

# MySQL - Using Join



Trong MySQL, `JOIN` được sử dụng để kết hợp các bảng khác nhau với nhau bằng cách sử dụng một hoặc nhiều cột chung. Có nhiều loại `JOIN` khác nhau, bao gồm:

- `INNER JOIN`: Trả về các bản ghi chỉ khi có các giá trị khớp trong cả hai bảng.
- `LEFT JOIN`: Trả về tất cả các bản ghi từ bảng bên trái và các bản ghi khớp từ bảng bên phải.
- `RIGHT JOIN`: Trả về tất cả các bản ghi từ bảng bên phải và các bản ghi khớp từ bảng bên trái.
- `FULL OUTER JOIN`: Trả về tất cả các bản ghi từ cả hai bảng, bao gồm các bản ghi không khớp.

## INNER JOIN

Giả sử bạn có hai bảng: `orders` và `customers`. Bảng `orders` chứa thông tin về các đơn hàng, bao gồm `order_id`, `customer_id` và `order_date`. Bảng `customers` chứa thông tin về khách hàng, bao gồm `customer_id`, `name` và `email`.

Để kết hợp hai bảng này để hiển thị thông tin về tên khách hàng và ngày đặt hàng cho mỗi đơn hàng, bạn có thể sử dụng câu lệnh SQL sau:

```
SELECT orders.order_id, customers.name, orders.order_date
FROM orders
INNER JOIN customers
ON orders.customer_id = customers.customer_id;
```

Trong ví dụ này, `orders` và `customers` là tên của các bảng cần kết hợp. `ON orders.customer_id = customers.customer_id` là điều kiện cần thiết để kết hợp các bảng này dựa trên cột `customer_id`. Các cột được chọn để hiển thị trong kết quả là `order_id`, `name` và `order_date`.

## LEFT JOIN

Giả sử bạn có hai bảng: `orders` và `customers`. Bảng `orders` chứa thông tin về các đơn hàng, bao gồm `order_id`, `customer_id` và `order_date`. Bảng `customers` chứa thông tin về khách hàng, bao gồm `customer_id`, `name` và `email`.

Để kết hợp hai bảng này để hiển thị thông tin về tên khách hàng và ngày đặt hàng cho mỗi đơn hàng, kể cả các đơn hàng không có khách hàng liên quan, bạn có thể sử dụng câu lệnh SQL sau:

```
SELECT orders.order_id, customers.name, orders.order_date
FROM orders
LEFT JOIN customers
ON orders.customer_id = customers.customer_id;
```

Trong ví dụ này, `orders` và `customers` là tên của các bảng cần kết hợp. `ON orders.customer_id = customers.customer_id` là điều kiện cần thiết để kết hợp các bảng này dựa trên cột `customer_id`. `LEFT JOIN` đảm bảo rằng các bản ghi từ bảng `orders` sẽ được trả về ngay cả khi không có bản ghi khớp trong bảng `customers`. Các cột được chọn để hiển thị trong kết quả là `order_id`, `name` và `order_date`.

## RIGHT JOIN

Giả sử bạn có hai bảng: `orders` và `customers`. Bảng `orders` chứa thông tin về các đơn hàng, bao gồm `order_id`, `customer_id` và `order_date`. Bảng `customers` chứa thông tin về khách hàng, bao gồm `customer_id`, `name` và `email`.

Để kết hợp hai bảng này để hiển thị thông tin về tên khách hàng và ngày đặt hàng cho mỗi đơn hàng, kể cả các đơn hàng không có khách hàng liên quan, bạn có thể sử dụng câu lệnh SQL sau:

```
SELECT orders.order_id, customers.name, orders.order_date
FROM orders
RIGHT JOIN customers
ON orders.customer_id = customers.customer_id;
```

Trong ví dụ này, `orders` và `customers` là tên của các bảng cần kết hợp. `ON orders.customer_id = customers.customer_id` là điều kiện cần thiết để kết hợp các bảng này dựa trên cột `customer_id`. `RIGHT JOIN` đảm bảo rằng tất cả các bản ghi từ bảng `customers` sẽ được trả về, kể cả khi không có bản ghi khớp trong bảng `orders`. Các cột được chọn để hiển thị trong kết quả là `order_id`, `name` và `order_date`.

## FULL OUTER JOIN

Giả sử bạn có hai bảng: `orders` và `customers`. Bảng `orders` chứa thông tin về các đơn hàng, bao gồm `order_id`, `customer_id` và `order_date`. Bảng `customers` chứa thông tin về khách hàng, bao gồm `customer_id`, `name` và `email`.

Để kết hợp hai bảng này để hiển thị thông tin về tên khách hàng và ngày đặt hàng cho mỗi đơn hàng, kể cả các đơn hàng không có khách hàng liên quan, bạn có thể sử dụng câu lệnh SQL sau:

```
SELECT orders.order_id, customers.name, orders.order_date
FROM orders
FULL OUTER JOIN customers
ON orders.customer_id = customers.customer_id;
```

Trong ví dụ này, `orders` và `customers` là tên của các bảng cần kết hợp. `ON orders.customer_id = customers.customer_id` là điều kiện cần thiết để kết hợp các bảng này dựa trên cột `customer_id`. `FULL OUTER JOIN` đảm bảo rằng tất cả các bản ghi từ cả hai bảng sẽ được trả về, kể cả các bản ghi không khớp. Các cột được chọn để hiển thị trong kết quả là `order_id`, `name` và `order_date`.

---

Revision #1

Created 4 July 2023 09:42:30 by Laptrinh.vn

Updated 4 July 2023 09:47:27 by Laptrinh.vn