

COUNT, DISTINCT, LIMIT

We will learn a few useful expressions that are used with SELECT statements. First, you will learn COUNT, which is an aggregate function that retrieves the number of rows that matches the query criteria. Next, you will learn DISTINCT, which is used to remove duplicate values from a specified result set and only return the unique values. Lastly, you will learn LIMIT, which is used for restricting the number of rows retrieved from the table.

Software Used in this Lab

In this lab, you 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

- Retrieving the number of rows that match a query criteria
- Removing duplicate values from a result set and return the unique values
- Restricting the number of rows retrieved from a table

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**.

2. `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

3. Click **Submit Query**.
4. 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.

Title	ReleaseYear	Locations	FunFacts	ProductionCompany	Distributor	Director	Writer	Actor1	Actor2	Actor3
180	2011	Epic Rounthouse (399 Embarcadero)		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarthi Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand
180	2011	Moore & California Streets (Nob Hill)		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarthi Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand
180	2011	Justin Herman Plaza		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarthi Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand
180	2011	200-Hayes Market Street		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarthi Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand
180	2011	City Hall		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarthi Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand
180	2011	Polk & Leavenworth Streets		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarthi Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand
180	2011	Russell Museum		SP Cinemas		Jayendra	Umarji Anuradha, Jayendra, Aarthi Sriram, & Suba	Siddharth	Nithya Menon	Priya Anand
180	2011	335 Market St.		SP Cinemas		Jayendra	Umarji	Siddharth	Nithya	Priya

5. These are the column attribute descriptions from the **FilmLocations** table:
6. FilmLocations(
7. Title: titles of the films,
8. ReleaseYear: time of public release of the films,
9. Locations: locations of San Francisco where the films were shot,

10. FunFacts: funny facts about the filming locations,
11. ProductionCompany: companies who produced the films,
12. Distributor: companies who distributed the films,
13. Director: people who directed the films,
14. Writer: people who wrote the films,
15. Actor1: person 1 who acted in the films,
16. Actor2: person 2 who acted in the films,
17. Actor3: person 3 who acted in the films
18.)

Exercise 1: COUNT

In this exercise, you will first go through some examples of using COUNT in queries and then solve some exercise problems by using it.

Task A: Example exercises on COUNT

Let us go through some examples of COUNT related queries:

1. In this example, suppose we want to count the number of records or rows of the "FilmLocations" table.

1. Problem:

Retrieve the number of rows from the "FilmLocations" table.

2. Solution: `SELECT COUNT(*) FROM FilmLocations;`

4. Output resultset:

The screenshot shows a web-based SQL practice environment. At the top, there's a breadcrumb trail: "home / Practice SQL / SanFranciscoFilmLocations". Below this is the title "Practice SQL" and the database name "Database: SanFranciscoFilmLocations". A text input field contains the query: `SELECT COUNT(*) FROM FilmLocations;`. Below the input field is a tip: "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 query `SELECT COUNT(*) FROM FilmLocations` is repeated, followed by the result: `COUNT(*)` and the value `3414`. At the bottom, there's a footer: "Powered by Datasette".

2. In this example, now we want to count the number of locations of the films. But we also want to restrict the output resultset in such a way that we only retrieve the number of locations of the films written by a certain writer.

1. Problem:

Retrieve the number of locations of the films which are written by James Cameron.

2. Solution:

```
SELECT COUNT(Locations) FROM FilmLocations WHERE Writer="James Cameron";
```

4. Output resultset:

The screenshot shows a web interface for practicing SQL. At the top, there's a breadcrumb trail: "home / Practice SQL / SanFranciscoFilmLocations". Below this is the title "Practice SQL" and the database name "Database: SanFranciscoFilmLocations". A text input field contains the SQL query: `SELECT COUNT(Locations) FROM FilmLocations WHERE Writer="James Cameron";`. Below the input field is a tip: "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 query is repeated: `SELECT COUNT(Locations) FROM FilmLocations WHERE Writer="James Cameron"`. The result is displayed in a table with one column, "COUNT(Locations)", and one row with the value "48". At the bottom, there's a footer that says "Powered by Datasette".

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT COUNT(Locations) FROM FilmLocations WHERE Writer="James Cameron";
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT COUNT(Locations) FROM FilmLocations WHERE Writer="James Cameron"
```

COUNT(Locations)
48

Powered by Datasette

Task B: Practice exercises on COUNT

Now, let us practice creating and running some COUNT related queries.

1. Problem:

Retrieve the number of locations of the films which are directed by Woody Allen.

Solution:

```
SELECT COUNT(Locations) FROM FilmLocations WHERE Director="Woody Allen";
```

The screenshot shows a web interface for practicing SQL. At the top, there is a breadcrumb trail: "home / Practice SQL / SanFranciscoFilmLocations". Below this is the title "Practice SQL" and the database name "Database: SanFranciscoFilmLocations". A text input field contains the SQL query: `SELECT COUNT(Locations) FROM FilmLocations WHERE Director="Woody Allen";`. Below the input field is a tip: "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 shown: `SELECT COUNT(Locations) FROM FilmLocations WHERE Director="Woody Allen"`. The results are displayed in a table with one column, "COUNT(Locations)", and one row with the value "62". At the bottom, there is a footer: "Powered by Datasette".

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT COUNT(Locations) FROM FilmLocations WHERE Director="Woody Allen";
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT COUNT(Locations) FROM FilmLocations WHERE Director="Woody Allen"
```

COUNT(Locations)
62

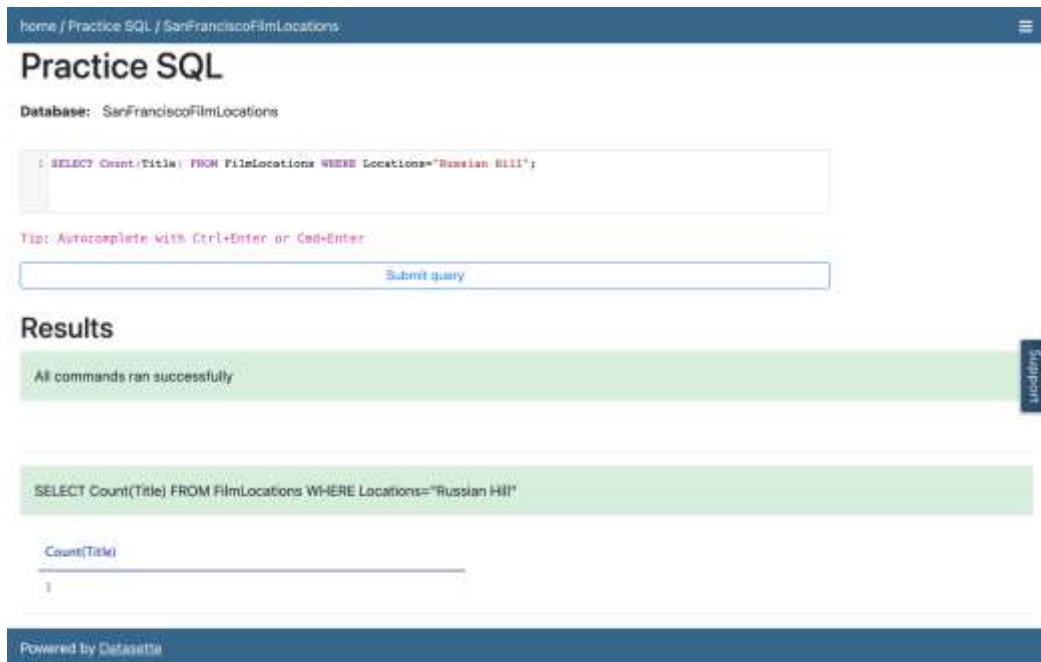
Powered by Datasette

2. Problem:

Retrieve the number of films shot at Russian Hill.

Solution:

```
SELECT Count(Title) FROM FilmLocations WHERE Locations="Russian Hill";
```



The screenshot shows a web-based SQL practice environment. At the top, a breadcrumb trail reads 'home / Practice SQL / SanFranciscoFilmLocations'. The main heading is 'Practice SQL'. Below it, the database is identified as 'SanFranciscoFilmLocations'. A text input field contains the SQL query: 'SELECT Count(Title) FROM FilmLocations WHERE Locations="Russian Hill";'. A tip below the input field suggests using 'Ctrl+Enter' or 'Cmd+Enter' for autocomplete. A 'Submit query' button is positioned to the right of the input field. Below the input field, the 'Results' section is displayed. It features a green status bar indicating 'All commands ran successfully'. The query is repeated in a light green box. Below this, a table with one column, 'Count(Title)', shows a single row with the value '1'. A 'Support' button is located on the right side of the results section. The footer of the interface states 'Powered by Datasette'.

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT Count(Title) FROM FilmLocations WHERE Locations="Russian Hill";
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT Count(Title) FROM FilmLocations WHERE Locations="Russian Hill"
```

Count(Title)
1

Powered by Datasette

3. Problem:

Retrieve the number of rows having a release year older than 1950 from the "FilmLocations" table.

Solution:

```
SELECT Count(*) FROM FilmLocations WHERE ReleaseYear<1950;
```

The screenshot shows a web-based SQL practice environment. At the top, a breadcrumb trail reads 'home / Practice SQL / SanFranciscoFilmLocations'. Below this is a header 'Practice SQL' and a sub-header 'Database: SanFranciscoFilmLocations'. A text input field contains the SQL query: 'SELECT Count(*) FROM FilmLocations WHERE ReleaseYear<1950;'. Below the input field is a tip: 'Tip: Autocomplete with Ctrl+Enter or Cmd+Enter'. A 'Submit query' button is located below the tip. The 'Results' section shows a green status bar with the text 'All commands ran successfully'. Below this, the same SQL query is displayed. A table with one column 'Count(*)' and one row containing the value '62' is shown. The footer of the interface reads 'Powered by Datasistia'.

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT Count(*) FROM FilmLocations WHERE ReleaseYear<1950;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT Count(*) FROM FilmLocations WHERE ReleaseYear<1950
```

Count(*)
62

Powered by Datasistia

Exercise 2: DISTINCT

In this exercise, you will first go through some examples of using DISTINCT in queries, and then solve some exercise problems by using it.

Task A: Example exercises of DISTINCT

Let us go through some examples of DISTINCT related queries:

1. In this example, we want to retrieve the title of all films in the table in such a way that duplicates will be discarded in the output resultset.

1. Problem:

Retrieve the name of all films without any repeated titles.

2. Solution:

```
SELECT DISTINCT Title FROM FilmLocations;
```

4. Output resultset:

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT DISTINCT Title FROM FilmLocations;
```

Tip! Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT DISTINCT Title FROM FilmLocations
```

Title
LEO
24 Hours on Craggill
A Night Full of Rain
About a Boy
Age of Adaline
After the Thin Man
Art-Man
Amertraha
Another 48 Hours
Around the Fire
Attack of the Killer Tomatoes

2. In this example, we want to retrieve the count of release years of the films produced by a specific company in such a way that duplicate release years of those films will be discarded in the count.

1. Problem:

Retrieve the number of release years of the films distinctly, produced by Warner Bros. Pictures.

2. Solution:

```
SELECT COUNT(DISTINCT ReleaseYear) FROM FilmLocations WHERE ProductionCompany="Warner Bros. Pictures";
```

4. Output result set:

The screenshot shows a web-based SQL practice environment. At the top, a breadcrumb trail reads 'home / Practice SQL / SanFranciscoFilmLocations'. Below this is a header 'Practice SQL' and a sub-header 'Database: SanFranciscoFilmLocations'. A text box contains the SQL query: `SELECT COUNT(DISTINCT ReleaseYear) FROM FilmLocations WHERE ProductionCompany="Warner Bros. Pictures";`. Below the text box is a tip: 'Tip: Autocomplete with Ctrl+Enter or Cmd+Enter'. A 'Submit query' button is located below the tip. The 'Results' section shows a green bar with the message 'All commands ran successfully'. Below this, the same SQL query is displayed. The result set is shown as a table with one column, 'COUNT(DISTINCT ReleaseYear)', and one row with the value '14'. At the bottom, a footer reads 'Powered by Datasette'.

home / Practice SQL / SanFranciscoFilmLocations

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT COUNT(DISTINCT ReleaseYear) FROM FilmLocations WHERE ProductionCompany="Warner Bros. Pictures";
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT COUNT(DISTINCT ReleaseYear) FROM FilmLocations WHERE ProductionCompany="Warner Bros. Pictures"
```

COUNT(DISTINCT ReleaseYear)
14

Powered by Datasette

Task B: Practice exercises on DISTINCT

Now, let us practice creating and running some DISTINCT related queries.

1. Problem:

Retrieve the name of all unique films released in the 21st century and onwards, along with their release years.

Solution:

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

Practice SQL

Database: SanFranciscoFilmLocations

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

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

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

Title	ReleaseYear
180	2011
24 Hours on Craigslist	2005
About a Boy	2014
Age of Adaline	2013
Ant-Man	2015
Americana	2015
Blue Jasmine	2013
Blue Velvet	2005
Big Eyes	2014
Big Sea	2013
Somewhere	2013

2. Problem:

Retrieve the names of all the directors and their distinct films shot at City Hall.

Solution:

```
SELECT DISTINCT Title, Director FROM FilmLocations WHERE Locations="City Hall";
```

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT DISTINCT Title, Director FROM FilmLocations WHERE Locations="City Hall";
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT DISTINCT Title, Director FROM FilmLocations WHERE Locations="City Hall"
```

Title	Director
180	Jayendra
Bedazzled	Harold Ramis
Westernized Man	Chris Columbus
Boys and Girls	Robert Lucove
Class Action	Michael Apted
Down of the Planet of the Apes	Matt Reeves
Final Analysis	Phil Joirens
The Rock	Michael Bay
The Wedding Planner	Adam Shankman
When We Rise	Gu Van Sant
The Night Stuff	Philip Kaufman

Support

3. Problem:

Retrieve the number of distributors distinctly who distributed films acted by Clint Eastwood as 1st actor.

Solution

```
SELECT COUNT(DISTINCT Distributor) FROM FilmLocations WHERE Actor1="Clint Eastwood";
```

[home](#) / [Practice SQL](#) / [SanFranciscoFilmLocations](#)

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT COUNT(DISTINCT Distributor) FROM FilmLocations WHERE Actor1="Clint Eastwood";
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
1 SELECT COUNT(DISTINCT Distributor) FROM FilmLocations WHERE Actor1="Clint Eastwood"
```

COUNT(DISTINCT Distributor)
7

Powered by [DataGrip](#)

Exercise 3: LIMIT

In this exercise, you will first go through some examples of using LIMIT in queries and then solve some exercise by using it.

Task A: Example exercises of LIMIT

Let us go through some examples of LIMIT related queries:

1. In this example, let us retrieve a specific number of rows from the top of the table in such a way that rows other than those are not in the output resultset.

1. Problem:

Retrieve the first 25 rows from the "FilmLocations" table.

2. Solution:

```
SELECT * FROM FilmLocations LIMIT 25;
```

4. Output result set:

Practice SQL

Database: SanFranciscoFilmLocations

```
SELECT * FROM FilmLocations LIMIT 25;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT * FROM FilmLocations LIMIT 25
```

Title	ReleaseYear	Locations	FunFacts	ProductionCompany	Distributor	Director	Writer	Actor1	Actor2	Actor3
180	2011	Epic Roushhouse (399 Erbslandstr.)		SPI Cinema		Jayendra	Unraj Anuradha, Jayendra, Aarti Sivan, & Subo	Siddharth	Nithya Menon	Priya Anand
180	2011	Mason & California Street (Rob Hill)		SPI Cinema		Jayendra	Unraj Anuradha, Jayendra, Aarti Sivan, & Subo	Siddharth	Nithya Menon	Priya Anand
180	2011	Justin Herman Plaza		SPI Cinema		Jayendra	Unraj Anuradha, Jayendra, Aarti Sivan, & Subo	Siddharth	Nithya Menon	Priya Anand

2. In this example, let us take the first example to a more advanced level. Now we want to retrieve a specific number of rows from the table, but this time, not from the top of the table. This time we want to retrieve a specific number of rows starting from a specific row in the table.

1. Problem:

Retrieve the first 15 rows from the "FilmLocations" table starting from row 11.

2. Solution:

```
SELECT * FROM FilmLocations LIMIT 15 OFFSET 10;
```

4. Output result set:

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT * FROM FilmLocations LIMIT 15 OFFSET 10;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT * FROM FilmLocations LIMIT 15 OFFSET 10;
```

Title	ReleaseYear	Locations	FunFacts	ProductionCompany	Distributor	Director	Writer	Actor1	Actor2	Actor3
A Night Full of Race	1978	Fairmont Hotel (950 Mason Street, Nob Hill)	In 1945 the Fairmont hosted the United Nations Conference on International Organization as delegates arrived to draft a charter for the organization. The U.S.	Liberty Film	Warner Bros. Atlanta	Lina Wertmuller	Lina Wertmuller	Candice Bergen	Giuseppe Gianni	

Task B: Practice exercises on LIMIT

Now, let us practice creating and running some LIMIT related queries.

1. Problem:

Retrieve the name of first 50 films distinctly.

Solution:

```
SELECT DISTINCT Title FROM FilmLocations LIMIT 50;
```

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT DISTINCT Title FROM FilmLocations LIMIT 50;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT DISTINCT Title FROM FilmLocations LIMIT 50
```

Title

180

24 Hours on Craiglist

A Night Full of Rain

About a Boy

Age of Adaline

After the Thin Man

Ace Man

Americana

Another 48 Hours

Around the Fire

Attack of the Killer Tomatoes

2. Problem:

Retrieve first 10 film names distinctly released in 2015.

Solution:

```
SELECT DISTINCT Title FROM FilmLocations WHERE ReleaseYear=2015 LIMIT 10;
```

Practice SQL

Database: SanFranciscoFilmLocations

```
1 SELECT DISTINCT Title FROM FilmLocations WHERE ReleaseYear=2015 LIMIT 10;
```

Tip: Autocomplete with Ctrl+Enter or Cmd+Enter

Submit query

Results

All commands ran successfully

```
SELECT DISTINCT Title FROM FilmLocations WHERE ReleaseYear=2015 LIMIT 10
```

Title

Age of Adam

Are-Mun

Americana

Supervetone

Cardinal II

I Am Michael

Steve Jobs

Quatern

Son Address

SonnetB

3. Problem:

Retrieve the next 3 film names distinctly after first 5 films released in 2015.

Solution:

```
SELECT DISTINCT Title FROM FilmLocations WHERE ReleaseYear=2015 LIMIT 3 OFFSET 5;
```

The screenshot shows the 'Practice SQL' interface on the Datasift website. The database is 'SanFranciscoFilmLocations'. The query entered is 'SELECT DISTINCT Title FROM FilmLocations WHERE ReleaseYear=2015 LIMIT 3 OFFSET 5;'. The results are displayed in a table with the following data:

Title
I Am Michael
Steve Jobs
Quitters

The interface also includes a 'Submit query' button, a 'Results' section, and a 'Powered by Datasift' footer.

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.