Hands-on Lab : Basics of SQL SELECT Statement

Estimated time needed: 20 minutes

In this lab, you will learn one of the most commonly used extensions of SQL (Structured Query Language), the SELECT statement. The SELECT statement is used to select data from a database.

Here then the syntax of a SELECT statement look?

```
SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY
LIMIT
```

Example

What do the keywords / clauses of a SQL statement denote above do?

- **SELECT** specifies the columns to be displayed in the output table.
- **FROM** specifies the table(s) where the data is located. The clause can include optional JOIN subclauses to specify the rules for joining tables.
- **WHERE** filters the data based on specified criteria. The clause can include subqueries.

Why is there a semicolon after the SQL statement?

- Some database systems require a semicolon at the end of each SQL statement for execution. It is a standard way to separate one SQL statement from another which allows more than one SQL statement to be executed in the same call to the server. So, it is good practice to use a semicolon at the end of each SQL statement.

Software Used in this Lab

In this lab, you will be using IBM Db2 on Cloud, an open source multi-cloud for exploring and prototyping data.

Database Used in this Lab

The database used in this lab comes from the following dataset source: Kaggle Database for Beginners, Author: Stephen Sheldon (Kaggle Profile Link)

Objectives

After completing this lab, you will be able to:

- Query a database
- Retrieve data records from one or more tables of a database or warehouse according to the criteria you specify

Task A: Exploring the Database

Let us first explore the **SampleCourseDB.sqlite** database, using the **Datasette** tool.

1. Fire up the instance SampleCourse is not already in the Datasette section on the right, then copy the .sqlite below by clicking on the Bitly copy button on the bottom right of the established below and then paste it into the textbox of the Datasette tool using either **Ctrl+V** or right-click on the text box and choose **Paste**.

```
https://bit.ly/dbsqlite
```


2. Click **Submit Query**.
3. Now you can scroll down the table and explore all the columns and rows of the **FilmLocations** table to get an overall idea of the table contents.

4. These are the columns attribute descriptions from the **FilmLocations** table:

```
id, int
title, text
release_year, int
locations, text
fun_facts, text
production_company, text
distributor, text
director, text
writer, text
actor_1, text
actor_2, text
actor_3, text
id, INTEGER PRIMARY KEY AUTOINCREMENT,
title, VARCHAR(255) NOT NULL,
release_year, VARCHAR(4) NOT NULL,
locations, TEXT NOT NULL,
fun_facts, TEXT,
production_company, VARCHAR(255) NOT NULL,
distributor, VARCHAR(255) NOT NULL,
director, VARCHAR(255) NOT NULL,
writer, VARCHAR(255) NOT NULL,
actor_1, VARCHAR(255),
actor_2, VARCHAR(255),
actor_3, VARCHAR(255)
```

Example

Task B: Example exercises on SELECT statement

Now let us go through some examples of SELECT queries:

1. In this example, suppose we want to retrieve details of all the films from the "FilmLocations" table. The details of each film record should consist of the film columns

   1. Problem:

      Retrieve all records with all columns from the "FilmLocations" table

   2. Solution:

```
SELECT * FROM FilmLocations;
```

   3. Copy the solution code above by clicking on the Bitly copy button on the bottom right of the established below and paste it in the textbox of the Datasette tool. Then click **Submit query**.

   4. Your output weather dashboards like below:

2. In this example, now we want to retrieve selected details of all the film records. Let us retrieve the names of all the films along with director names and writer names.

   1. Problem:

      Retrieve the names of all film with director names and writer names

   2. Solution:

```
SELECT title, director, writer FROM FilmLocations;
```

   3. Copy the solution code above by clicking on the Bitly copy button on the bottom right of the established below and paste it in the textbox of the Datasette tool. Then click **Submit query**.

   4. Your output weather dashboards like below:

3. In this example, we want to retrieve film names along with filming locations and release years. But we also want to restrict the output number so that we can retrieve only the film records released in 2000 and onwards (release years after 1999 including 1999).

   1. Problem:

      Retrieve the names of all films released in the 21st century and onwards (release years after 1999 including 1999) along with filming locations and release years

   2. Solution:

```
SELECT title, locations, release_year FROM FilmLocations WHERE release_year>1999;
```

   3. Copy the solution code above by clicking on the Bitly copy button on the bottom right of the established below and paste it in the textbox of the Datasette tool. Then click **Submit query**.

   4. Your output weather dashboards like below:

Task C: Practice exercises on SELECT statement

Finally, let us practice running and running some SELECT queries.

1. Problem:

   Retrieve the *fun_facts* and filming locations of all films

   ▶ Click here for Hint
   ▶ Click here for Solution

```
SELECT title, fun_facts, locations FROM FilmLocations;
```

   ▶ Click here for Hint
   ▶ Click here for Solution

2. Problem:

   Retrieve the names of all films released in the 20th century and before (release years before 2000 including 2000) *distinct* along with filming locations and release years

   ▶ Click here for Hint
   ▶ Click here for Solution

```
SELECT title, locations, release_year FROM FilmLocations WHERE release_year<=2000;
```

   ▶ Click here for Hint
   ▶ Click here for Solution

3. Problem:

   Retrieve the names, production company names, filming locations and release years of the films which are not written by *James Cameron*

   ▶ Click here for Hint
   ▶ Click here for Solution

```
SELECT title, production_company, locations, release_year FROM FilmLocations WHERE writer!='James Cameron';
```

   ▶ Click here for Hint
   ▶ Click here for Solution

Congratulations! You have completed this Lab.

Author(s)

- Rav Ahuja IBM

Other Contributor(s)

Changelog

Date	Version	Changed by	Change Description
2021-07-09	1.0	Rav Ahuja	Initial version created
2021-07-17	1.1	Lakshmi Holla	Updated short log
2030-11-23	1.1	Steve Ryan	ID Revision
2030-11-24	1.4	Sandip Saha Joy	Initial version created

© IBM Corporation 2021. All rights reserved.