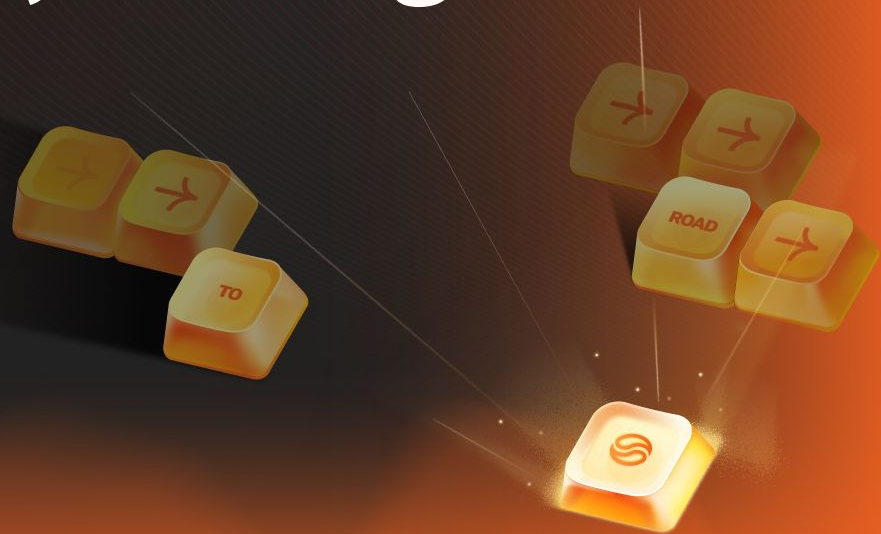


Join, Order, Group, Having

Nguyễn Anh Tuấn

KTECH
COLLEGE

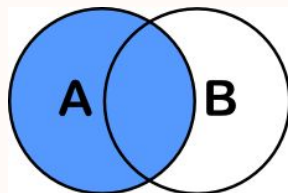


Nội dung bài giảng

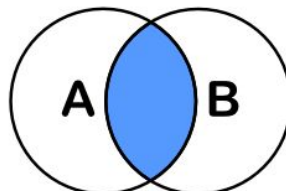
- 1 Ghép nối nhiều bảng
- 2 Sắp xếp kết quả truy vấn
- 3 Nhóm kết quả truy vấn
- 4 Nhóm kết quả truy vấn có điều kiện

Ghép nối nhiều bảng

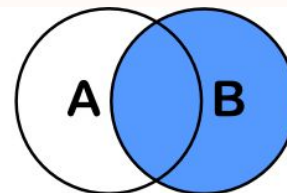
Ghép nối nhiều bảng - JOIN



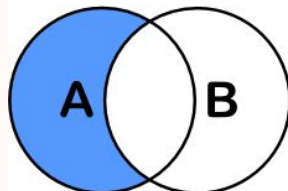
```
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
```



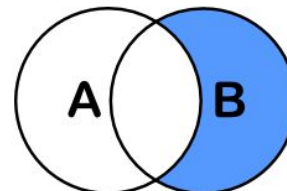
```
SELECT <auswahl>
FROM tabelleA A
INNER JOIN tabelleB B
ON A.key = B.key
```



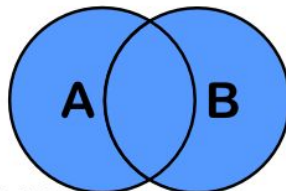
```
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
```



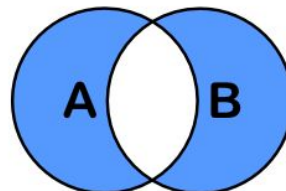
```
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
WHERE B.key IS NULL
```



```
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
```



```
SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
```



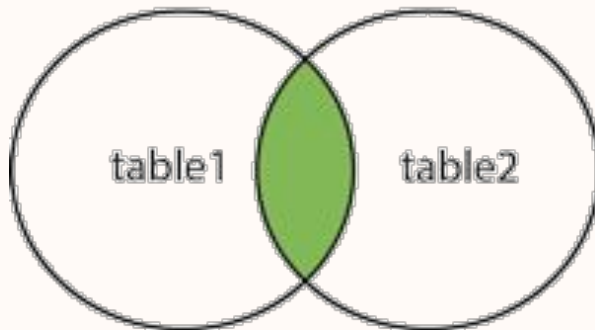
```
SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```

Ghép nối nhiều bảng - JOIN

- Các phép nối (join) được sử dụng để kết hợp dữ liệu từ nhiều bảng để tạo thành một tập kết quả duy nhất.
- Oracle cung cấp hai cách tiếp cận để ghép nối các bảng:
 - Cú pháp nối **non-ANSI**:
 - Sử dụng mệnh đề **FROM** và điều kiện **WHERE**
`SELECT * FROM table1, table2 WHERE table1.id = table2.id;`
 - Cú pháp nối **ANSI**:
 - Sử dụng từ khoá **JOIN**
`SELECT * FROM table1 JOIN table2 ON table1.id = table2.id;`

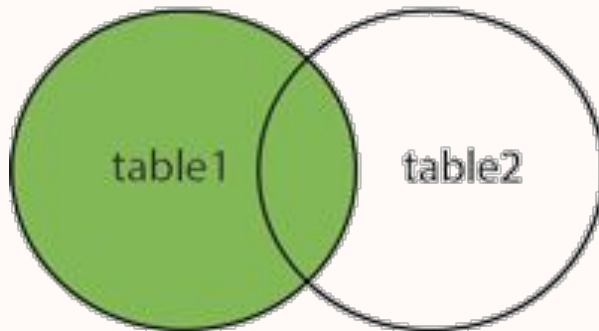
[INNER] JOIN ... ON

- Một dữ liệu **INNER JOIN** kết hợp từ hai bảng trong đó có một kết quả khớp trên (các) cột tham gia trong cả hai bảng.
 - Cú pháp nối **non-ANSI**:
 - Sử dụng mệnh đề **FROM** và điều kiện **WHERE**
`SELECT * FROM table1, table2 WHERE table1.id = table2.id;`
 - Cú pháp nối **ANSI**:
 - Sử dụng từ khoá **JOIN**
`SELECT * FROM table1 JOIN table2 ON table1.id = table2.id;`



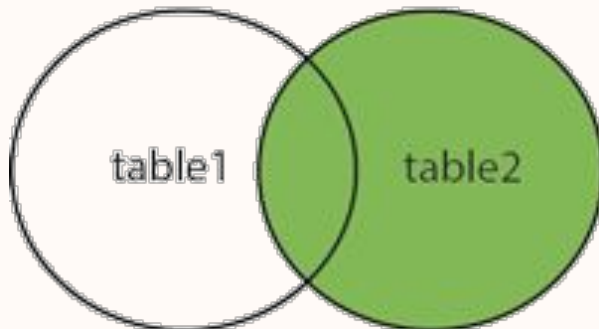
LEFT [OUTER] JOIN

- Một dữ liệu **LEFT OUTER JOIN** trả về tất cả các hàng hợp lệ từ bảng ở phía bên trái của từ khóa JOIN, cùng với các giá trị từ bảng ở phía bên phải hoặc NULL nếu một hàng phù hợp không tồn tại.
 - Cú pháp nối **non-ANSI**:
 - Sử dụng mệnh đề **FROM** và điều kiện **WHERE**
SELECT * **FROM** table1, table2 **WHERE** table1.id = table2.id (+);
 - Cú pháp nối **ANSI**:
 - Sử dụng từ khoá **JOIN**
SELECT * FROM table1 **LEFT JOIN** table2 **ON** table1.id = table2.id;



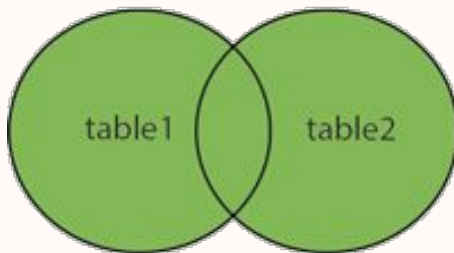
RIGHT [OUTER] JOIN

- Một dữ liệu **RIGHT OUTER JOIN** trả về tất cả các hàng hợp lệ từ bảng ở phía bên phải của từ khóa JOIN, cùng với các giá trị từ bảng ở phía bên trái hoặc NULL nếu một hàng phù hợp không tồn tại.
 - Cú pháp nối **non-ANSI**:
 - Sử dụng mệnh đề **FROM** và điều kiện **WHERE**
 - Cú pháp nối **ANSI**:
 - Sử dụng từ khoá **JOIN**
`SELECT * FROM table1 RIGHT JOIN table2 ON table1.id = table2.id;`



FULL [OUTER] JOIN

- Một dữ liệu **FULL OUTER JOIN** kết hợp tất cả các hàng từ các bảng ở bên trái và bên phải của phép join. Nếu một trong hai bên bị thiếu dữ liệu, nó sẽ được thay thế bằng NULL, thay vì bỏ hàng đi.
 - Cú pháp nối **non-ANSI**:
 - Sử dụng mệnh đề **FROM**, điều kiện **WHERE** và **UNION ALL**
SELECT * **FROM** table1, table2 **WHERE** table1.id = table2.id (+);
UNION ALL
SELECT * **FROM** table1, table2 **WHERE** table2.id = table1.id (+);
 - Cú pháp nối **ANSI**:
 - Sử dụng từ khoá **JOIN**
SELECT * FROM table1 **FULL JOIN** table2 **ON** table1.id = table2.id;



Ghép nối nhiều bảng - JOIN

- Các phương pháp ghép nối hiếm gặp:
 - **CROSS JOIN:**
 - Giống như câu lệnh **SELECT * FROM** table1, table2; nhưng không có **WHERE**
 - **NATURAL JOIN**
 - **[INNER] JOIN ... USING**
 - **CROSS APPLY**
 - **OUTER APPLY**



Sắp xếp kết quả truy vấn

Sắp xếp kết quả truy vấn - ORDER BY

- **ORDER BY** sẽ sắp xếp dựa vào bảng mã ASCII, kí tự nào đứng đầu tiên sẽ được xem là nhỏ nhất, kí tự nằm cuối là lớn nhất.
- Cú pháp sắp xếp **ORDER BY**:

```
SELECT column_name  
FROM table_name  
WHERE condition  
ORDER BY column_name ASC | DESC;
```

// ASC là giá trị tăng dần từ nhỏ tới lớn

// DESC là giá trị giảm dần từ lớn tới nhỏ

Sắp xếp kết quả truy vấn - ORDER BY

EmployeeID	EmployeeLastName	EmployeeFirstName	EmailID
003	Jones	Amy	amy@gmail.com
006	Brown	Dan	dan@gmail.com
001	Donald	Jo	jo@gmail.com



SELECT *
FROM Employee
ORDER BY
EmployeeLastName;



Result

EmployeeID	EmployeeLastName	EmployeeFirstName	EmailID
006	Brown	Dan	dan@gmail.com
001	Donald	Jo	jo@gmail.com
003	Jones	Amy	amy@gmail.com

Sắp xếp kết quả truy vấn - ORDER BY

// Ví dụ sắp xếp tăng dần:

```
SELECT *  
FROM Employee  
ORDER BY EmployeeLastName ASC;
```

// Nếu sử dụng ORDER BY mà không có ASC | DESC, cơ sở dữ liệu sẽ tự hiểu là kiểu ASC

// Ví dụ sắp xếp giảm dần:

```
SELECT *  
FROM Employee  
ORDER BY EmployeeLastName DESC;
```

Nhóm kết quả truy vấn

Nhóm kết quả truy vấn - GROUP BY

- **GROUP BY** kết hợp với lệnh SELECT để lấy dữ liệu, sau đó gộp dữ liệu lại theo một hoặc nhiều cột.
- **GROUP BY** sẽ gộp dữ liệu trùng dựa vào columns được SELECT
GROUP BY phân biệt với DISTINCT

Name	Age	Location	ID
John	35	California	12698
Harry	24	Los Angeles	12699
Smith	32	California	12700
Gary	45	New Jersey	12701

SELECT Location
COUNT(*)AS number
FROM Details
GROUP BY Location;

Location	number
California	2
Los Angeles	1
New Jersey	1

Nhóm kết quả có điều kiện

Nhóm kết quả truy vấn có điều kiện - HAVING

- Lệnh **SELECT** có thể **WHERE** dùng để lọc dữ liệu thoả mãn điều kiện, ở **GROUP BY** cũng sử dụng lệnh **HAVING** giống với lệnh WHERE để lọc dữ liệu thoả mãn điều kiện.

// Cú pháp câu điều kiện HAVING

```
SELECT column_name  
FROM table_name  
WHERE condition  
GROUP BY column_name1, column_name2, ...  
HAVING having_condition;
```

// Ví dụ về HAVING

```
SELECT product, SUM(amount) AS total_sales  
FROM sales  
GROUP BY product  
HAVING SUM(amount) > 300;
```

Thứ tự thực hiện câu lệnh truy vấn

- Thứ tự thực hiện câu lệnh truy vấn:
 - **FROM > WHERE > GROUP BY > HAVING > DISTINCT > SELECT > ORDER BY**
 - **From:** Xác định bảng dữ liệu cần truy vấn.
 - **WHERE:** Lọc các bản ghi dựa trên điều kiện.
 - **GROUP BY:** Nhóm các bản ghi dựa trên một hoặc nhiều cột.
 - **HAVING:** Lọc các nhóm dữ liệu dựa trên điều kiện áp dụng cho nhóm.
 - **DISTINCT:** Loại bỏ các bản ghi trùng lặp.
 - **SELECT:** Chọn các cột hoặc biểu thức cần hiển thị.
 - **ORDER BY:** Sắp xếp kết quả.

ROAD TO KOREA

Nếu có bất kỳ thắc mắc nào, hãy đặt câu hỏi qua

mail@mail.com hoặc Zalo 0xxx xxx xxx