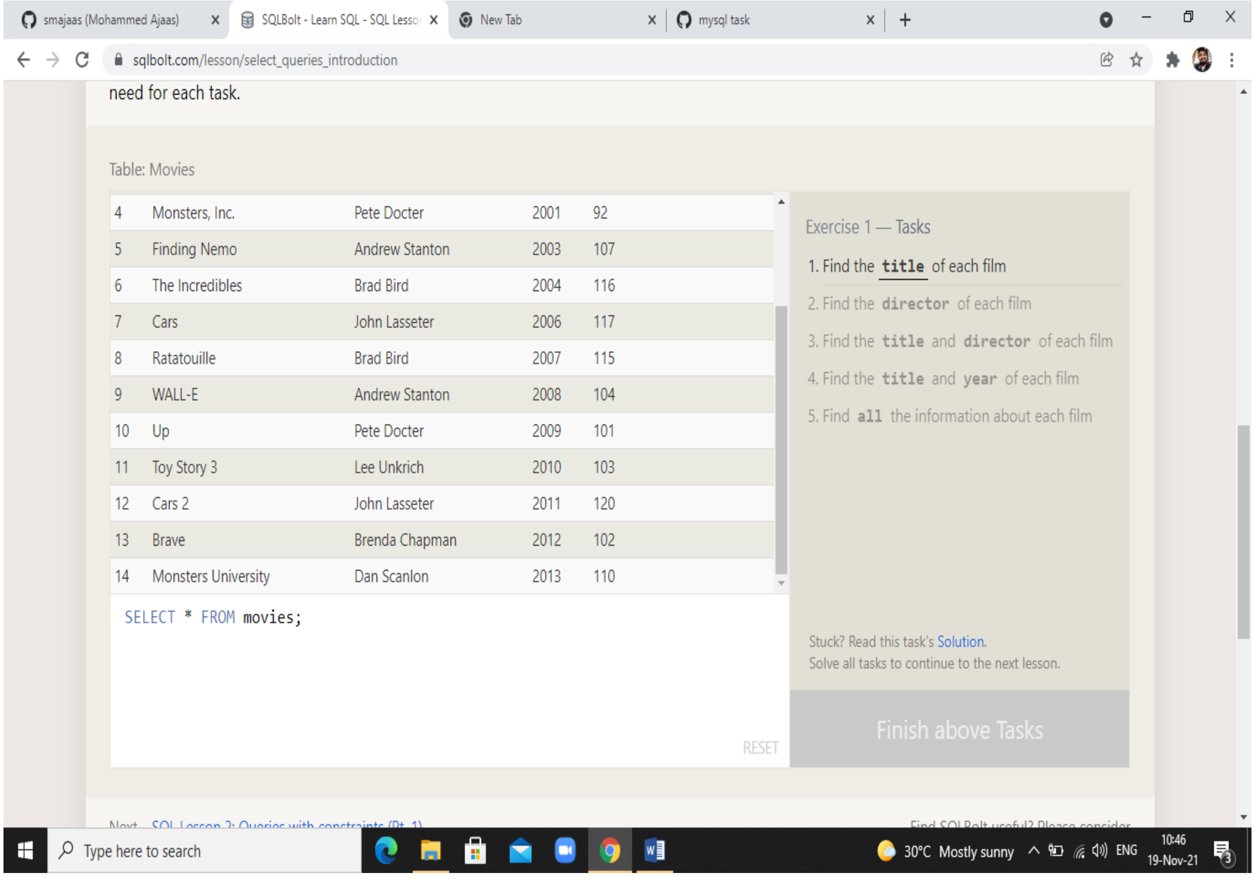
# This Exercise from https://sqlbolt.com/

# SQL Lesson 1: SELECT queries 101


smajaas (Mohammed Ajaas)
SQLBolt - Learn SQL - SQL Lesso
New Tab
mysql task
sqlbolt.com/lesson/select_queries_introduction
RESET
Finish above Tasks
Next - SQL Lesson 2: Queries with constraints (Pt. 1)
Find SQLBolt useful? Please consider
Type here to search
30°C Mostly sunny
ENG
10:46
19-Nov-21

need for each task.

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 |

```
SELECT * FROM movies;
```

Exercise 1 — Tasks

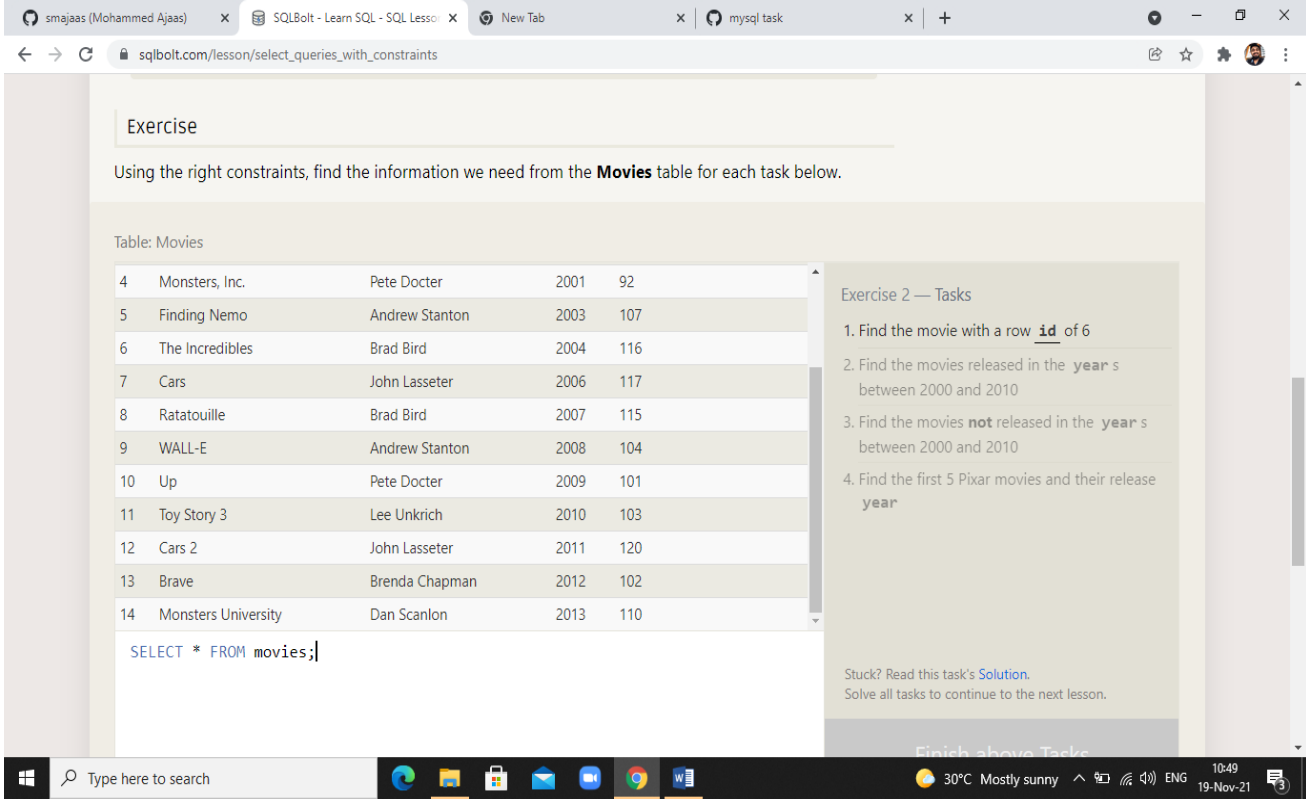
1. Find the `title` of each film
2. Find the `director` of each film
3. Find the `title` and `director` of each film
4. Find the `title` and `year` of each film
5. Find `all` the information about each film

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

Answers :

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

# SQL Lesson 2: Queries with constraints (Pt. 1)



Exercise

Using the right constraints, find the information we need from the **Movies** table for each task 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 |
| 14 | Monsters University | Dan Scanlon | 2013 | 110 |

```
SELECT * FROM movies;
```

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.

Answers :

1. SELECT * FROM movies where id >= 6
2. SELECT * FROM movies where year between 2000 and 2010
3. SELECT * FROM movies where year Not between 2000 and 2010
4. SELECT * FROM movies WHERE id <=5;

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

Answers :

1. SELECT title FROM movies where title Like "Toy Story%";

2. SELECT *FROM movies where director= "John Lasseter";
3. SELECT *FROM movies where director!= "John Lasseter";
4. SELECT * FROM movies where title like "WALL-%";

## SQL Lesson 4: Filtering and sorting Query results

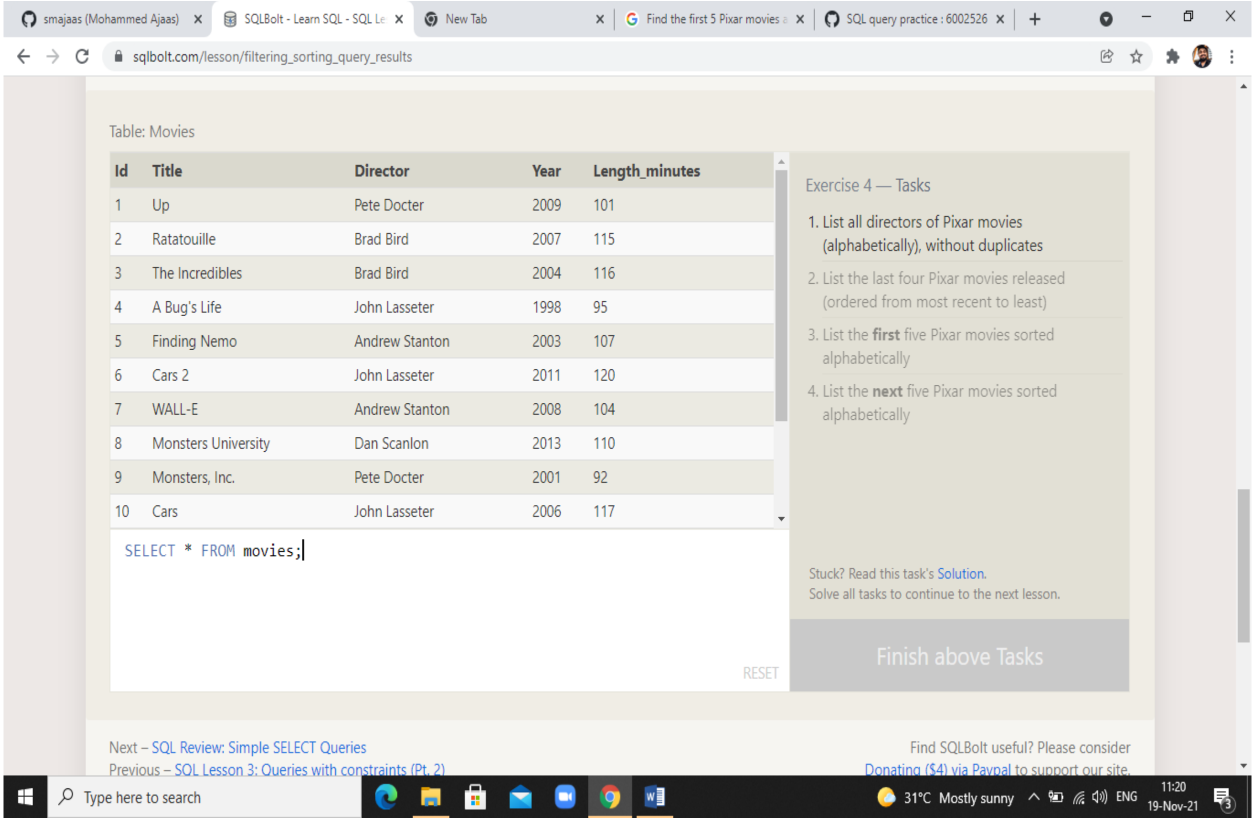

smajaas (Mohammed Ajaas)
SQLBolt - Learn SQL - SQL Le
New Tab
Find the first 5 Pixar movies a
SQL query practice : 6002526
sqlbolt.com/lesson/filtering_sorting_query_results

Table: Movies

| Id | Title | Director | Year | Length_minutes |
|---|---|---|---|---|
| 1 | Up | Pete Docter | 2009 | 101 |
| 2 | Ratatouille | Brad Bird | 2007 | 115 |
| 3 | The Incredibles | Brad Bird | 2004 | 116 |
| 4 | A Bug's Life | John Lasseter | 1998 | 95 |
| 5 | Finding Nemo | Andrew Stanton | 2003 | 107 |
| 6 | Cars 2 | John Lasseter | 2011 | 120 |
| 7 | WALL-E | Andrew Stanton | 2008 | 104 |
| 8 | Monsters University | Dan Scanlon | 2013 | 110 |
| 9 | Monsters, Inc. | Pete Docter | 2001 | 92 |
| 10 | Cars | John Lasseter | 2006 | 117 |

```
SELECT * FROM movies;
```

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.

## Answers :

1. SELECT distinct director FROM movies
   order by director;
2. SELECT DISTINCT title FROM movies
   ORDER BY year DESC LIMIT 4;
3. SELECT title FROM movies
   ORDER BY title LIMIT 5;

4. SELECT title FROM movies
   ORDER BY title LIMIT 5 OFFSET 5;

## Lesson:5 SQL Review: Simple SELECT Queries

## Answers :

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

## SQL Lesson 6: Multi-table queries with JOINs

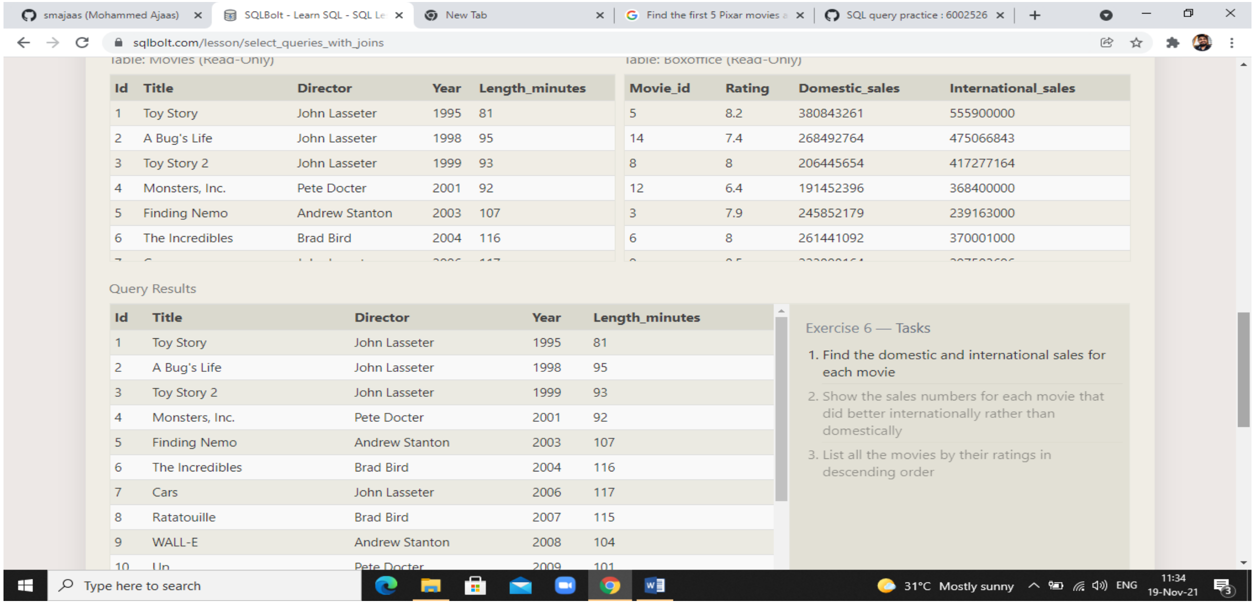

smajaas (Mohammed Ajaas)
SQLBolt - Learn SQL - SQL Le
New Tab
Find the first 5 Pixar movies a
SQL query practice : 6002526
sqlbolt.com/lesson/select_queries_with_joins
Type here to search
31°C Mostly sunny
ENG
11:34
19-Nov-21

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

| 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 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

## Answers :

1. SELECT title, domestic_sales, international_sales FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id;
2. SELECT title, domestic_sales, international_sales 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;

## SQL Lesson 7: OUTER JOINs

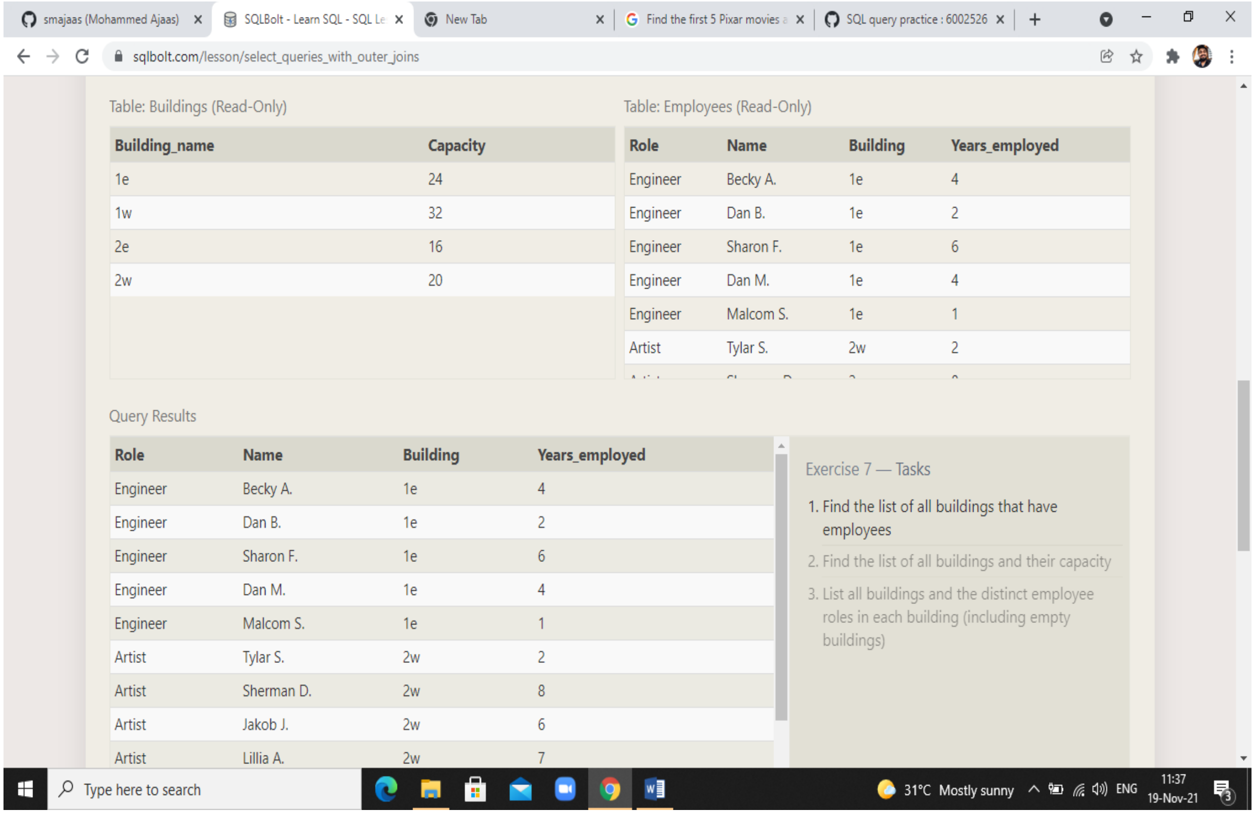

smajaas (Mohammed Ajaas)
SQLBolt - Learn SQL - SQL Le
New Tab
Find the first 5 Pixar movies a
SQL query practice : 6002526
sqlbolt.com/lesson/select_queries_with_outer_joins
Type here to search
31°C Mostly sunny
ENG
11:37
19-Nov-21

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

| Role | Name | Building | Years_employed |
|---|---|---|---|
| Engineer | Becky A. | 1e | 4 |
| Engineer | Dan B. | 1e | 2 |
| Engineer | Sharon F. | 1e | 6 |
| Engineer | Dan M. | 1e | 4 |
| Engineer | Malcom S. | 1e | 1 |
| Artist | Tylar S. | 2w | 2 |
| Artist | Sherman D. | 2w | 8 |
| Artist | Jakob J. | 2w | 6 |
| Artist | Lillia A. | 2w | 7 |

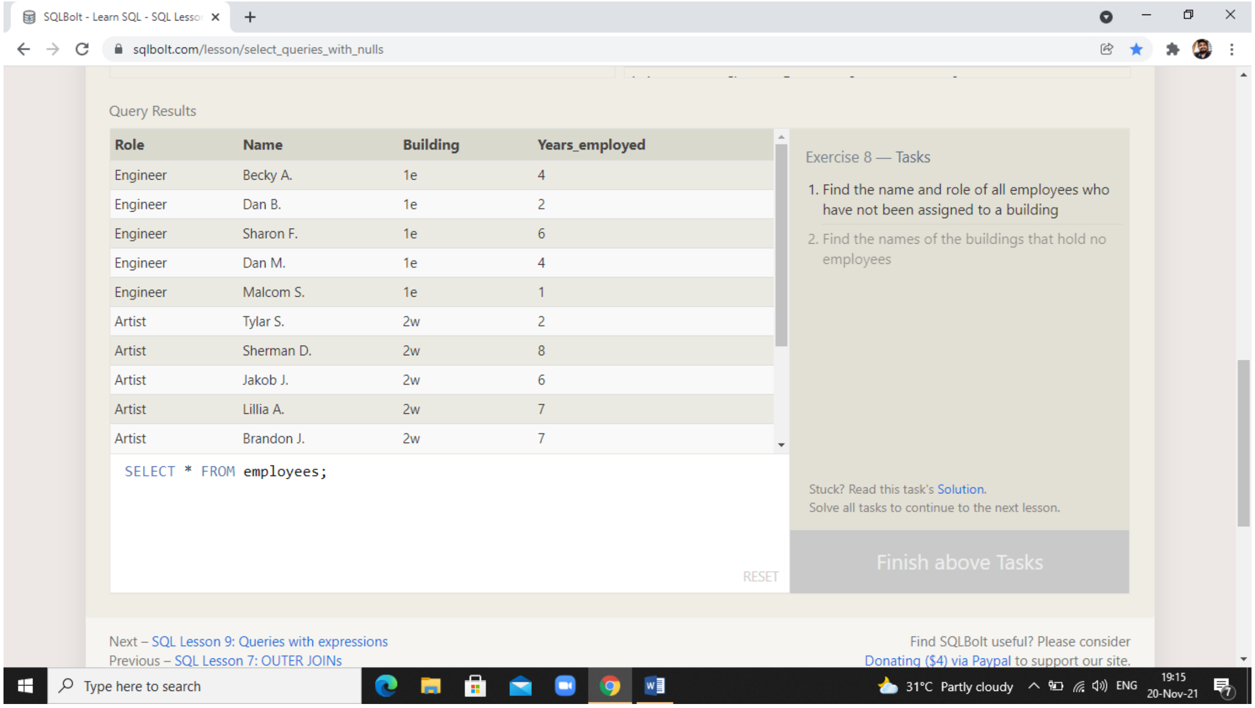
Exercise 7 — Tasks

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)

# Answers :

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

# SQL Lesson 8: A short note on NULLs


SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/select_queries_with_nulls
Type here to search
31°C Partly cloudy
ENG
19:15
20-Nov-21

Query Results

| Role | Name | Building | Years_employed |
|---|---|---|---|
| Engineer | Becky A. | 1e | 4 |
| Engineer | Dan B. | 1e | 2 |
| Engineer | Sharon F. | 1e | 6 |
| Engineer | Dan M. | 1e | 4 |
| Engineer | Malcom S. | 1e | 1 |
| Artist | Tylar S. | 2w | 2 |
| Artist | Sherman D. | 2w | 8 |
| Artist | Jakob J. | 2w | 6 |
| Artist | Lillia A. | 2w | 7 |
| Artist | Brandon J. | 2w | 7 |

```
SELECT * FROM employees;
```

RESET

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.

Finish above Tasks

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.

# Answers :

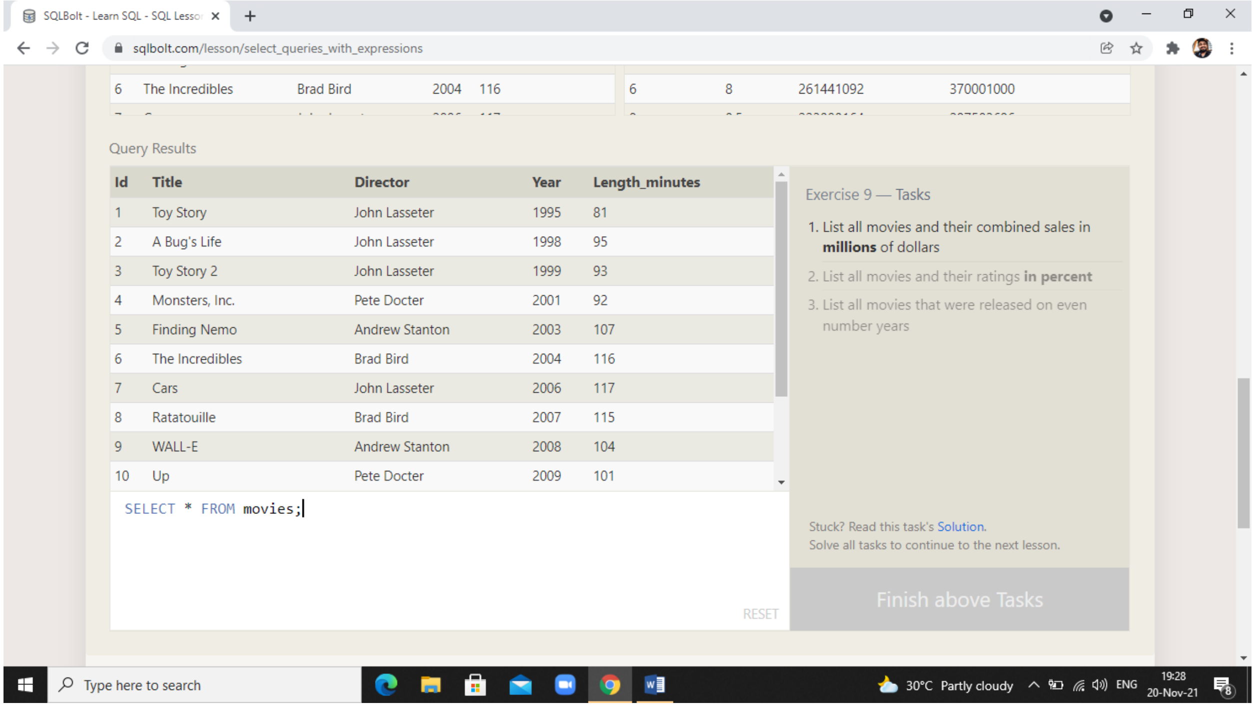
## 1. Find the name and role of all employees who have not been assigned to a building

SELECT * FROM employees where building is null

## 2. Find the names of the buildings that hold no employees

SELECT Building_name from buildings left join employees on building_name=building where role is null;

# SQL Lesson 9: Queries with expressions:



SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/select_queries_with_expressions

|   |   |   |   |   |
|---|---|---|---|---|
| 6 | The Incredibles | Brad Bird | 2004 | 116 |

|   |   |   |   |
|---|---|---|---|
| 6 | 8 | 261441092 | 370001000 |

Query Results

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

```
SELECT * FROM movies;
```

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.

# Answers :

## 1. List all movies and their combined sales in millions of dollars:

```
SELECT title,(Domestic_sales+International_sales)/1000000 as Total_sales_In_Million FROM movies inner join Boxoffice on Movies.id = movie_id;
```

2. **List all movies and their ratings in percent:**

```
SELECT title, rating*10 as Percentage FROM movies inner join Boxoffice on Movies.id = movie_id;
```

3. **List all movies that were released on even number years:**

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

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

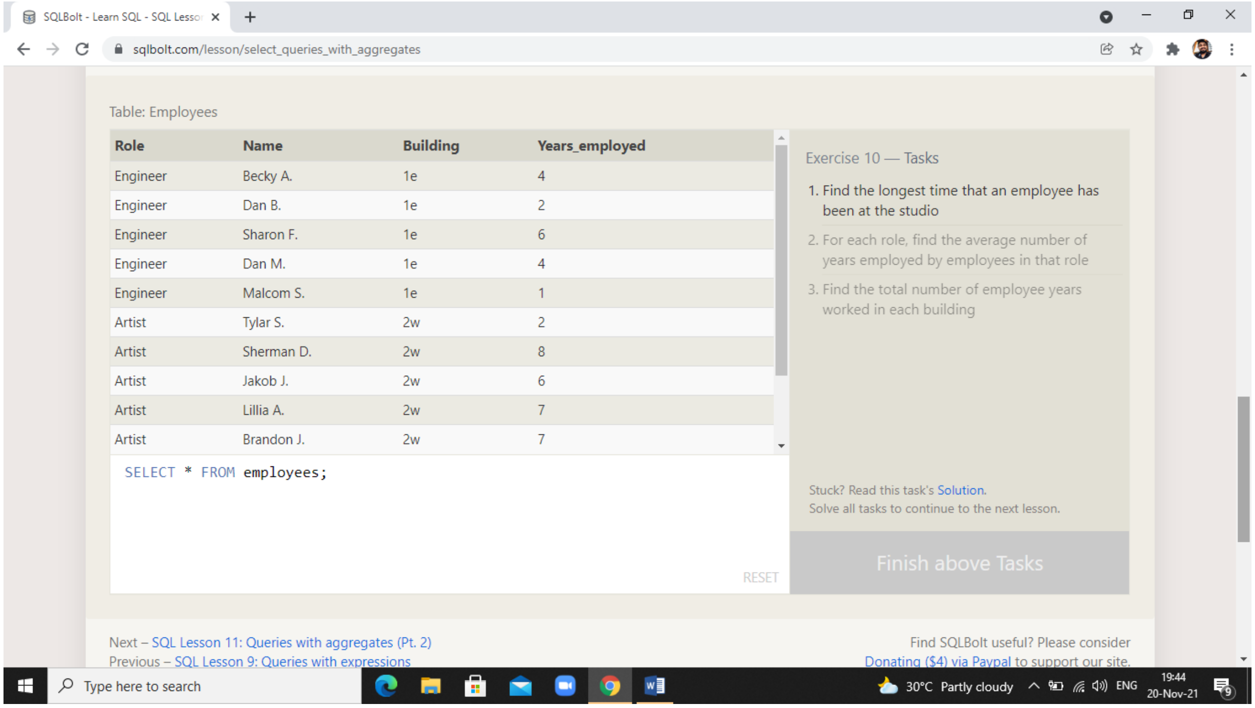

SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/select_queries_with_aggregates
RESET
Finish above Tasks
Type here to search
30°C Partly cloudy
ENG
19:44
20-Nov-21

Table: Employees

| Role | Name | Building | Years_employed |
|---|---|---|---|
| Engineer | Becky A. | 1e | 4 |
| Engineer | Dan B. | 1e | 2 |
| Engineer | Sharon F. | 1e | 6 |
| Engineer | Dan M. | 1e | 4 |
| Engineer | Malcom S. | 1e | 1 |
| Artist | Tylar S. | 2w | 2 |
| Artist | Sherman D. | 2w | 8 |
| Artist | Jakob J. | 2w | 6 |
| Artist | Lillia A. | 2w | 7 |
| Artist | Brandon J. | 2w | 7 |

```
SELECT * FROM employees;
```

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.

# Answers :

1. **Find the longest time that an employee has been at the studio**

```
SELECT distinct name,Max(Years_employed) as longest_service
FROM employees;
```

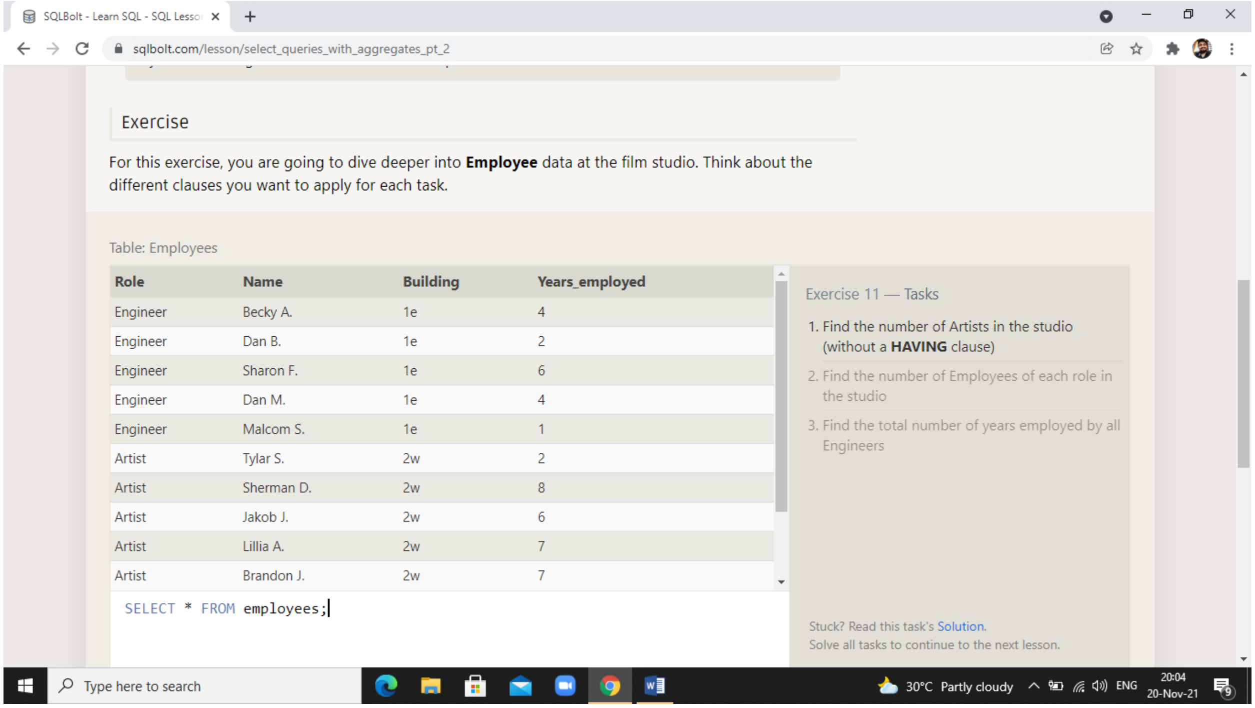
2. For each role, find the average number of years employed by employees in that role

```
SELECT role,avg(Years_employed) as Average_service FROM
employees group by role;
```

3. Find the total number of employee years worked in each building:

```
SELECT building,sum(Years_employed) as Sum_of_years FROM
employees group by building;
```

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


SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/select_queries_with_aggregates_pt_2
Type here to search
30°C Partly cloudy
ENG
20:04
20-Nov-21

Exercise

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 | Name | Building | Years_employed |
|---|---|---|---|
| Engineer | Becky A. | 1e | 4 |
| Engineer | Dan B. | 1e | 2 |
| Engineer | Sharon F. | 1e | 6 |
| Engineer | Dan M. | 1e | 4 |
| Engineer | Malcom S. | 1e | 1 |
| Artist | Tylar S. | 2w | 2 |
| Artist | Sherman D. | 2w | 8 |
| Artist | Jakob J. | 2w | 6 |
| Artist | Lillia A. | 2w | 7 |
| Artist | Brandon J. | 2w | 7 |

```
SELECT * FROM employees;
```

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.

## Answers :

1. **Find the number of Artists in the studio (without a** HAVING **clause)**

**SELECT role, count(role) FROM employees where Role="Artist";**

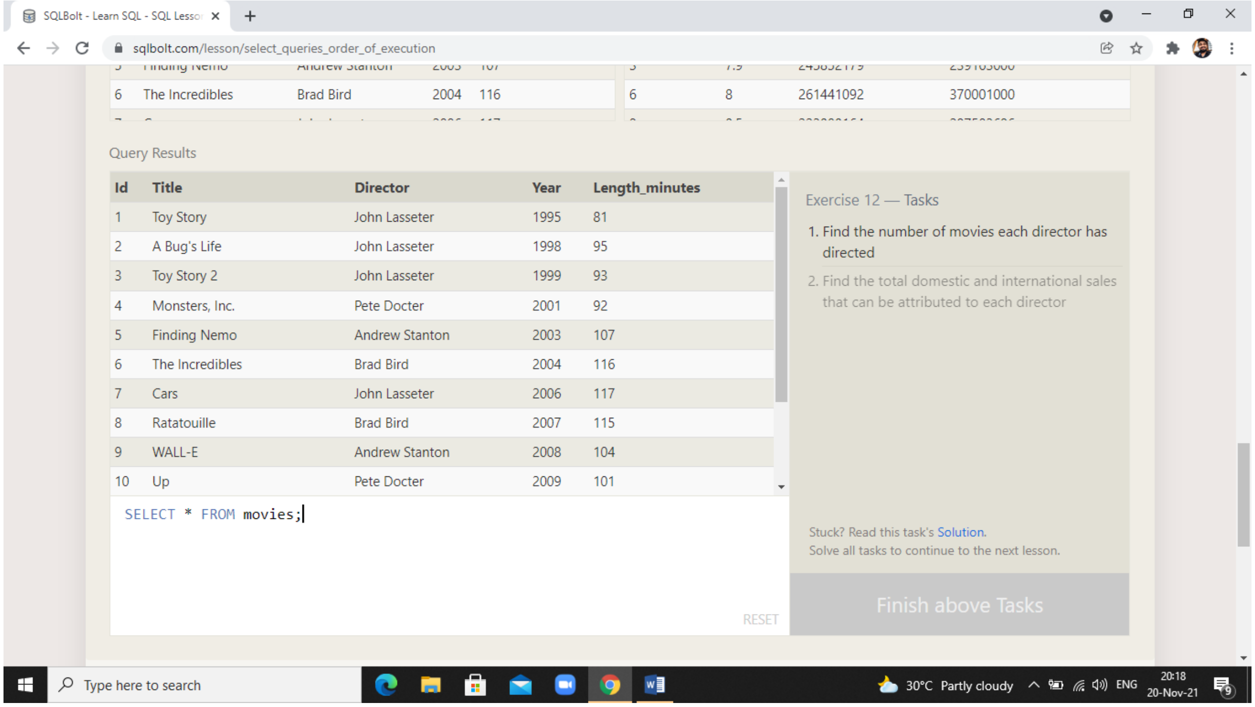
2. **Find the number of Employees of each role in the studio**

**SELECT role,count(role) FROM employees group by role;**

3. **Find the total number of years employed by all Engineers**

   **SELECT role,sum(years_employed) FROM employees where Role="Engineer";**

## SQL Lesson 12: Order of execution of a Query



SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/select_queries_order_of_execution

| | | | | |
|---|---|---|---|---|
| 6 | The Incredibles | Brad Bird | 2004 | 116 |

| | | | |
|---|---|---|---|
| 6 | 8 | 261441092 | 370001000 |

Query Results

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

```
SELECT * FROM movies;
```

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.

## Answers :

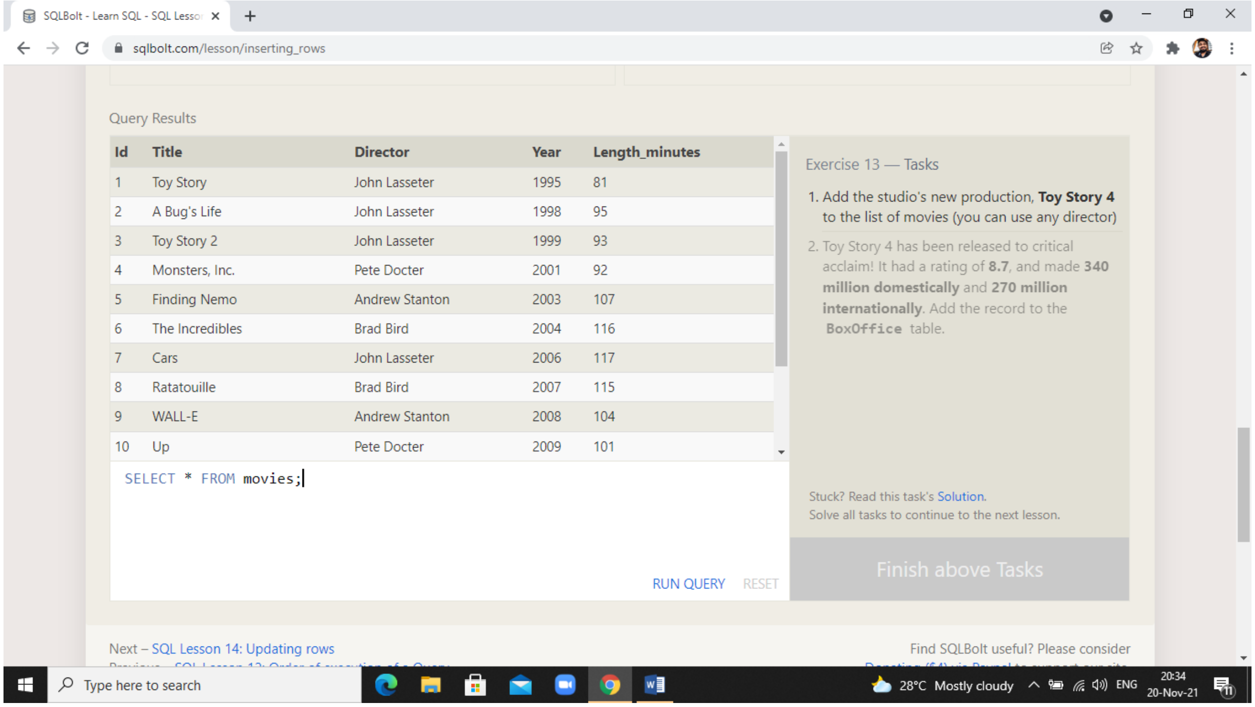
1. **Find the number of movies each director has directed**

```
SELECT Director,count(Director) as count FROM movies group by director order by count desc ;
```

2. Find the total domestic and international sales that can be attributed to each director:

```
SELECT director, sum(Domestic_sales+International_sales)
/1000000 as Total_Collection FROM movies inner join Boxoffice on
Movies.id=Movie_id group by director order by Total_Collection desc ;
```

## SQL Lesson 13: Inserting rows


SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/inserting_rows
Type here to search
28°C Mostly cloudy
ENG
20:34
20-Nov-21

Query Results

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

```
SELECT * FROM movies;
```

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

## Answers :

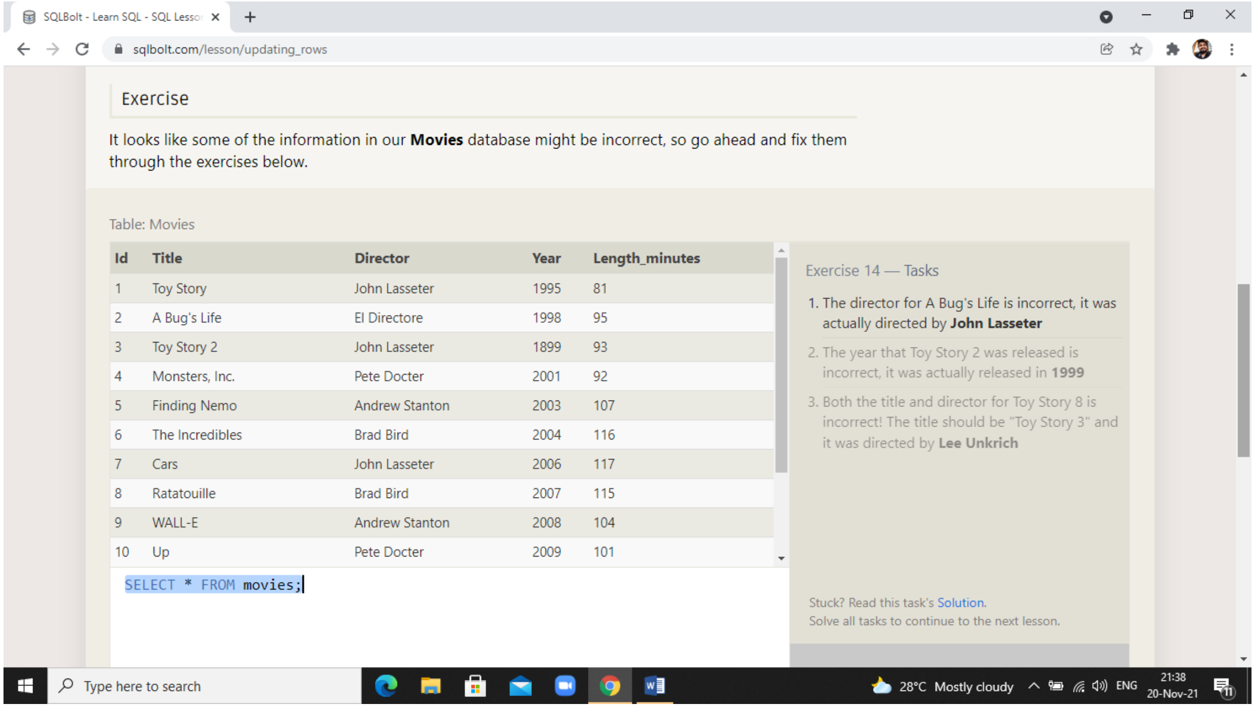
1. Add the studio's new production, Toy Story 4 to the list of movies (you can use any director)

```
INSERT INTO Movies
VALUES (4, "Toy Story 4", "Ajaas", 2021, 120)
```

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.**

```
INSERT INTO Boxoffice
VALUES (4,8.7,340000000,270000000)
```

## SQL Lesson 14: Updating rows:


SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/updating_rows
Type here to search
28°C Mostly cloudy
ENG
21:38
20-Nov-21

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 |
|---|---|---|---|---|
| 1 | Toy Story | John Lasseter | 1995 | 81 |
| 2 | A Bug's Life | El Directore | 1998 | 95 |
| 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 |

```
SELECT * FROM movies;
```

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.

## Answers:

1. **The director for A Bug's Life is incorrect, it was actually directed by John Lasseter**

```
update movies
 set director = "John Lasseter"
 where id=2;
```

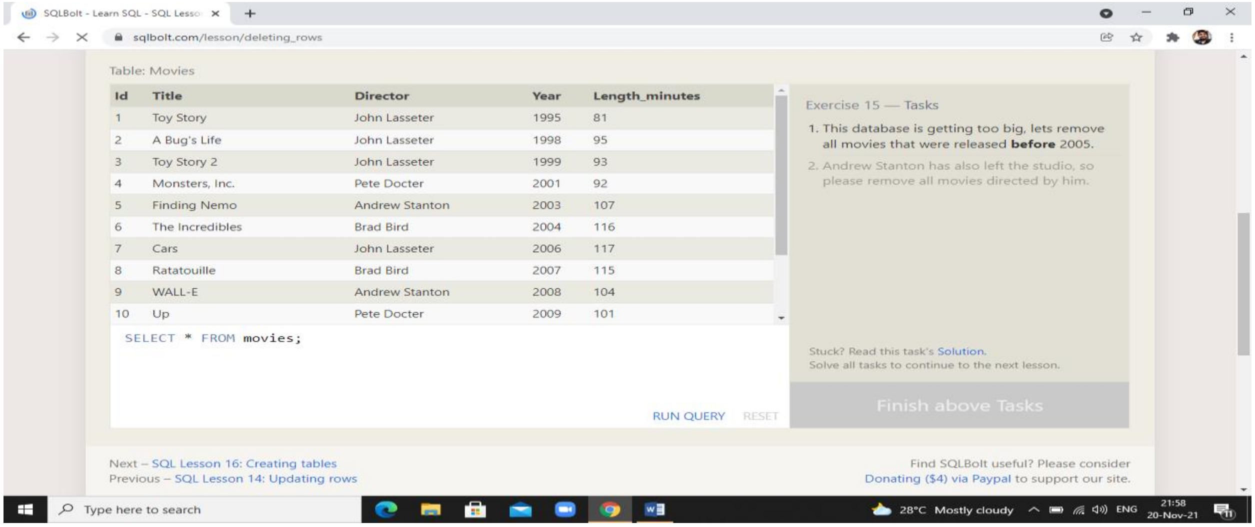
2. The year that Toy Story 2 was released is incorrect, it was actually released in 1999.

```
update movies
set year = 1999
where id=3;
```

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 id = 11;
```

## SQL Lesson 15: Deleting rows



SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/deleting_rows

Table: Movies

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

```
SELECT * FROM movies;
```

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.

## Answers:

1. This database is getting too big, lets remove all movies that were released before 2005.

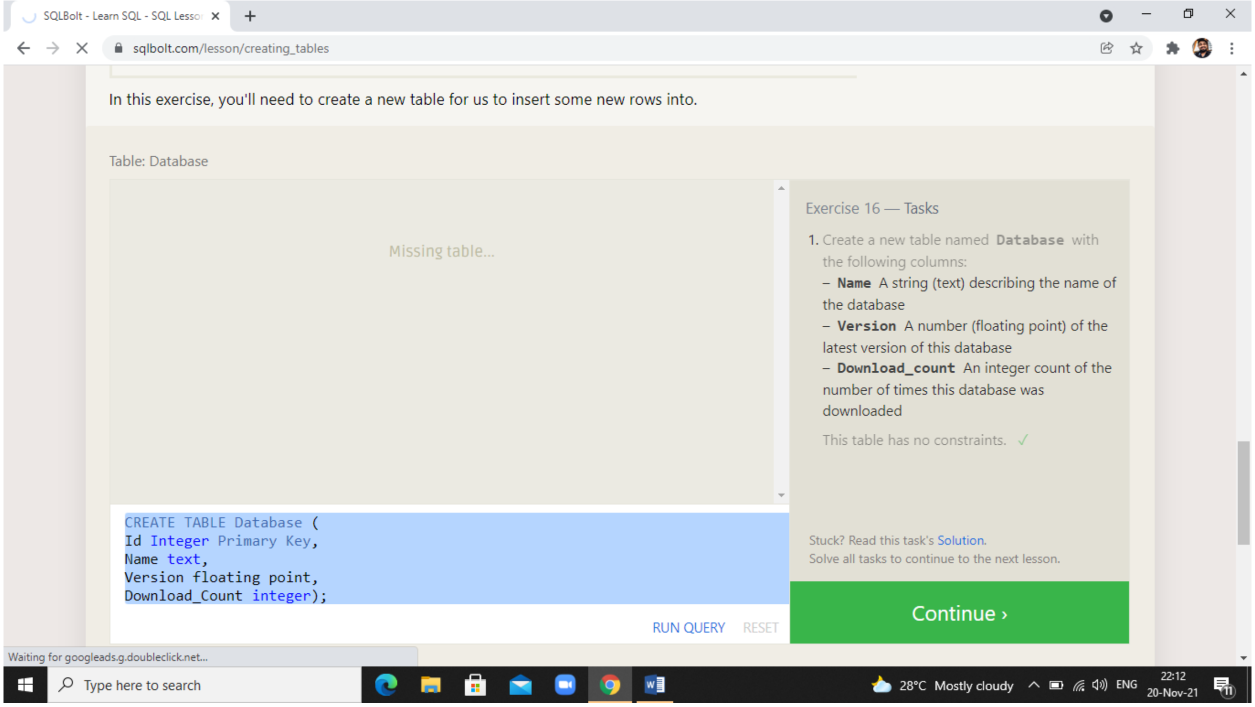
```
DELETE FROM Movies

where title >2005;
```

2. Andrew Stanton has also left the studio, so please remove all movies directed by him.

```
DELETE FROM Movies
where director = "Andrew Stanton";
```

## SQL Lesson 16: Creating tables



SQLBolt - Learn SQL - SQL Lesso
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

Missing table...

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. ✓

```
CREATE TABLE Database (
Id Integer Primary Key,
Name text,
Version floating point,
Download_Count integer);
```

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

## Answers :

**CREATE TABLE Database (**

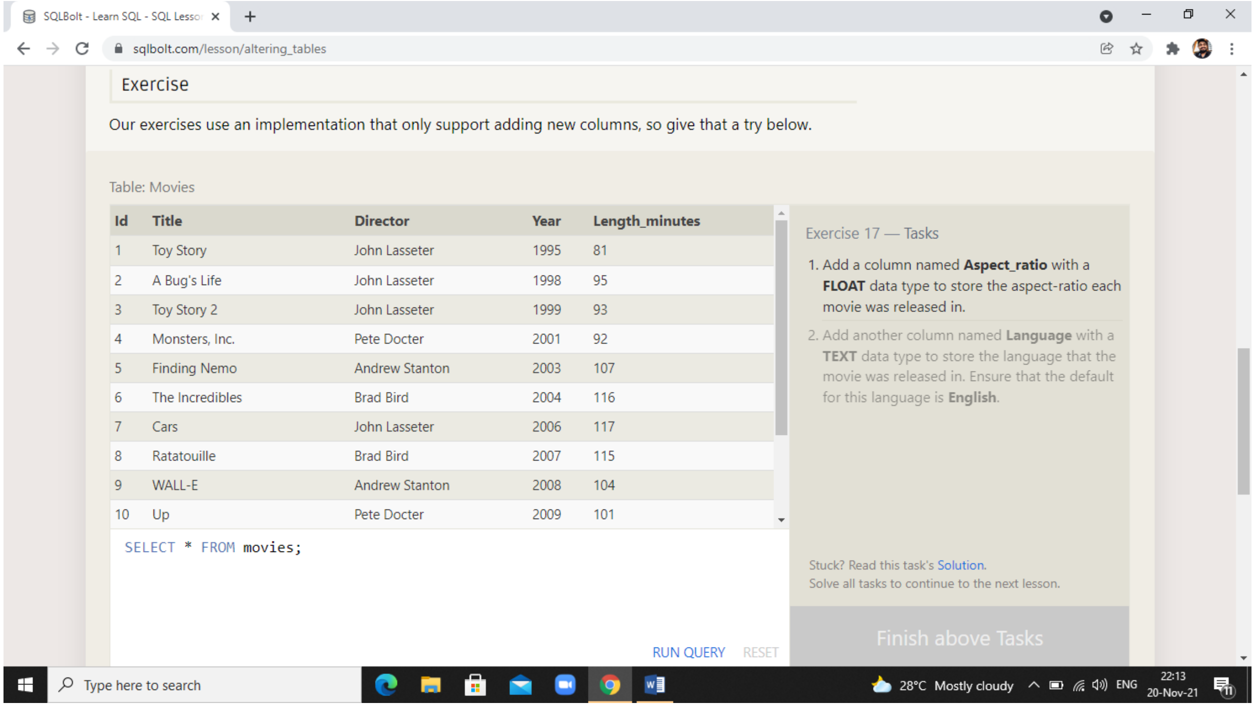
**Id Integer Primary Key,**

**Name text,**

**Version floating point,**

**Download_Count integer);**

# SQL Lesson 17: Altering tables:


SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/altering_tables
Type here to search
28°C Mostly cloudy
ENG
22:13
20-Nov-21

Exercise

Our exercises use an implementation that only support adding new columns, so give that a try 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 |
| 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;
```

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

## Answers :

1. **Add a column named Aspect_ratio with a FLOAT data type to store the aspect-ratio each movie was released in.**

**ALTER TABLE Movies**

**ADD Aspect_ratio Float;**

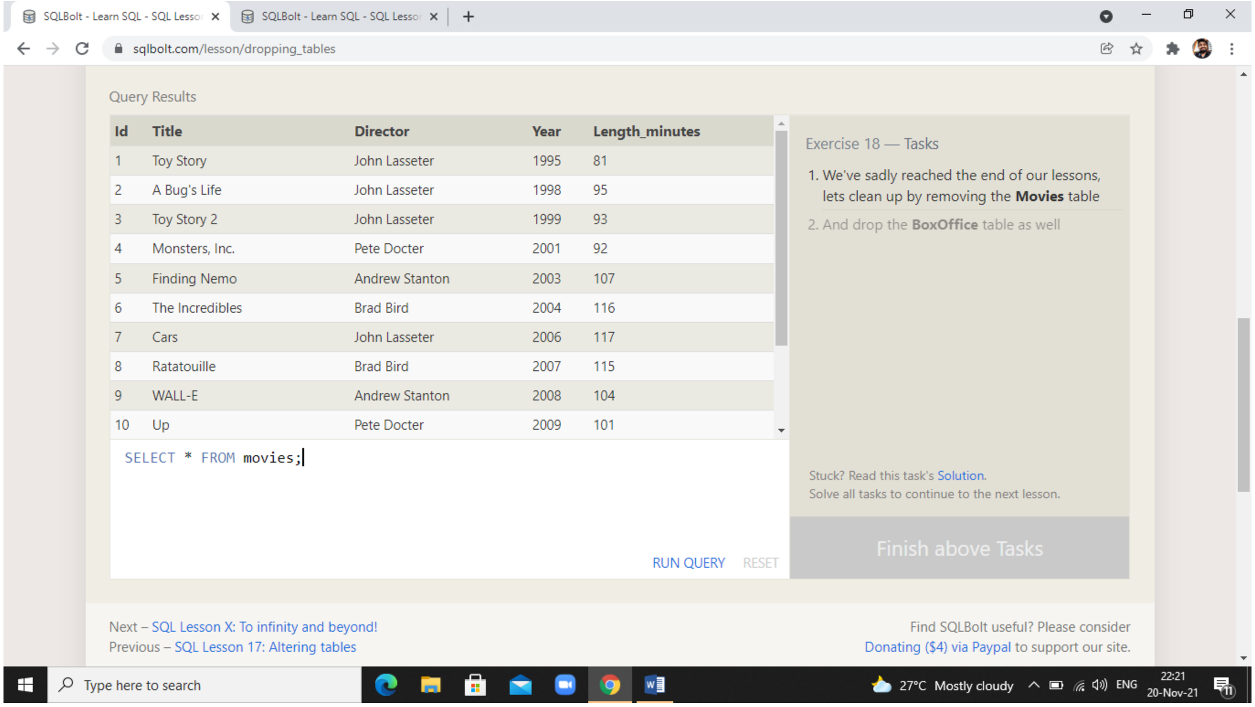
2. **Add another column named Language with a TEXT data type to store the language that the movie was released in. Ensure that the default for this language is** English.

**ALTER TABLE Movies**

**ADD Language Text**

**DEFAULT English;**

# SQL Lesson 18: Dropping tables


SQLBolt - Learn SQL - SQL Lesso
SQLBolt - Learn SQL - SQL Lesso
sqlbolt.com/lesson/dropping_tables
RUN QUERY
RESET
Finish above Tasks
Type here to search
27°C Mostly cloudy
ENG
22:21
20-Nov-21

Query Results

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

```
SELECT * FROM movies;
```

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. **We've sadly reached the end of our lessons, lets clean up by removing the Movies table**

```
DROP TABLE IF EXISTS MOVIES
```

2. **And drop the BoxOffice table as well**

```
DROP TABLE IF EXISTS BOXOFFICE
```