

SQL Lesson 1: SELECT queries 101

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 ›

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;

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

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 ›

Answers:

1. `SELECT * FROM movies WHERE id = 6;`
2. `SELECT * FROM movies WHERE year BETWEEN 2000 AND 2010;`
3. `SELECT * FROM movies WHERE year NOT BETWEEN 2000 AND 2010;`
4. `SELECT * FROM movies WHERE id BETWEEN 1 AND 5;`

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

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 ›

Answers:

1. `SELECT * FROM movies where Title Like '%Toy Story%';`
2. `SELECT * FROM movies where Director Like '%john Lasseter%';`
3. `SELECT * FROM movies where Director NOT Like '%john Lasseter%';`
4. `SELECT * FROM movies where Title Like '%Wall%';`

SQL Lesson 4: Filtering and sorting Query results

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 ›

Answers:

1. `SELECT distinct Director FROM movies ORDER BY Director ASC;`
2. `SELECT Title, Year 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;`

SQL Review: Simple SELECT Queries

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 ›

Answers:

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

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 ›

Answers:

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

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 ›

Answers:

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

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 ›

Answers:

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

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 ›

Answers:

1.

```
SELECT Title, (domestic_sales + international_sales) / 1000000  
AS Gross_sales_millions FROM Movies JOIN Boxoffice ON  
movies.id = Boxoffice.Movie_id;
```
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)

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 ›

Answers:

1. `SELECT MAX(years_employed) as 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_employed FROM Employees GROUP BY Building;`

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

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 ›

Answers:

1. `SELECT Role, COUNT(*) as Number_of_artists FROM Employees WHERE Role = "Artist";`
2. `SELECT Role, COUNT(*)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

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 >](#)

Answers:

1. `SELECT Director, COUNT(id) as Num_movies_directed FROM Movies GROUP BY Director;`
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

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 ›

Answers:

1. INSERT INTO Movies VALUES (4, "Toy Story 4", "El Directore", 2015, 90);
2. INSERT INTO Boxoffice VALUES (4, 8.7, 340000000, 270000000);

SQL Lesson 14: Updating rows

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 ›

Answers:

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;

SQL Lesson 15: Deleting rows

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 >

Answers:

1. DELETE FROM Movies where Year < 2005;
2. DELETE FROM Movies where Director = "Andrew Stanton";

SQL Lesson 16: Creating tables

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 ›

Answers:

1. CREATE TABLE Database (Name TEXT, Version FLOAT, Download_count INTEGER);

SQL Lesson 17: Altering tables

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 >

Answers:

1. ALTER TABLE Movies ADD COLUMN Aspect_ratio FLOAT DEFAULT 2.39;
2. ALTER TABLE Movies ADD COLUMN Language TEXT DEFAULT "English";

SQL Lesson 18: Dropping tables

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 ›

Answers:

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

SQL Lesson X: To infinity and beyond!



SQLBolt

Learn SQL with simple, interactive exercises.

SQL Lesson X: To infinity and beyond!



You've finished the tutorial!