SQL Lesson 1: SELECT queries 101:

1)SELECT(title) FROM movies;

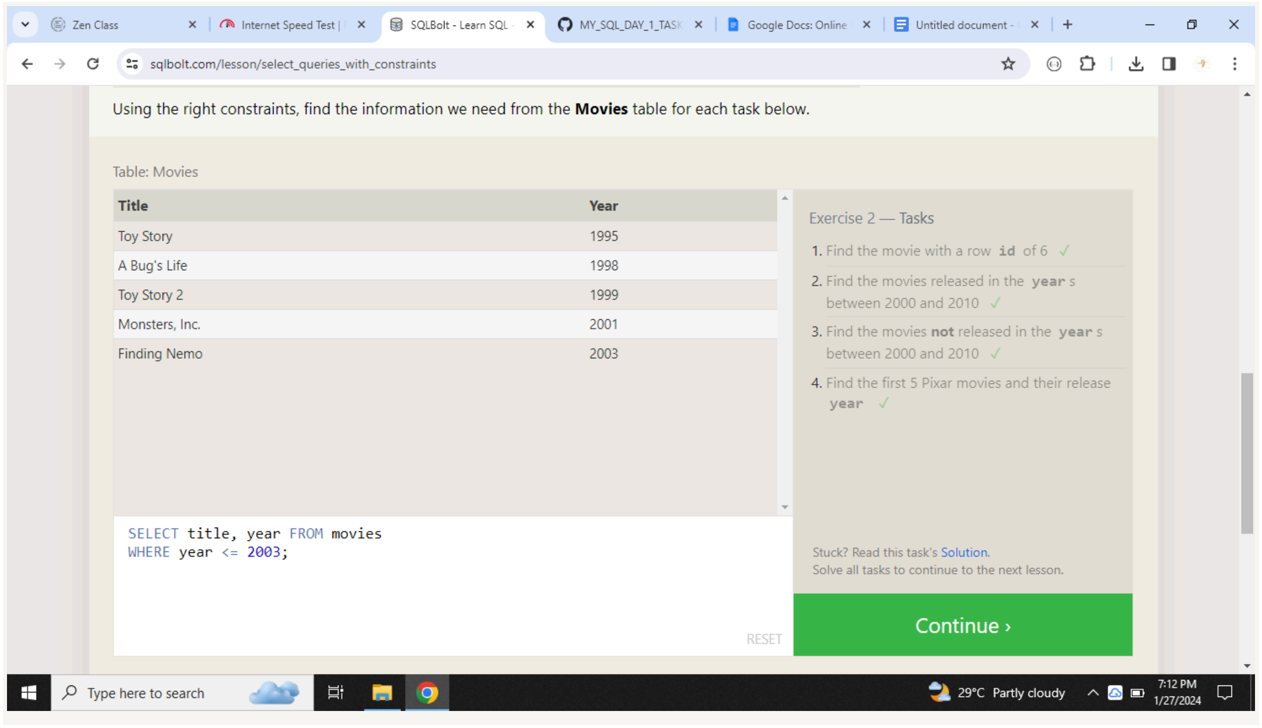
2) SELECT(Director) FROM movies;

3) SELECT title, director FROM movies;

4) SELECT title,year FROM movies;

5) SELECT * FROM movies;

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



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, year FROM movies
WHERE year <= 2003;
```

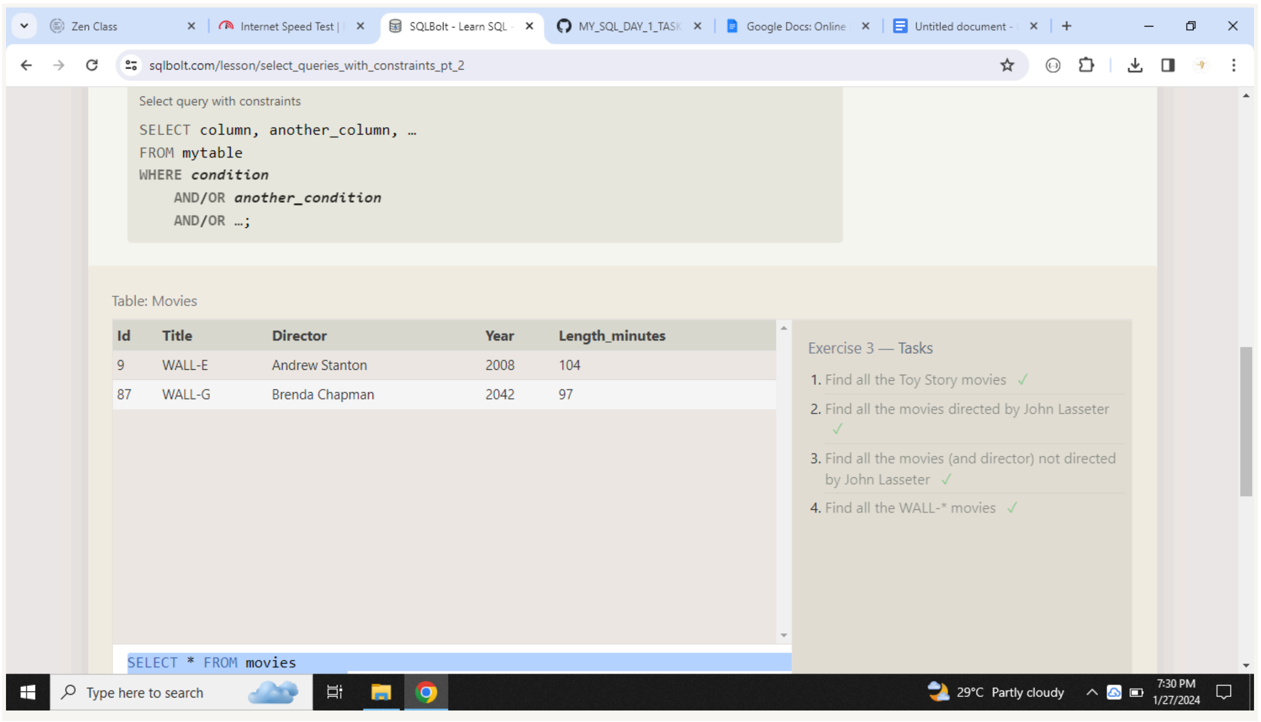
Exercise 2 — Tasks

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.

```
1)SELECT title FROM movies where id=6;
2)SELECT title,year FROM movies where year>2000 or year2010   ;
3)SELECT title, year FROM movies
  WHERE year < 2000 OR year > 2010;
4)SELECT title, year FROM movies
    WHERE year <= 2003;
```

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


sqlbolt.com/lesson/select_queries_with_constraints_pt_2
Select query with constraints

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

Table: Movies

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

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 * FROM movies
```

1)SELECT title, director FROM movies

WHERE title LIKE "Toy Story%";

2)SELECT title, director FROM movies

WHERE director LIkE "john Lasseter%";

3)SELECT title, director FROM movies

WHERE director != "john Lasseter";

4)SELECT * FROM movies

WHERE title LIKE "WALL-%";

SQL Lesson 4: Filtering and sorting Query results

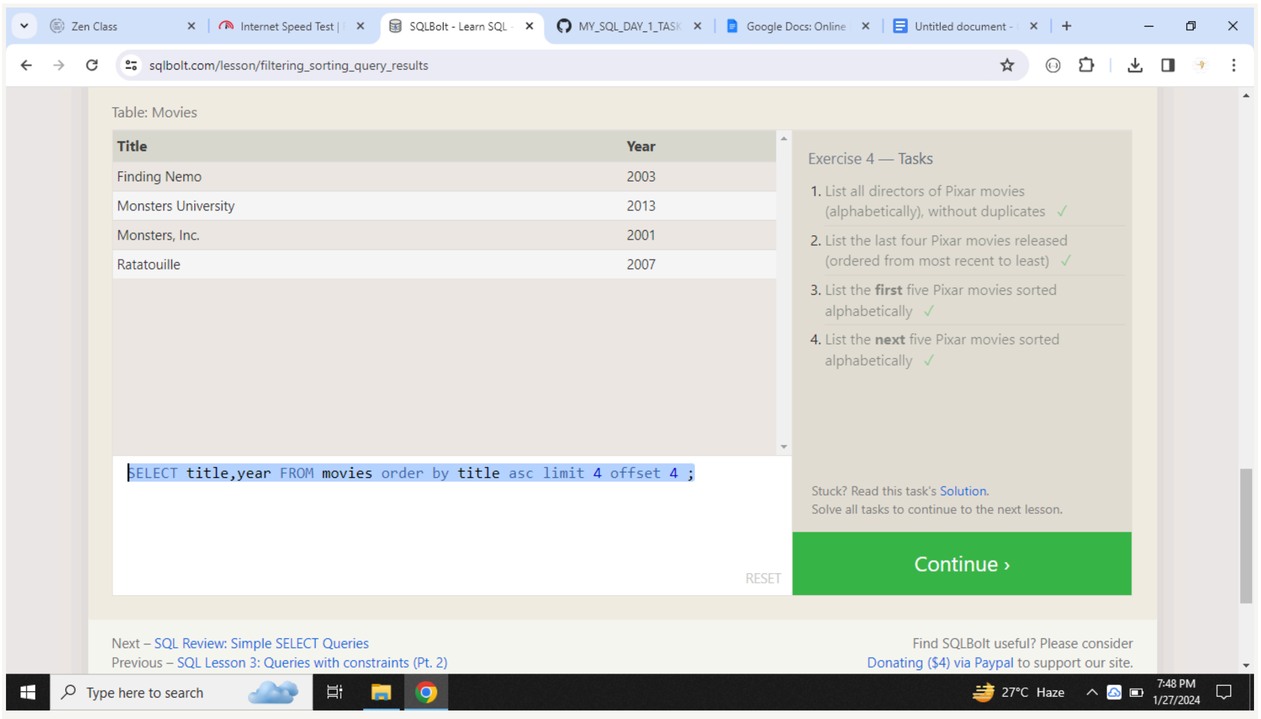


Table: Movies

Title	Year
Finding Nemo	2003
Monsters University	2013
Monsters, Inc.	2001
Ratatouille	2007

```
SELECT title,year FROM movies order by title asc limit 4 offset 4 ;
```

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 asc;
2)SELECT title FROM movies ORDER BY title asc LIMIT5;
3)SELECT title,year FROM movies ORDER BY year desc LIMIT5;
4)SELECT title,year FROM movies order by title asc limit 4 offset 4 ;

SQL Review: Simple SELECT Queries

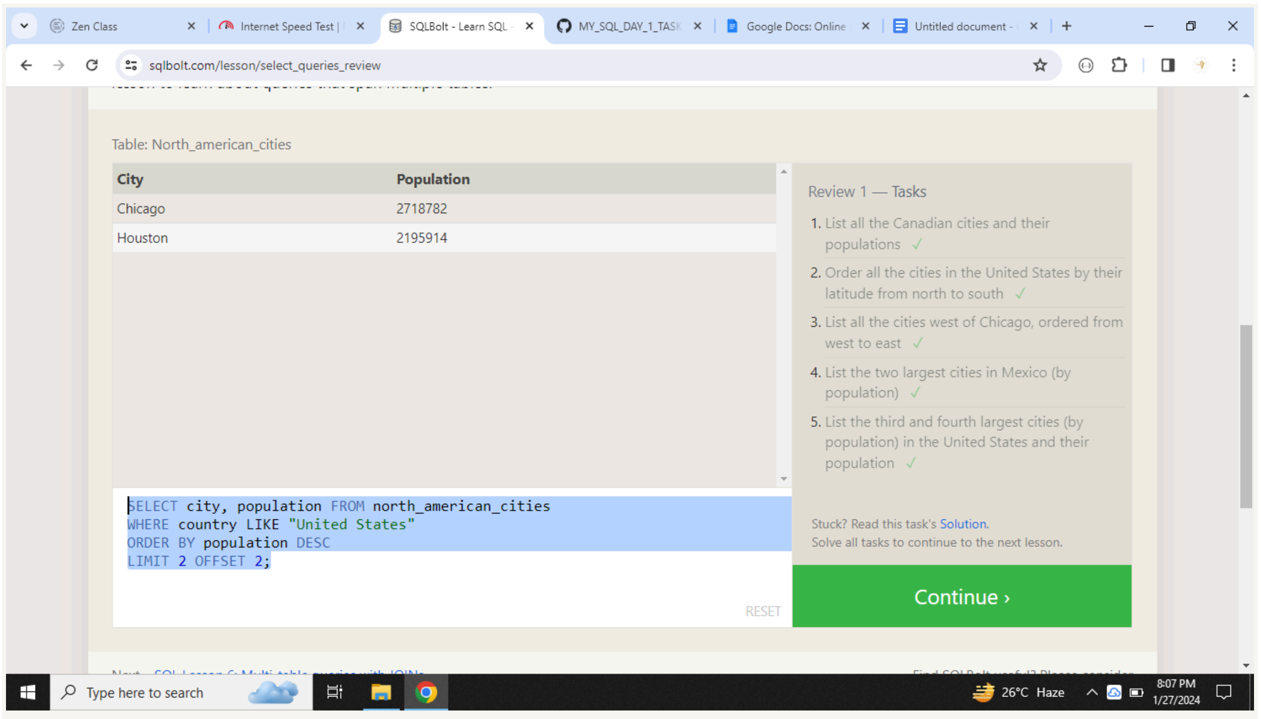


Table: North_american_cities

City	Population
Chicago	2718782
Houston	2195914

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

Review 1 — Tasks

1. List all the Canadian cities and their populations ✓
2. Order all the cities in the United States by their latitude from north to south ✓
3. List all the cities west of Chicago, ordered from west to east ✓
4. List the two largest cities in Mexico (by population) ✓
5. List the third and fourth largest cities (by population) in the United States and their population ✓

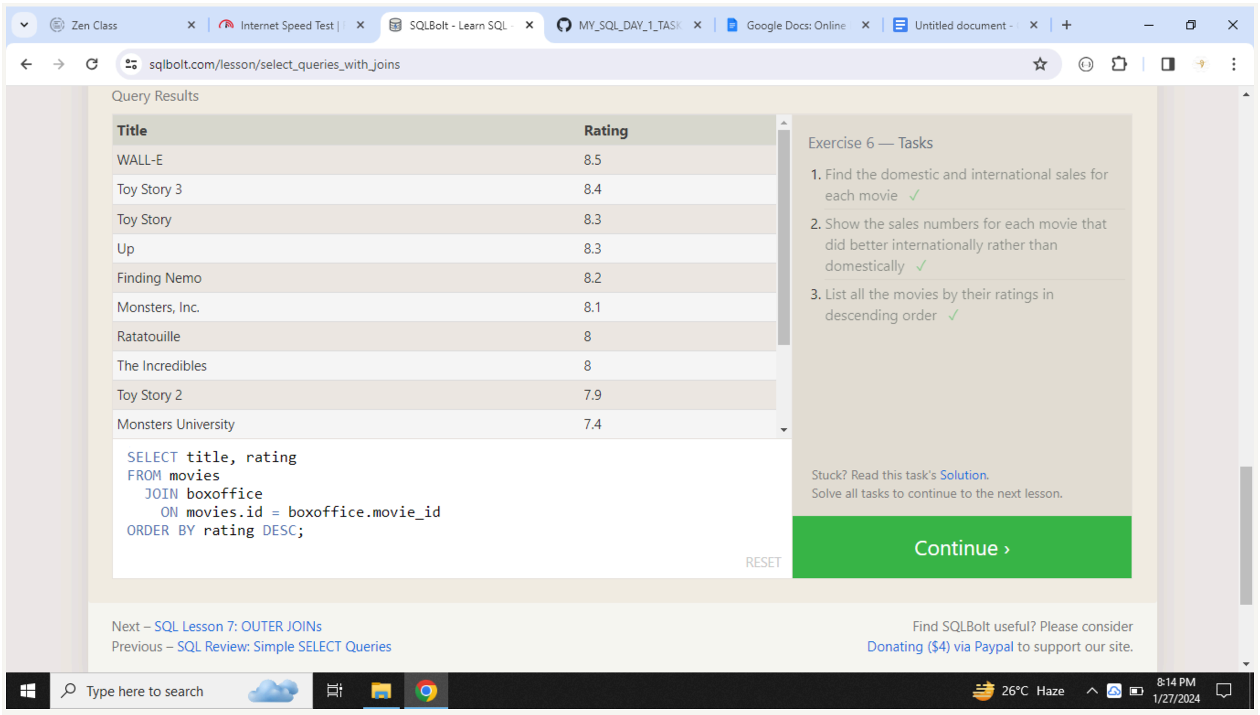
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

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

ORDER BY population DESC
LIMIT 2 OFFSET 2;

SQL Lesson 6: Multi-table queries with JOINs

`



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
  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.

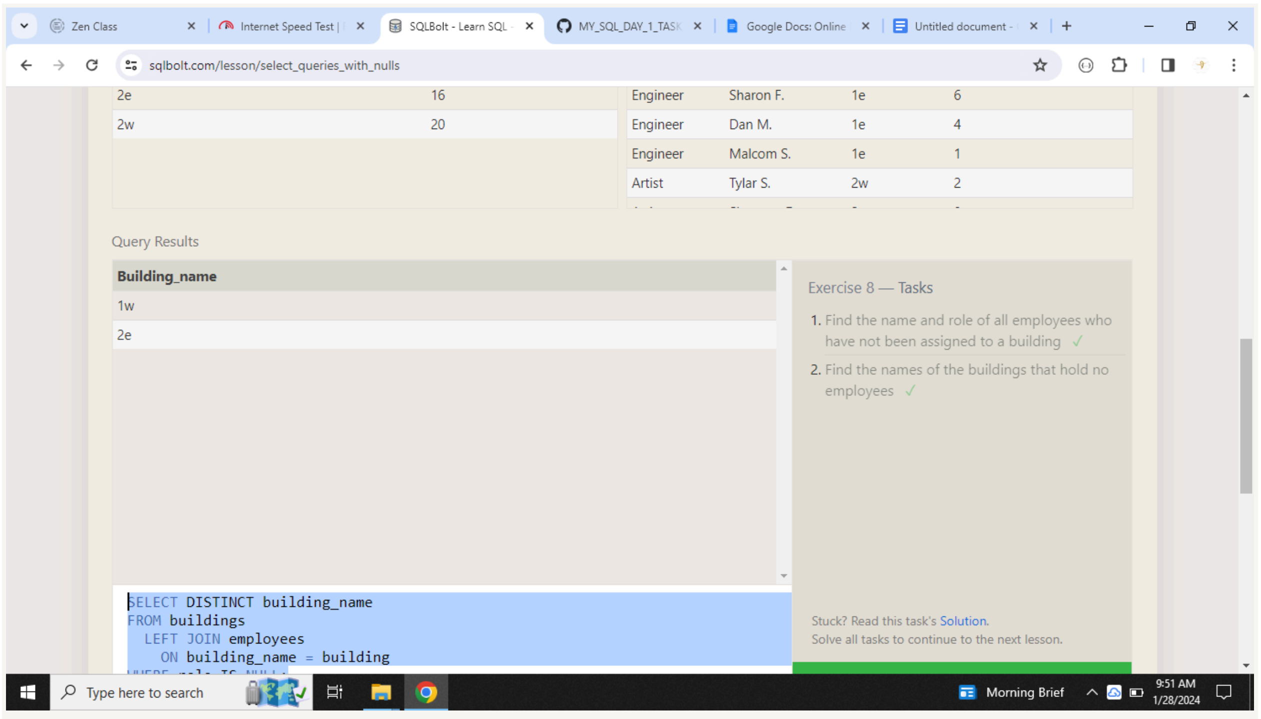
1)SELECT title, domestic_sales, international_sales
FROM movies
 JOIN boxoffice
ON movies.id = boxoffice.movie_id;
2)SELECT title, domestic_sales, international_sales
   FROM movies
   JOIN boxoffice
    ON movies.id = boxoffice.movie_id
    WHERE international_sales > domestic_sales;
3)SELECT title, rating

```
FROM movies
JOIN boxoffice
ON movies.id = boxoffice.movie_id
 ORDER BY rating DESC;
```

SQL Lesson 7: OUTER JOINs

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

SQL Lesson 8: A short note on 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 ✓

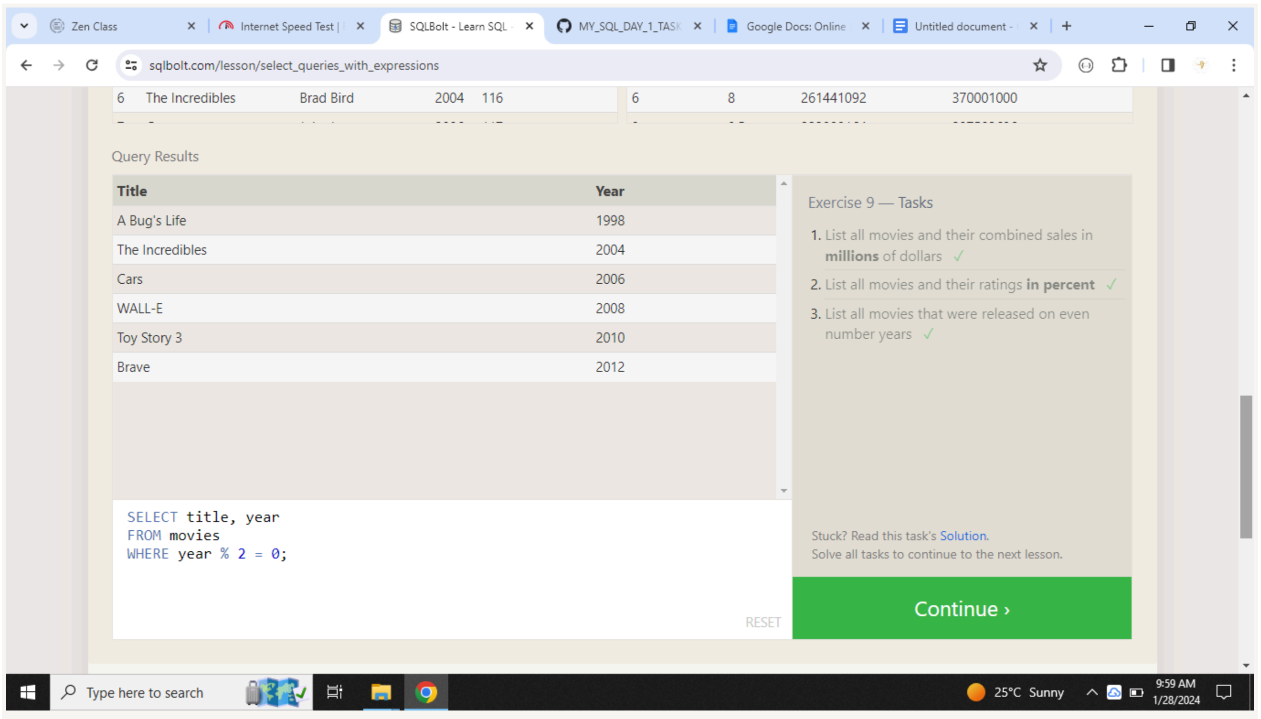
Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

```
SELECT DISTINCT building_name
FROM buildings
  LEFT JOIN employees
    ON building_name = building
```

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

SQL Lesson 9: Queries with expressions

```
 1)SELECT title, (domestic_sales + international_sales)
/ 1000000 AS gross_sales_millions
        FROM movies
        JOIN boxoffice
        ON movies.id = boxoffice.movie_id;
```



6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000

Query Results

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

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

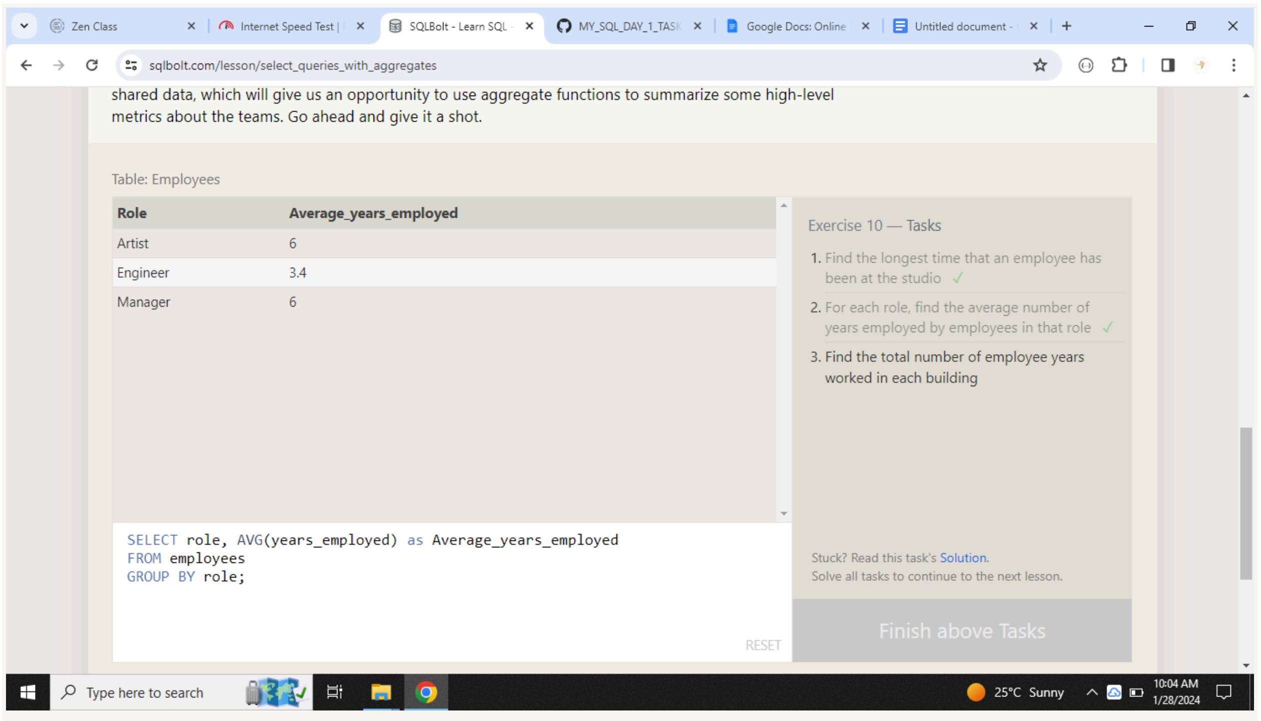
Exercise 9 — Tasks

1. List all movies and their combined sales in **millions** of dollars ✓
2. List all movies and their ratings **in percent** ✓
3. List all movies that were released on even number years ✓

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

```
2)SELECT title, rating *  10 as rating_percent
        FROM movies
         JOIN boxoffice
           ON movies.id = boxoffice.movie_id;
3)SELECT title, year
      FROM movies
      WHERE year % 2 = 0;
```

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



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

Role	Average_years_employed
Artist	6
Engineer	3.4
Manager	6

```
SELECT role, AVG(years_employed) as Average_years_employed
FROM employees
GROUP BY role;
```

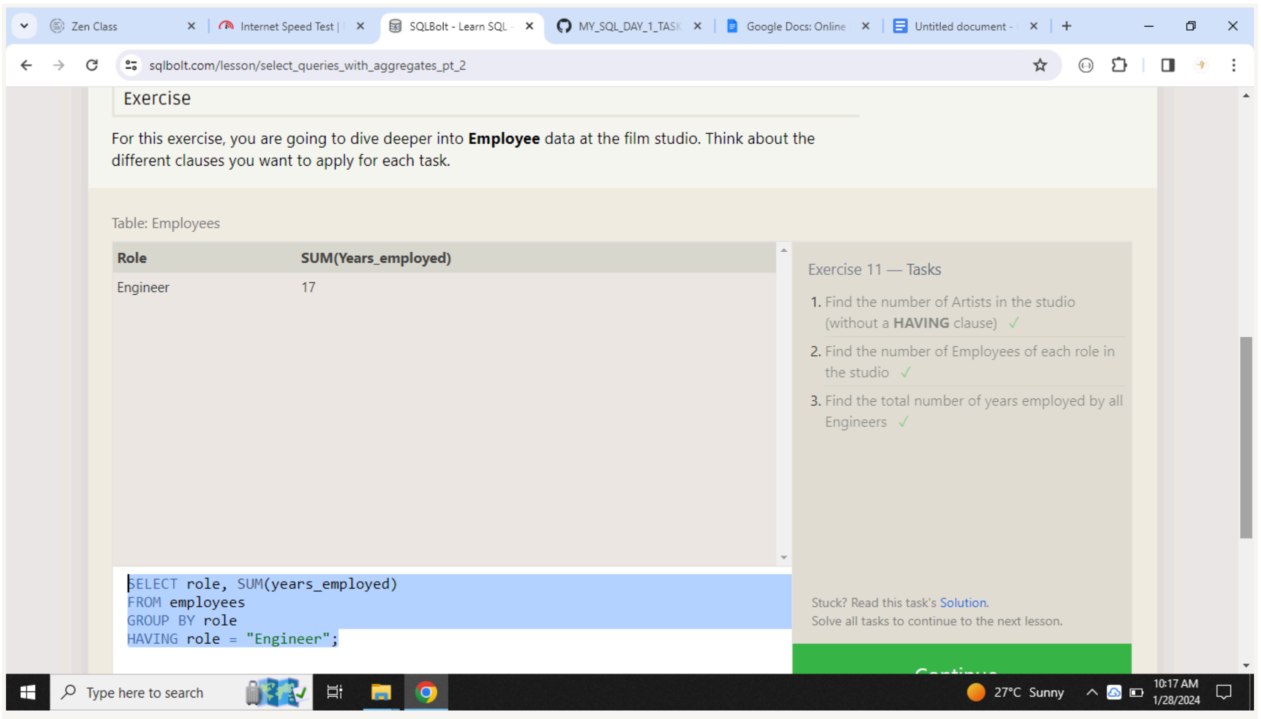
Exercise 10 — Tasks

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.

1)SELECT max(Years_employed) FROM employees;
2)SELECT role, AVG(years_employed) as
Average_years_employed
 FROM employees
GROUP BY role;

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



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	SUM(Years_employed)
Engineer	17

```
SELECT role, SUM(years_employed)
FROM employees
GROUP BY role
HAVING role = "Engineer";
```

Exercise 11 — Tasks

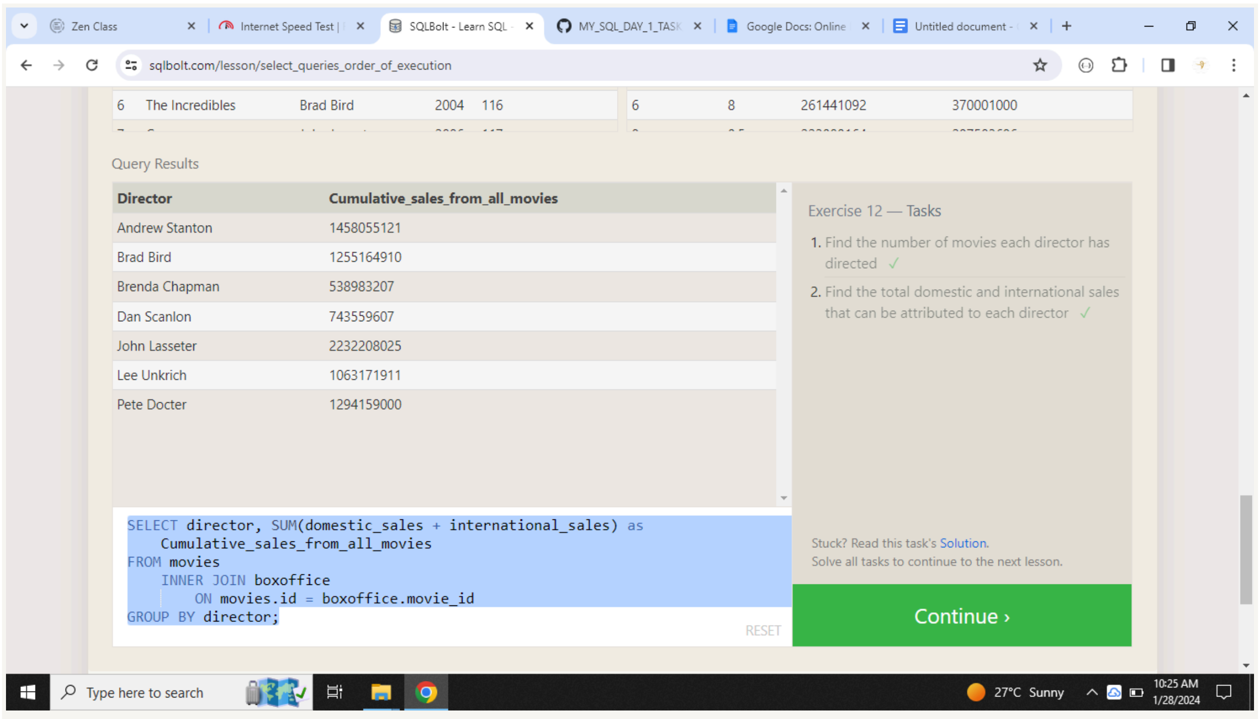
1. Find the number of Artists in the studio (without a **HAVING** clause) ✓
2. Find the number of Employees of each role in the studio ✓
3. Find the total number of years employed by all Engineers ✓

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

```
1)SELECT role, COUNT(*) as Number_of_artists
FROM employees
WHERE role = "Artist";
2)SELECT role, COUNT(*) as Number_of_artists
FROM employees
group by role;
3)SELECT role, SUM(years_employed)
FROM employees
GROUP BY role
HAVING role = "Engineer";
```

SQL Lesson 12: Order of execution of a Query

```
1)SELECT director,count(director) as couut
FROM movies GROUP BY director;
```



6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000

Query Results

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

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

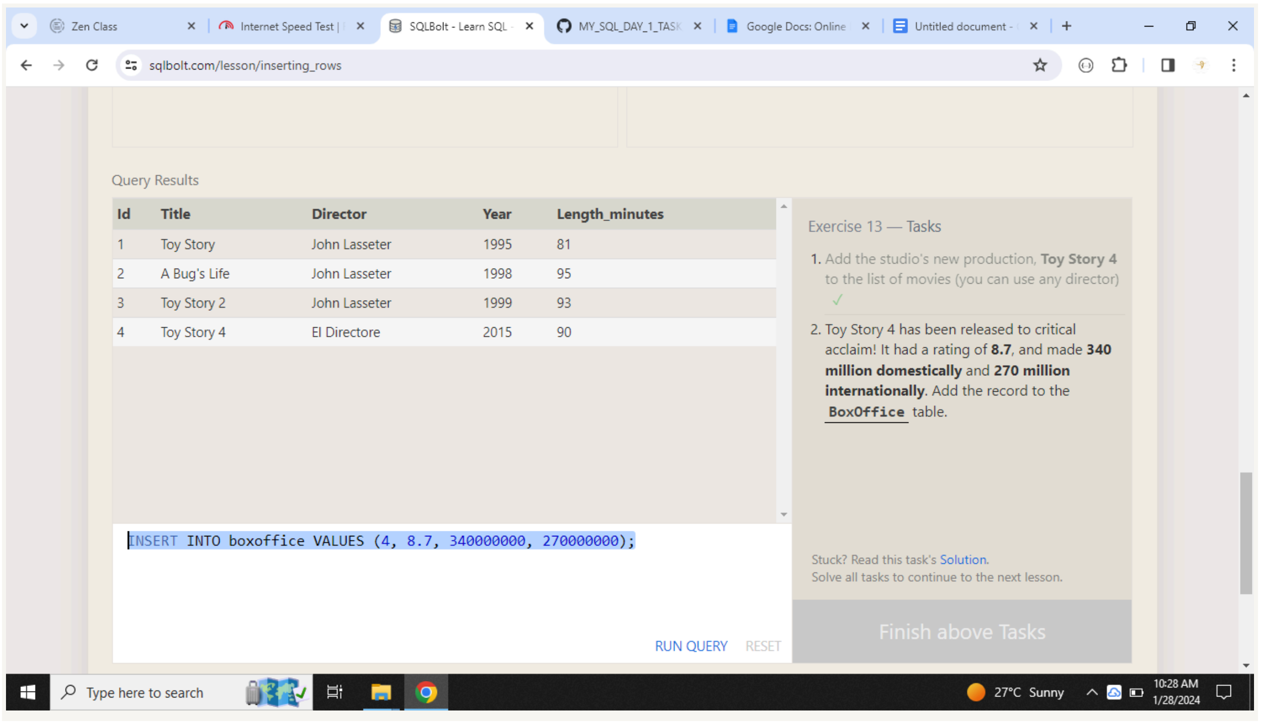
Exercise 12 — Tasks

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.

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

SQL Lesson 13: Inserting rows



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	El Directore	2015	90

```
INSERT INTO boxoffice VALUES (4, 8.7, 340000000, 270000000);
```

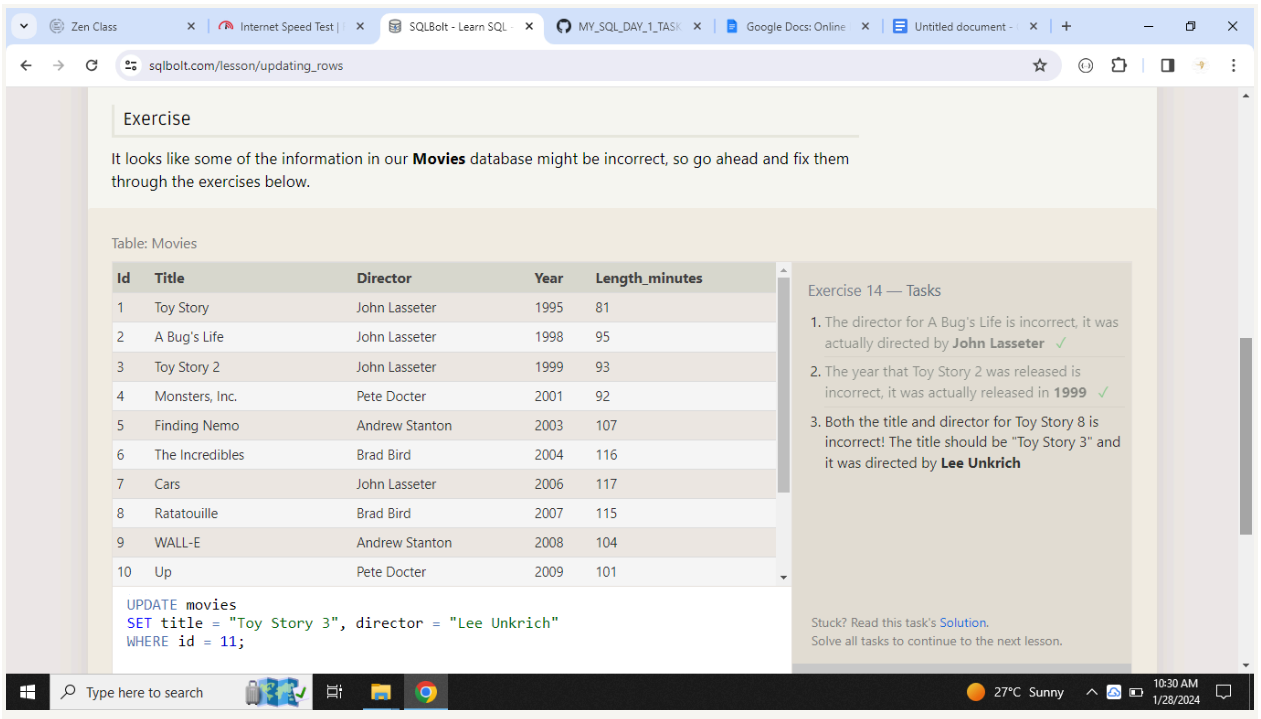
Exercise 13 — Tasks

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 VALUES (4, "Toy Story 4", "El Directore", 2015, 90);
2)INSERT INTO movies VALUES (4, "Toy Story 4", "El Directore", 2015, 90);
3)INSERT INTO boxoffice VALUES (4, 8.7, 340000000, 270000000);

SQL Lesson 14: 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
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

```
UPDATE movies
SET title = "Toy Story 3", director = "Lee Unkrich"
WHERE id = 11;
```

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.

```
1)UPDATE movies
SET director = "John Lasseter"
WHERE id = 2;
2)UPDATE movies
SET year = 1999
WHERE id = 3;
3)UPDATE movies
SET title = "Toy Story 3", director = "Lee Unkrich"
WHERE id = 11;
```