

Contents

MySQL MIN() and MAX() Functions	2
SQL COUNT(), AVG() and SUM() Functions	2
SQL LIKE Operator	3
SQL Wildcards	4
SQL IN Operator	5
SQL BETWEEN Operator	5
MySQL Aliases	6

MySQL MIN , MAX ,COUNT, AVG, SUM,
SQL LIKE SQL Wildcards,SQL IN Operator,SQL BETWEEN Operator,
MySQL Aliases,
MIN , MAX ,COUNT, AVG, SUM,LIKE, Wildcards , IN,
BETWEEN,Aliases,

```
CREATE TABLE demo.customers(  
  CustomerID INT,  
  CustomerName NVARCHAR(100),  
  ContactName NVARCHAR(100),  
  Address NVARCHAR(100),  
  City VARCHAR(100),  
  PostalCode VARCHAR(100),  
  Country VARCHAR(100)  
);
```

```
SELECT * FROM demo.customers;  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (1,'Alfreds Futterkiste','Maria Anders','Obere Str. 57','Berlin','12209','Germany');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (2,'Ana Trujillo Emparedados y helados','Ana Trujillo','Avda. de la Constituci3n 2222','M3ximo D.F.','05021','Mexico');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (3,'Antonio Moreno Taquer3a','Antonio Moreno','Mataderos 2312','M3ximo D.F.','05023','Mexico');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (4,'Around the Horn','Thomas Hardy','120 Hanover Sq.','London','WA1 1DP','UK');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (5,'Berglunds snabbk3p','Christina Berglund','Berguvsv3ngen 8','Lule3','S-958 22','Sweden');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (6,'Blauer See Delikatessen','Hanna Moos','Forsterstr. 57','Mannheim','68306','Germany');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (7,'Blondel p3re et fils','Fr3d3rique Citeaux','24, place Kl3ber','Strasbourg','67000','France');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (8,'B3lido Comidas preparadas','Mart3n Sommer','C/ Araquil, 67','Madrid','28023','Spain');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (9,'Bon app','Laurence Lebihans','12, rue des Bouchers','Marseille','13008','France');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (10,'Bottom-Dollar Marketse','Elizabeth Lincoln','23 Tsawassen Blvd.','Tsawassen','T2F 8M4','Canada');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (11,'B's Beverages','Victoria Ashworth','Fauntleroy Circus','London','EC2 5NT','UK');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (12,'Cactus Comidas para llevar','Patricio Simpson','Cerrito 333','Buenos Aires','1010','Argentina');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)  
  values (13,'Centro comercial Moctezuma','Francisco Chang','Sierras de Granada 9993','M3ximo D.F.','05022','Mexico');  
insert into customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)
```

```

values (14,'Chop-suey Chinese','Yang Wang','Hauptstr. 29','Bern','3012','Switzerland') ;
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)
values (15,'ComÃ©rcio Mineiro','Pedro Afonso','Av. dos LusÃ-adas, 23','SÃ£o Paulo','05432-043','Brazil') ;
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)
values (16,'Consolidated Holdings','Elizabeth Brown','Berkeley Gardens 12 Brewery','London','WX1 6LT','UK') ;
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)
values (17,'Drachenblut Delikatessend','Sven Ottlieb','Walserweg 21','Aachen','52066','Germany') ;
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)
values (18,'Du monde entier','Janine Labrune','67, rue des Cinquante Otages','Nantes','44000','France') ;
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)
values (19,'Eastern Connection','Ann Devon','35 King George','London','WX3 6FW','UK') ;
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`City`,`PostalCode`,`Country`)
values (20,'Ernst Handel','Roland Mendel','Graz','8010','Austria') ;

```

MySQL MIN() and MAX() Functions

Các **MIN()** chức năng trả về giá trị nhỏ nhất của cột được chọn.

Các **MAX()** hàm trả về giá trị lớn nhất của cột được chọn.

Cú pháp MIN ()

```

SELECT MIN(column_name)
FROM table_name
WHERE condition;

```

Cú pháp MAX ()

```

SELECT MAX(column_name)
FROM table_name
WHERE condition;

```

Vd:

```

SELECT MIN(CustomerID) AS SmallestPrice
FROM customers;

```

Vd:

```

SELECT MAX(ContactName) AS SmallestPrice
FROM customers;

```

SQL COUNT(), AVG() and SUM() Functions

*Cú pháp COUNT ()

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

Vd:

```
SELECT COUNT(City)
FROM customers ;
```

Cú pháp AVG ()

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

Vd:

```
SELECT AVG(CustomerID)
FROM customers ;
```

Cú pháp SUM ()

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

Vd:

```
-- SELECT * FROM customers;
SELECT SUM(CustomerID)
FROM customers ;
```

SQL LIKE Operator

https://www.w3schools.com/sql/sql_like.asp

Cú pháp LIKE

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

Các **LIKE** nhà điều hành được sử dụng trong một **WHERE** khoản để tìm kiếm một kieeru mẫu quy định trong một cột.

Có hai ký tự đại diện thường được sử dụng cùng với **LIKE** toán tử:

- Dấu phần trăm (%) đại diện cho không, một hoặc nhiều ký tự
- Dấu gạch dưới (_) đại diện cho một, một ký tự

Vd:

-- Câu lệnh SQL sau chọn tất cả khách hàng có Tên khách hàng bắt đầu bằng "a":

```
SELECT * FROM Customers  
WHERE CustomerName LIKE 'a%';
```

Vd:

-- Câu lệnh SQL sau đây chọn tất cả khách hàng có Tên khách hàng có "r" ở vị trí thứ hai:

```
SELECT * FROM Customers  
WHERE CustomerName LIKE '_r%';
```

SQL Wildcards

Vd:

-- Câu lệnh SQL sau đây chọn tất cả các khách hàng có Thành phố bắt đầu bằng "L", theo sau là bất kỳ ký tự nào, tiếp theo là "n", theo sau là bất kỳ ký tự nào, tiếp theo là "on":

```
SELECT * FROM Customers  
WHERE City LIKE 'L_n_on';
```

Vd:

-- Câu lệnh SQL sau đây chọn tất cả khách hàng có Thành phố chứa mẫu "es":

```
SELECT * FROM Customers  
WHERE City LIKE '%es%';
```

SQL IN Operator

Các **IN** nhà điều hành cho phép bạn chỉ định nhiều giá trị trong một **WHERE** mệnh đề.

Các **IN** nhà điều hành là một cách viết tắt cho nhiều **OR** điều kiện.

Cú pháp IN

```
SELECT column_name(s) FROM table_name  
WHERE column_name IN (value1, value2, ...);
```

hoặc:

```
SELECT column_name(s) FROM table_name  
WHERE column_name IN (SELECT STATEMENT);  
-- SELECT STATEMENT tức là có thể chọn từ 1 bảng khác
```

Vd:

-- Câu lệnh SQL sau chọn tất cả khách hàng ở "Đức", "Pháp" hoặc "Vương quốc Anh":

```
SELECT * FROM Customers  
WHERE Country IN ('Germany', 'France', 'UK');  
-- WHERE Country NOT IN ('Germany', 'France', 'UK');
```

Vd:

```
SELECT * FROM Customers  
WHERE Country IN (SELECT Country FROM Customers);  
-- WHERE Country IN (SELECT Country FROM Suppliers); lấy giá trị country từ 1 bảng khác
```

SQL BETWEEN Operator

Các **BETWEEN** giá trị khai thác Selects trong một phạm vi nhất định. Các giá trị có thể là số, văn bản hoặc ngày tháng.

Các **BETWEEN** nhà điều hành đã bao gồm: bắt đầu và giá trị cuối được bao gồm.

Cú pháp

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

Vd:

```
SELECT * FROM Customers
WHERE CustomerID BETWEEN 5 AND 10;
-- WHERE CustomerID NOT BETWEEN 5 AND 10;
```

Vd:

```
-- GIỮA Ví dụ về giá trị văn bản
-- Câu lệnh SQL sau chọn tất cả các sản phẩm có Tên sản phẩm giữa
Carnarvon Tigers và Mozzarella di Giovanni:
SELECT * FROM Customers
WHERE ContactName BETWEEN 'Christina Berglund' AND 'Elizabeth
Lincoln'
-- WHERE ContactName NOT BETWEEN 'Christina Berglund' AND
'Elizabeth Lincoln'
```

```
ORDER BY ContactName;
```

MySQL Aliases

Bí danh được sử dụng để đặt tên tạm thời cho một bảng, hoặc một cột trong bảng.

Bí danh thường được sử dụng để làm cho tên cột dễ đọc hơn.

Bí danh chỉ tồn tại trong khoảng thời gian của truy vấn đó. Một bí danh được tạo bằng **AS** từ khóa.

Cú pháp cột bí danh

```
SELECT column_name AS alias_name
FROM table_name;
```

Cú pháp bảng bí danh

```
SELECT column_name(s)
FROM table_name AS alias_name;
```

Vd:

```
SELECT CustomerID AS ID, CustomerName AS Customer  
FROM Customers;
```

Vd: note : bảng có 2 từ thì phải để ngoặc kép

```
SELECT CustomerName AS Customer, ContactName AS  
"Contact Person"  
FROM Customers;
```

Vd:

Câu lệnh SQL sau tạo một bí danh có tên "Địa chỉ" kết hợp bốn cột (Địa chỉ, Mã bưu điện, Thành phố và Quốc gia):

```
SELECT CustomerName, CONCAT_WS(' ', Address,  
PostalCode, City, Country) AS Address  
FROM Customers;
```