



Bài 10

Câu lệnh truy vấn SQL

Module: BOOTCAMP WEB-BACKEND DEVELOPMENT

Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài “Thiết kế và tạo cơ sở dữ liệu”
Tóm tắt lại các phần đã học từ bài “Thiết kế và tạo cơ sở dữ liệu”



Mục tiêu

- Trình bày được cú pháp sử dụng các hàm trong SQL
- Trình bày được cú pháp sử dụng câu lệnh GROUP BY và HAVING
- Sử dụng được câu lệnh WHERE
- Sử dụng được câu lệnh JOIN
- Sử dụng được toán tử AND, OR và NOT
- Sử dụng được các hàm tập hợp, làm việc với chuỗi cơ bản
- Sử dụng được câu lệnh GROUP BY, HAVING
- Luyện tập tạo khoá chính trong bảng
- Luyện tập thiết kế cơ sở dữ liệu
- Luyện tập triển khai khoá ngoại

Câu lệnh WHERE



- Cú pháp:

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

- Ví dụ:

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

Các toán tử trong câu lệnh WHERE



Toán tử	Mô tả
=	So sánh bằng
<>	Khác nhau
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
BETWEEN	Nằm trong khoảng (bao gồm cả 2 giá trị biên)
LIKE	So sánh theo mẫu (pattern)
IN	So sánh theo một danh sách các giá trị

Toán tử AND

- Toán tử AND được sử dụng để quy định trả về đúng nếu 2 điều kiện ở hai vế đều trả về giá trị TRUE
- Cú pháp:

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

- Ví dụ:

```
SELECT * FROM customers  
WHERE country='Germany' AND city='Berlin';
```

Toán tử OR



- Toán tử OR được sử dụng để quy định trả về đúng nếu 1 trong 2 điều kiện ở hai vế trả về giá trị TRUE
- Cú pháp:

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

- Ví dụ:

```
SELECT * FROM customers  
WHERE city='Berlin' OR city='München';
```



Toán tử NOT

- Toán tử NOT được sử dụng để quy định trả về đúng nếu giá trị ở về sau là FALSE
- Cú pháp:

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

- Ví dụ:

```
SELECT * FROM customers  
WHERE NOT country='Germany';
```


Kết hợp AND, OR và NOT



- Ví dụ kết hợp AND và OR:

```
SELECT * FROM customers  
WHERE country='Germany' AND (city='Berlin' OR city='München');
```

- Ví dụ kết hợp AND và NOT:

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



Câu lệnh JOIN

- Câu lệnh JOIN được sử dụng để truy vấn dữ liệu kết hợp từ nhiều bảng.
- Chẳng hạn:
 - Bảng orders: id, customer_id, orderDate
 - Bảng customers: id, name, country
- Câu lệnh truy vấn sau sẽ trả về danh sách các order cùng với tên khách hàng tương ứng:

```
SELECT orders.id, customers.name, orders.orderDate  
FROM orders  
JOIN customers ON orders.customer_id=customers.id;
```



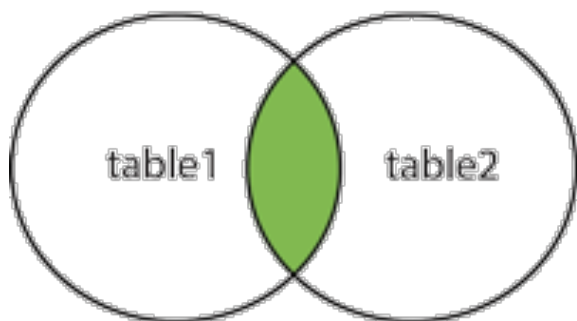
Các loại câu lệnh JOIN (1)

- **INNER JOIN** (hoặc **JOIN**): Chỉ trả về các record có mặt ở cả 2 bảng
- **LEFT JOIN**: Trả về tất cả các record có mặt ở bảng bên trái, và những record tương ứng ở bảng bên phải
- **RIGHT JOIN**: Trả về tất cả các record có mặt ở bảng bên phải, và những record tương ứng ở bảng bên trái
- **FULL JOIN**: Trả về tất cả các record ở cả hai bảng

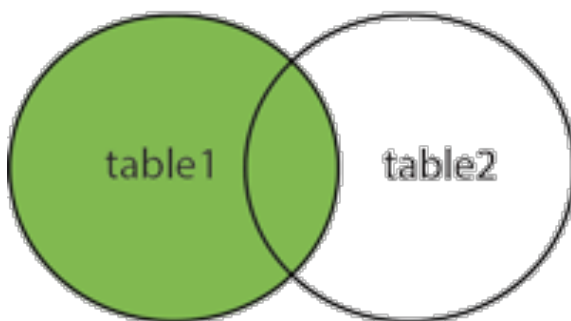
Các loại câu lệnh JOIN (2)



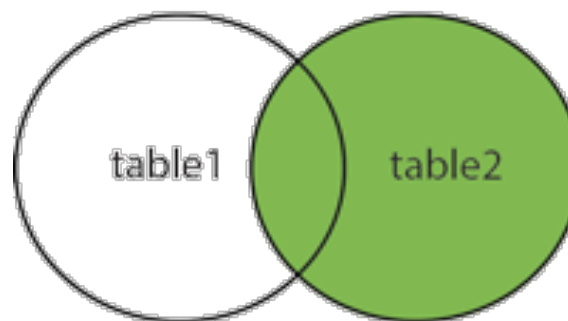
INNER JOIN



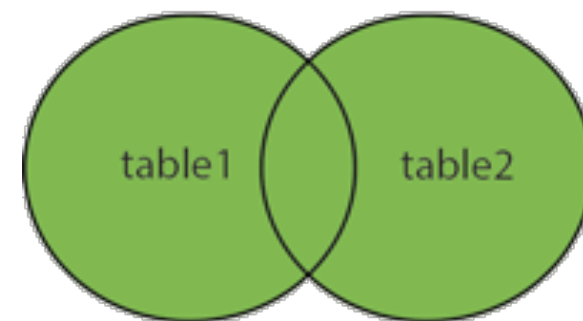
LEFT JOIN



RIGHT JOIN



FULL OUTER JOIN

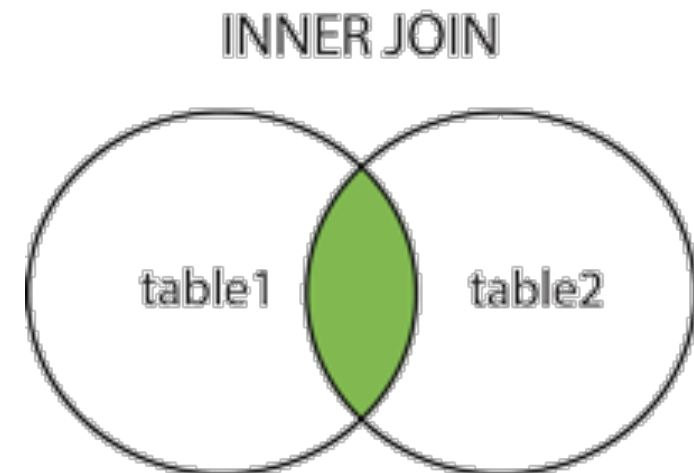


Inner Join



- Inner join cho phép chọn những bản ghi phù hợp ở cả 2 bảng

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

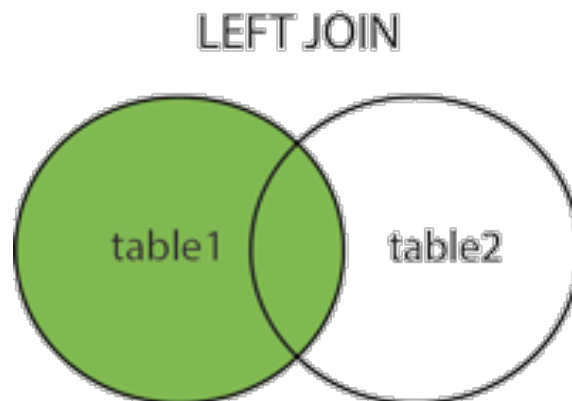


Left Join



- Left Join cho phép lựa chọn tất cả các bản ghi từ bảng bên trái (table 1) và những bản ghi phù hợp từ bảng bên phải (table 2)

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

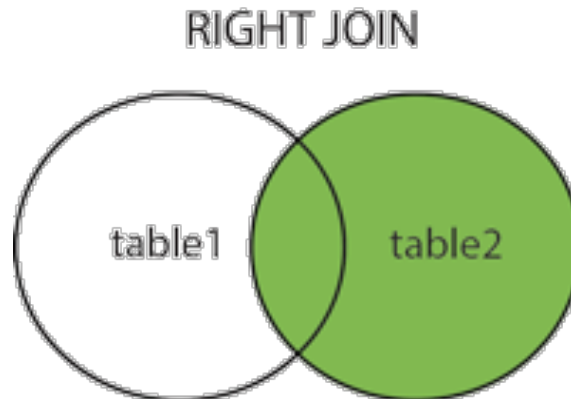


Right Join



- Right Join cho phép lựa chọn những bản ghi phù hợp từ bảng bên trái (table 1) và tất cả các bản ghi từ bảng bên phải (table 2)

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

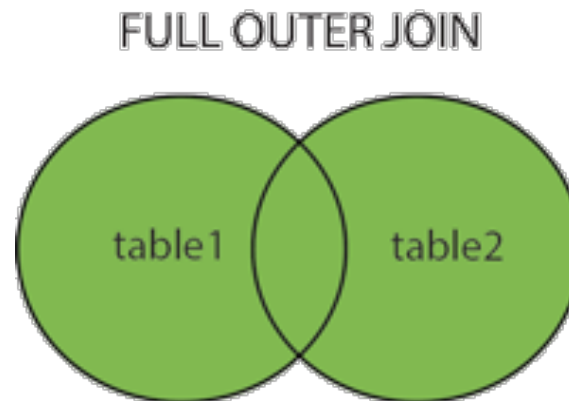


Full Outer Join



- Full Outer Join cho phép tất cả các bản ghi từ cả hai bảng bên trái và bên phải

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





JOIN nhiều hơn 2 bảng

- Có thể JOIN nối tiếp giữa nhiều bảng
- Ví dụ:

```
SELECT orders.id, customers.name, shippers.name AS shipper
FROM (orders
INNER JOIN customers ON orders.customer_id = customers.id)
INNER JOIN shippers ON orders.shipper_id = shippers.id);
```

- **Lưu ý:** Từ khoá **AS** được sử dụng để đổi tên trường khi truy vấn.

Các hàm SQL



- SQL cung cấp sẵn một số hàm để thao tác với dữ liệu chuỗi, số, ngày tháng
- Chẳng hạn:
 - SUM(), AVG(), COUNT(), MIN(), MAX(), UCASE(), LCASE()...

Hàm SUM()



- Hàm SUM() được sử dụng để tính giá trị tổng của một cột chứa giá trị số
- Cú pháp:

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

- Ví dụ:

```
SELECT SUM(quantity)  
FROM orderDetails;
```

Hàm AVG()



- Hàm AVG() tính giá trị trung bình của một cột chứa giá trị số
- Cú pháp:

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

- Ví dụ:

```
SELECT AVG(price)
FROM products;
```

Hàm COUNT()



- Hàm COUNT() được sử dụng để đếm số bản ghi
- Cú pháp:

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

- Ví dụ:

```
SELECT COUNT(id)  
FROM products;
```

Hoặc

```
SELECT COUNT(*)  
FROM products;
```

Hàm MAX()



- Hàm MAX() trả về giá trị lớn nhất của một cột
- Cú pháp:

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

- Ví dụ:

```
SELECT MAX(price) AS largestPrice
FROM products;
```

Hàm MIN()



- Hàm MIN() trả về giá trị nhỏ nhất của một cột
- Cú pháp:

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

- Ví dụ:

```
SELECT MIN(price) AS smallestPrice  
FROM products;
```

Hàm UCASE()



- Hàm UCASE() được sử dụng để chuyển một chuỗi thành chữ viết hoa (uppercase)
- Cú pháp:

```
SELECT UCASE(column_name) FROM table_name;
```

- Ví dụ:

```
SELECT UCASE(name) AS customer  
FROM customers;
```




Hàm LCASE()

- Hàm LCASE() được sử dụng để chuyển một chuỗi thành chữ viết thường (lowercase)
- Cú pháp:

```
SELECT LCASE(column_name) FROM table_name;
```

- Ví dụ:

```
SELECT LCASE(name) AS customer, city  
FROM customers;
```

Hàm LEN()

- Hàm LEN() trả về độ dài (số lượng ký tự) của một chuỗi
- Cú pháp:

```
SELECT LEN(column_name) FROM table_name;
```

- Ví dụ:

```
SELECT name, LEN(address) as lengthOfAddress  
FROM customers;
```

Hàm NOW()



- Hàm NOW() trả về ngày giờ hiện tại của hệ thống
- Cú pháp:

```
SELECT NOW() FROM table_name;
```

- Ví dụ:

```
SELECT name, price, Now() AS perDate  
FROM products;
```

Câu lệnh ORDER BY

- Câu lệnh ORDER BY sắp xếp các bản ghi theo trật tự dựa vào giá trị của một cột hoặc nhiều cột
- Cú pháp:

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

- Trong đó:
 - **ASC**: Trật tự tăng dần
 - **DESC**: Trật tự giảm dần



Câu lệnh ORDER BY: Ví dụ

- Sắp xếp các khách hàng theo trật tự tên A-Z:

```
SELECT * FROM customers  
ORDER BY name;
```

- Sắp xếp các khách hàng theo trật tự tên Z-A:

```
SELECT * FROM customers  
ORDER BY name DESC;
```

Câu lệnh GROUP BY

- Câu lệnh GROUP BY được dùng để nhóm các tập kết quả dựa theo giá trị của một cột hoặc nhiều cột
- Câu lệnh GROUP BY thường được dùng chung với các hàm khác của SQL như: COUNT(), MIN(), MAX(), SUM(), AVG()
- Cú pháp:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
```

Câu lệnh GROUP BY: Ví dụ



- Số lượng khách hàng thuộc từng quốc gia:

```
SELECT COUNT(id), country  
FROM customers  
GROUP BY country;
```

- Hoặc:

```
SELECT COUNT(id) AS customerCount, country  
FROM customers  
GROUP BY country  
ORDER BY customerCount DESC;
```

Câu lệnh HAVING

- Câu lệnh HAVING được sử dụng để quy định các điều kiện trong trường hợp sử dụng các hàm SQL (không thể sử dụng câu lệnh WHERE)
- Cú pháp:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```




Câu lệnh HAVING: Ví dụ

- Chỉ liệt kê các quốc gia có nhiều hơn 5 khách hàng:

```
SELECT COUNT(id), country
FROM customers
GROUP BY country
HAVING COUNT(id) > 5;
```

- Hoặc:

```
SELECT COUNT(id) AS customerCount, country
FROM customers
GROUP BY country
HAVING customerCount > 5;
```

- Lệnh WHERE là một tùy chọn, để hạn chế dữ liệu được xử lý trong câu lệnh SQL
- Lệnh JOIN để kết nối hai hay nhiều bảng với nhau
- Toán tử AND, OR và NOT sử dụng trong WHERE để kết hợp các biểu thức
- Mỗi bảng chỉ có duy nhất một khoá chính.
- Khoá ngoại là khoá chính (hoặc trường được xác định với ràng buộc unique) của một bảng khác.



Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập

Chuẩn bị bài tiếp theo: *Tổng quan về Ứng dụng Web*