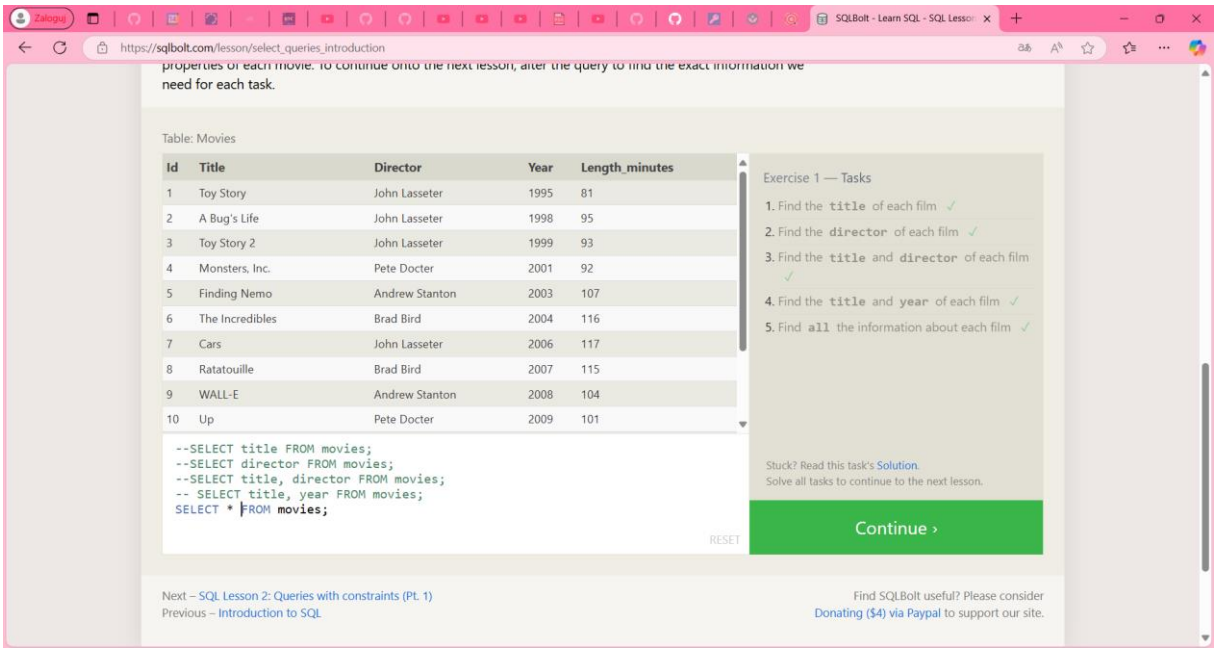
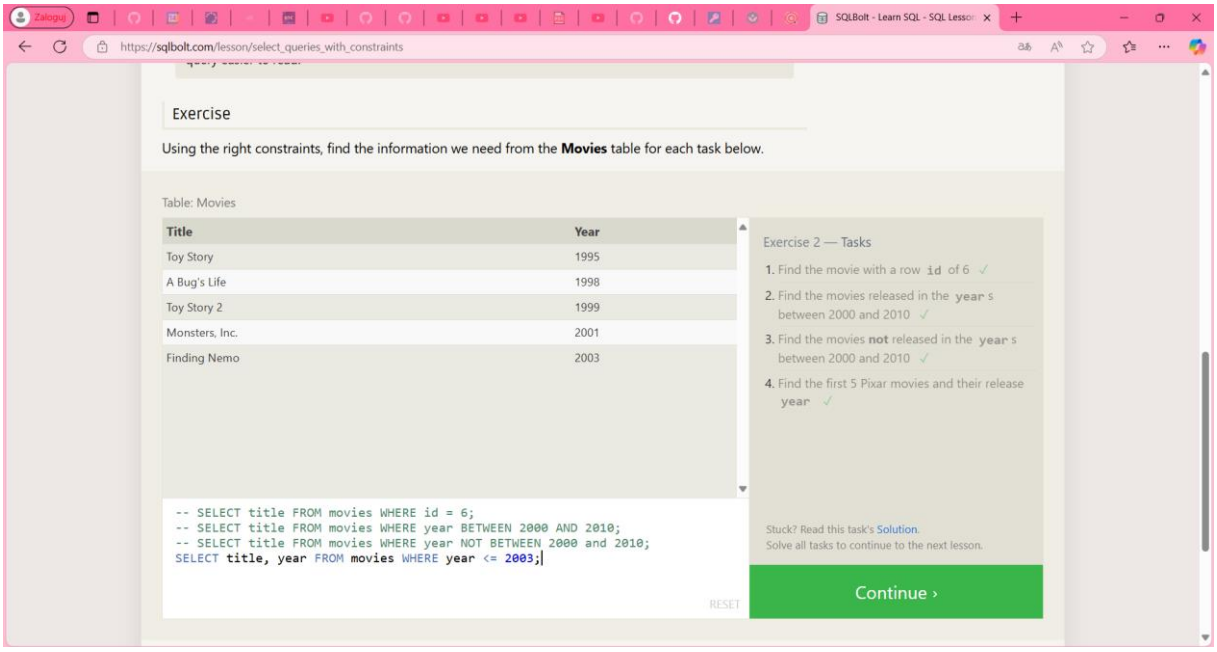
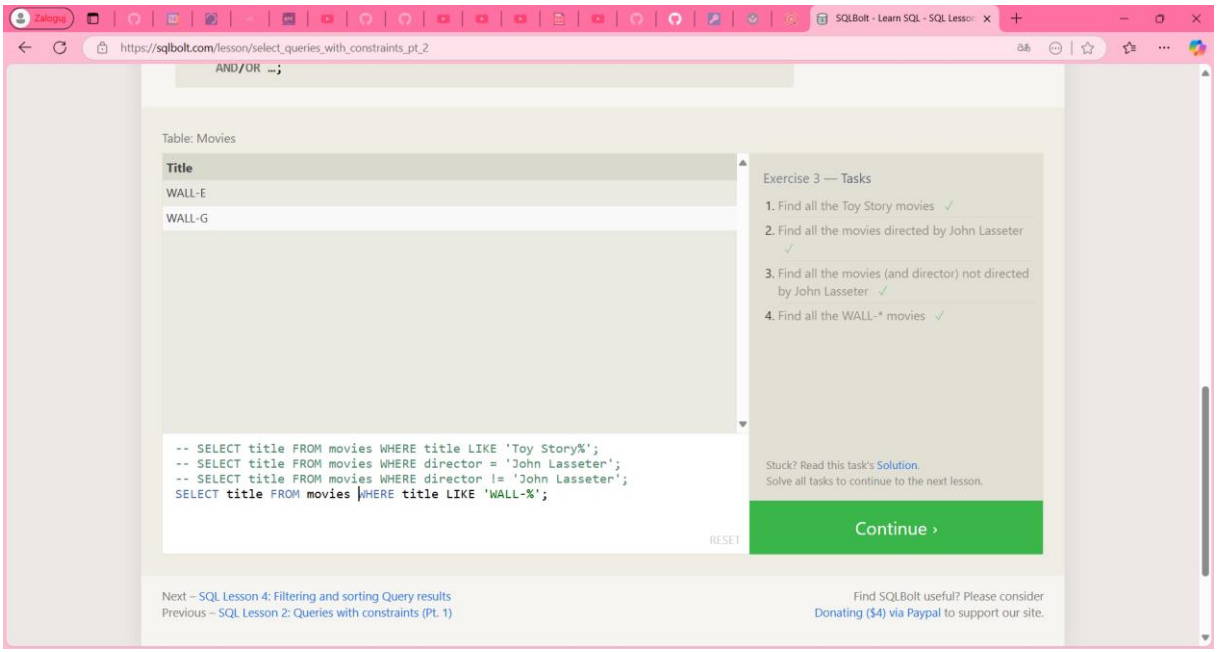
1


https://sqlbolt.com/lesson/select_queries_introduction
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 title FROM movies;
--SELECT director FROM movies;
--SELECT title, director FROM movies;
-- SELECT title, year FROM movies;
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 ›
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.

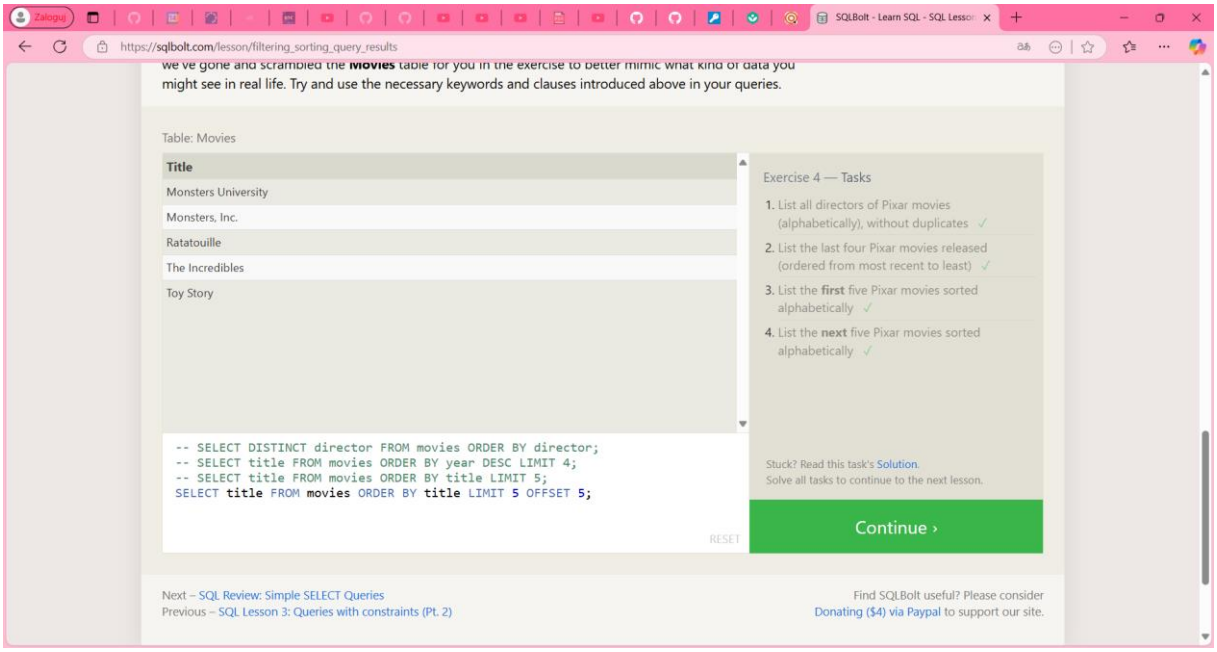
2


https://sqlbolt.com/lesson/select_queries_with_constraints
Exercise
Using the right constraints, find the information we need from the Movies table for each task below.
Table: Movies
Title Year
Toy Story 1995
A Bug's Life 1998
Toy Story 2 1999
Monsters, Inc. 2001
Finding Nemo 2003
-- SELECT title FROM movies WHERE id = 6;
-- SELECT title FROM movies WHERE year BETWEEN 2000 AND 2010;
-- SELECT title FROM movies WHERE year NOT BETWEEN 2000 and 2010;
SELECT title, year FROM movies WHERE year <= 2003;
RESET
Exercise 2 — Tasks
1. Find the movie with a row id of 6
2. Find the movies released in the years between 2000 and 2010
3. Find the movies not released in the years 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 ›

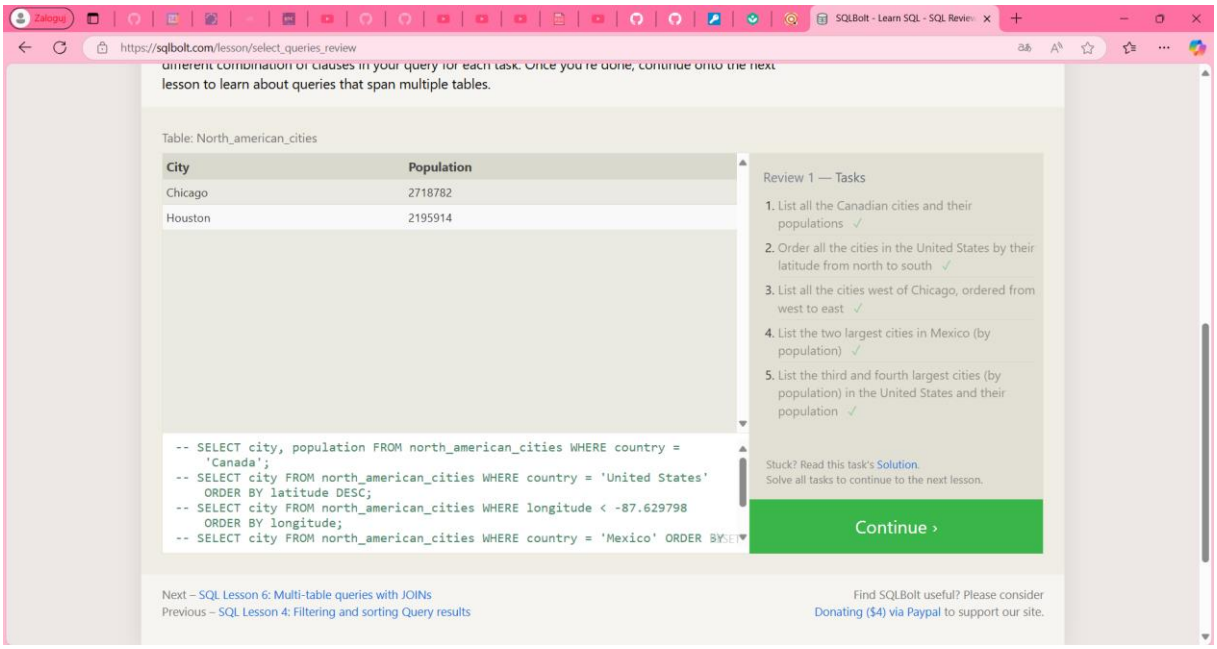
3

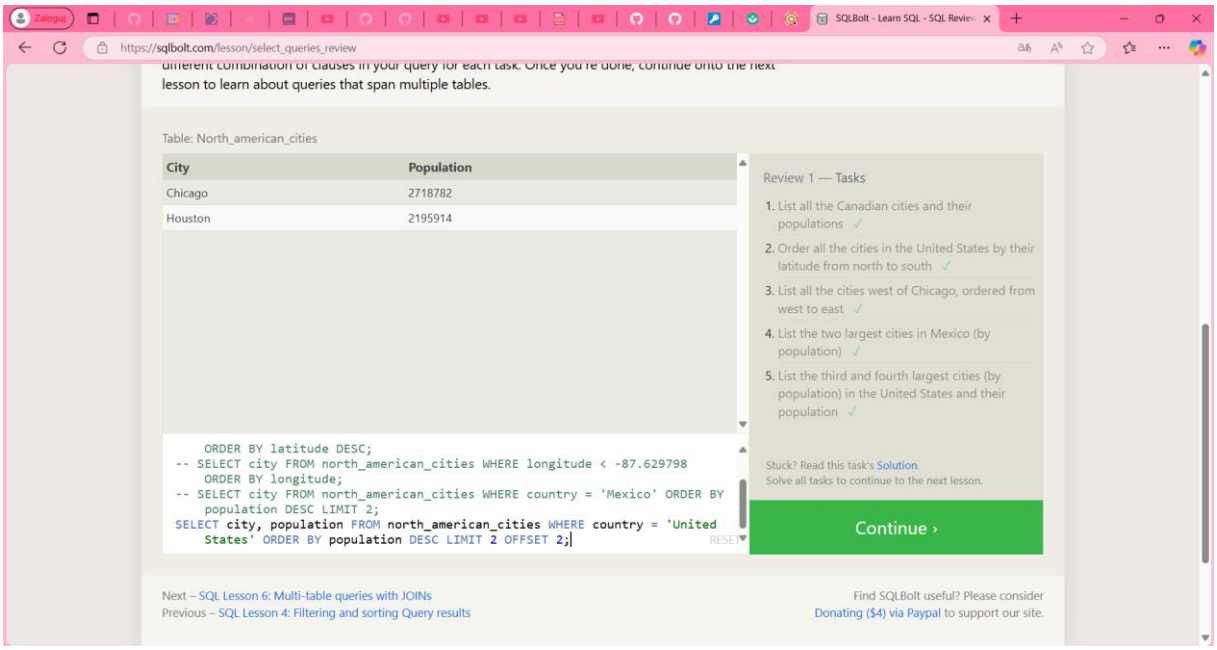

https://sqlbolt.com/lesson/select_queries_with_constraints_pt_2
AND/OR ...;
Table: Movies
Title
WALL-E
WALL-G
Exercise 3 — Tasks
1. Find all the Toy Story movies
2. Find all the movies directed by John Lasseter
3. Find all the movies (and director) not directed by John Lasseter
4. Find all the WALL-* movies
-- SELECT title FROM movies WHERE title LIKE 'Toy Story%';
-- SELECT title FROM movies WHERE director = 'John Lasseter';
-- SELECT title FROM movies WHERE director != 'John Lasseter';
SELECT title FROM movies WHERE title LIKE 'WALL-%';
RESET
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.
Continue ›
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.

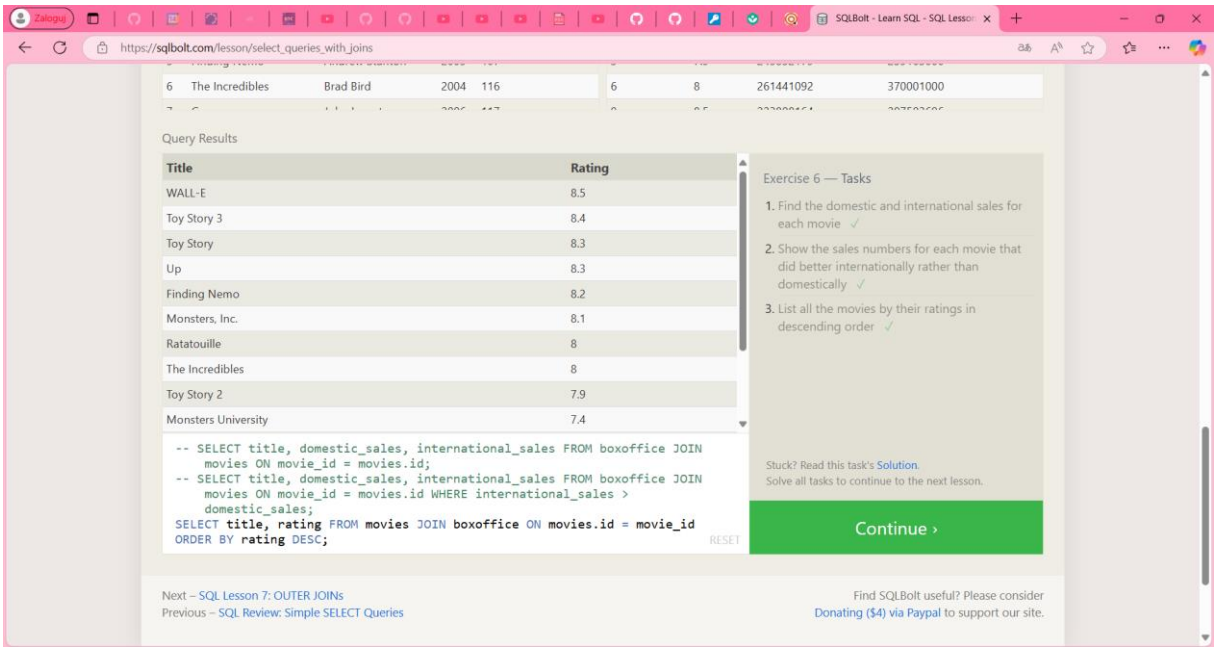
4


https://sqlbolt.com/lesson/filtering_sorting_query_results
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
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
-- SELECT DISTINCT director FROM movies ORDER BY director;
-- SELECT title FROM movies ORDER BY year DESC LIMIT 4;
-- SELECT title FROM movies ORDER BY title LIMIT 5;
SELECT title FROM movies ORDER BY title LIMIT 5 OFFSET 5;
RESET
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.
Continue ›
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.

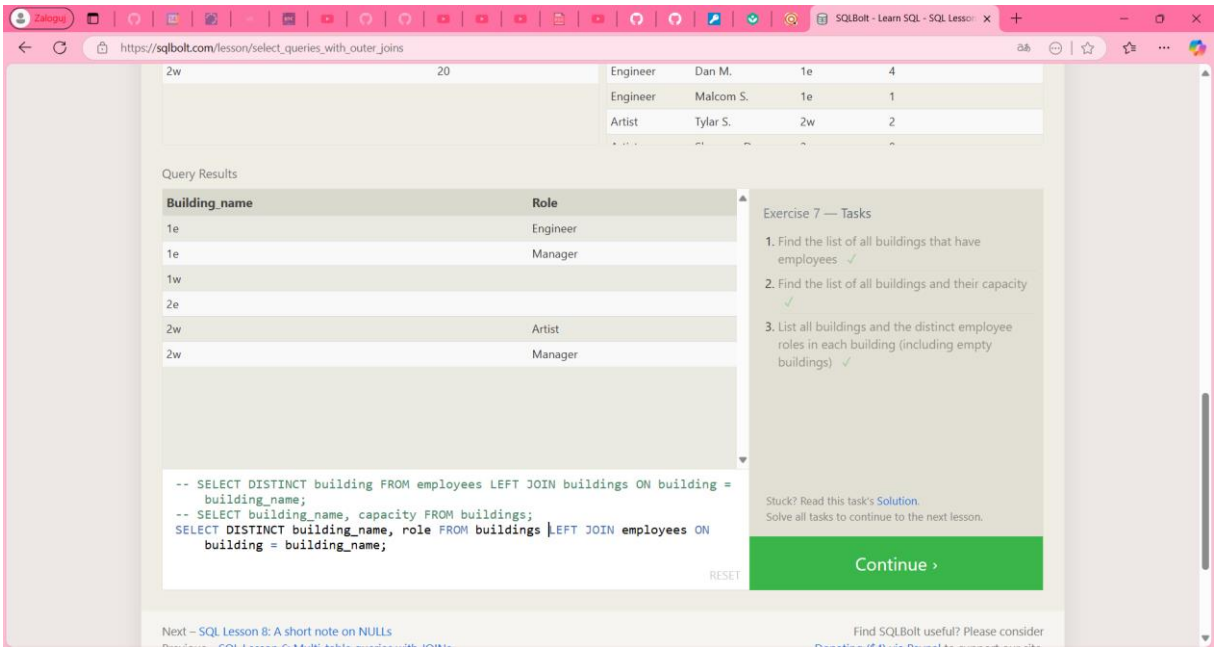
5


https://sqlbolt.com/lesson/select_queries_review
lesson to learn about queries that span multiple tables.
Table: North_american_cities
City
Population
Chicago
2718782
Houston
2195914
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
-- SELECT city, population FROM north_american_cities WHERE country = 'Canada';
-- SELECT city FROM north_american_cities WHERE country = 'United States' ORDER BY latitude DESC;
-- SELECT city FROM north_american_cities WHERE longitude < -87.629798 ORDER BY longitude;
-- SELECT city FROM north_american_cities WHERE country = 'Mexico' ORDER BY
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.
Continue ›
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.

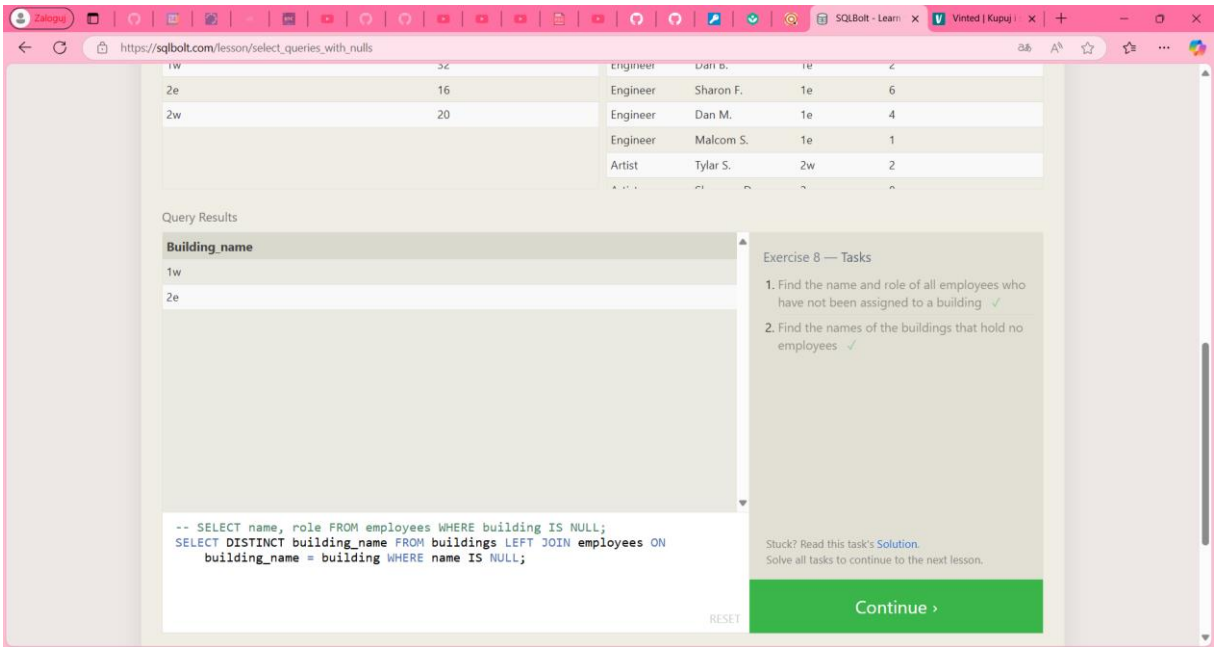

https://sqlbolt.com/lesson/select_queries_review
lesson to learn about queries that span multiple tables.
Table: North_american_cities
City
Population
Chicago
2718782
Houston
2195914
Review 1 — Tasks
1. List all the Canadian cities and their populations
2. Order all the cities in the United States by 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
ORDER BY latitude DESC;
-- SELECT city FROM north_american_cities WHERE longitude < -87.629798 ORDER BY longitude;
-- SELECT city FROM north_american_cities WHERE country = 'Mexico' ORDER BY population DESC LIMIT 2;
SELECT city, population FROM north_american_cities WHERE country = 'United States' ORDER BY population DESC LIMIT 2 OFFSET 2;
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.
Continue ›
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.


https://sqlbolt.com/lesson/select_queries_with_joins
6 The Incredibles Brad Bird 2004 116
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, domestic_sales, international_sales FROM boxoffice JOIN movies ON movie_id = movies.id;
-- SELECT title, domestic_sales, international_sales FROM boxoffice JOIN movies ON movie_id = movies.id WHERE international_sales > domestic_sales;
SELECT title, rating FROM movies JOIN boxoffice ON movies.id = movie_id ORDER BY rating DESC;
RESET
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.

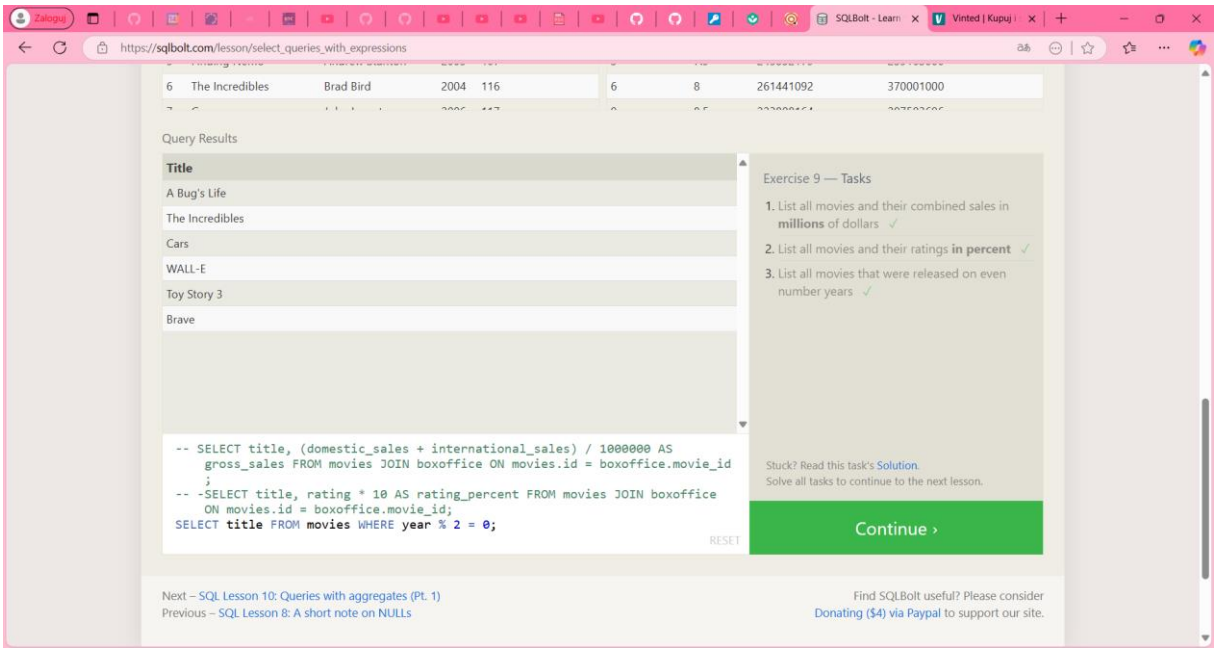
7


https://sqlbolt.com/lesson/select_queries_with_outer_joins
2w 20
Engineer Dan M. 1e 4
Engineer Malcom S. 1e 1
Artist Tylar S. 2w 2
Query Results
Building_name Role
1e Engineer
1e Manager
1w
2e
2w Artist
2w Manager
-- SELECT DISTINCT building FROM employees LEFT JOIN buildings ON building = building_name;
-- SELECT building_name, capacity FROM buildings;
SELECT DISTINCT building_name, role FROM buildings LEFT JOIN employees ON building = building_name;
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 ›
Next – SQL Lesson 8: A short note on NULLs
Find SQLBolt useful? Please consider

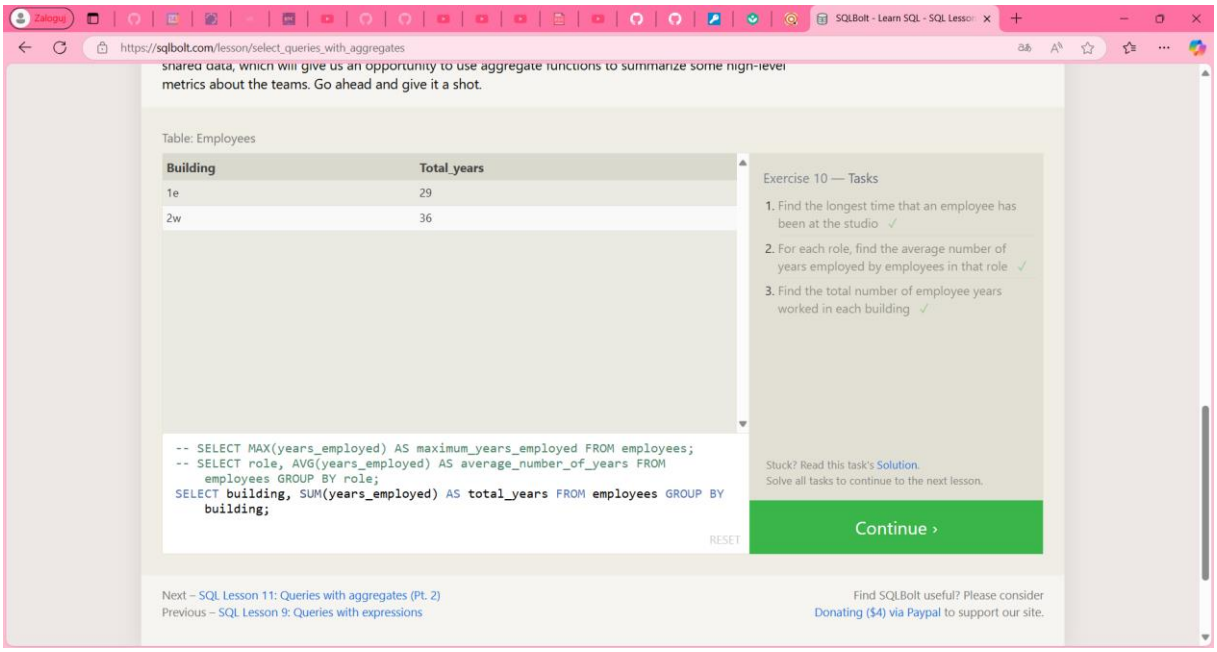
8


https://sqlbolt.com/lesson/select_queries_with_nulls
2e 16
2w 20
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
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
-- SELECT name, role FROM employees WHERE building IS NULL;
SELECT DISTINCT building_name FROM buildings LEFT JOIN employees ON
building_name = building WHERE name IS NULL;
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.
RESET
Continue ›

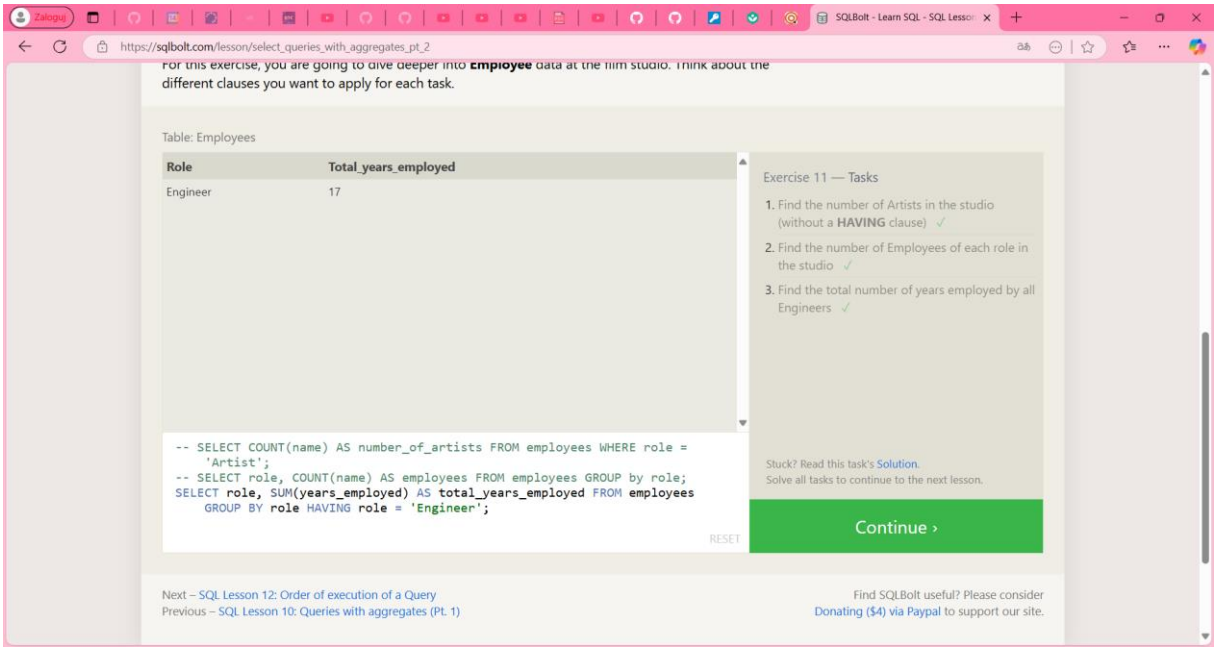
9


https://sqlbolt.com/lesson/select_queries_with_expressions
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
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
-- SELECT title, (domestic_sales + international_sales) / 1000000 AS
gross_sales FROM movies JOIN boxoffice ON movies.id = boxoffice.movie_id
;
-- -SELECT title, rating * 10 AS rating_percent FROM movies JOIN boxoffice
ON movies.id = boxoffice.movie_id;
SELECT title FROM movies WHERE year % 2 = 0;
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.
RESET
Continue ›
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.

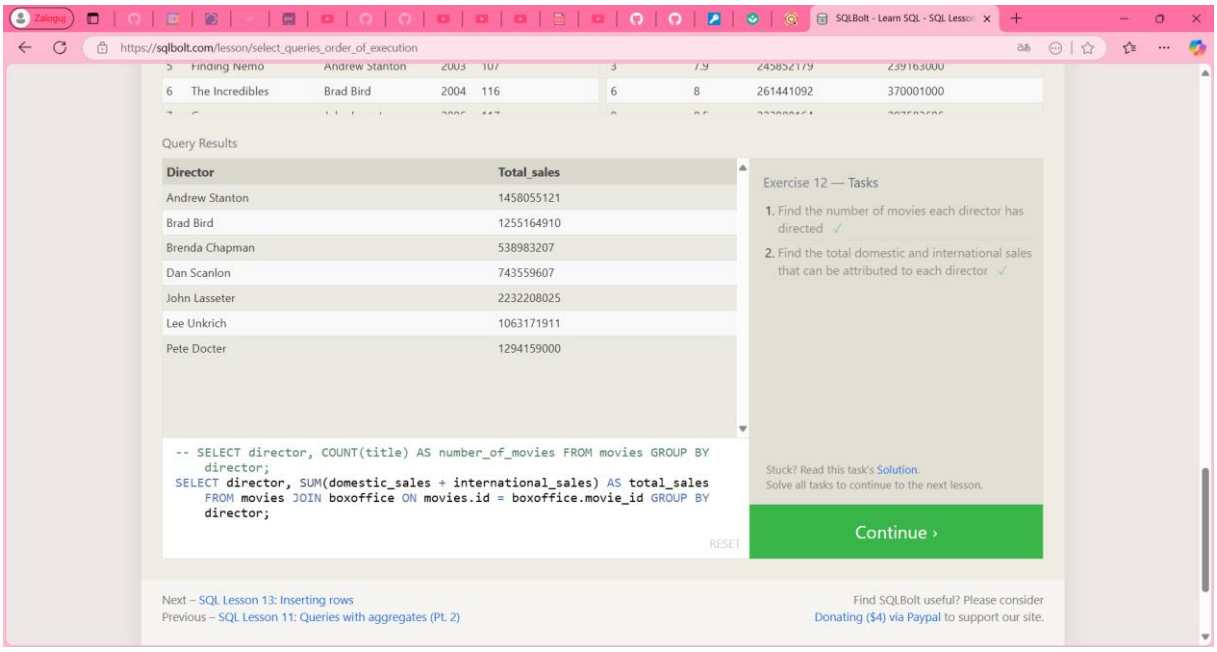
10


SQLBolt - Learn SQL - SQL Lesson
https://sqlbolt.com/lesson/select_queries_with_aggregates
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
Total_years
1e
29
2w
36
-- SELECT MAX(years_employed) AS maximum_years_employed FROM employees;
-- SELECT role, AVG(years_employed) AS average_number_of_years FROM employees GROUP BY role;
SELECT building, SUM(years_employed) AS total_years FROM employees GROUP BY building;
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 ›
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.

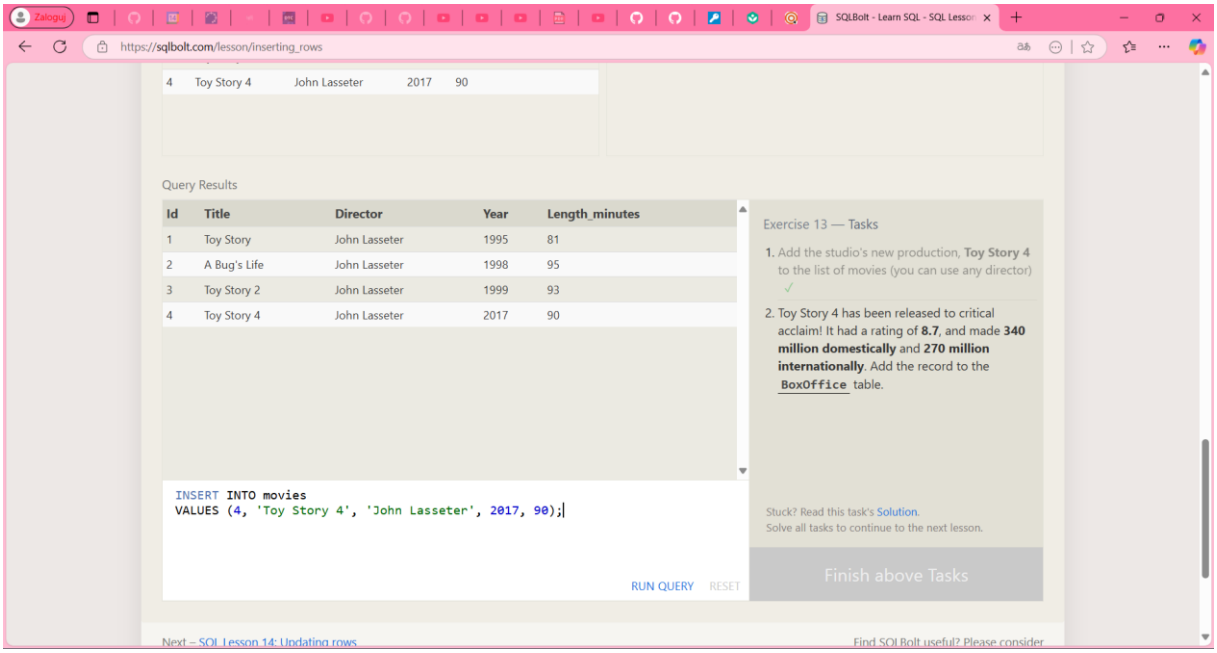
11

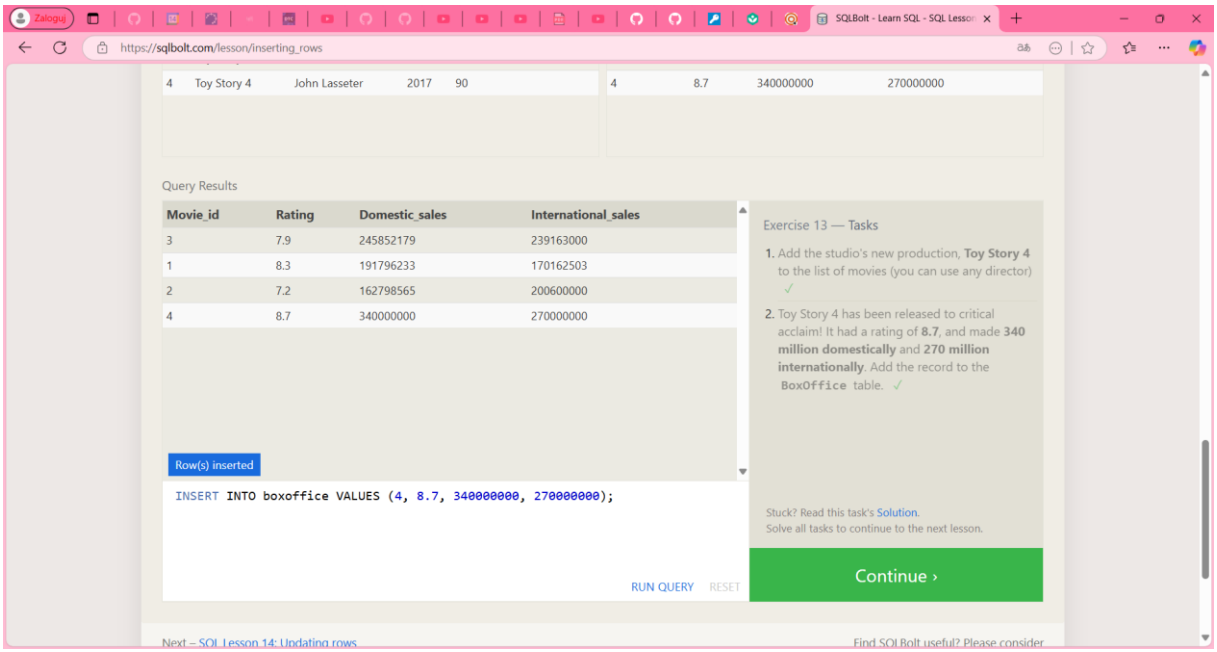

SQLBolt - Learn SQL - SQL Lesson
https://sqlbolt.com/lesson/select_queries_with_aggregates_pt_2
For this exercise, you are going to dive deeper into employee data at the film studio. Think about the different clauses you want to apply for each task.
Table: Employees
Role
Total_years_employed
Engineer
17
-- SELECT COUNT(name) AS number_of_artists FROM employees WHERE role = 'Artist';
-- SELECT role, COUNT(name) AS employees FROM employees GROUP by role;
SELECT role, SUM(years_employed) AS total_years_employed FROM employees GROUP BY role HAVING role = 'Engineer';
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 ›
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.

12

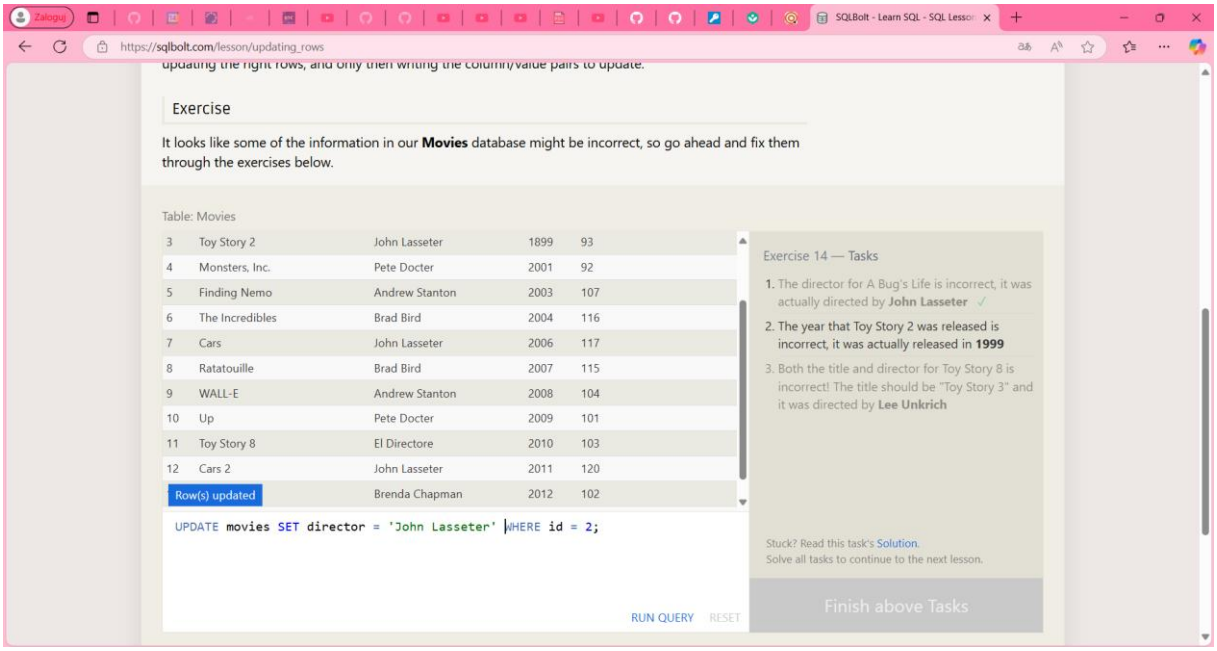

SQLBolt - Learn SQL - SQL Lesson
https://sqlbolt.com/lesson/select_queries_order_of_execution
6 The Incredibles Brad Bird 2004 116 6 8 261441092 370001000
Query Results
Director Total_sales
Andrew Stanton 1458055121
Brad Bird 1255164910
Brenda Chapman 538983207
Dan Scanlon 743559607
John Lasseter 2232208025
Lee Unkrich 1063171911
Pete Docter 1294159000
-- SELECT director, COUNT(title) AS number_of_movies FROM movies GROUP BY director;
SELECT director, SUM(domestic_sales + international_sales) AS total_sales FROM movies 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.

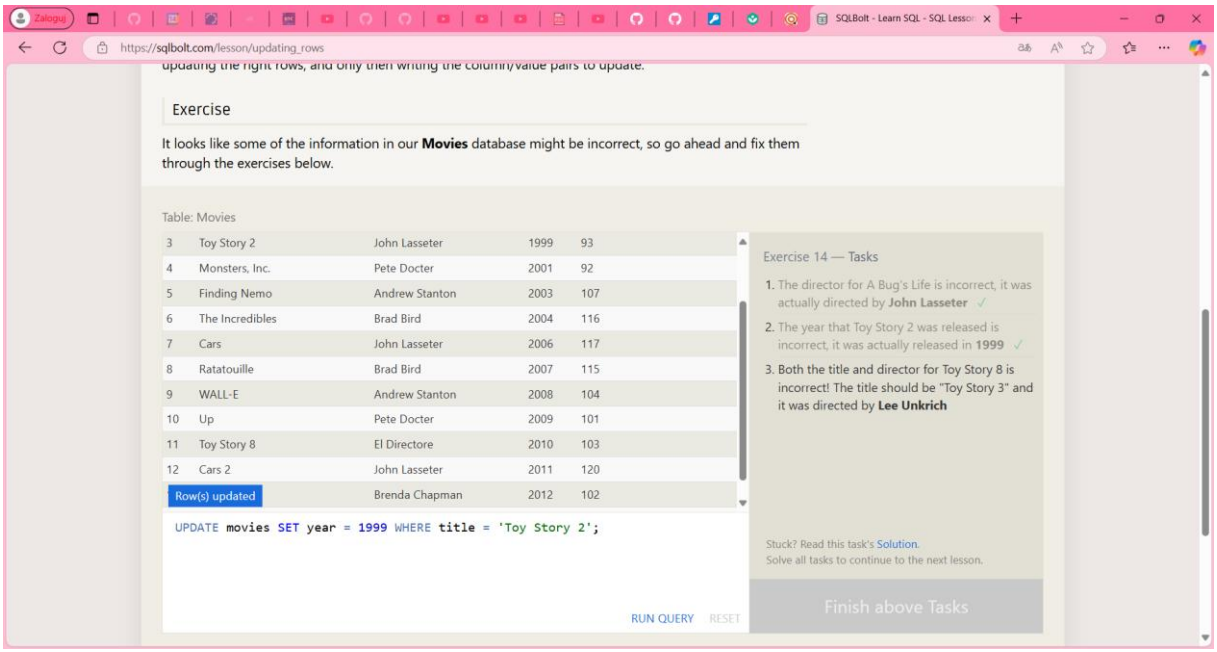
13

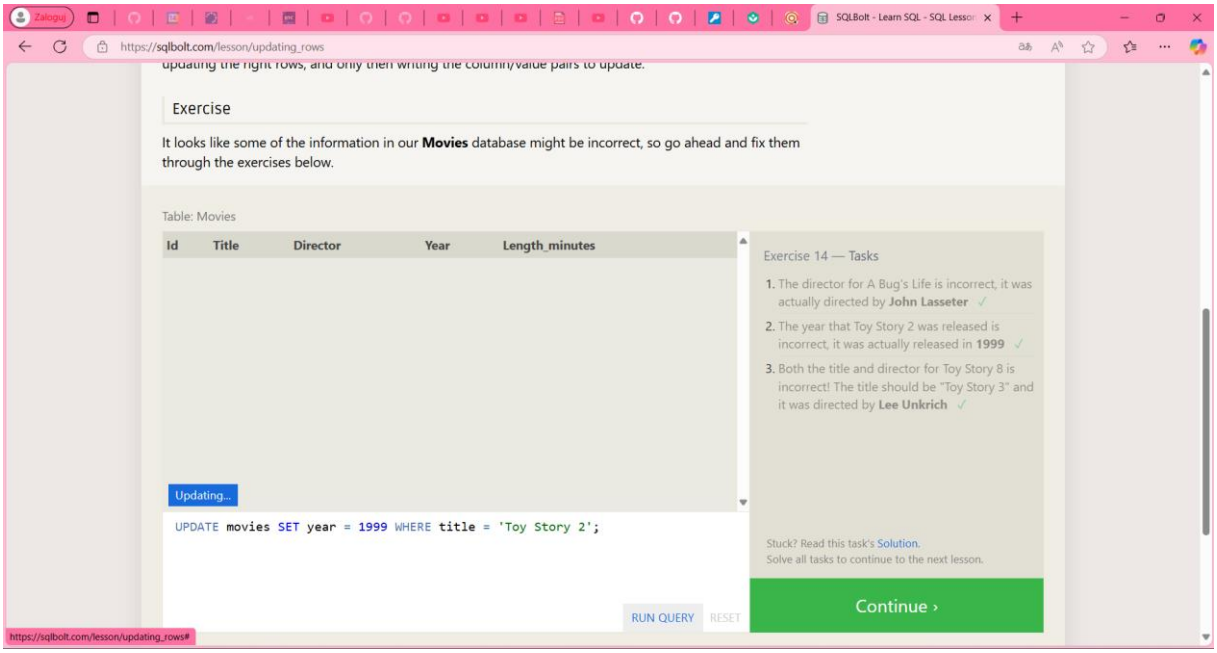

SQLBolt - Learn SQL - SQL Lesson
https://sqlbolt.com/lesson/inserting_rows
4 Toy Story 4 John Lasseter 2017 90
Query Results
Id Title Director Year Length_minutes
1 Toy Story John Lasseter 1995 81
2 A Bug's Life John Lasseter 1998 95
3 Toy Story 2 John Lasseter 1999 93
4 Toy Story 4 John Lasseter 2017 90
INSERT INTO movies
VALUES (4, 'Toy Story 4', 'John Lasseter', 2017, 90);
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.
Finish above Tasks
Next – SQL Lesson 14: Updating rows
Find SQLBolt useful? Please consider

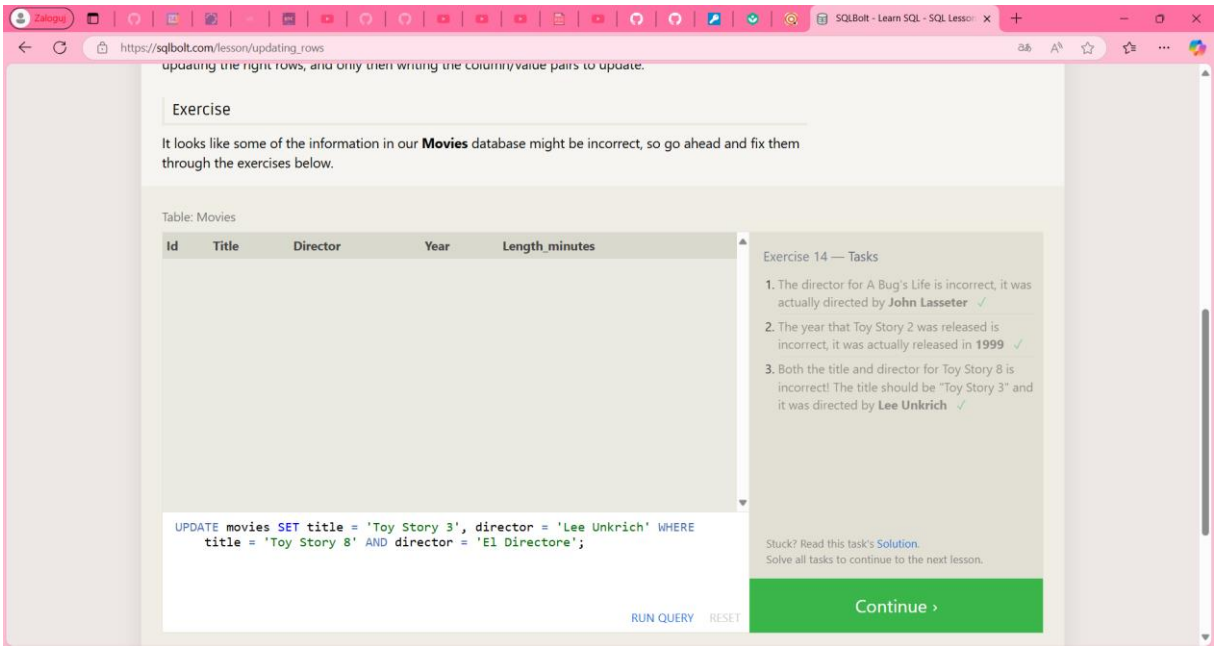

SQLBolt - Learn SQL - SQL Lesson
https://sqlbolt.com/lesson/inserting_rows
4 Toy Story 4 John Lasseter 2017 90
4 8.7 340000000 270000000
Query Results
Movie_id Rating Domestic_sales International_sales
3 7.9 245852179 239163000
1 8.3 191796233 170162503
2 7.2 162798565 200600000
4 8.7 340000000 270000000
Row(s) inserted
INSERT INTO boxoffice VALUES (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
Find SQLBolt useful? Please consider

14

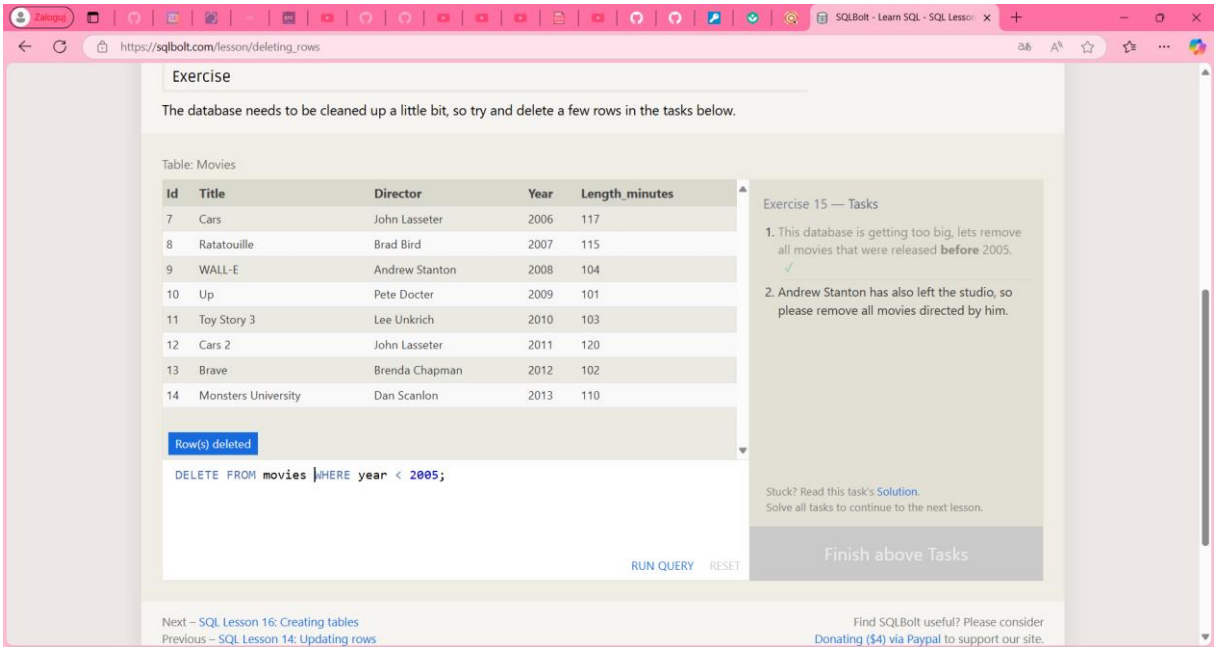

SQLBolt - Learn SQL - SQL Lesson
https://sqlbolt.com/lesson/updating_rows
Exercise
It looks like some of the information in our Movies database might be incorrect, so go ahead and fix them through the exercises below.
Table: Movies
3 Toy Story 2 John Lasseter 1899 93
4 Monsters, Inc. Pete Docter 2001 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
11 Toy Story 8 El Directore 2010 103
12 Cars 2 John Lasseter 2011 120
Row(s) updated Brenda Chapman 2012 102
UPDATE movies SET director = 'John Lasseter' WHERE id = 2;
RUN QUERY RESET
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.
Finish above Tasks

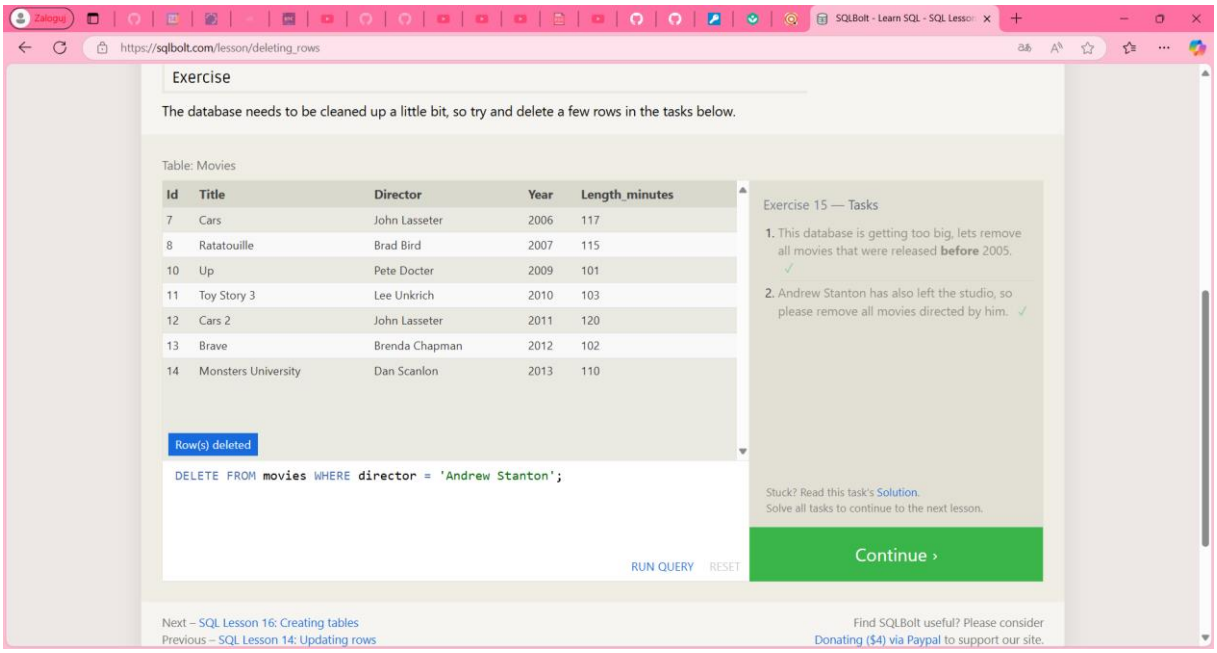

https://sqlbolt.com/lesson/updating_rows
updating the right rows, and only then writing the column/value pairs to update.
Exercise
It looks like some of the information in our Movies database might be incorrect, so go ahead and fix them through the exercises below.
Table: Movies
3 Toy Story 2 John Lasseter 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
11 Toy Story 8 El Directore 2010 103
12 Cars 2 John Lasseter 2011 120
Row(s) updated
Brenda Chapman 2012 102
UPDATE movies SET year = 1999 WHERE title = 'Toy Story 2';
RUN QUERY
RESET
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.
Finish above Tasks


https://sqlbolt.com/lesson/updating_rows
updating the right rows, and only then writing the column/value pairs to update.
Exercise
It looks like some of the information in our Movies database might be incorrect, so go ahead and fix them through the exercises below.
Table: Movies
Id Title Director Year Length_minutes
Updating...
UPDATE movies SET year = 1999 WHERE title = 'Toy Story 2';
RUN QUERY
RESET
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.
Continue ›

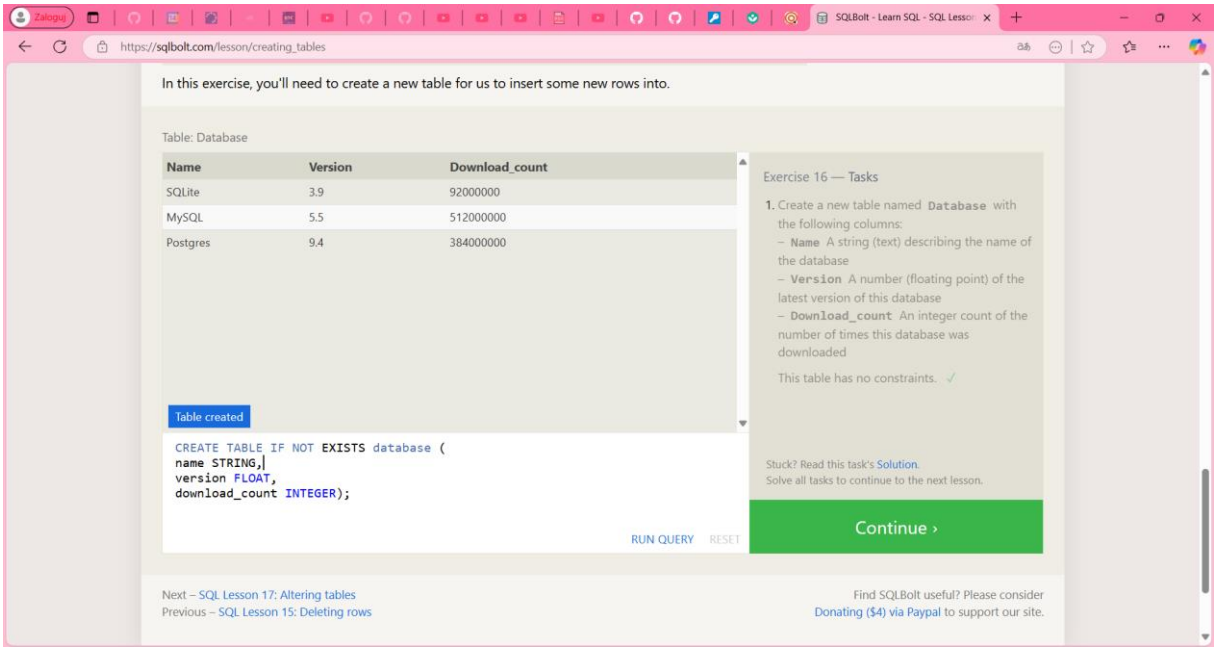

https://sqlbolt.com/lesson/updating_rows
Exercise
It looks like some of the information in our **Movies** database might be incorrect, so go ahead and fix them through the exercises below.
Table: Movies
Id Title Director Year Length_minutes
Exercise 14 — Tasks
1. The director for A Bug's Life is incorrect, 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**
UPDATE movies SET title = 'Toy Story 3', director = 'Lee Unkrich' WHERE title = 'Toy Story 8' AND director = 'El Directore';
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.
RUN QUERY
RESET
Continue ›

15

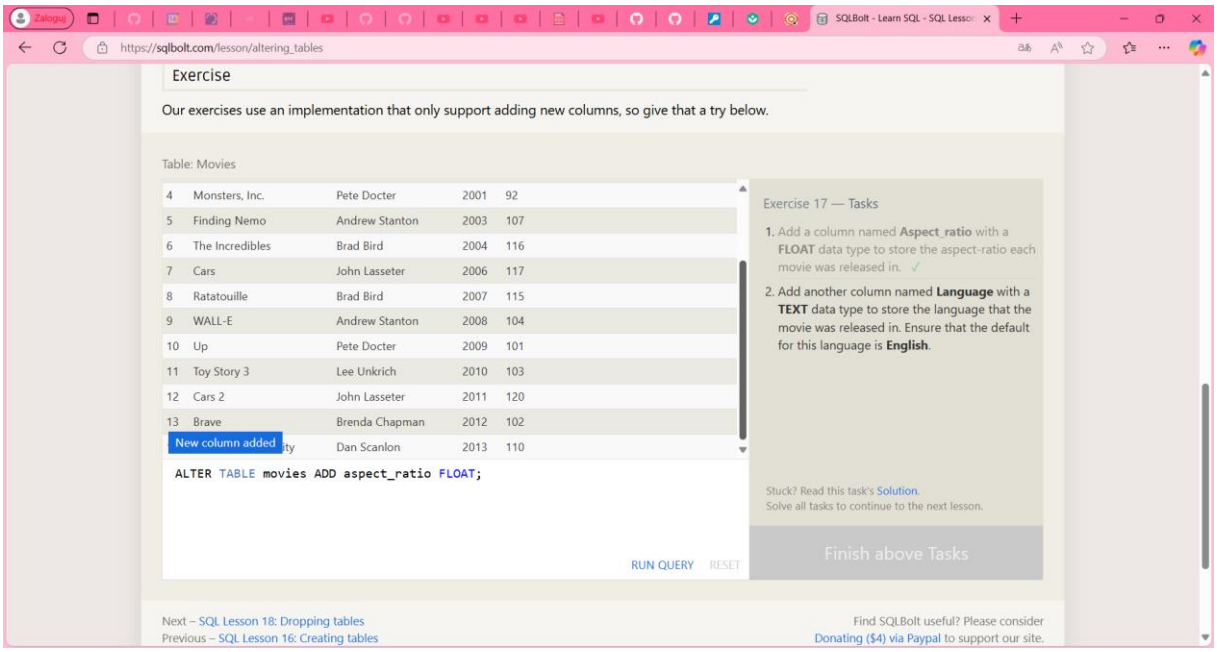

https://sqlbolt.com/lesson/deleting_rows
Exercise
The database needs to be cleaned up a little bit, so try and delete a few rows in the tasks below.
Table: Movies
Id Title Director Year Length_minutes
7 Cars John Lasseter 2006 117
8 Ratatouille Brad Bird 2007 115
9 WALL-E Andrew Stanton 2008 104
10 Up 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
Row(s) deleted
DELETE FROM movies WHERE year < 2005;
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.
RUN QUERY
RESET
Finish above Tasks
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.

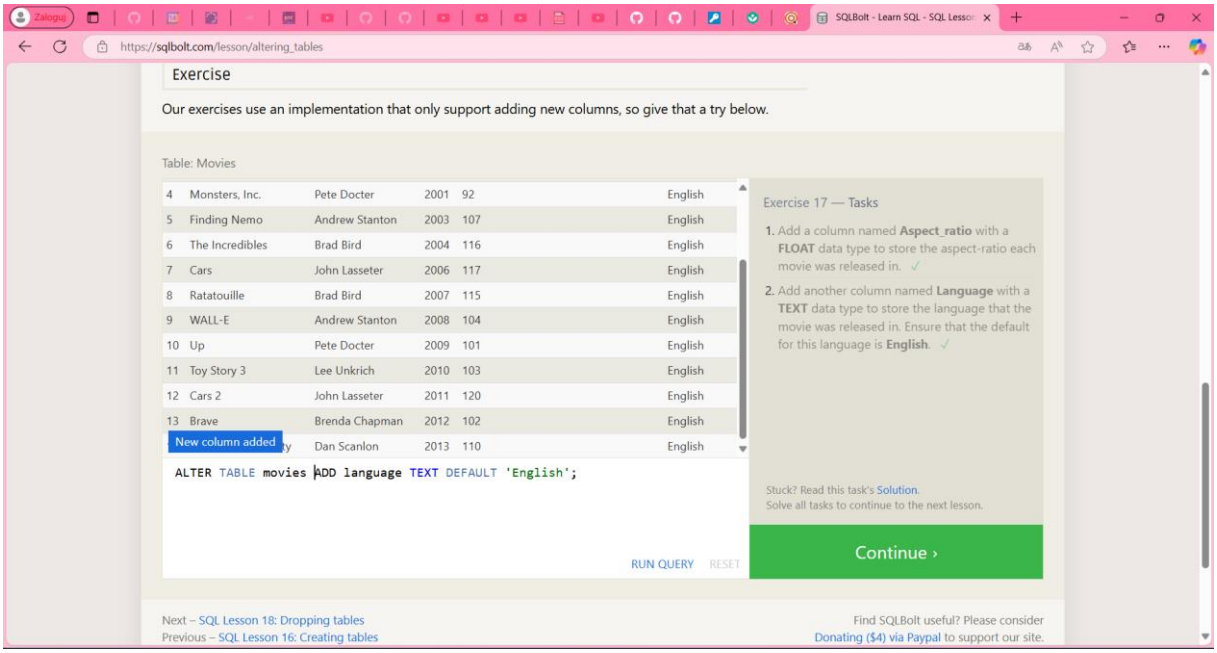

https://sqlbolt.com/lesson/deleting_rows
Exercise
The database needs to be cleaned up a little bit, so try and delete a few rows in the tasks below.
Table: Movies
Id Title Director Year Length_minutes
7 Cars John Lasseter 2006 117
8 Ratatouille Brad Bird 2007 115
10 Up Pete Docter 2009 101
11 Toy Story 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
Row(s) deleted
DELETE FROM movies WHERE director = 'Andrew Stanton';
RUN QUERY
RESET
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.
Continue ›
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.

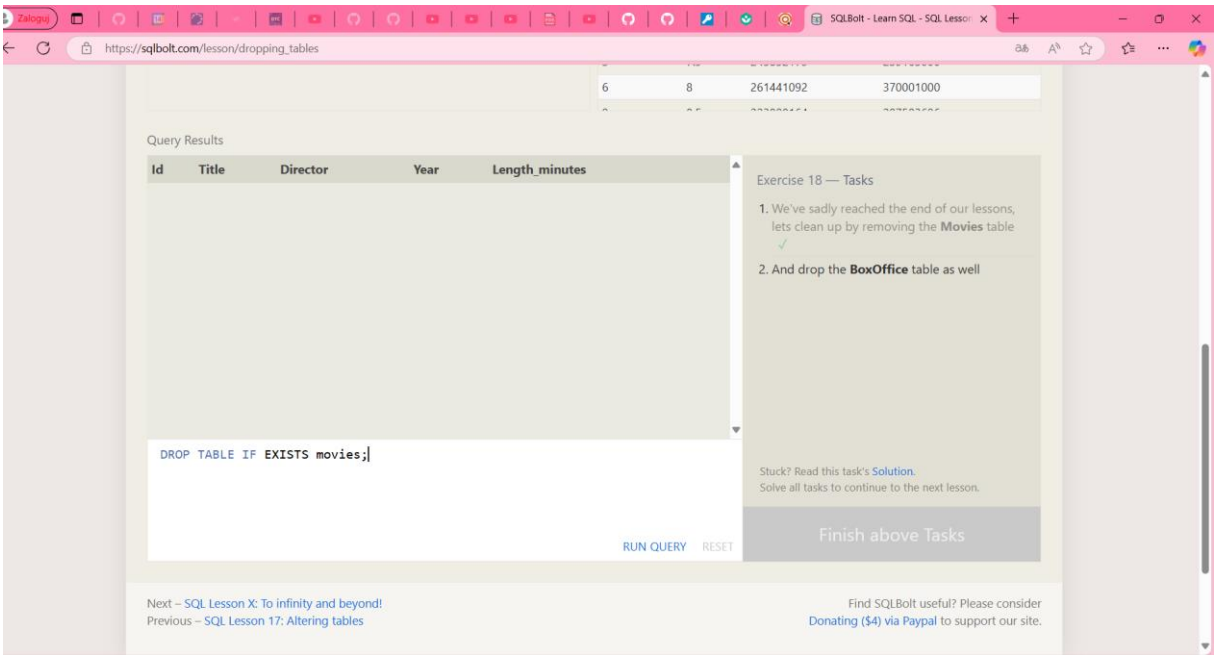
16

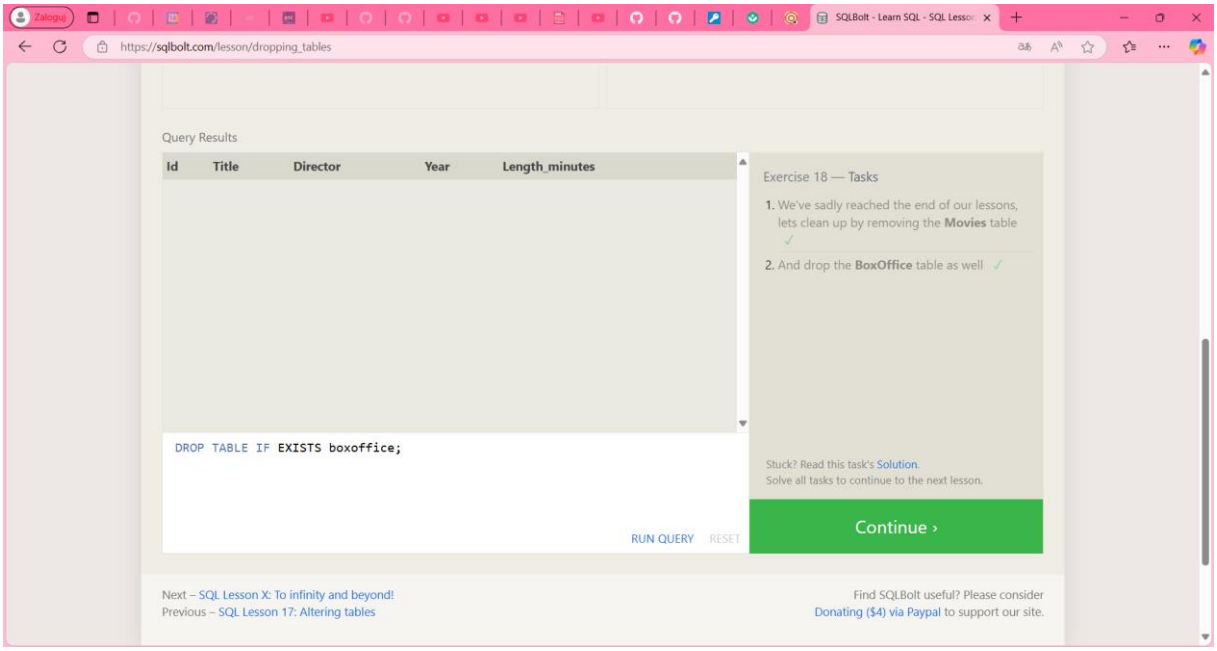

https://sqlbolt.com/lesson/creating_tables
In this exercise, you'll need to create a new table for us to insert some new rows into.
Table: Database
Name Version Download_count
SQLite 3.9 92000000
MySQL 5.5 512000000
Postgres 9.4 384000000
Table created
CREATE TABLE IF NOT EXISTS database (
name STRING,
version FLOAT,
download_count INTEGER);
RUN QUERY
RESET
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.
Continue ›
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.

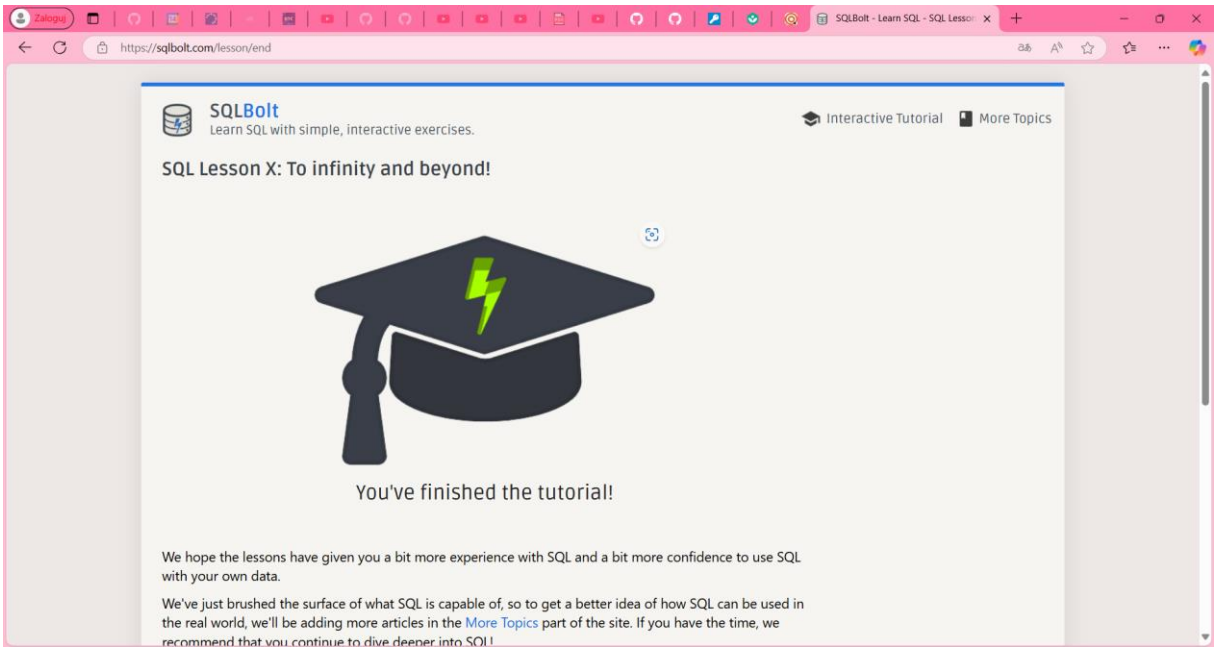
17


https://sqlbolt.com/lesson/altering_tables
Exercise
Our exercises use an implementation that only support adding new columns, so give that a try below.
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
New column added
Dan Scanlon 2013 110
ALTER TABLE movies ADD aspect_ratio FLOAT;
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.
Finish above Tasks
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.


https://sqlbolt.com/lesson/altering_tables
Exercise
Our exercises use an implementation that only support adding new columns, so give that a try below.
Table: Movies
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
11 Toy Story 3 Lee Unkrich 2010 103 English
12 Cars 2 John Lasseter 2011 120 English
13 Brave Brenda Chapman 2012 102 English
New column added
Dan Scanlon 2013 110 English
ALTER TABLE movies ADD language TEXT DEFAULT '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 ›
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.


SQLBolt - Learn SQL - SQL Lesson
https://sqlbolt.com/lesson/dropping_tables
6 8 261441092 370001000
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
DROP TABLE IF EXISTS movies;
RUN QUERY RESET
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.
Finish above Tasks
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.


SQLBolt - Learn SQL - SQL Lesson
https://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
DROP TABLE IF EXISTS boxoffice;
RUN QUERY RESET
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.
Continue ›
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.


SQLBolt - Learn SQL - SQL Lesson
https://sqlbolt.com/lesson/end
SQLBolt
Learn SQL with simple, interactive exercises.
Interactive Tutorial
More Topics
SQL Lesson X: To infinity and beyond!
You've finished the tutorial!
We hope the lessons have given you a bit more experience with SQL and a bit more confidence to use SQL with your own data.
We've just brushed the surface of what SQL is capable of, so to get a better idea of how SQL can be used in the real world, we'll be adding more articles in the More Topics part of the site. If you have the time, we recommend that you continue to dive deeper into SQL!