

Table: Movies

Monsters, Inc.

Finding Nemo

The Incredibles

Cars

Ratatouille

WALL-E

Up

Toy Story 3

Cars 2

Brave

Monsters University

```
SELECT title| FROM movies;
```

#### Exercise 1 — Tasks

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

Stuck? Read this task's [Solution](#).

Solve all tasks to continue to the next lesson.

RESET

Finish above Tasks

Table: Movies

Pete Docter
Andrew Stanton
Brad Bird
John Lasseter
Brad Bird
Andrew Stanton
Pete Docter
Lee Unkrich
John Lasseter
Brenda Chapman
Dan Scanlon

```
SELECT director 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.

RESET

Finish above Tasks

Table: Movies

Monsters, Inc.	Pete Docter
Finding Nemo	Andrew Stanton
The Incredibles	Brad Bird
Cars	John Lasseter
Ratatouille	Brad Bird
WALL-E	Andrew Stanton
Up	Pete Docter
Toy Story 3	Lee Unkrich
Cars 2	John Lasseter
Brave	Brenda Chapman
Monsters University	Dan Scanlon

`SELECT title, director 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.

RESET

Finish above Tasks

Table: Movies

Monsters, Inc.	2001
Finding Nemo	2003
The Incredibles	2004
Cars	2006
Ratatouille	2007
WALL-E	2008
Up	2009
Toy Story 3	2010
Cars 2	2011
Brave	2012
Monsters University	2013

```
SELECT title, year| 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.

RESET

Finish above Tasks

Table: Movies

<b>Id</b>	<b>Title</b>	<b>Director</b>	<b>Year</b>	<b>Length_minutes</b>
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.

RESET

Continue ›

Table: Movies

<b>Id</b>	<b>Title</b>	<b>Director</b>	<b>Year</b>	<b>Length_minutes</b>
6	The Incredibles	Brad Bird	2004	116

```
SELECT * FROM movies WHERE id=6;
```

#### 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.

RESET

Finish above Tasks

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

Previous – SQL Lesson 1: SELECT queries 101

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

Table: Movies

Title	Year
Monsters, Inc.	2001
Finding Nemo	2003
The Incredibles	2004
Cars	2006
Ratatouille	2007
WALL-E	2008
Up	2009
Toy Story 3	2010

```
SELECT title,year FROM movies WHERE year between 2000 and 2010
```

### 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.

RESET

Finish above Tasks

Next – [SQL Lesson 3: Queries with constraints \(Pt. 2\)](#)

Previous – [SQL Lesson 1: SELECT queries 101](#)

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

Table: Movies

Title	Year
Toy Story	1995
A Bug's Life	1998
Toy Story 2	1999
Cars 2	2011
Brave	2012
Monsters University	2013

```
SELECT title,year FROM movies WHERE year not between 2000 and 2010
```

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.

Finish above Tasks

[Next – SQL Lesson 3: Queries with constraints \(Pt. 2\)](#)

[Previous – SQL Lesson 1: SELECT queries 101](#)

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

Table: Movies

Title
Toy Story
A Bug's Life
Toy Story 2
Monsters, Inc.
Finding Nemo

```
SELECT title FROM movies WHERE id in |(1,2,3,4,5)
```

### Exercise 2 — Tasks

1. Find the movie with a row `id` of 6 ✓
2. Find the movies released in the `year`s between 2000 and 2010 ✓
3. Find the movies **not** released in the `year`s between 2000 and 2010 ✓
4. Find the first 5 Pixar movies and their release `year` ✓

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

RESET

Continue >

[Next – SQL Lesson 3: Queries with constraints \(Pt. 2\)](#)

[Previous – SQL Lesson 1: SELECT queries 101](#)

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

Table: Movies

Title	Director
Toy Story	John Lasseter
Toy Story 2	John Lasseter
Toy Story 3	Lee Unkrich

```
SELECT title,director FROM movies where title like "toy story%";
```

### 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.

RESET

Finish above Tasks

Table: Movies

Director	Title
John Lasseter	Toy Story
John Lasseter	A Bug's Life
John Lasseter	Toy Story 2
John Lasseter	Cars
John Lasseter	Cars 2

```
SELECT director ,title FROM movies where director like "john%";
```

### 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.

RESET

Finish above Tasks

[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.

Table: Movies

Director	Title
Pete Docter	Monsters, Inc.
Andrew Stanton	Finding Nemo
Brad Bird	The Incredibles
Brad Bird	Ratatouille
Andrew Stanton	WALL-E
Pete Docter	Up
Lee Unkrich	Toy Story 3
Brenda Chapman	Brave
Dan Scanlon	Monsters University
Brenda Chapman	WALL-E

```
SELECT director ,title FROM movies where director not like "john%";
```

### 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.

RESET

Finish above Tasks

**Title**

WALL-E

WALL-G

```
SELECT title 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.

RESET

**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.

Table: Movies

**Director**

Andrew Stanton

Brad Bird

Brenda Chapman

Dan Scanlon

John Lasseter

Lee Unkrich

Pete Docter

```
SELECT distinct director FROM movies order by director;
```

**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.

RESET

Finish above Tasks

[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.

Table: Movies

Title
Monsters University
Brave
Cars 2
Toy Story 3

```
SELECT DISTINCT title| FROM movies ORDER BY year DESC LIMIT 4;
```

#### Exercise 4 — Tasks

1. List all directors of Pixar (alphabetically), without duplicates
2. List the last four Pixar movies (ordered from most recent)
3. List the **first** five Pixar movies (ordered alphabetically)
4. List the **next** five Pixar movies (ordered alphabetically)

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

Table: Movies

Title
Monsters University
Brave
Cars 2
Toy Story 3

```
SELECT DISTINCT title FROM movies ORDER BY year DESC LIMIT 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.

RESET

Finish above Tasks

[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

Table: Movies

Title
A Bug's Life
Brave
Cars
Cars 2
Finding Nemo

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

RESET

Finish above Tasks

[Next – SQL Review: Simple SELECT Queries](#)

[Previous – SQL Lesson 3: Queries with constraints \(Pt. 2\)](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$1\) via PayPal](#) to support our site

Table: Movies

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

```
SELECT title FROM movies ORDER BY title LIMIT 5 OFFSET 5;|
```

#### Exercise 4 — Tasks

1. List all directors of Pixar movies (alphabetically), without duplicates ✓
2. List the last four Pixar movies released (ordered from most recent to least) ✓
3. List the **first** five Pixar movies sorted alphabetically ✓
4. List the **next** five Pixar movies sorted alphabetically ✓

Stuck? Read this task's [Solution](#).

Solve all tasks to continue to the next lesson.

RESET

Continue >

Table: North\_american\_cities

City	Population
Toronto	2795060
Montreal	1717767

`SELECT city, population  
FROM north_american_cities  
WHERE country = "Canada";|`

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.

RESET

Finish above Tasks

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.

Table: North\_american\_cities

### City

Chicago

New York

Philadelphia

Los Angeles

Phoenix

Houston

```
SELECT city
FROM north_american_cities
WHERE country = "United States"
ORDER BY latitude DESC;
```

### 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.

RESET

Finish above Tasks

[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.

Table: North\_american\_cities

### City

Los Angeles

Phoenix

Guadalajara

Mexico City

Ecatepec de Morelos

Houston

```
SELECT city
FROM north_american_cities
WHERE longitude < -87.629798
ORDER BY longitude;
```

### 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.

RESET

Finish above Tasks

[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.

Table: North\_american\_cities

### City

Mexico City

Ecatepec de Morelos

```
SELECT city
FROM north_american_cities
WHERE country = "Mexico"
ORDER BY population DESC
LIMIT 2;
```

RESET

### 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.

[Finish above Tasks](#)

[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.

Table: North\_american\_cities

City
Chicago
Houston

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

RESET

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](#)

[Previous – SQL Lesson 4: Filtering and sorting Query results](#)

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

## Query Results

Title	Domestic_sales	International_sales
Finding Nemo	380843261	555900000
Monsters University	268492764	475066843
Ratatouille	206445654	417277164
Cars 2	191452396	368400000
Toy Story 2	245852179	239163000
The Incredibles	261441092	370001000
WALL-E	223808164	297503696
Toy Story 3	415004880	648167031
Toy Story	191796233	170162503
Cars	244082982	217900167

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

## Exercise 6 — Tasks

- Find the domestic and international sales for each movie ✓
- Show the sales numbers for each movie that did better internationally rather than domestically
- 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.

RESET

Finish above Tasks

[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.

## Query Results

Title	Domestic_sales	International_sales
Finding Nemo	380843261	555900000
Monsters University	268492764	475066843
Ratatouille	206445654	417277164
Cars 2	191452396	368400000
The Incredibles	261441092	370001000
WALL-E	223808164	297503696
Toy Story 3	415004880	648167031
Up	293004164	438338580
A Bug's Life	162798565	200600000
Brave	237283207	301700000

```
SELECT title, domestic_sales, international_sales
FROM movies
INNER JOIN boxoffice
ON movies.id = boxoffice.movie_id
WHERE international_sales > domestic_sales;
```

## 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.

RESET

Finish above Tasks

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.

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

RESET

Continue >

### 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.

Artist	Tylar S.	2w	2

## Query Results

### Building

1e

2w

```
SELECT DISTINCT building FROM employees;
```

RESET

### Exercise 7 — Tasks

1. Find the list of all buildings that have 1 employee ✓
2. Find the list of all buildings and their count
3. List all buildings and the distinct roles in each building (including employees) of each building

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

Finish above Task

[Next – SQL Lesson 8: A short note on NULLs](#)

[Previous – SQL Lesson 6: Multi-table queries with JOINs](#)

Find SQLBolt useful?

[Donating \(\\$4\) via PayPal](#) to support SQLBolt



Search



## Query Results

### Building

1e

2w

```
SELECT DISTINCT building FROM employees;
```

### 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.

RESET

Finish above Tasks

## Query Results

### Building\_name

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

```
SELECT * FROM buildings;
```

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.

[Finish above Tasks](#)

[Next – SQL Lesson 8: A short note on NULLs](#)

[Previous – SQL Lesson 6: Multi-table queries with JOINs](#)

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

## Query Results

### Building\_name

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

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

RESET

Continue >

### 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.

Next – [SQL Lesson 8: A short note on NULLs](#)

Previous – [SQL Lesson 6: Multi-table queries with JOINS](#)

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

## Query Results

Name	Role
Yancy I.	Engineer
Oliver P.	Artist

```
SELECT name, role FROM employees WHERE building IS NULL;
```

### Exercise 8 — Tasks

1. Find the name and role of all employees who have not been assigned to a building ✓
2. Find the names of the buildings that hold no employees

Stuck? Read this task's [Solution](#).

Solve all tasks to continue to the next lesson.

RESET

Finish above Tasks

Query results

**Building\_name**

1w

2e

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

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.

**Continue ›**

[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.

## Query Results

Title	Sales
Finding Nemo	936.743261
Monsters University	743.559607
Ratatouille	623.722818
Cars 2	559.852396
Toy Story 2	485.015179
The Incredibles	631.442092
WALL-E	521.31186
Toy Story 3	1063.171911
Toy Story	361.958736
Cars	461.983149

```
SELECT DISTINCT
    title,(domestic_sales + international_sales) / 1000000 AS sales
FROM movies|
INNER JOIN boxoffice
ON movies.id = boxoffice.movie_id;
```

### 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.

RESET

Finish above Tasks

## Query Results

Title	Rate_percent
Finding Nemo	82
Monsters University	74
Ratatouille	80
Cars 2	64
Toy Story 2	79
The Incredibles	80
WALL-E	85
Toy Story 3	84
Toy Story	83
Cars	72

`SELECT DISTINCT title,  
(rating * 10) AS rate_percent  
FROM movies  
INNER JOIN boxoffice  
ON movies.id = boxoffice.movie_id;`

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.

**Finish above Tasks**

RESET

## Query Results

### Title

A Bug's Life

The Incredibles

Cars

WALL-E

Toy Story 3

Brave

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

### Exercise 9 — Tasks

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

Stuck? Read this task's [Solution](#).

Solve all tasks to continue to the next lesson.

RESET

Continue ›

Table: Employees

MAX(Years\_employed)

9

```
SELECT MAX(Years_employed)  
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.

RESET

Finish above Tasks

Table: Employees

Role	AVG(Years_Employed)
Artist	6
Engineer	3.4
Manager	6

```
SELECT Role, AVG(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.

RESET

Finish above Tasks

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.

Table: Employees

Building	SUM(Years_Employed)
1e	29
2w	36

```
SELECT Building, SUM(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.

RESET

Continue >

Table: Employees

Role	Number_of_Artists
Artist	5

```
SELECT Role, COUNT(*) AS Number_of_Artists
FROM Employees
WHERE Role = "Artist";
```

### 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.

RESET

Finish above Tasks

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.

Table: Employees

Role	COUNT(*)
Artist	5
Engineer	5
Manager	3

`SELECT Role, COUNT(*)  
FROM Employees  
GROUP BY Role;`

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.

Finish above Tasks

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.

EMPLOYEES

Role	SUM(Years_Employed)
Engineer	17

```
SELECT Role, SUM(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.

## Query Results

<b>Id</b>	<b>Title</b>	<b>Director</b>	<b>Year</b>	<b>Length_minutes</b>	<b>COUNT(Title)</b>
9	WALL-E	Andrew Stanton	2008	104	2
8	Ratatouille	Brad Bird	2007	115	2
13	Brave	Brenda Chapman	2012	102	1
14	Monsters University	Dan Scanlon	2013	110	1
12	Cars 2	John Lasseter	2011	120	5
11	Toy Story 3	Lee Unkrich	2010	103	1
10	Up	Pete Docter	2009	101	2

```
SELECT *, COUNT(Title)
FROM Movies
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.

RESET

Finish above Tasks

[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.

## 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, sum(Domestic_sales + International_Sales) as Total_Sales
FROM Movies
LEFT JOIN Boxoffice ON Id = 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.

RESET

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.

## Query Results

<b>Id</b>	<b>Title</b>	<b>Director</b>	<b>Year</b>	<b>Length_minutes</b>
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	EI DIRECTORE	2015	90

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

RUN QUERY    RESET

Finish above Tasks

### 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.

## 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

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

RUN QUERY    RESET

Continue >

### 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.

Next – SQL Lesson 14: Updating rows

Previous – SQL Lesson 12: Order of execution of a Query

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

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 8	El Directore	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

```
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](#)

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 8	El Directore	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

```
UPDATE Movies
SET Year = "1999"
WHERE Id = 3;
```

[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](#)

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 8	El Directore	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

```
UPDATE Movies
SET Title = "Toy Story 3", Director = "Lee Unkrich"
WHERE Id = 11;
```

[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](#)

## Exercise

The database needs to be cleaned up a little bit, so try and delete a few rows in the tasks below.

Table: Movies

<b>Id</b>	<b>Title</b>	<b>Director</b>	<b>Year</b>	<b>Length_minutes</b>
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

```
DELETE FROM Movies  
WHERE Year < 2005;
```

[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.

[Finish above Tasks](#)

## EXERCISE

The database needs to be cleaned up a little bit, so try and delete a few rows in the tasks below.

Table: Movies

<b>Id</b>	<b>Title</b>	<b>Director</b>	<b>Year</b>	<b>Length_minutes</b>
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

```
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 ›**

Table: Database

Name	Version	Download_Count
SQLite	3.9	92000000
MySQL	5.5	512000000
Postgres	9.4	384000000

`CREATE TABLE Database ( Name TEXT, 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.

Table: Movies

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

```
ALTER TABLE Movies
ADD COLUMN Aspect_ratio FLOAT DEFAULT 3;
```

[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.

Table: Movies

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

```
ALTER TABLE Movies
ADD COLUMN 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 >](#)

## Query Results

<b>Id</b>	<b>Title</b>	<b>Director</b>	<b>Year</b>	<b>Length_minutes</b>
1	The Empire Strikes Back	Irvin Kershner	1980	128
2	Return of the Jedi	Richard Marquand	1983	134
3	Star Wars: Episode V - The Empire Strikes Back	Irvin Kershner	1980	128
4	Star Wars: Episode VI - Return of the Jedi	Richard Marquand	1983	134

```
DROP TABLE Movies;|
```

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

[Finish above Tasks](#)

### 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.

## Query Results

ID	Title	Director	Year	Length_minutes
1	The Shawshank Redemption	Frank Darabont	1994	142
2	The Godfather	Francis Ford Coppola	1972	175
3	The Godfather: Part II	Francis Ford Coppola	1974	163
4	The Dark Knight	Christopher Nolan	2008	152
5	Inception	Christopher Nolan	2010	148
6	The Godfather: Part III	Francis Ford Coppola	1990	165
7	The Lord of the Rings: The Return of the King	Peter Jackson	2003	162
8	The Pianist	Janusz Kamiński	2002	135
9	Schindler's List	Spielberg	1993	190
10	The Godfather: Part I	Francis Ford Coppola	1972	167

```
DROP TABLE BoxOffice;
```

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 ›