

# 1.SELECT queries 101

First exercise will only involve the **movies** table, and the default query below currently shows all the properties of each movie. To continue onto the next lesson, alter the query to find the exact information we need for each task.

Table: Movies

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

```
SELECT * FROM movies;
```

Exercise 1 — Tasks

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

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Continue >

## 2.Queries with constraints

query easier to read.

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, 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.

Continue >

### 3.Queries with constraints (Pt. 2)

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.

Continue >

Next — SQL Lesson 4: Filtering and sorting Query results

### 4.Filtering and sorting Query results

There are a few concepts in this lesson, but all are pretty straight-forward to apply. To spice things up, we've gone and scrambled the **Movies** table for you in the exercise to better mimic what kind of data you might see in real life. Try and use the necessary keywords and clauses introduced above in your queries.

Table: Movies

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

```
SELECT title FROM movies
ORDER BY title ASC
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.

Continue >

## 5.Simple SELECT Queries

lesson to learn about queries that span multiple tables.

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.

Continue >

Next — SQL Lesson 6: Multi-table queries with JOINS  
Find SQLBolt useful? Please consider Donating (\$4) via Paypal to support our site.

## 6.Multi-table queries with JOINS

Query Results

Title	Rating
WALL-E	8.5
Toy Story 3	8.4
Toy Story	8.3
Up	8.3
Finding Nemo	8.2
Monsters, Inc.	8.1
Ratatouille	8
The Incredibles	8
Toy Story 2	7.9
Monsters University	7.4

```
SELECT title, rating
FROM movies
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.

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. OUTER JOINS

The screenshot shows the SQLBolt lesson interface for 'Outer Joins'. At the top, there's a browser tab for 'sqlbolt.com/lesson/select\_queries\_with\_outer\_joins'. Below the browser, a small table shows employee data:

Role	Employee	Building	Count
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2

The 'Query Results' section displays the following data:

Building_name	Role
1e	Engineer
1e	Manager
1w	
2e	
2w	Artist
2w	Manager

The query editor contains the following SQL code:

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

The 'Exercise 7 — Tasks' section lists three 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) ✓

Below the tasks, there's a green 'Continue >' button and a 'RESET' button.

## 8. A short note on NULLs

The screenshot shows the SQLBolt lesson interface for 'A short note on NULLs'. The browser tab is 'sqlbolt.com/lesson/select\_queries\_with\_nulls'. The small table at the top is the same as in the previous screenshot.

The 'Query Results' section displays the following data:

Building_name
1w
2e

The query editor contains the following SQL code:

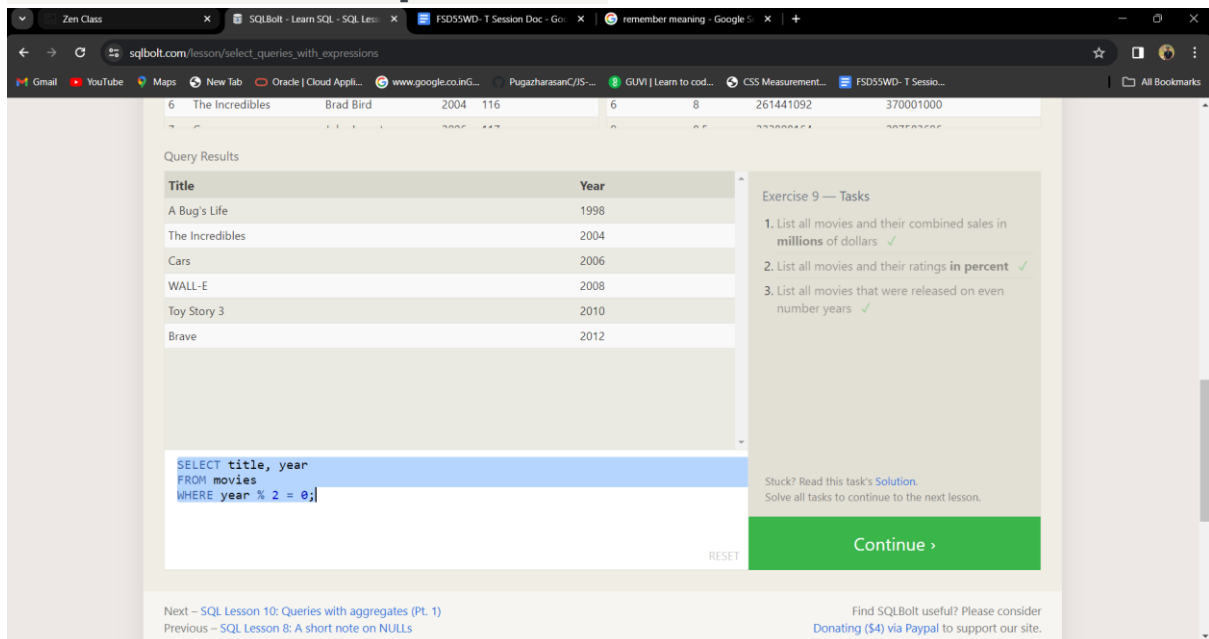
```
SELECT DISTINCT building_name
FROM buildings
LEFT JOIN employees
ON building_name = building
WHERE role IS NULL;
```

The 'Exercise 8 — Tasks' section lists two 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 ✓

Below the tasks, there's a green 'Continue >' button and a 'RESET' button.

## 9. Queries with expressions



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 ✓

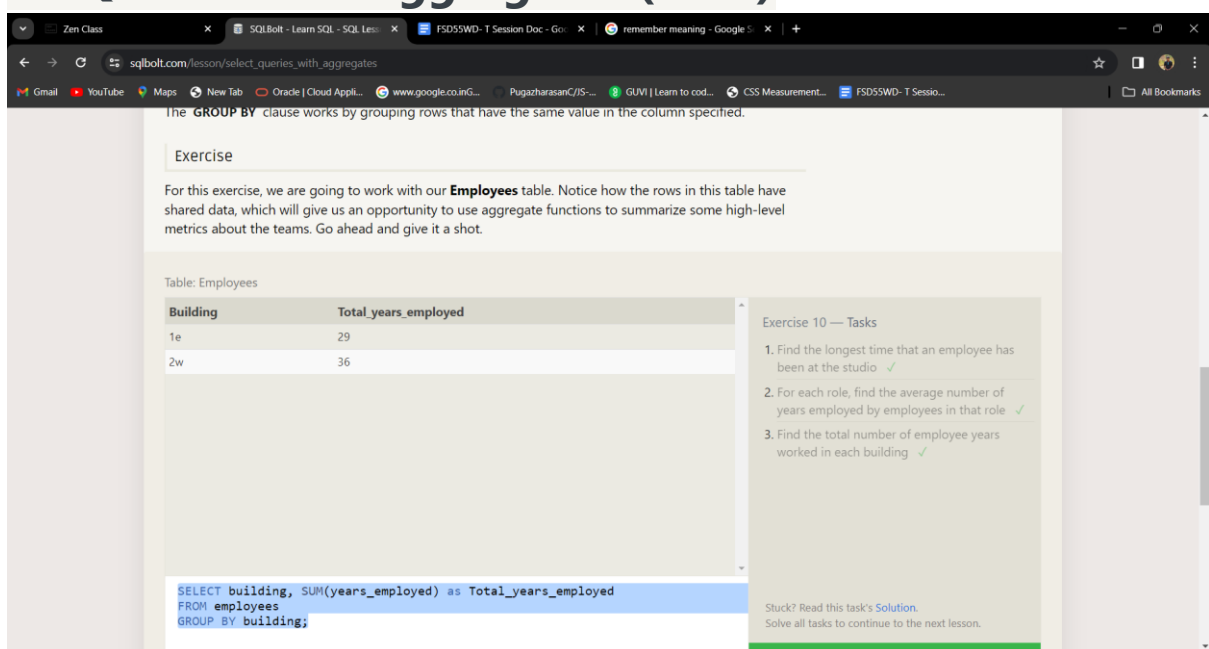
Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

[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. Queries with aggregates (Pt. 1)



The **GROUP BY** clause works by grouping rows that have the same value in the column specified.

Exercise

For this exercise, we are going to work with our **Employees** table. Notice how the rows in this table have 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_employed
1e	29
2w	36

```
SELECT building, SUM(years_employed) as Total_years_employed
FROM employees
GROUP BY building;
```

Exercise 10 — Tasks

1. Find the longest time that an employee has been at the studio ✓
2. For each role, find the average number of years employed by employees in that role ✓
3. Find the total number of employee years worked in each building ✓

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

## 11.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	SUM(Years_employed)
Engineer	17

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 ✓

SELECT role, SUM(years\_employed)  
FROM employees  
GROUP BY role  
HAVING role = "Engineer";

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

Continue >

## 12.Order of execution of a Query

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

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 ✓

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;

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. Inserting rows

SQLBolt - Learn SQL - SQL Les...

FSD55WD- T Session Doc - Go...

remember meaning - Google S...

sqlbolt.com/lesson/inserting\_rows

Gmail YouTube Maps New Tab Oracle | Cloud Appli... www.google.co.inG... PugazharasanC/JS... GUMI | Learn to cod... CSS Measurement... FSD55WD- T Sessio...

All Bookmarks

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

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 [Donating \(\\$4\) via Paypal](#) to support our site.

## 14. Updating rows

SQLBolt - Learn SQL - SQL Les...

FSD55WD- T Session Doc - Go...

remember meaning - Google S...

sqlbolt.com/lesson/inserting\_rows

Gmail YouTube Maps New Tab Oracle | Cloud Appli... www.google.co.inG... PugazharasanC/JS... GUMI | Learn to cod... CSS Measurement... FSD55WD- T Sessio...

All Bookmarks

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

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 >

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.

## 15.Deleting rows

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

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: Inserting rows](#)

Find SQLBolt useful? Please consider [Donating \(€4\)](#) to support our site.

## 16.Creating tables

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 downloadedThis 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. Altering tables

Table: Movies

Id	Title	Director	Year	Length_minutes	Aspect_ratio	Language
1	Toy Story	John Lasseter	1995	81	2.39	English
2	A Bug's Life	John Lasseter	1998	95	2.39	English
3	Toy Story 2	John Lasseter	1999	93	2.39	English
4	Monsters, Inc.	Pete Docter	2001	92	2.39	English
5	Finding Nemo	Andrew Stanton	2003	107	2.39	English
6	The Incredibles	Brad Bird	2004	116	2.39	English
7	Cars	John Lasseter	2006	117	2.39	English
8	Ratatouille	Brad Bird	2007	115	2.39	English
9	WALL-E	Andrew Stanton	2008	104	2.39	English
10	Up	Pete Docter	2009	101	2.39	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.

# 18. Dropping tables

Query Results

Id	Title	Director	Year	Length_minutes
----	-------	----------	------	----------------

[RUN QUERY](#) [RESET](#)

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.

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.

