

## Contents

MySQL Joins .....	2
MySQL INNER JOIN Keyword .....	3
MySQL LEFT JOIN Keyword .....	4
MySQL RIGHT JOIN Keyword .....	4
MySQL CROSS JOIN Keyword .....	4
MySQL Self Join .....	5
MySQL UNION Operator .....	6

## Database : bảng 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 Constitución 2222','MÃ©xico D.F.','05021','Mexico') ;  
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)  
  values (3,'Antonio Moreno TaquerÃ­a','Antonio Moreno','Mataderos 2312','MÃ©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 snabbÃ©p','Christina Berglund','BerguvsvÃ©gen 8','LuleÃ¥','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 pÃ¢re et fils','FrÃ©dÃ©rique Citeaux','24, place KlÃ©ber','Strasbourg','67000','France') ;  
insert into customers(`CustomerID`,`CustomerName`,`ContactName`,`Address`,`City`,`PostalCode`,`Country`)  
  values (8,'BÃ³lido Comidas preparadas','MartÃ­n 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','MÃ©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 LusÃ­adas, 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') ;

```

## DT2:

```

CREATE TABLE demo.Orders(
  OrderID INT,
  CustomerID INT,
  EmployeeID INT,
  OrderDate VARCHAR(100) ,
  ShipperID INT
);
insert into `Orders` (`OrderID`,`CustomerID`,`EmployeeID`,`OrderDate`,`ShipperID`)
  values (10248,5,5,'1996-07-04',3) ;
insert into `Orders` (`OrderID`,`CustomerID`,`EmployeeID`,`OrderDate`,`ShipperID`)
  values (10249,8,6,'1996-07-05',1) ;
insert into `Orders` (`OrderID`,`CustomerID`,`EmployeeID`,`OrderDate`,`ShipperID`)
  values (10250,9,4,'1996-07-08',2) ;
insert into `Orders` (`OrderID`,`CustomerID`,`EmployeeID`,`OrderDate`,`ShipperID`)
  values (10251,10,3,'1996-07-08',1) ;
insert into `Orders` (`OrderID`,`CustomerID`,`EmployeeID`,`OrderDate`,`ShipperID`)
  values (10252,11,4,'1996-07-09',2) ;
insert into `Orders` (`OrderID`,`CustomerID`,`EmployeeID`,`OrderDate`,`ShipperID`)
  values (10253,12,3,'1996-07-10',2) ;
insert into `Orders` (`OrderID`,`CustomerID`,`EmployeeID`,`OrderDate`,`ShipperID`)
  values (10254,13,5,'1996-07-11',2) ;
insert into `Orders` (`OrderID`,`CustomerID`,`EmployeeID`,`OrderDate`,`ShipperID`)
  values (10255,100,9,'1996-07-12',3) ;

```

## MySQL Joins

Một **JOIN** mệnh đề được sử dụng để kết hợp các hàng từ hai hoặc nhiều bảng, dựa trên một cột có liên quan giữa chúng.

Vd:

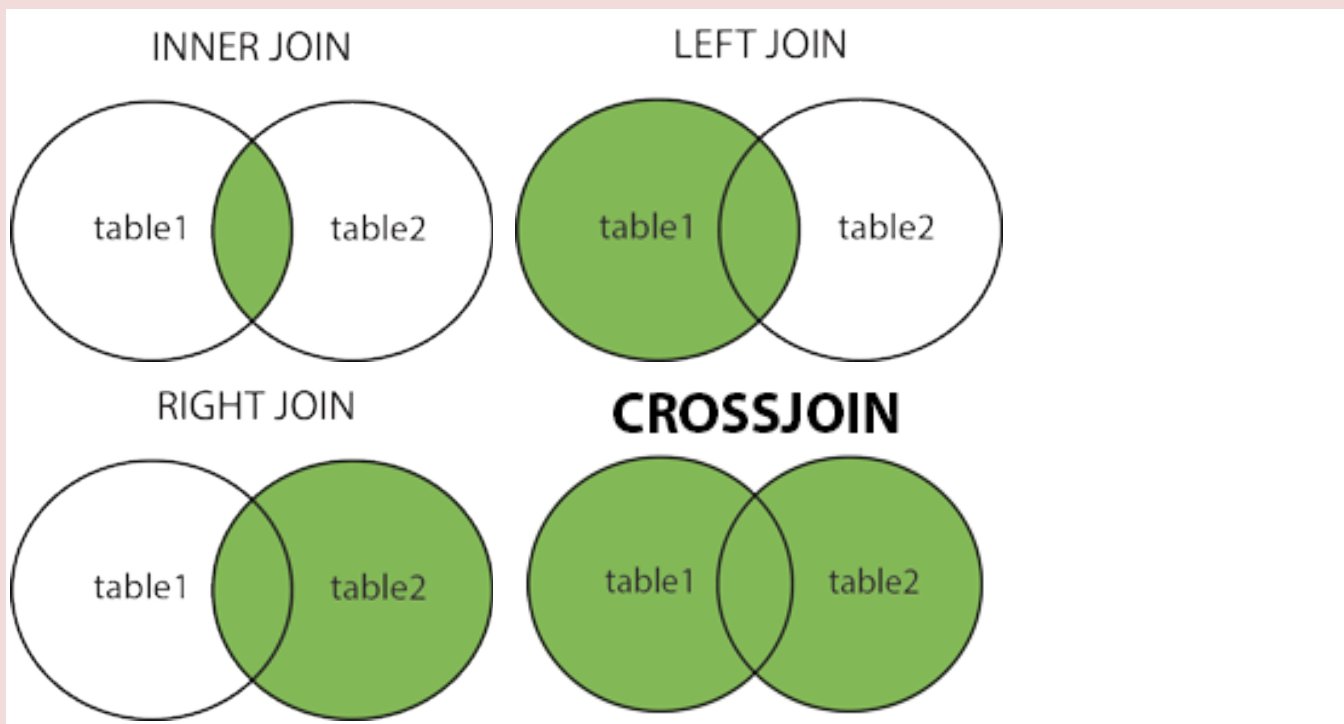
```

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

```

Các loại tham gia được hỗ trợ trong MySQL

- **INNER JOIN**: Trả về các bản ghi có giá trị phù hợp trong cả hai bảng
- **LEFT JOIN**: Trả về tất cả các bản ghi từ bảng bên trái và các bản ghi phù hợp từ bảng bên phải
- **RIGHT JOIN**: Trả về tất cả các bản ghi từ bảng bên phải và các bản ghi phù hợp từ bảng bên trái
- **CROSS JOIN**: Trả về tất cả các bản ghi từ cả hai bảng



## MySQL INNER JOIN Keyword

### Cú pháp INNER JOIN

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

### Vd:

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

=>

**Lưu ý:** Các **INNER JOIN** Selects từ khóa tất cả các dòng từ hai bảng như miễn là có một trận đấu giữa các cột. Nếu có bản ghi trong bảng "Đơn đặt hàng" không khớp trong "Khách hàng", những đơn đặt hàng này sẽ không được hiển thị!

## MySQL LEFT JOIN Keyword

### Cú pháp LEFT JOIN

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

#### Vd:

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerName;
```

**Lưu ý:** Các **LEFT JOIN** từ khóa trả về tất cả các bản ghi từ bảng bên trái (Khách hàng), ngay cả khi không có trận đấu ở bảng bên phải (Orders).

## MySQL RIGHT JOIN Keyword

### Cú pháp RIGHT JOIN

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

#### Vd:

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
right JOIN Orders ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerName;
```

**Lưu ý:** Các **right JOIN** từ khóa trả về tất cả các bản ghi từ bảng bên phải (Orders), ngay cả khi không có trận đấu ở bảng bên phải (khách hàng).

## MySQL CROSS JOIN Keyword

## Cú pháp CROSS JOIN

```
SELECT column_name(s)  
FROM table1  
CROSS JOIN table2;
```

### Vd:

```
SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
CROSS JOIN Orders;
```

Câu lệnh SQL sau chọn tất cả khách hàng và tất cả các đơn đặt hàng:

=> **Lưu ý:** Các **CROSS JOIN** lợi nhuận từ khóa tất cả hồ sơ phù hợp từ hai bảng dù bảng khác phù hợp hay không. Vì vậy, nếu có hàng trong "Khách hàng" không khớp trong "Đơn đặt hàng" hoặc nếu có hàng trong "Đơn hàng" không khớp trong "Khách hàng", những hàng đó cũng sẽ được liệt kê.

Nếu bạn thêm một **WHERE** mệnh đề (nếu table1 và table2 có mối quan hệ), mệnh đề **CROSS JOIN** sẽ tạo ra kết quả tương tự như **INNER JOIN** mệnh đề:

### Thí dụ

```
SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
CROSS JOIN Orders  
WHERE Customers.CustomerID=Orders.CustomerID;
```

## MySQL Self Join

Tự tham gia là một tham gia thông thường, nhưng bảng được kết hợp với chính nó.

### Cú pháp

```
SELECT column_name(s)  
FROM table1 T1, table1 T2  
WHERE condition
```

T1 và T2 là các bí danh bảng khác nhau cho cùng một bảng.

### Vd:

```
SELECT A.CustomerName AS CustomerName1,  
B.CustomerName AS CustomerName2, A.City  
FROM Customers A, Customers B  
WHERE A.CustomerID <> B.CustomerID  
AND A.City = B.City  
ORDER BY A.City;
```

=> Câu lệnh SQL sau phù hợp với khách hàng đến từ cùng một thành phố:

### MySQL UNION Operator

Các **UNION** nhà điều hành được sử dụng để kết hợp các kết quả-tập hợp của hai hay nhiều **SELECT** câu lệnh.

- Mọi **SELECT** câu lệnh bên trong **UNION** phải có cùng số cột
- Các cột cũng phải có kiểu dữ liệu tương tự
- Các cột trong mọi **SELECT** câu lệnh cũng phải theo cùng một thứ tự

Cú pháp UNION

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

**Vd:**

```
SELECT CustomerID FROM Customers  
UNION -- all  
SELECT CustomerID FROM Orders  
ORDER BY CustomerID;
```

### UNION ALL Cú pháp

Các **UNION** nhà khai thác lựa chọn duy nhất giá trị khác biệt theo mặc định. Để cho phép các giá trị trùng lặp, hãy sử dụng **UNION ALL**:

```
SELECT column_name(s) FROM table1  
UNION ALL  
SELECT column_name(s) FROM table2;
```

### UNION SQL có thể đi với WHERE

**vd:**

```
SELECT CustomerID FROM Customers
```

```
WHERE Country='Germany'  
UNION all  
SELECT CustomerID FROM Orders  
WHERE shipperid>2  
ORDER BY CustomerID;
```