BÀI 4:

Truy vấn nâng cao

Module: Fundamental Database

Phiên bản: 1.0

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

NỘI DUNG

1. Tổng quan về Aliases - Tên tạm thời trong SQL
2. Tổng quan về các Aggregate Function trong SQL
3. Group by – Having
4. Nested Select - Truy vấn lồng trong SQL
5. Truy vấn dữ liệu trên nhiều bảng

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

1. Tổng quan về Aliases - Đặt tên tạm thời

ALIASES - Đặt tên tạm thời trong SQL

- ***Aliases*** *trong SQL được sử dụng để đặt cho bảng hoặc cột trong bảng một tên tạm thời.*
- *Aliases thường được sử dụng để làm cho tên cột dễ đọc hơn.*
- *Alias chỉ tồn tại trong suốt thời gian truy vấn đó.*
- *Alias được tạo ra với từ khóa* ***AS****.*

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

1. Tổng quan về Aliases - Đặt tên tạm thời

Aliases được sử dụng trên cột

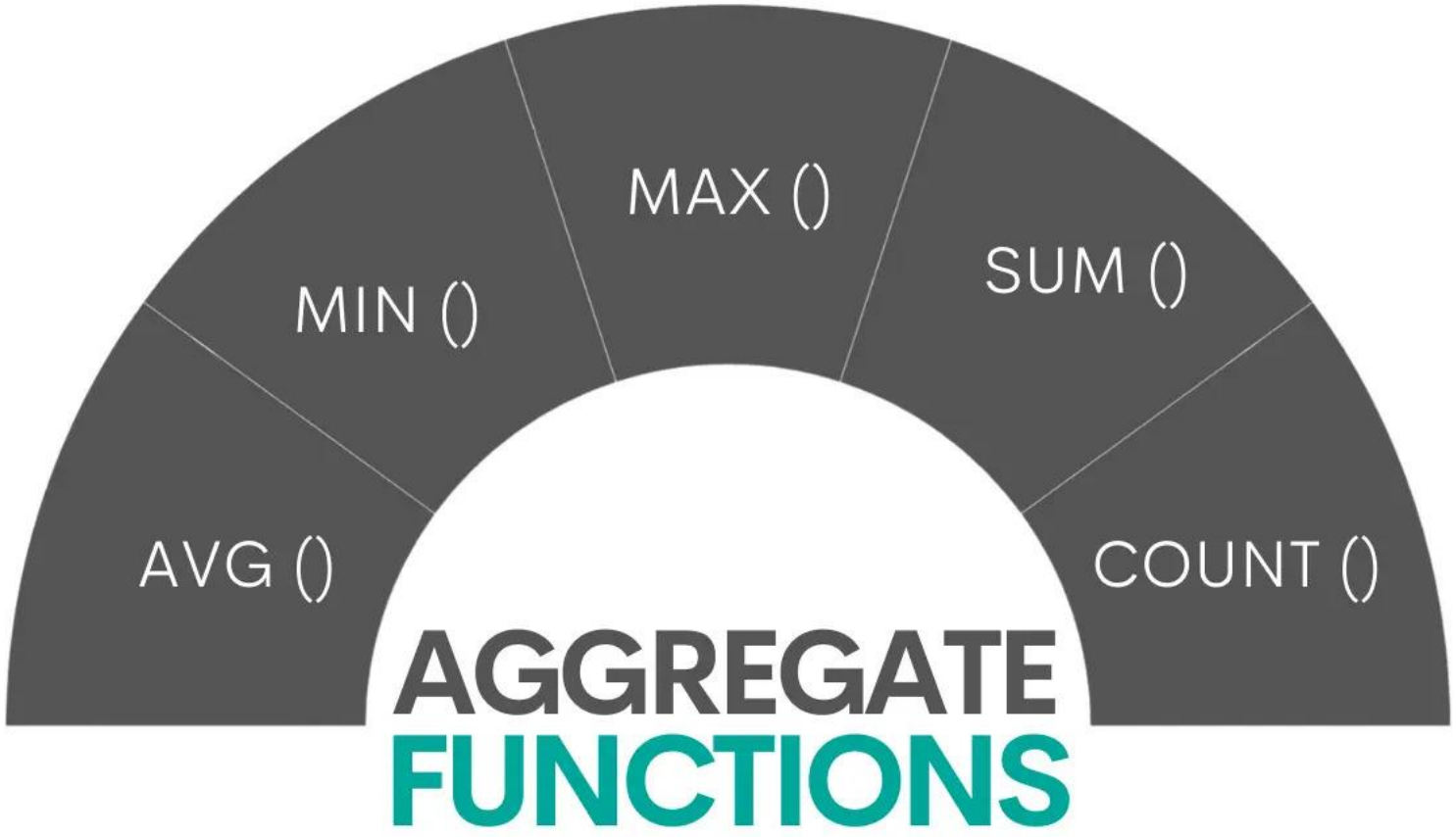
```
SELECT column_name AS alias_name
FROM table_name;
```

Aliases được sử dụng trên bảng

```
SELECT column_name(s)
FROM table_name AS alias_name;
```

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

2. Tổng quan về các Aggregate Function



© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

2. Tổng quan về các Aggregate Function

Aggregate Function

Data Aggregation *(Tổng hợp dữ liệu) là quá trình thu thập dữ liệu và trình bày nó dưới dạng tóm tắt (Tổng hợp), phù hợp cho việc phân tích và thống kê dữ liệu*

Aggregate Function là các hàm được xây dựng sẵn trong SQL giúp cho chúng ta có thể thực hiện các tác vụ tổng hợp dữ liệu

Các Aggregate Function thông dụng hay được sử dụng:
MAX(), MIN(), COUNT(), SUM(), AVG()

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

2. Tổng quan về các Aggregate Function

Các Aggregate Function thông dụng

MAX()	Truy vấn và lấy ra một trường dữ liệu có giá trị lớn nhất **SELECT MAX**(age) **FROM** Student;
MIN()	Truy vấn và lấy ra một trường dữ liệu có giá trị nhỏ nhất **SELECT MIN**(age) **FROM** Student;
COUNT()	Đếm và trả về số lượng bản ghi. Nếu đếm các trường dữ liệu cụ thể thì giá trị **NULL** sẽ không được tính **SELECT COUNT**(*) **FROM** Student; **SELECT COUNT**(age) **FROM** Student; → Bản ghi có **age = NULL** sẽ không được đếm
SUM()	Tính toán và trả về tổng của một cột dữ liệu số (number). Có thể tiến hành tính toán trong quá trình **SUM**() **SELECT SUM**(quantity) **FROM** Product; **SELECT SUM**(quantity * 10) **FROM** Product;
AVG()	Tính toán và trả về giá trị trung bình của một cột dữ liệu số (number). **SELECT AVG**(quantity) **FROM** Product

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

3. Group by - Having

GROUP BY - Mệnh đề nhóm các bản ghi

- ***Câu lệnh GROUP BY*** *nhóm các hàng có cùng giá trị vào các hàng tóm tắt, ví dụ như "tìm số lượng khách hàng ở mỗi quốc gia".*
- ***Câu lệnh GROUP BY*** *thường được sử dụng cùng với các hàm tổng hợp* ***(COUNT(), MAX(), MIN(), SUM(), AVG())*** *để nhóm tập kết quả theo một hoặc nhiều cột*

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

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

3. Group by - Having

Câu lệnh SQL sau liệt kê ra số lượng customers theo mỗi country

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

Câu lệnh SQL sau liệt kê số lượng customers theo mỗi country, sắp xếp từ cao tới thấp

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
ORDER BY COUNT(CustomerID) DESC;
```

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

3. Group by - Having

HAVING - Mệnh đề sử dụng thay thế cho WHERE

- ***Having -*** *Mệnh đề* ***HAVING*** *được thêm vào SQL vì từ khóa* ***WHERE*** *không thể được sử dụng với các hàm tổng hợp (Aggregate Function).*

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

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

3. Group by - Having

Câu lệnh SQL sau liệt kế số lượng customers ở mỗi thành phố sao cho số lượng customers theo thành phố phải lớn hơn 5

```sql
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5;
```

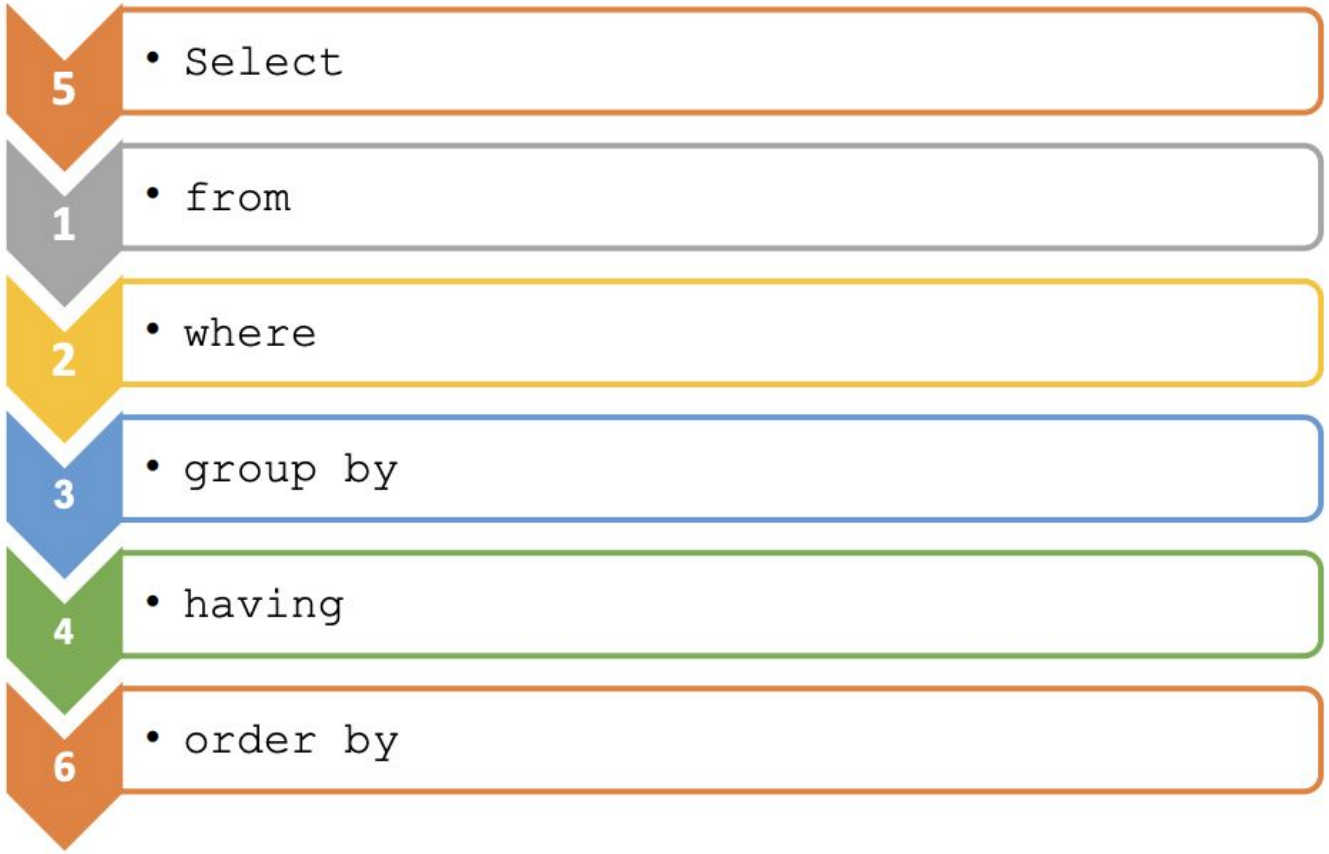
Câu lệnh SQL sau liệt kê số lượng customers theo mỗi country, sắp xếp từ cao tới thấp sao cho số lượng customers theo thành phố phải lớn hơn 5

```sql
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5
ORDER BY COUNT(CustomerID) DESC;
```

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

3. Group by - Having

Thứ tự thực hiện các mệnh đề trong câu lệnh SELECT



© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

4. Nested SELECT - Truy vấn lồng

Rikkei Academy
where the dream begins

Truy vấn lồng (Nested Query)

- Truy vấn lồng là các truy vấn nâng cao, cho phép chúng ta thực hiện các thao tác mà sẽ rất khó hoặc không thể làm được chỉ với một câu lệnh SELECT duy nhất
- Truy vấn lồng giúp chia nhỏ các vấn đề phức tạp thành các tác vụ đơn giản hơn, làm cho các truy vấn trở nên dễ quản lý và hiệu quả hơn
- Bằng cách sử dụng các truy vấn SELECT lồng nhau, bạn có thể lọc, tổng hợp và chuyển đổi dữ liệu một cách chính xác hơn

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

4. Nested SELECT - Truy vấn lồng

<Tên cột> <So sánh> <Select con>
Điều kiện đúng khi **giá trị của cột** so sánh **đúng** với **kết quả duy nhất** trả về từ select con

<Tên cột> <So sánh> ALL <Select con>
Điều kiện đúng khi **giá trị của cột** so sánh **đúng** với **toàn bộ giá trị** trả về từ select con

<Tên cột> <So sánh> ANY I SOME <Select con>
Điều kiện đúng khi **giá trị của cột** so sánh **đúng** với **bất kỳ một giá trị nào** trả về từ select con

<Tên cột> [NOT] IN <Select con>
Điều kiện đúng khi **giá trị của cột nằm trong tập giá trị** trả về từ select con

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

4. Nested SELECT - Truy vấn lồng

<Tên cột> <So sánh> <Select con>

- ***Select con*** *- Trả về một kết quả duy nhất*
- ***cost*** *được so sánh với* ***1 kết quả duy nhất trả về từ select con***

```
SELECT name, cost
FROM building
WHERE cost =
    (
        SELECT MAX(cost) FROM building
    )
```

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

4. Nested SELECT - Truy vấn lồng

<Tên cột> <So sánh> ALL <Select con>

- ***Select con*** *- Trả về một tập giá trị*
- ***cost*** *được so sánh với* ***Toàn bộ (ALL) kết quả trả về từ select con***

```
SELECT * FROM building
WHERE cost > ALL
    (
        SELECT cost FROM building
        WHERE city = 'can tho'
    );
```

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

4. Nested SELECT - Truy vấn lồng

<Tên cột> <So sánh> ANY | SOME <Select con>

- ***Select con*** *- Trả về một tập giá trị*
- ***cost*** *được so sánh với* ***bất kỳ một kết quả trong tập kết quả trả về từ select con***

```
SELECT * FROM building
WHERE city <> "can tho" AND cost > ANY
    (
        SELECT cost FROM building
        WHERE city = 'can tho'
    );
```

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

4. Nested SELECT - Truy vấn lồng

<Tên cột> <So sánh> [NOT] IN <Select con>

- ***Select con*** *- Trả về một tập giá trị*
- ***id*** *được kiểm tra xem có nằm trong* ***tập kết quả trả về từ select con*** *hay không?*

```
SELECT * FROM building
WHERE id IN
    (
        SELECT building_id FROM design
    );
```

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

5. Truy vấn dữ liệu trên nhiều bảng

Kết nối ngoại - outer join

- *Nếu một dòng của một bảng được kết nối không khớp với dòng nào bên bảng còn lại → dòng đó sẽ không xuất hiện ở bảng kết quả*
- ***Kết nối ngoại** cho phép giữ lại các dòng không thỏa mãn điều kiện kết nối*

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

5. Truy vấn dữ liệu trên nhiều bảng

Kết nối ngoại - outer join

- *MySQL cung cấp các loại kết nối ngoại (join) sau*
- ***INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN***

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

5. Truy vấn dữ liệu trên nhiều bảng

Kết nối ngoại - outer join

- *MySQL cung cấp các loại kết nối ngoại (join) sau*
- ***INNER JOIN, LEFT JOIN, RIGHT JOIN, UNION JOIN***
- ***ON -** Từ khoá thể hiện điều kiện kết nối giữa các bảng với nhau*

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

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

TỔNG KẾT

- Tổng quan về Aliases - Tên tạm thời trong SQL
- Tổng quan về các Aggregate Function trong SQL
- Group by – Having
- Truy vấn dữ liệu trên nhiều bảng
- Nested Select - Truy vấn lồng trong SQL

© 2022 By Rikkei Academy - Rikkei Education - All rights reserved.

RIKEI Academy®
where the dream begins

KẾT THÚC

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN