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 statements of SQL (Structured Query Language), the SELECT statement. The SELECT statement is used to select data from a database.

Objectives

After completing this lab, you will be able to:

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

Concepts covered

How does the syntax of a SELECT statement look?

```
1
2
3
4
```

```
SELECT column1, column2, ...
FROM table_name
WHERE condition
;
```

Copied!

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

- **FROM**: Specifies from which table to get the data. The clause can include optional JOIN subclauses to specify the rules for joining tables.
- [Optional Clause] **WHERE** : Specifies which rows to retrieve.

Why is there a semicolon after the SQL statements?

- 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.

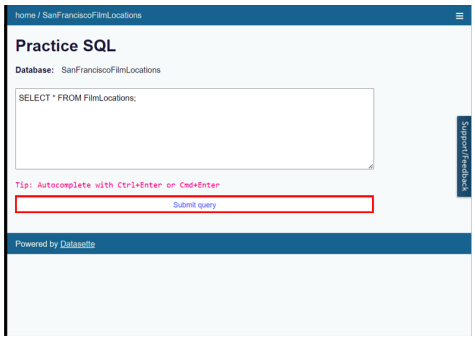
Introduction to Lab Environment

Software Used in this Lab

In this lab, you will use Datasette, an open source tool for exploring and publishing data. You can visit the Datasette GitHub repository here.

Working with Datasette

The **Datasette** tool offers a platform to input and execute SQL queries. By clicking the **Submit query** button, you can execute the SQL query.



Database Used in this Lab

The database used in this lab comes from the following dataset source: Film Locations in San Francisco under a PDDL: Public Domain Dedication and License.

Task A: Exploring the Database

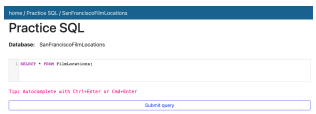
Let us first explore the **SanFranciscoFilmLocations** database using the **Datasette** tool:

1. If the first statement listed below is not already in the Datasette textbox on the right, then copy the code below by clicking on the little copy button on the bottom right of the codeblock below and then paste it into the textbox of the Datasette tool using either **Ctrl+V** or right-click in the text box and choose **Paste**.

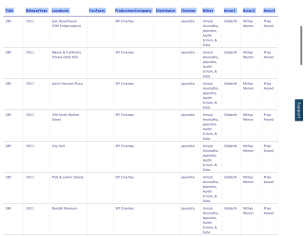
```
1
```

```
SELECT * FROM FilmLocations;
```

Copied!



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 column attribute descriptions from the **FilmLocations** table:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
```

```
FilmLocations(
    Title:              titles of the films,
    ReleaseYear:        time of public release of the films,
    Locations:          locations of San Francisco where the films were shot,
    FunFacts:           funny facts about the filming locations,
    ProductionCompany:  companies who produced the films,
    Distributor:        companies who distributed the films,
    Director:           people who directed the films,
    Writer:             people who wrote the films,
    Actor1:             person 1 who acted in the films,
    Actor2:             person 2 who acted in the films,
    Actor3:             person 3 who acted in the films
```

```
13. )
```
Copied!

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 contain all the film columns.
    1. Problem:

        *Retrieve all records with all columns from the **FilmLocations** table.*

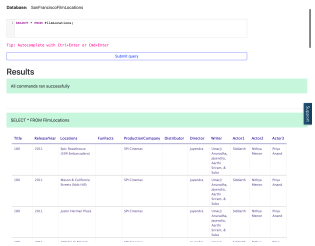
    2. Solution:

        ```
        1. SELECT * FROM FilmLocations;
        ```
        Copied!

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

    4. Your output resultset should match like below:

2. In this example, now we want to retrieve selective 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 films with director names and writer names.

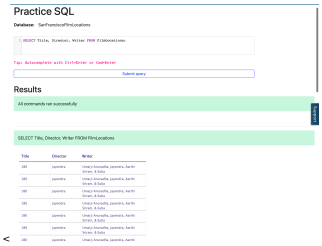
    2. Solution:

        ```
        1. SELECT Title, Director, Writer FROM FilmLocations;
        ```
        Copied!

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

    4. Your output resultset should match 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 resultset so that we can retrieve only the film records released in 2001 and onwards (release years after 2001 including 2001).
    1. Problem:

        Retrieve the names of all films released in the 21st century and onwards (release years after 2001 including 2001), along with filming locations and release years.

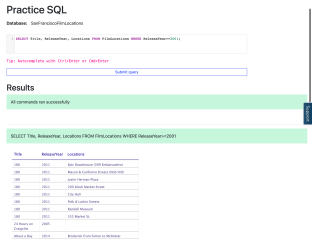
    2. Solution:

        ```
        1. SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE ReleaseYear>=2001;
        ```
        Copied!

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

    4. Your output resultset should match like below:

Task C: Practice exercises on SELECT statement

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

1. Problem:

    Retrieve the fun facts and filming locations of all films.

▾ Click here for Hint
Follow example 2 of SELECT where records have been retrieved containing details of some particular columns. Like,

```
1. SELECT [column1],[column2] FROM table_name;
```
Copied!

▾ Click here for Solution

```
1. 1
1. SELECT Locations, FunFacts FROM FilmLocations;
```
Copied!

▾ Click here for Output

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT Locations, FunFacts FROM FilmLocations;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

Support

SELECT Locations, FunFacts FROM FilmLocations

Locations	FunFacts
Epic Roasthouse (399 Embarcadero)	
Mason & California Streets (Nob Hill)	
Justin Herman Plaza	
200 block Market Street	
City Hall	
Polk & Larkin Streets	
Randall Museum	
555 Market St.	
Embarcadero	Embarcadero Freeway, which was featured in the film

2. Problem:

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

▾ Click here for Hint

Follow example 3 of SELECT where we restricted the output resultset so that we can retrieve only the film records with certain release years. Use WHERE clause comparsion operator <= which means **"Less than or equal to"**.

```
1. 1
1. SELECT [column1],[column2], [column3] FROM table_name WHERE [specifiedcolumn] <=2000;
```
Copied!

▾ Click here for Solution

```
1. 1
1. SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE ReleaseYear<=2000;
```
Copied!

▸ Click here for Output

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

Use WHERE clause comparsion operator <> which means **"Not equal to"**.

```
1. 1
1.  SELECT [column1], [column2], [column3], [column4] FROM table_name WHERE [specifiedcolumnname] <>"James Cameron";
```
Copied!

▾ Click here for Solution

```
1. 1
1. SELECT Title, ProductionCompany, Locations, ReleaseYear FROM FilmLocations WHERE Writer<>"James Cameron";
```
Copied!

▾ Click here for Output

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT Title, ProductionCompany, Locations, ReleaseYear FROM FilmLocations WHERE Writer<>"James Cameron";
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

Support

SELECT Title, ProductionCompany, Locations, ReleaseYear FROM FilmLocations WHERE Writer<>"James Cameron"

Title	ProductionCompany	Locations	ReleaseYear
180	SPI Cinemas	Epic Roasthouse (399 Embarcadero)	2011
180	SPI Cinemas	Mason & California Streets (Nob Hill)	2011
180	SPI Cinemas	Justin Herman Plaza	2011
180	SPI Cinemas	200 block Market Street	2011
180	SPI Cinemas	City Hall	2011
180	SPI Cinemas	Polk & Larkin Streets	2011
180	SPI Cinemas	Randall Museum	2011

Project Completion

Thank you for completing this lab where you learnt to query a database, and retrieve data records according to the criteria specified!

Author(s)

- Sandip Saha Joy

Changelog

Date	Version	Changed by	Change Description
2023-07-11	1.6	Lakshmi Holla	Updated labs
2023-06-02	1.5	Eric Hao	Fixed Page Styles
2023-05-10	1.4	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-04	1.3	Benny Li	Republished
2022-07-27	1.2	Lakshmi Holla	Updated html tag
2020-11-23	1.1	Steve Ryan	ID Review
2020-11-20	1.0	Sandip Saha Joy	Initial version created