

MYSQL DAY 1 TASK

TASK 1:

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;
```

RESET

Exercise 1 — Tasks

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

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

Continue >

Next – [SQL Lesson 2: Queries with constraints \(Pt. 1\)](#)
Previous – [Introduction to SQL](#)

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

TASK 2:

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;
```

RESET

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 >

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.

TASK 3:

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-%";
```

RESET

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](#)
Previous – [SQL Lesson 2: Queries with constraints \(Pt. 1\)](#)

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

MYSQL DAY 1 TASK

TASK 4:

Table: Movies

Id	Title	Director	Year	Length_minutes
13	Monsters University	Dan Scanlon	2013	110
10	Monsters, Inc.	Pete Docter	2001	92
1	Ratatouille	Brad Bird	2007	115
2	The Incredibles	Brad Bird	2004	116
4	Toy Story	John Lasseter	1995	81

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 >

```
SELECT * FROM movies
ORDER BY Title ASC
LIMIT 5
OFFSET 5;
```

RESET

TASK 5:

Table: North_american_cities

City
Chicago
Houston

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 >

```
SELECT city FROM north_american_cities
where country = "United States"
ORDER BY population DESC
limit 2
offset 2;
```

RESET

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.

TASK 6:

2	A Bug's Life	John Lasseter	1998	95	7	7.2	244082982	217900167
3	Toy Story 2	John Lasseter	1999	93	10	8.3	293004164	438338580
4	Monsters, Inc.	Pete Docter	2001	92	4	8.1	289916256	272900000
5	Finding Nemo	Andrew Stanton	2003	107	2	7.2	162798565	200600000
6	The Incredibles	Brad Bird	2004	116	13	7.2	237283207	301700000

Query Results

Id	Title	Director	Year	Length_minutes	Movie_id	Rating	Domestic_sales	International_sales
9	WALL-E	Andrew Stanton	2008	104	9	8.5	223808164	297503696
11	Toy Story 3	Lee Unkrich	2010	103	11	8.4	415004880	648167031
1	Toy Story	John Lasseter	1995	81	1	8.3	191796233	170162503
10	Up	Pete Docter	2009	101	10	8.3	293004164	438338580
5	Finding Nemo	Andrew Stanton	2003	107	5	8.2	380843261	555900000
4	Monsters, Inc.	Pete Docter	2001	92	4	8.1	289916256	272900000

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 >

```
SELECT *
FROM movies
JOIN boxoffice
ON movies.id = boxoffice.movie_id
order by Rating desc ;
```

RESET

MYSQL DAY 1 TASK

TASK 7:

Query Results

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 = building;
```

RESET

Continue >

Exercise 7 — Tasks

- Find the list of all buildings that have employees ✓
- Find the list of all buildings and their capacity ✓
- 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.

TASK 8:

2w	2w	Manager	Daria O.	2w	6
		Engineer	Yancy I.		0
		Artist	Oliver P.		0

Query Results

Building_name
1w
2e

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

RESET

Continue >

Exercise 8 — Tasks

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

TASK 9:

3	Toy Story 2	John Lasseter	1999	93	8	8	206445654	417277164
4	Monsters, Inc.	Pete Docter	2001	92	12	6.4	191452396	368400000
5	Finding Nemo	Andrew Stanton	2003	107	3	7.9	245852179	239163000
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;
```

RESET

Continue >

Exercise 9 — Tasks

- List all movies and their combined sales in **millions of dollars** ✓
- List all movies and their ratings **in percent** ✓
- 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.

MYSQL DAY 1 TASK

TASK 10:

Table: Employees

Building	Total_years_employed
1e	29
2w	36

Exercise 10 — Tasks

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

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

Continue ›

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

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.

TASK 11:

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

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 ›

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

TASK 12:

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 ✓

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

Continue ›

```
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;
```

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.

MYSQL DAY 1 TASK

TASK 13:

3	Toy Story 2	John Lasseter	1999	93	2	7.2	162798565	200600000
15	Toy Story 4	El Directore	2015	90	4	8.7	340000000	270000000
					4	8.7	340000000	270000000
					4	8.7	340000000	270000000

Query Results

Movie_id	Rating	Domestic_sales	International_sales
3	7.9	245852179	239163000
1	8.3	191796233	170162503
2	7.2	162798565	200600000
4	8.7	340000000	270000000
4	8.7	340000000	270000000
4	8.7	340000000	270000000
15	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.

RUN QUERY RESET

Continue >

TASK 14:

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

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.

RUN QUERY RESET

Continue >

TASK 15:

Exercise

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

Table: Movies

Id	Title	Director	Year	Length_minutes
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
10	Up	Pete Docter	2009	101
11	Toy Story 3	Lee Unkrich	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

Row(s) deleted

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

Exercise 15 — Tasks

1. This database is getting too big, lets remove all movies that were released **before 2005**. ✓
2. Andrew Stanton has also left the studio, so please remove all movies directed by him. ✓

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

RUN QUERY RESET

Continue >

Next – SQL Lesson 16: Creating tables
Previous – SQL Lesson 14: Updating rows

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

MYSQL DAY 1 TASK

TASK 16:

Exercise

In this exercise, you'll need to create a new table for us to insert some new rows into.

Table: Database

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

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 >

RUN QUERY RESET

TASK 17:

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	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

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 >

RUN QUERY RESET

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.

TASK18:

Query Results

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

Exercise 18 — Tasks

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

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

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

Continue >

RUN QUERY RESET

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.