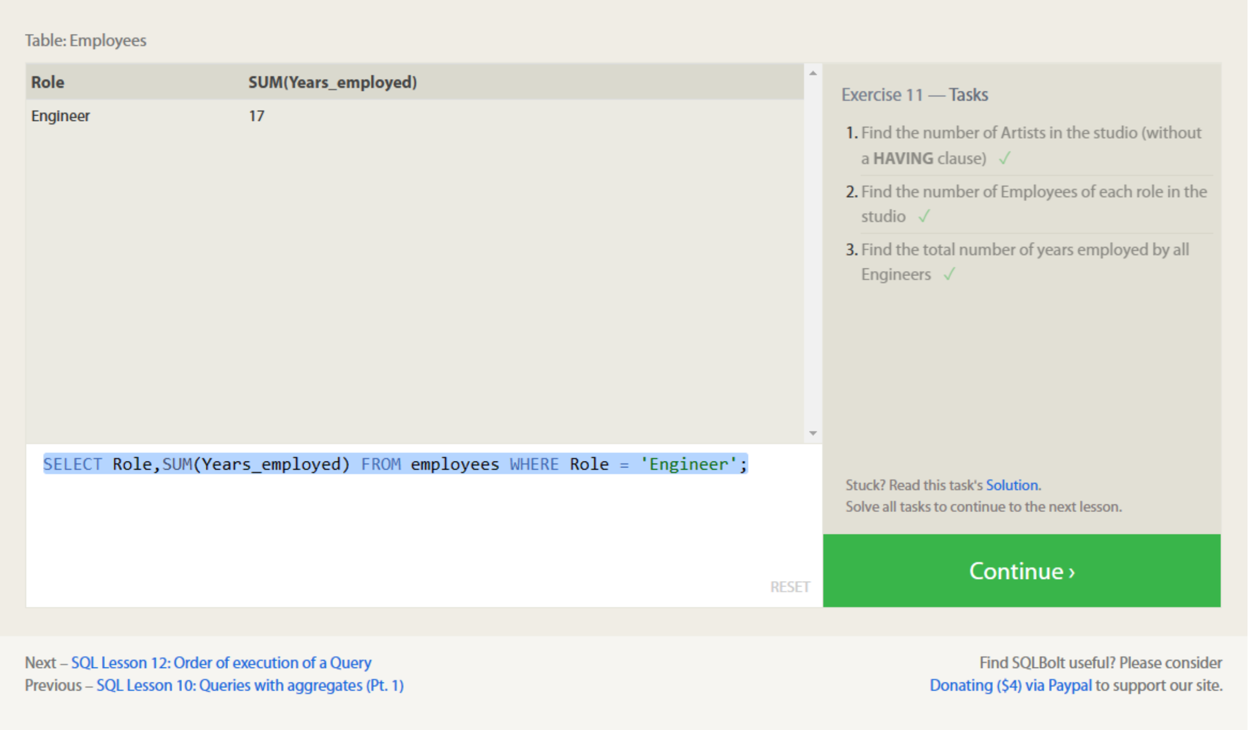


Table: Employees

Role	SUM(Years_employed)
Engineer	17

```
SELECT Role,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
Previous – SQL Lesson 10: Queries with aggregates (Pt. 1)

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

1. SELECT COUNT(Role) FROM employees WHERE Role = 'Artist';
2. SELECT Role,COUNT(Name) FROM employees GROUP BY Role;
3. SELECT Role,SUM(Years_employed) FROM employees WHERE Role = 'Engineer';

Lesson 12

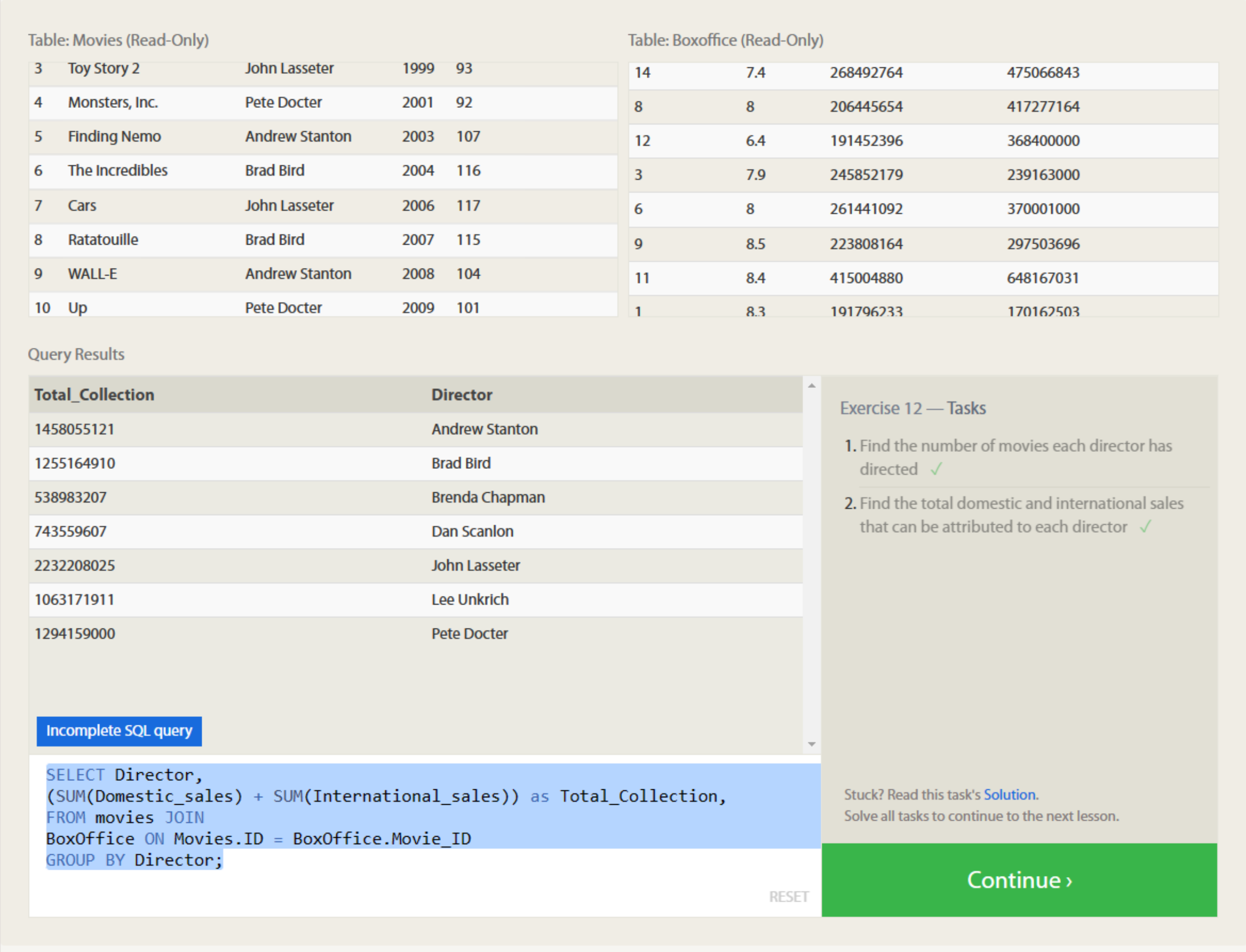


Table: Movies (Read-Only)

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

Table: Boxoffice (Read-Only)

14	7.4	268492764	475066843
8	8	206445654	417277164
12	6.4	191452396	368400000
3	7.9	245852179	239163000
6	8	261441092	370001000
9	8.5	223808164	297503696
11	8.4	415004880	648167031
1	8.3	191796233	170162503

Query Results

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

```
SELECT Director,
(SUM(Domestic_sales) + SUM(International_sales)) as Total_Collection,
FROM movies JOIN
BoxOffice ON Movies.ID = BoxOffice.Movie_ID
GROUP BY Director;
```

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.

1. SELECT COUNT(Title),Director FROM movies GROUP BY Director;
2. SELECT Director,
   (SUM(Domestic_sales) + SUM(International_sales)) as Total_Collection,
   FROM movies JOIN
   BoxOffice ON Movies.ID = BoxOffice.Movie_ID
   GROUP BY Director;

Lesson 13

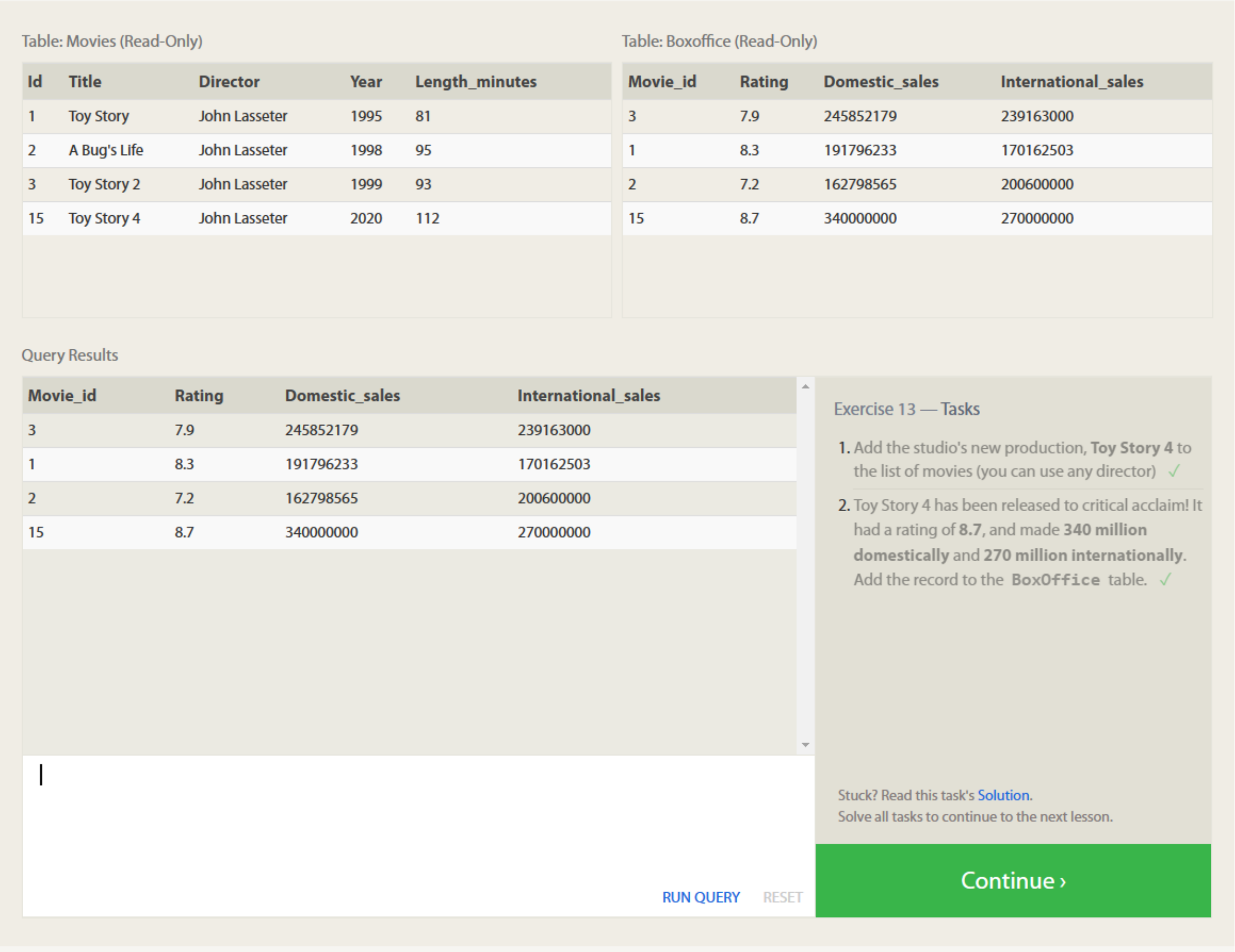


Table: Movies (Read-Only)

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
15	Toy Story 4	John Lasseter	2020	112

Table: Boxoffice (Read-Only)

Movie_id	Rating	Domestic_sales	International_sales
3	7.9	245852179	239163000
1	8.3	191796233	170162503
2	7.2	162798565	200600000
15	8.7	340000000	270000000

Query Results

Movie_id	Rating	Domestic_sales	International_sales
3	7.9	245852179	239163000
1	8.3	191796233	170162503
2	7.2	162798565	200600000
15	8.7	340000000	270000000

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.

1. INSERT INTO Movies (Id, Title, Director, Year , Length_minutes) VALUES ('15', 'Toy Story 4', 'John Lasseter', '2020', '112');
2. INSERT INTO BoxOffice (Movie_Id,Rating,Domestic_sales,International_sales) VALUES (15, 8.7, 340000000, 270000000);

Lesson 14

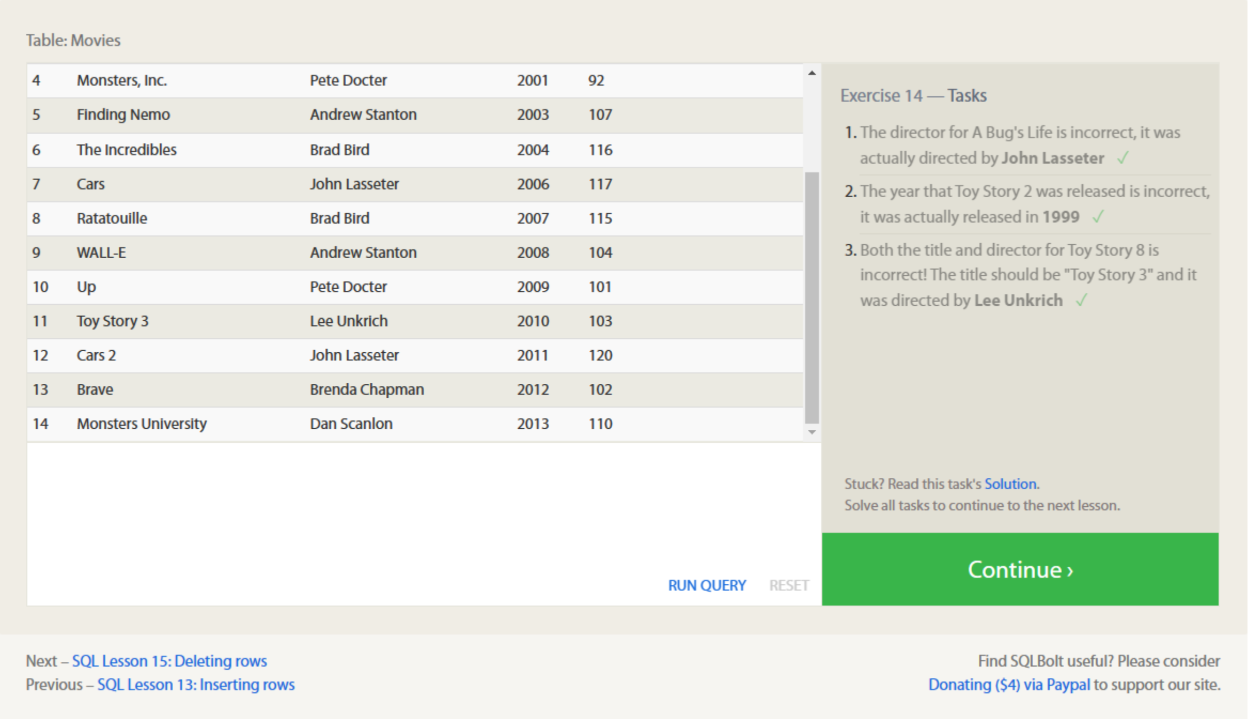


Table: Movies

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
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 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
Previous – SQL Lesson 13: Inserting rows

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

1. UPDATE Movies SET Director = 'John Lasseter' WHERE Title like '%Bug%';
2. UPDATE Movies SET Year = '1999' WHERE Title = 'Toy Story 2';
3. UPDATE Movies SET Title = 'Toy Story 3',Director ='Lee Unkrich' WHERE id = 11;

Lesson 15

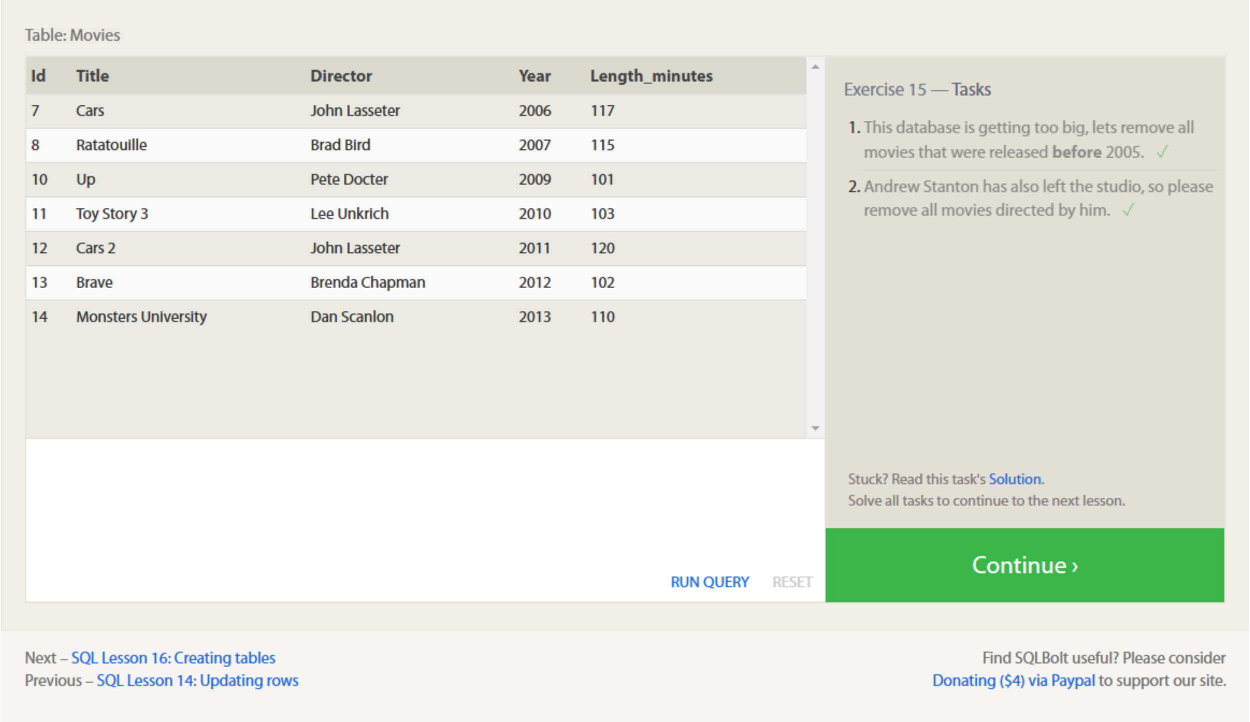


Table: Movies

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.

1. DELETE FROM Movies WHERE Year < 2005;
2. DELETE FROM Movies WHERE Director = 'Andrew Stanton';

Lesson 16

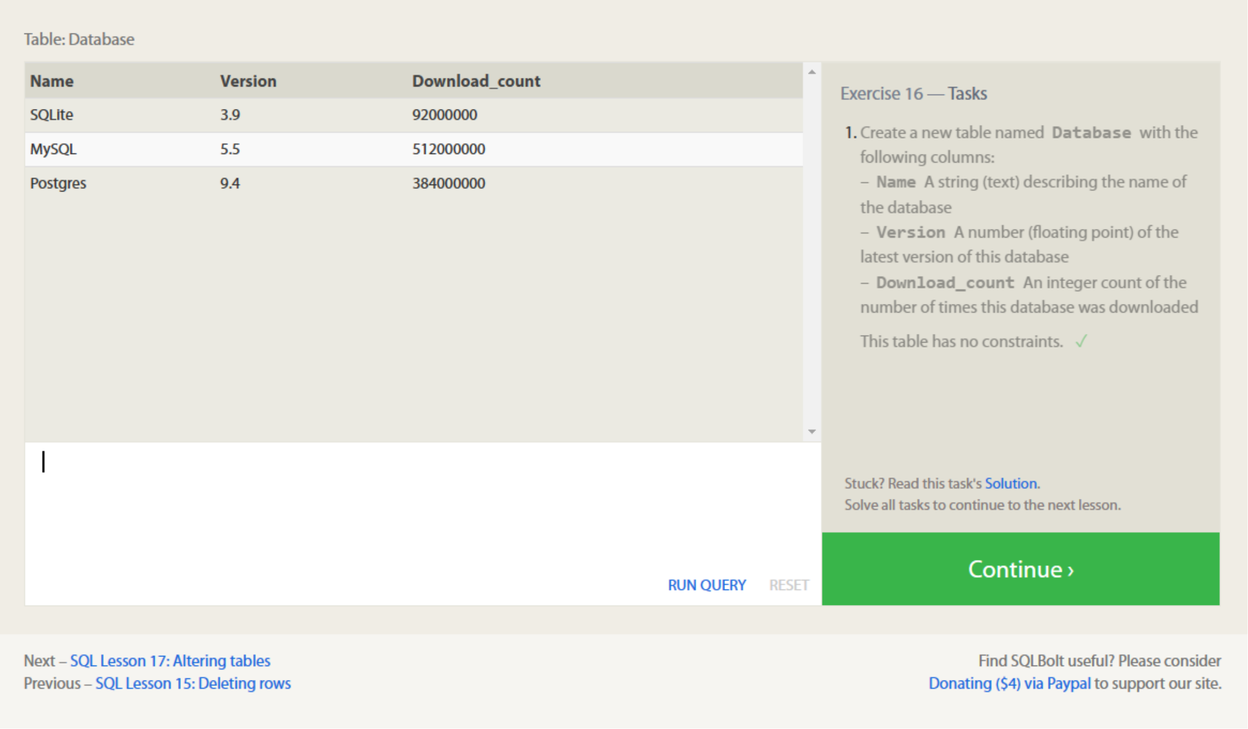


Table: Database

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.

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.

1.
   ```
   CREATE TABLE Database(
   Name TEXT,
   Version FLOAT,
   Download_count INTEGER
   );
   ```

Lesson 17

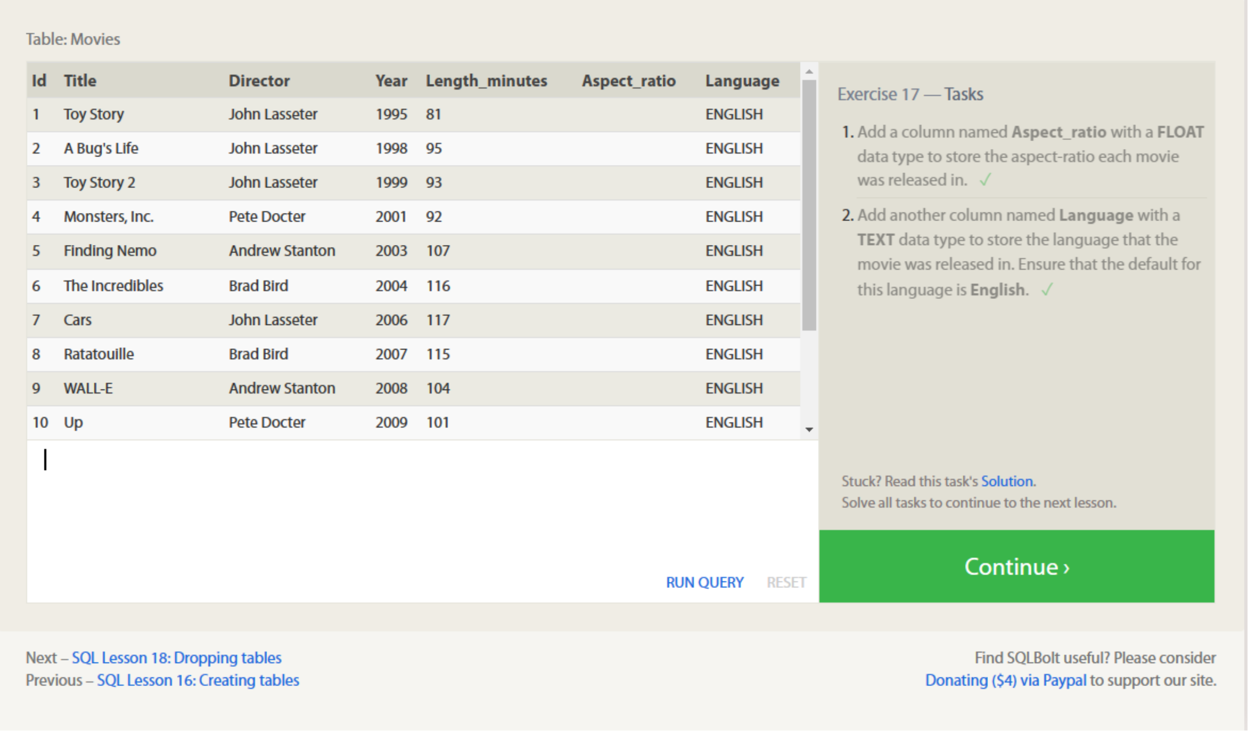


Table: Movies

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

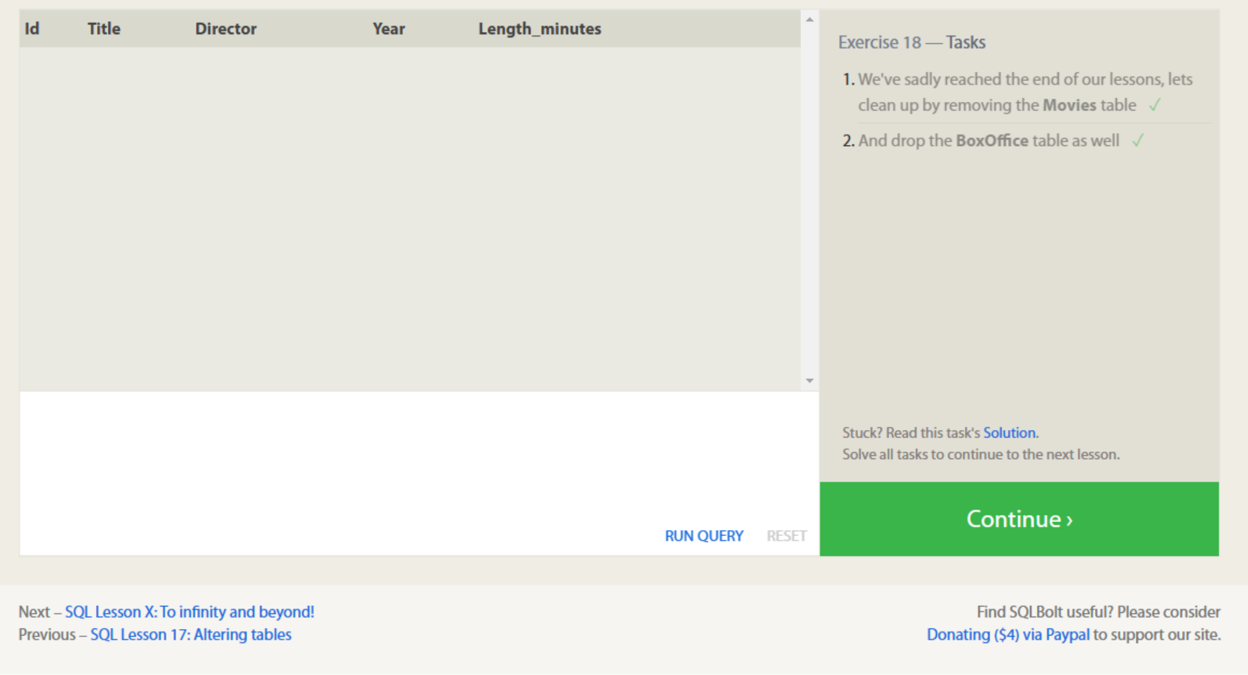
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

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.

1. ALTER TABLE Movies
   ADD Aspect_ratio FLOAT;
2. ALTER TABLE Movies
   ADD Language TEXT Default 'ENGLISH';

Lesson 18



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.

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.

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