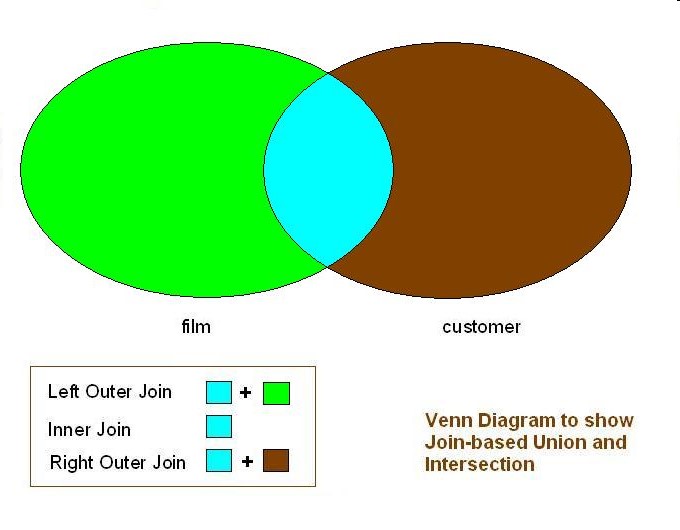
# Bài thực hành số 7

# Các phép nối bảng dữ liệu

 **Nội dung chính:** *Trong các bài thực hành trước, các truy vấn được thực hiện trên một bảng dữ liệu. Không ngạc nhiên khi rất nhiều truy vấn yêu cầu thông tin từ nhiều bảng dữ liệu khác nhau. Ví dụ muốn đưa ra thông tin khách hàng của các đơn hàng, cần kết hợp thông tin từ hai bảng dữ liệu là customers và orders. Kết hợp các bảng dữ liệu để tạo ra một bảng suy diễn được gọi là phép nối (join). Trong bài này, chúng ta sẽ làm quen với phép toán nối để truy vấn dữ liệu từ nhiều bảng: INNER JOIN, LEFT JOIN, SELF JOIN*



## 1. PHÉP NỐI TRONG (INNER JOIN)

INNER JOIN hay c n gọi là phép nối trong, là một phần tùy chọn của câu lệnh SELECT. Nó xuất hiện liền ngay sau mệnh đề FROM. Trước khi sử dụng INNER JOIN, phải xác định rõ các tiêu chí sau đây:

* Trước tiên, cần phải xác định các bảng mà muốn liên kết với bảng chính. Bảng chính xuất hiện trong mệnh đề FROM. Bảng muốn nối với bảng chính phải xuất hiện sau từ khóa INNER JOIN. Về mặt lý thuyết, có thể nối một bảng với số lượng không giới hạn các bảng khác, tuy nhiên, để có hiệu suất tốt hơn, nên hạn chế số lượng bảng tham gia phép nối dựa trên các điều kiện nối và khối lượng dữ liệu trong các bảng.
* Thứ hai, cần phải xác định điều kiện nối. Điều kiện nối xuất hiện sau từ khóa ON. Điều kiện nối chính là nguyên tắc để tìm được các bản ghi phù hợp trong các bảng và nối chúng lại với nhau.

Cú pháp INNER JOIN như sau:

SELECT column\_list

FROM table1

INNER JOIN table2 ON join\_condition1

INNER JOIN table3 ON join\_condition2

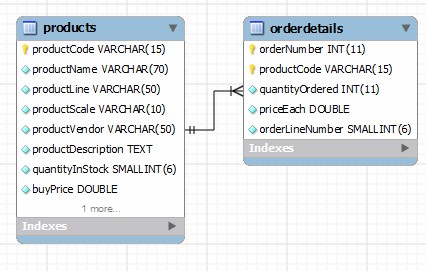
...

WHERE WHERE\_conditions;

Ví dụ, nếu nối hai bảng A và B, INNER JOIN so sánh mỗi bản ghi của bảng A với mỗi bản ghi của bảng B để tìm tất cả các cặp bản ghi đáp ứng được điều kiện nối. Khi điều kiện nối được thoả mãn, giá trị cột cho mỗi cặp bản ghi phù hợp của bảng A và bảng B được kết hợp thành một bản ghi trong kết quả trả về.

Hạn chế sự trùng tên cột khi sử dụng INNER JOIN: Nếu nối nhiều bảng có cột với tên tương tự, phải chỉ rõ tên bảng có chứa cột dữ liệu định lấy để tránh lỗi cột không rõ ràng. Giả sử nếu bảng *tbl\_A* và *tbl\_B* có các cột tương tự *M* . Trong câu lệnh SELECT với INNER JOIN, phải tham chiếu tới cột *M* bằng cách sử dụng cú pháp như *tbl\_A.M* .

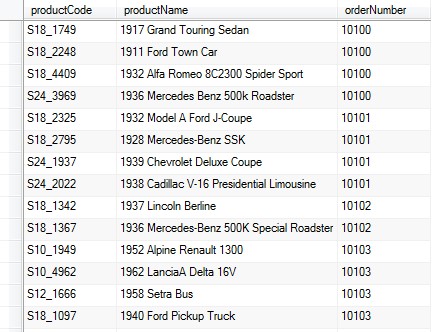
**Ví dụ**: Hãy xem xét hai bảng products và orderDetails. Bảng products là bảng dữ liệu tổng thể lưu trữ tất cả các sản phẩm. Bất cứ khi nào một sản phẩm được bán ra, nó được lưu trữ trong bảng OrderDetails cùng với các thông tin khác. Liên kết giữa các bảng này là cột productCode



**Ví dụ**: muốn biết những sản phẩm đã được bán, có thể sử dụng *INNER JOIN* như sau:

SELECT products.productCode, products.productName, orderDetails.orderNumber FROM products

INNER JOIN orderDetails on products.productCode = orderDetails.productCode;



INNER JOIN so sánh từng d ng trong bảng products và OrderDetails để tìm một cặp bản ghi có cùng productCode. Nếu một cặp bản ghi có cùng mã sản phẩm, khi đó tên sản phẩm và số thứ tự cũng sẽ được kết hợp thành một hàng để trả lại kết quả.

*Bí danh (Alias)*: có thể tạo bí danh của bảng *tbl\_A* là A và tham chiếu đến cột *M* là *A.M* , như vậy không mất công gõ lại tên bảng nữa. Ví dụ trên có thể viết lại như sau:

SELECT p.productCode, p.productName, o.orderNumber

FROM products p

INNER JOIN orderDetails o on p.productCode = o.productCode;

*Lưu ý:* Bên cạnh phép nối trong sử dụng mệnh đề INNER JOIN .. ON, có thể nối trong hai bảng bằng cách đưa điều kiện nối vào mệnh đề WHERE. Ví dụ trên có thể viết lại như sau:

SELECT p.productCode, p.productName, o.orderNumber

FROM products p, orderDetails o

WHERE p.productCode = o.productCode;

Chúng ta sẽ xem xét một số ví dụ khác sử dụng phép nối dưới đây:

**Ví dụ**: Bảng Employees là bảng lưu giữ thông tin về các nhân viên của công ty; bảng Customers là bảng lưu giữ thông tin của các khách hàng, trong đó có thông tin liên quan đến mã số của nhân viên chăm sóc khách hàng. Như vậy liên kết giữa hai bảng này được thực hiện thông qua cột employeeNumber của bảng Employees và cột salesRep employeeNumber của bảng Customers.

Để biết thông tin về khách hàng và tên nhân viên chăm sóc khách hàng đó, có thể viết truy vấn sử dụng INNER JOIN như sau:

SELECT customerName, firstname as EmployeeName FROM customers C join employees E on C.salesrepemployeenumber = e.employeenumber

Kết quả trả về như sau:



**Ví dụ:** Đưa ra thông tin về các d ng sản phẩm và tổng số hàng có trong d ng sản phẩm đó.

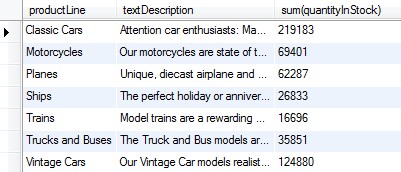
SELECT pl.productLine, pl.textDescription, sum(quantityInStock)

FROM productlines pl JOIN products p ON pl.productLine

=p.productLine

GROUP by pl.productLine;

Kết quả trả về như sau:



**Ví dụ:** Đưa ra thông tin về các sản phẩm và tổng giá trị đã đặt hàng cho sản phẩm

SELECT P.productCode,

P.productName,

SUM(priceEach \* quantityOrdered) total

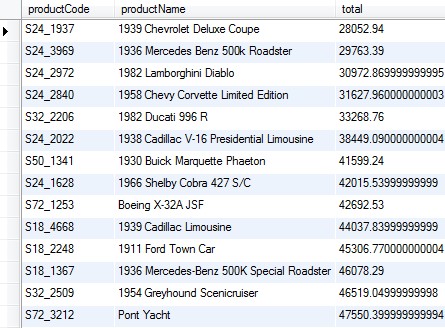
FROM orderdetails O

INNER JOIN products P ON O.productCode = P.productCode

GROUP by productCode

ORDER BY total

Kết quả trả về như sau:



Bên cạnh phép nối hai bảng dữ liệu, ta có thể nối nhiều bảng dữ liệu trong cùng một câu lệnh SELECT.

**Ví dụ**: Đưa ra tên các khách hàng và tổng giá trị các đơn hàng của các khách hàng đó.

SELECT C.customerName, sum(OD.priceEach\*OD.quantityOrdered) as total

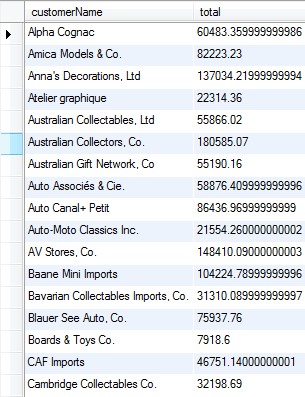
FROM customers C

INNER JOIN orders O on C.customerNumber = O.customerNumber

INNER JOIN orderdetails OD on O.orderNumber =

OD.orderNumber GROUP BY C.customerName

Như trong ví dụ trên thông tin cần kết hợp từ ba bảng dữ liệu là customers, orders và orderdetails.



**Ví dụ**: Đưa ra các đơn hàng, tên các khách hàng và tổng giá trị của đơn hàng đó.

SELECT O.orderNumber,C.customerName, sum(OD.priceEach\*OD.quantityOrdered) as total

FROM customers C

INNER JOIN orders O on C.customerNumber = O.customerNumber INNER JOIN orderdetails OD on O.orderNumber =

OD.orderNumber

GROUP BY O.orderNumber;



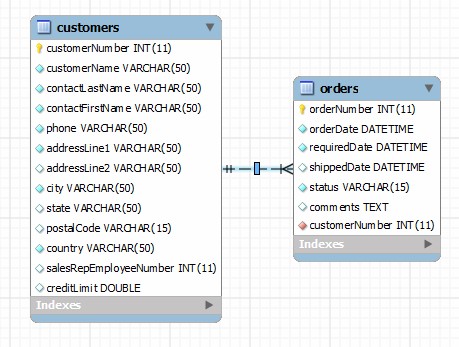
## 2. PHÉP NỐI TRÁI (LEFT JOIN)

LEFT JOIN cũng là một tùy chọn của câu lệnh SELECT cho phép lấy thêm dữ liệu từ các bảng khác. LEFT JOIN bao gồm các từ khóa LEFT JOIN, tiếp theo là bảng thứ hai muốn thực hiện nối. Yếu tố tiếp theo là từ khóa ON và theo sau bởi các điều kiện nối.

Mệnh đề LEFT JOIN sẽ được thực hiện như sau: khi một hàng từ bảng bên trái phù hợp với một hàng từ bảng bên phải dựa trên điều kiện nối, nội dung của hàng đó sẽ được lựa chọn như một d ng trong kết quả đầu ra. Khi một hàng trong bảng bên trái không tìm được hàng nào phù hợp trong bảng nối, nó vẫn được xuất hiện trong kết quả đầu ra, nhưng kết hợp với một hàng "giả" từ bảng bên phải với giá trị NULL cho tất cả các cột. Tóm lại, LEFT JOIN cho phép chọn tất cả các hàng từ bảng bên trái ngay cả khi không có bản ghi nào phù hợp với nó trong bảng bên phải.

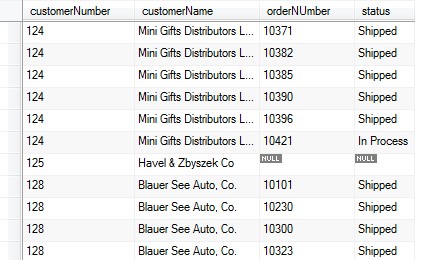
**Ví dụ:** sử dụng LEFT JOIN

Chúng ta hãy xét vào hai bảng customers và orders. Nếu muốn biết một khách hàng với hoá đơn nào đó của họ và tình trạng hoá đơn đó thế nào, có thể sử dụng MySQL LEFT JOIN như sau:



SELECT c.customerNumber, customerName,orderNUmber, o.status FROM customers c

LEFT JOIN orders o ON c.customerNumber = o.customerNumber;



Ở bảng kết quả trên, có thể nhìn thấy tất cả các khách hàng được liệt kê. Tuy nhiên, có những bản ghi có thông tin khách hàng nhưng tất cả các thông tin về đơn hàng là NULL. Điều này có nghĩa là những khách hàng này không có bất kỳ một đơn đặt hàng nào được lưu trong cơ sở dữ liệu của chúng ta.

LEFT JOIN rất hữu ích khi muốn tìm các bản ghi trong bảng bên trái mà không phù hợp với bất kỳ một bản ghi nào trong bảng bên phải. có thể thực hiện điều này bằng cách thêm một mệnh đề WHERE để lựa chọn các hàng chỉ có giá trị NULL trong một cột ở bảng bên phải . Vì vậy, để tìm thấy tất cả các khách hàng không có bất kỳ đơn đặt hàng nào trong cơ sở dữ liệu của chúng ta, có thể sử dụng LEFT JOIN như sau:

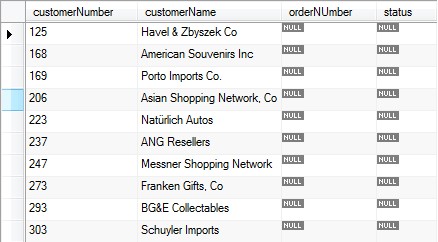
SELECT c.customerNumber, customerName,orderNUmber, o.status

FROM customers c

LEFT JOIN orders o ON c.customerNumber = o.customerNumber

WHERE orderNumber is NULL

Kết quả trả về như sau:



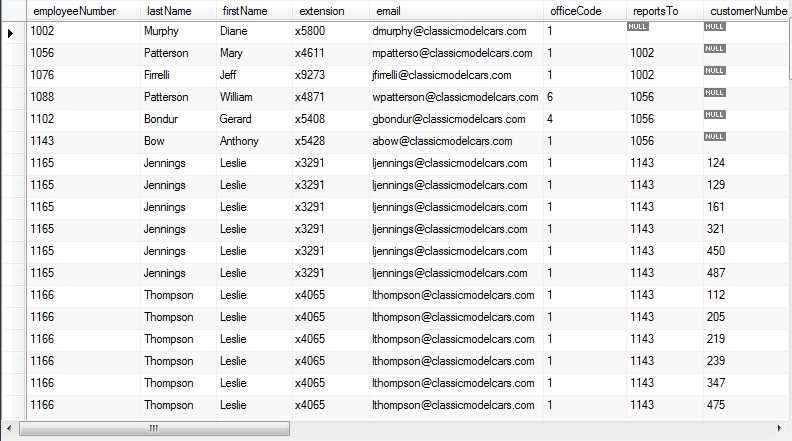
Như vậy, truy vấn chỉ trả về các khách hàng mà không có bất kỳ đơn hàng nào nhờ vào các giá trị NULL.

Tương tự như vậy, để tìm ra những nhân viên không làm nhiệm vụ chăm sóc khách hàng, bước đầu, thực hiện truy vấn như sau:

Select \* from employees e

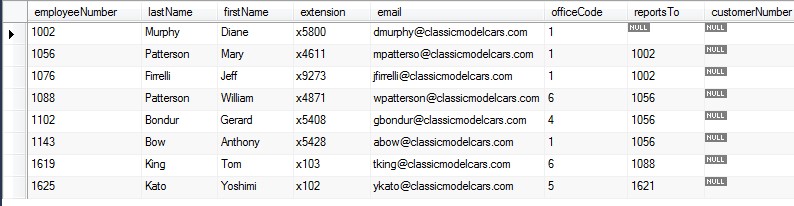
left join customers c

on e.employeenumber=c.salesrepemployeenumber



Sau đó lọc ra những bản ghi nhận giá trị null tại cột customerNumber, đó chính là kết quả của truy vấn.

Select \* from employees e left join customers c on e.employeenumber=c.salesrepemployeenumber where customerNumber is null



## 3. PHÉP TỰ NỐI (Self Join)

Một phép tự nối là một kiểu nối trong đó một bảng được nối với chính nó, cụ thể khi một bảng có một khóa ngoài tham chiếu tới khóa chính của nó.

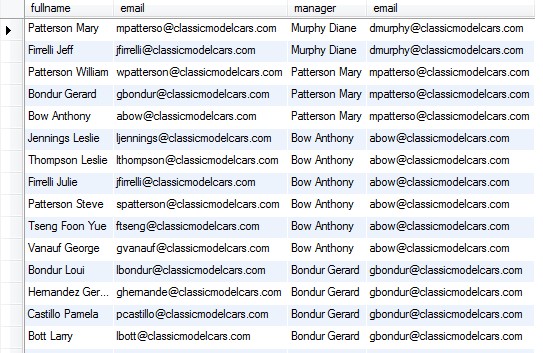
Ví dụ: Bảng *employees* có một khóa ngoài là *reportsTo* tham chiếu tới khóa chính *employeeNumber* của chính bảng *employees.*

Cần thiết phải sử dụng bí danh cho mỗi bản sao của bảng đó để tránh nhập nhằng

SELECT concat (e1.lastName ," ",e1.firstName) as fullname, e1.email, concat (e2.lastName ," ",e2.firstName) as manager, e2.email

FROM employees e1, employees e2 WHERE e1.reportsTo = e2.employeeNumber;

Kết quả trả về như sau:



##  Bài tập thực hành:

1. Đưa ra thông tin về các nhân viên và tên văn ph ng nơi họ làm việc.
2. Đưa ra thông tin về các mặt hàng chưa có ai đặt mua.
3. Đưa ra thông tin về các đơn hàng trong tháng 3/2003 (gồm orderDate, requiredDate, Status) và tổng giá trị của đơn hàng.
4. Đưa ra thông tin về các d ng sản phẩm và số lượng sản phẩm của d ng sản phẩm đó. Sắp xếp theo thứ tự số lượng giảm dần.
5. Đưa ra tên khách hàng và số tiền hàng họ đã mua