

Road-Map: Day-33: Database: MySQL – Day-1 : Tasks

SQL Lesson 1: SELECT queries 101:

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;

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 ›

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

Answers:

1. SELECT * FROM Movies where id=6;
2. SELECT Title FROM Movies WHERE Year BETWEEN 2000 AND 2010;
3. SELECT Title FROM Movies WHERE Year NOT BETWEEN 2000 AND 2010;
4. SELECT Title FROM Movies LIMIT 5;

Table: Movies

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

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 ›

```
SELECT Title FROM Movies LIMIT 5;
```

RESET

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

Answers:

1. SELECT Title FROM Movies WHERE Title LIKE "%Toy%";
2. SELECT Title FROM Movies WHERE Director = "John Lasseter";
3. SELECT Title FROM Movies WHERE Director != "John Lasseter";
4. SELECT Title FROM Movies WHERE Title LIKE "%WALL%";

Table: Movies

Title
WALL-E
WALL-G

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 ›

```
SELECT Title FROM Movies WHERE Title LIKE "%WALL%";
```

RESET

SQL Lesson 4: Filtering and sorting Query results :

Answers :

1. SELECT DISTINCT Director FROM Movies ORDER BY Director;
2. SELECT Title FROM Movies ORDER BY Year DESC LIMIT 4;
3. SELECT Title FROM Movies ORDER BY Title ASC LIMIT 5;
4. SELECT Title FROM Movies ORDER BY Title ASC LIMIT 5 OFFSET 5;

Table: Movies

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

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 Title FROM Movies ORDER BY Title ASC LIMIT 5 OFFSET 5;
```

RESET

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_cities WHERE Longitude < -87.629798 ORDER BY Longitude;
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;

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 ›

SQL Lesson 6: Multi-table queries with JOINS :

Answers :

1. SELECT Title, Domestic_sales, International_sales FROM Movies INNER JOIN Boxoffice ON Movies.id = Boxoffice.Movie_id;
2. SELECT m.Title, bo.Domestic_sales, bo.International_sales FROM Movies AS m INNER JOIN Boxoffice AS bo ON m.id = bo.Movie_id WHERE bo.International_sales > bo.Domestic_sales;
3. SELECT m.Title, bo.Rating FROM Movies AS m INNER JOIN Boxoffice AS bo ON m.id = bo.Movie_id ORDER BY bo.Rating DESC;

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 m.Title, bo.Rating FROM Movies AS m INNER JOIN Boxoffice AS bo ON m.id = bo.Movie_id ORDER BY bo.Rating DESC;
```

RESET

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 ›

SQL Lesson 7: OUTER JOINS :

Answers :

1. SELECT DISTINCT building FROM employees;
2. SELECT Building_name, Capacity FROM Buildings;
3. SELECT DISTINCT b.Building_name, e.Role FROM Buildings AS b LEFT JOIN Employees AS e ON b.Building_name = e.Building;

Query Results

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

```
SELECT DISTINCT b.Building_name, e.Role FROM Buildings AS b LEFT JOIN Employees AS e ON b.Building_name = e.Building;
```

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.

Continue ›

SQL Lesson 8: A short note on NULLs :

Answers :

1. SELECT e.Name, e.Role FROM employees AS e LEFT JOIN Buildings AS b ON e.Building = B.Building_name WHERE B.Building_name IS NULL ;
2. SELECT DISTINCT b.Building_name FROM Buildings AS b LEFT JOIN Employees AS e ON b.Building_name = e.Building WHERE Role IS NULL;

Query Results

Building_name
1w
2e

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.

Continue ›

```
SELECT DISTINCT b.Building_name FROM Buildings AS b LEFT JOIN Employees AS e
ON b.Building_name = e.Building WHERE Role IS NULL;
```

RESET

SQL Lesson 9: Queries with expressions :

Answers :

- SELECT DISTINCT m.Title, (b.Domestic_sales+b.International_sales)/1000000 AS Millions
from Movies as m JOIN Boxoffice AS b ON m.Id = b.Movie_id ORDER BY Millions DESC;
- SELECT m.Title, (b.Rating)*10 AS Rating from Movies as m JOIN Boxoffice AS b ON m.Id =
b.Movie_id ORDER BY b.Rating DESC;
- SELECT m.Title, m.Year from Movies as m WHERE (M.YEAR%2)==0;

Query Results

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

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.

Continue ›

```
SELECT m.Title, m.Year from Movies as m WHERE (M.YEAR%2)==0;
```

RESET

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

Answers :

1. SELECT Name,MAX(Years_employed) FROM Employees;
2. SELECT Role, AVG(Years_employed) as Average_years_employed FROM Employees GROUP BY Role;
3. SELECT Building, SUM(Years_employed) AS Total_Years FROM employees GROUP BY Building;

Table: Employees

Building	Total_Years
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 FROM employees GROUP BY Building;
```

RESET

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

Answers :

1. SELECT Role, COUNT(Role) AS Artist_Count From employees WHERE Role = 'Artist';
2. SELECT Role, COUNT(Role) AS Total_Employees FROM Employees GROUP BY Role;
3. SELECT Role, SUM(Years_employed) AS Total_Years FROM employees WHERE Role='Engineer';

Table: Employees

Role	Total_Years
Engineer	17

```
SELECT Role, SUM(Years_employed) AS Total_Years FROM employees WHERE 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.

RESET Continue ›

SQL Lesson 12: Order of execution of a Query :

Answers :

1. SELECT Director, COUNT(Director) AS Total_Movies FROM movies GROUP BY Director;
2. SELECT m.Director, SUM(b.Domestic_sales) + SUM(b.International_sales) AS Total_Sales FROM Movies AS m JOIN Boxoffice AS b ON m.Id = b.Movie_id GROUP BY m.Director;

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 m.Director, SUM(b.Domestic_sales) + SUM(b.International_sales) AS Total_Sales FROM Movies AS m JOIN Boxoffice AS b ON m.Id = b.Movie_id GROUP BY m.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 ›

SQL Lesson 13: Inserting rows :

Answers :

1. INSERT INTO Movies VALUES(15, 'Toy Story 4', 'John Lasseter', 2002, 90);
2. INSERT INTO BoxOffice VALUES(15, 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
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 >

SQL Lesson 14: Updating rows :

Answers :

1. UPDATE movies SET director = "John Lasseter" WHERE Title = "A Bug's Life";
2. UPDATE movies SET Year = 1999 WHERE Title = "Toy Story 2";
3. UPDATE movies SET Title = "Toy Story 3", Director = "Lee Unkrich" WHERE ID = 11;

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

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 >

SQL Lesson 15: Deleting rows :

Answers :

1. DELETE FROM Movies WHERE Year < 2005;
2. DELETE FROM Movies WHERE Director = 'Andrew Stanton';

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.

RUN QUERY RESET

Continue ›

SQL Lesson 16: Creating tables :

Answers :

1. CREATE TABLE Database (Name varchar(20), Version float ,Download_count int);

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.

RUN QUERY RESET

Continue ›

SQL Lesson 17: Altering tables :

Answers :

1. ALTER TABLE Movies ADD COLUMN Aspect_ratio FLOAT DEFAULT 4.5;
2. ALTER TABLE Movies ADD COLUMN Language TEXT DEFAULT "English";

Table: Movies

Id	Title	Director	Year	Length_minutes	Aspect_ratio	Language
1	Toy Story	John Lasseter	1995	81	4.5	English
2	A Bug's Life	John Lasseter	1998	95	4.5	English
3	Toy Story 2	John Lasseter	1999	93	4.5	English
4	Monsters, Inc.	Pete Docter	2001	92	4.5	English
5	Finding Nemo	Andrew Stanton	2003	107	4.5	English
6	The Incredibles	Brad Bird	2004	116	4.5	English
7	Cars	John Lasseter	2006	117	4.5	English
8	Ratatouille	Brad Bird	2007	115	4.5	English
9	WALL-E	Andrew Stanton	2008	104	4.5	English
10	Up	Pete Docter	2009	101	4.5	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.

RUN QUERY RESET

Continue ›

SQL Lesson 18: Dropping tables :

Answers :

1. DROP TABLE Movies;
2. DROP TABLE BoxOffice;

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.

RUN QUERY RESET

Continue ›



SQLBolt

Learn SQL with simple, interactive exercises.

SQL Lesson X: To infinity and beyond!



You've finished the tutorial!

-----End-----