

Contents

MySQL SELECT Statement	2
MySQL WHERE Clause	3
MySQL AND, OR and NOT Operators	4
MySQL ORDER BY Keyword	5
MySQL INSERT INTO Statement	6
MySQL NULL Values	6
MySQL UPDATE Statement	7
MySQL DELETE Statement	8
MySQL LIMIT Clause	9

SELECT ,WHERE ,AND, OR and NOT ,ORDER BY ,INSERT INTO , NULL Values
,UPDATE,DELETE ,LIMIT

/Database : b ng customer

```
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 Constituci n 2222','M xico D.F.','05021','Mexico');  
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)  
  values (3,'Antonio Moreno Taquer a','Antonio Moreno','Mataderos 2312','M xico 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 snabb  p','Christina Berglund','Berguvsv gen 8','Lule  ','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 p  re et fils','Fr  d  rique Citeaux','24, place Kl  ber','Strasbourg','67000','France');  
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)  
  values (8,'B  lido Comidas preparadas','Mart  n Sommer','C/ Araquil, 67','Madrid','28023','Spain');  
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)  
  values (9,'Bon app  l','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','M xico 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 Labruno','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 SELECT Statement

Câu **SELECT** lệnh được sử dụng để chọn dữ liệu từ cơ sở dữ liệu.

Dữ liệu trả về được lưu trữ trong một bảng kết quả, được gọi là tập kết quả.

Cú pháp SELECT

```

SELECT column1, column2, ...
FROM table_name;

```

Ở đây, column1, column2, ... là tên trường của bảng mà bạn muốn chọn dữ liệu. Nếu bạn muốn chọn tất cả các trường có sẵn trong bảng, hãy sử dụng cú pháp sau:

```

SELECT * FROM table_name;

```

Vd:

```

SELECT CustomerName, City, Country FROM Customers;

```

Vd:

```

SELECT * FROM Customers;

```

Câu lệnh MySQL SELECT DISTINCT

Câu **SELECT DISTINCT** lệnh chỉ được sử dụng để trả về các giá trị riêng biệt (khác nhau).

Bên trong một bảng, một cột thường chứa nhiều giá trị trùng lặp; và đôi khi bạn chỉ muốn liệt kê các giá trị khác nhau (riêng biệt).

Cú pháp SELECT DISTINCT

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

Vd:

```
SELECT DISTINCT Country FROM Customers;
```

```
SELECT COUNT(DISTINCT Country) FROM Customers;
```

MySQL WHERE Clause

Các **WHERE** khoản được sử dụng để ghi chép lọc.
Nó chỉ được sử dụng để trích xuất những bản ghi đáp ứng một điều kiện cụ thể.

Cú pháp WHERE

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

Vd:

```
SELECT * FROM Customers  
WHERE Country = 'Mexico';
```

Vd:

```
SELECT * FROM Customers  
WHERE CustomerID = 1;
```

Các toán tử trong Mệnh đề WHERE

Các toán tử sau có thể được sử dụng trong **WHERE** mệnh đề:

Operator	Description	Example
=	Equal	Try it
>	Greater than	Try it
<	Less than	Try it
>=	Greater than or equal	Try it

<=	Less than or equal	Try it
<>	Not equal. Note: In some versions of SQL this operator may be written as !=	Try it
BETWEEN	Between a certain range	Try it
LIKE	Search for a pattern	Try it
IN	To specify multiple possible values for a column	Try it

MySQL AND, OR and NOT Operators

Các **WHERE** điều khoản có thể được kết hợp với **AND**, **OR** và **NOT**

AND Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

OR Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

NOT Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;
```

Vd:

```
SELECT * FROM Customers
WHERE Country = 'Germany' AND City = 'Berlin';
```

Vd:

```
SELECT * FROM Customers  
WHERE City = 'Berlin' OR City = 'Stuttgart';
```

Vd:

```
SELECT * FROM Customers  
WHERE NOT Country = 'Germany';
```

Vd:

```
SELECT * FROM Customers  
WHERE Country = 'Germany' AND (City  
= 'Berlin' OR City = 'Stuttgart');
```

MySQL ORDER BY Keyword

Các **ORDER BY** từ khóa được sử dụng để sắp xếp các kết quả-set trong tăng dần hoặc giảm dần.

Các **ORDER BY** từ khóa sắp xếp các bản ghi trong thứ tự tăng dần theo mặc định. Để sắp xếp các bản ghi theo thứ tự giảm dần, hãy sử dụng **DESC** từ khóa.

ORDER BY Syntax

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

Vd:

```
SELECT * FROM Customers  
ORDER BY Country;
```

Vd:

```
SELECT * FROM Customers  
ORDER BY Country, CustomerName;
```

Vd:

```
SELECT * FROM Customers  
ORDER BY Country DESC;
```

Vd:

```
SELECT * FROM Customers  
ORDER BY Country ASC, CustomerName DESC;
```

MySQL INSERT INTO Statement

Có thể viết **INSERT INTO** câu lệnh theo hai cách:

1. Chỉ định cả tên cột và giá trị được chèn:

```
INSERT INTO table_name (column1, column2, column3,  
...)
```

```
VALUES (value1, value2, value3, ...);
```

2. Nếu bạn đang thêm giá trị cho tất cả các cột của bảng, bạn không cần chỉ định tên cột trong truy vấn SQL. Tuy nhiên, hãy đảm bảo thứ tự của các giá trị theo thứ tự với các cột trong bảng. Ở đây, **INSERT INTO** cú pháp sẽ như sau:

```
INSERT INTO table_name
```

```
VALUES (value1, value2, value3, ...);
```

Vd:

```
insert into customers
```

```
(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`Postal  
Code`,`Country`)
```

```
values (20,'Ernst Handel','Roland Mendel','Kirchgasse  
6','Graz','8010','Austria');
```

vd:

```
INSERT INTO Customers (CustomerName, City, Country)
```

```
VALUES ('Cardinal', 'Stavanger', 'Norway');
```

MySQL NULL Values

Lưu ý: Giá trị NULL khác với giá trị 0 hoặc trường chứa khoảng trắng. Trường có giá trị NULL là trường đã bị bỏ trống trong quá trình tạo bản ghi!

IS NULL Syntax

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```

IS NOT NULL Syntax

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NOT NULL;
```

Vd:

```
SELECT CustomerName, ContactName, Address  
FROM Customers  
WHERE Address IS NULL;
```

Vd:

```
SELECT CustomerName, ContactName, Address  
FROM Customers  
WHERE Address IS NOT NULL;
```

MySQL UPDATE Statement

Câu **UPDATE** lệnh được sử dụng để sửa đổi các bản ghi hiện có trong một bảng.

Cú pháp UPDATE:

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Lưu ý: Hãy cẩn thận khi cập nhật các bản ghi trong một bảng! Chú ý **WHERE** mệnh đề trong **UPDATE** câu lệnh. Các **WHERE** quy định cụ thể khoản mà kỷ lục (s) cần được cập nhật. Nếu bạn bỏ qua **WHERE** mệnh đề, tất cả các bản ghi trong bảng sẽ được cập nhật!

Vd:

-- chú ý lỗi chặn update : edit=> preference=>sql
editor=> xuống cuối=> safe update => tắt đi connect
lại

UPDATE Customers

SET ContactName = 'Alfred Schmidt', City
= 'Frankfurt'

WHERE CustomerID = 1;

Vd:

UPDATE Customers

SET PostalCode = 00000

WHERE Country = 'Mexico';

Vd:

UPDATE Customers

SET PostalCode = 00000;

MySQL DELETE Statement

Câu **DELETE** lệnh được sử dụng để xóa các bản ghi hiện có trong một bảng.

Cú pháp DELETE

DELETE FROM *table_name* **WHERE** *condition*;

Lưu ý: Hãy cẩn thận khi xóa các bản ghi trong bảng! Chú ý **WHERE** mệnh đề trong **DELETE** câu lệnh. Các **WHERE** quy định cụ thể khoản mà kỷ lục (s) sẽ bị xóa. Nếu bạn bỏ qua **WHERE** mệnh đề, tất cả các bản ghi trong bảng sẽ bị xóa!

Vd:

DELETE FROM Customers **WHERE** CustomerName='Alfreds
Futterkiste';

DELETE FROM Customers; -- xóa bảng

MySQL LIMIT Clause

Các **LIMIT** khoản được sử dụng để xác định số lượng hồ sơ để trả lại.

Các **LIMIT** khoản rất hữu ích trên bảng lớn với hàng ngàn hồ sơ. Trả lại một số lượng lớn bản ghi có thể ảnh hưởng đến hiệu suất.

Cú pháp LIMIT

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number;
```

Vd:

```
SELECT * FROM Customers
LIMIT 3; -- hay LIMIT 0,3;
```

Vd:

```
SELECT * FROM Customers
LIMIT 4,3; -- 3 cái bắt đầu từ cái thứ 4
```