

3 PILLAR GLOBAL ASSIGNMENT

SQL

1. Select Queries 101

The screenshots illustrate the following steps in the SQLBolt tutorial:

- Step 1:** A table of movies with columns Title and Year. The SQL query is `SELECT title, year FROM movies;`
- Step 2:** The same table of movies. The SQL query is `SELECT title, year FROM movies;`
- Step 3:** A table of movies with columns Title and Director. The SQL query is `SELECT title, director FROM movies;`
- Step 4:** The same table of movies with columns Title and Director. The SQL query is `SELECT Director FROM movies;`
- Step 5:** A table of movies with columns Id, Title, Director, Year, and Length (minutes). The SQL query is `SELECT * FROM movies;`

The tasks listed in the exercises are:

- Find the title of each film ✓
- Find the director of each film ✓
- Find the title and director of each film ✓
- Find the title and year of each film ✓
- Find all the information about each film ✓

2. Queries with constraints

The screenshots show a web-based SQL learning interface with four exercises. Each exercise includes a 'Table: Movies' and a list of tasks.

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

Table: Movies

Id	Title	Director	Year	Length minutes
6	The Incredibles	Brad Bird	2004	116

SQL Query:

```
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

Table: Movies

Title
Monsters, Inc.
Finding Nemo
The Incredibles
Cars
Ratatouille
WALL-E
Up
Toy Story 3

SQL Query:

```
SELECT title FROM movies where year > 2000 and year < 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

Table: Movies

Title
Toy Story
A Bug's Life
Toy Story 2
Cars 2
Brave
Monsters University

SQL Query:

```
SELECT title FROM movies where year not 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

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

SQL Query:

```
SELECT * FROM movies where year limit 5;
```

3. Queries with Constraints

The image displays four screenshots of a SQL learning interface, likely from a website like SQLBolt. The interface is divided into sections for exercises, tables, and queries.

Top Left Screenshot: Shows a query editor with the text `AND/OR another_condition` and `AND/OR ...`. Below it, a table named "Movies" is shown with columns "Title" and "Director". The table contains three rows: "Toy Story", "Toy Story 2", and "Toy Story 3". To the right, "Exercise 3 — Tasks" lists four 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, and 4. Find all the WALL* movies. Below the tasks, a query is shown: `SELECT title FROM movies where title like 'toy story';`. A "Solution" link is provided.

Top Right Screenshot: Shows a query editor with the text `SELECT * FROM movies where director = "John Lasseter";`. To the right, a table named "Movies" is shown with columns "Id", "Title", "Director", "Year", and "Length minutes". The table contains six rows: "Toy Story", "A Bug's Life", "Toy Story 2", "Cars", "Cars 2", and "Cars 3". To the right, "Exercise 3 — Tasks" lists four 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, and 4. Find all the WALL* movies. Below the tasks, a "Solution" link is provided.

Bottom Left Screenshot: Shows a query editor with the text `SELECT title,director FROM movies where director != "John Lasseter";`. To the right, a table named "Movies" is shown with columns "Title" and "Director". The table contains eight rows: "Monsters, Inc.", "Finding Nemo", "The Incredibles", "Ratatouille", "WALL-E", "Up", "Toy Story 3", and "Brave". To the right, "Exercise 3 — Tasks" lists four 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, and 4. Find all the WALL* movies. Below the tasks, a "Solution" link is provided.

Bottom Right Screenshot: Shows a query editor with the text `SELECT * FROM movies where title like 'wall';`. To the right, a table named "Movies" is shown with columns "Id", "Title", "Director", "Year", and "Length minutes". The table contains two rows: "WALL-E" and "WALL-G". To the right, "Exercise 3 — Tasks" lists four 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, and 4. Find all the WALL* movies. Below the tasks, a "Solution" link is provided.

4. Filtering and Sorting query results

The screenshots show a series of SQL exercises on a movie database. The exercises are divided into two main sections: 'Exercise 4 - Tasks' and 'Exercise 5 - Tasks'.

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 ✓

Exercise 5 - 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 ✓

SQL Queries:

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

```
SELECT distinct title FROM movies order by year desc limit 4;
```

```
SELECT title FROM movies order by title limit 5;
```

The screenshots also show the 'MOVIES' table structure and the results of the queries. The table structure is as follows:

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

Title
Monsters University
Brave
Cars 2
Toy Story 3

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

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

5. Multi-table queries with joins

The screenshots show the SQLBolt website interface for a multi-table query exercise. The exercise involves finding movies not released in 2004 and their domestic and international sales.

Query Results (Screenshot 1):

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

Query Results (Screenshot 2):

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

Query Results (Screenshot 3):

Title
WALL-E
Toy Story 3
Toy Story
Up
Finding Nemo
Monsters, Inc.
Ratatouille
The Incredibles
Toy Story 2
Monsters University

6. Outer Joins

The first screenshot shows the SQLBolt interface with the following query results:

Building
1e
2w

The query entered is: `SELECT distinct building FROM employees;`

The second screenshot shows the following query results:

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

The query entered is: `SELECT * FROM buildings;`

The third screenshot shows the following query results:

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

The query entered is: `SELECT DISTINCT building_name, role FROM buildings LEFT JOIN employees ON building_name = employees.building;`