

Basics of SQL SELECT Statement

Here we 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.

How does the syntax of a SELECT statement look?

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

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.

Software Used in this Lab

In this lab, we will use [Datasette](#), an open source multi-tool for exploring and publishing data.

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](#).

Objectives

- Querying a database.
- Retrieving data records from one or more tables of a database as result set according to the criteria specified.

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**.

```
SELECT * FROM FilmLocations;
```

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT * FROM FilmLocations;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

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.

Time	ReleaseYear	Locations	FunFacts	ProductionCompany	Distributor	Director	Writer	Actor1	Actor2	Actor3
180	2012	Epic Roadhouse (199 Embarcadero)		SP Cinemas		Jayendra	Umaji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddarth	Nitya Meenon	Priya Arund
180	2011	Mason & California Streets (Web Hill)		SP Cinemas		Jayendra	Umaji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddarth	Nitya Meenon	Priya Arund
180	2012	Justin Herman Plaza		SP Cinemas		Jayendra	Umaji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddarth	Nitya Meenon	Priya Arund
180	2011	200 Stock Market Street		SP Cinemas		Jayendra	Umaji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddarth	Nitya Meenon	Priya Arund
180	2012	City Hall		SP Cinemas		Jayendra	Umaji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddarth	Nitya Meenon	Priya Arund
180	2011	Polo & Larkin Streets		SP Cinemas		Jayendra	Umaji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddarth	Nitya Meenon	Priya Arund
180	2011	Randall Museum		SP Cinemas		Jayendra	Umaji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddarth	Nitya Meenon	Priya Arund

4. These are the column attribute descriptions from the **FilmLocations** table:
5. FilmLocations(
6. Title: titles of the films,
7. ReleaseYear: time of public release of the films,
8. Locations: locations of San Francisco where the films were shot,
9. FunFacts: funny facts about the filming locations,
10. ProductionCompany: companies who produced the films,
11. Distributor: companies who distributed the films,
12. Director: people who directed the films,
13. Writer: people who wrote the films,
14. Actor1: person 1 who acted in the films,
15. Actor2: person 2 who acted in the films,
16. Actor3: person 3 who acted in the films
17.)

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: `SELECT * FROM FilmLocations;`

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT * FROM FilmLocations;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

4. Output result set:

Database: SanFranciscoFilmLocations

```
1 SELECT * FROM FilmLocations;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

SELECT * FROM FilmLocations

Title	ReleaseYear	Locations	FunFacts	ProductionCompany	Distributor	Director	Writer	Actor1	Actor2	Actor3
180	2011	Epic Roadhouse (199 Embarkadero)		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand
180	2011	Mason & California Street (Nob Hill)		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand
180	2011	Justin Herman Plaza		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarti Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand

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: `SELECT Title, Director, Writer FROM FilmLocations;`

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT Title, Director, Writer FROM FilmLocations;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

4. Output result set:

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT Title, Director, Writer FROM FilmLocations;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT Title, Director, Writer FROM FilmLocations
```

Title	Director	Writer
180	Jayendra	Umari Anuradha, Jayendra, Aarthi Srinam, & Suba
180	Jayendra	Umari Anuradha, Jayendra, Aarthi Srinam, & Suba
180	Jayendra	Umari Anuradha, Jayendra, Aarthi Srinam, & Suba
180	Jayendra	Umari Anuradha, Jayendra, Aarthi Srinam, & Suba
180	Jayendra	Umari Anuradha, Jayendra, Aarthi Srinam, & Suba
180	Jayendra	Umari Anuradha, Jayendra, Aarthi Srinam, & Suba
180	Jayendra	Umari Anuradha, Jayendra, Aarthi Srinam, & Suba

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: `SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE ReleaseYear>=2001;`

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE ReleaseYear>=2001;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

4. Output result set:

Practice SQL

Database: SanFranciscoFilmLocations

```
1: SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE ReleaseYear>=2001;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE ReleaseYear>=2001
```

Title	ReleaseYear	Locations
180	2012	Gate Roadhouse (330 Embarcadero)
180	2012	Market & California Streets (Walk 148)
180	2012	Julien-Herman Plaza
180	2012	200 block Market Street
180	2012	City Hall
180	2012	Park & Larkin Streets
180	2012	Randall Museum
180	2012	155 Market St.
24 Hours on Craigslust	2001	
About a Boy	2014	Broderick from Fulton to McAlister

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.

Solution: **SELECT** Locations, FunFacts **FROM** FilmLocations;

Output:

Practice SQL

Database: SanFranciscoFilmLocations

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

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT Locations, FunFacts FROM FilmLocations
```

Locations	FunFacts
2pm Rusthouse	1999 (reunited)
Mason & California	Streets (Nov 1988)
Justin Herman Plaza	
200 block Market	Street
City Hall	
Pink & Larkin Streets	
Randall Museum	
333 Market St	

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.

Solution: `SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE ReleaseYear<=2000;`

Output:

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE ReleaseYear<=2000;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE ReleaseYear<=2000;
```

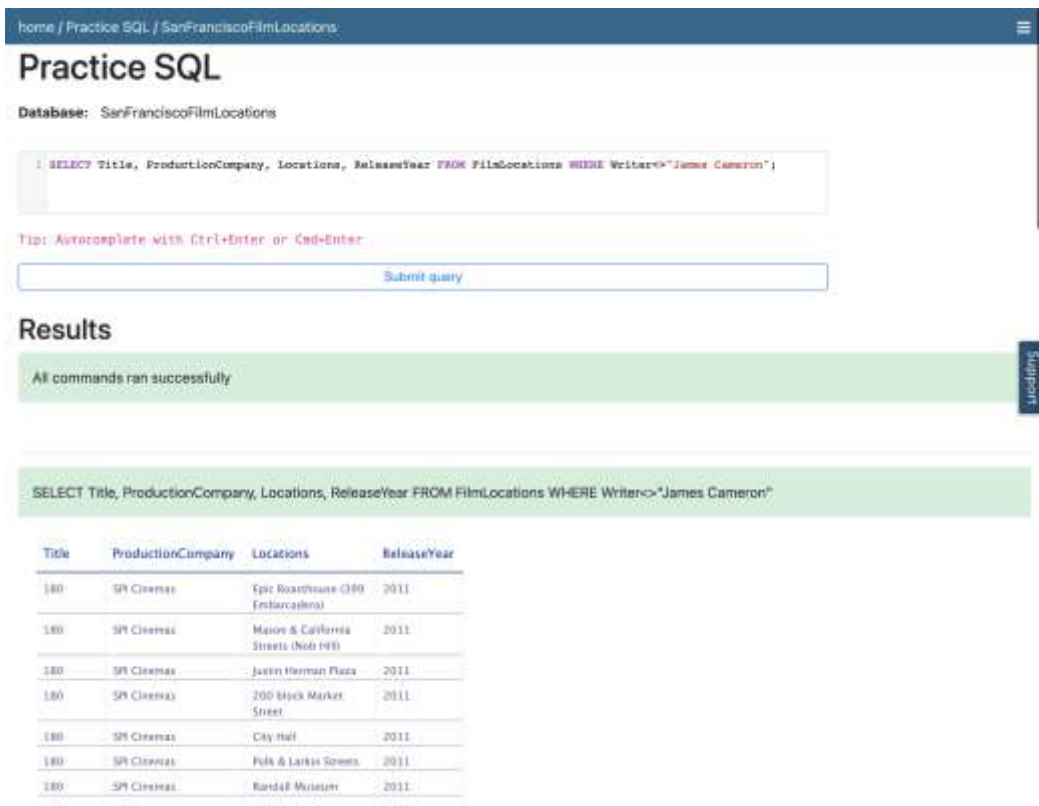
Title	ReleaseYear	Locations
A Night Full of Rain	1978	Embarcadero Freeway
A Night Full of Rain	1978	Farmers Hotel (950 Mason Street, Nob Hill)
A Night Full of Rain	1978	San Francisco Chronicle (601 Mission Street at 15th Street)
A Night Full of Rain	1978	Broadway (North Beach)
After the Thin Man	1936	Cable Tower
Another 48 Hours	1990	
Around the Fire	1998	Ocean Beach
Attack of the Killer Tomatoes	1978	Hyde Street Cable Car
Basic Instinct	1992	Yerba Buena Center for the Arts

3. Problem:

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

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

Output:



The screenshot shows a web-based SQL practice environment. At the top, a blue header bar contains the text "home / Practice SQL / SanFranciscoFilmLocations" and a hamburger menu icon. Below the header, the title "Practice SQL" is displayed in a large, bold font. Underneath the title, the database name "Database: SanFranciscoFilmLocations" is shown. A text input field contains the SQL query: `SELECT Title, ProductionCompany, Locations, ReleaseYear FROM FilmLocations WHERE Writer<>"James Cameron";`. Below the input field, a tip reads: "Tip: Autocomplete with Ctrl+Enter or Cmd+Enter". A "Submit query" button is located below the tip. The "Results" section shows a green message: "All commands ran successfully". Below this, the executed query is displayed: `SELECT Title, ProductionCompany, Locations, ReleaseYear FROM FilmLocations WHERE Writer<>"James Cameron";`. The results are presented in a table with four columns: Title, ProductionCompany, Locations, and ReleaseYear. The table contains six rows of data, all from the year 2011.

Title	ProductionCompany	Locations	ReleaseYear
180	SP Cinemas	Epic Roadhouse (180 Embarcadero)	2011
180	SP Cinemas	Major & California Streets (Nob Hill)	2011
180	SP Cinemas	Joann Herman Plaza	2011
180	SP Cinemas	200 block Market Street	2011
180	SP Cinemas	City Hall	2011
180	SP Cinemas	Polk & Larkin Streets	2011
180	SP Cinemas	Randall Museum	2011

This is the end of the notebook. In this code notebook some of knowledge required for a data scientist and some of the skills used by data scientists on a daily basis were shown and applied. The code and images examples were provided by IBM, and the development of the code, solutions and outputs, as well as some notes and editions, were carried out by me, Saulo Villaseñor (<https://www.linkedin.com/in/saulo-villase%C3%B1or-60669610a>), so that this notebook is available to everybody and work as a reference for anyone who wishes to learn new skills.