(Advanced Query Techniques

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

Trong phần này chúng ta sẽ đào sâu một số câu lệnh nâng cao như SELECT, INSERT...

Có thể nói hầu như ai cũng biết qua câu lệnh căn bản kiểu như "SELECT * FROM TABLENAME WHERE..." nhưng có thể có nhiều người không biết đến những tính chất nâng cao của nó.

Cú pháp đầy đủ của một câu lệnh SELECT rất phức tạp tuy nhiên ở đây chỉ trình bày những nét chính của lệnh này mà thôi:

SELECT select_list [INTO new_table] FROM table_source [WHERE search_condition] [GROUP BY group_by_expression] [HAVING search_condition] [ORDER BY order_expression [ASC | DESC]]

Chúng ta sẽ lần lượt nghiên cứu từng clause (mệnh đề) trong câu lệnh này.

SELECT Clause

Sau keyword (từ khóa) SELECT ta sẽ có một danh sách các cột mà ta muốn select được cách nhau bằng dấu ",". Có 3 Keywords cần nhấn mạnh trong phần SELECT.

- **Distinct** : Khi có keyword này vào thì sẽ cho kết quả các cột không trùng nhau. Ví dụ trong Orders table của Norwind database (database mẫu di kèm với SQL Server) chứa giá trị trùng lập (duplicate value) trong cột ShipCity. Nếu ta muốn select một danh sách ShipCity trong đó mỗi city chỉ xuất hiện một lần trong kết quả nhận được ta dùng như sau:

SELECT DISTINCT ShipCity, ShipRegion

FROM Orders

ORDER BY ShipCity

- **Top***n* : Nếu ta muốn select *n* hàng đầu tiên mà thôi ta có thể dùng Top keyword. Nếu có thêm ORDER BY thì kết quả sẽ được order trước sau đó mới select. Chúng ta cũng có thể select số hàng dựa trên phần trăm bằng cách thêm Keyword Percent vào. Ví dụ sau sẽ select 10 hàng đầu tiên theo thứ tự:

```
SELECT DISTINCT TOP 10 ShipCity, ShipRegion

FROM Orders

ORDER BY ShipCity
```

- **As** : Đôi khi chúng ta muốn cho SELECT statement dễ đọc hơn một chút ta có thể dùng một alias (tức là từ thay thế hay từ viết tắt) với keyword As hay không có keyword As: *table_name* As *table_alias* hay *table_nametable_alias*. Ví dụ:

```
USE pubs

SELECT p.pub_id, p.pub_name AS PubName

FROM publishers AS p
```

Ngoài ra trong Select list ta có thể select dưới dạng một expression như sau:

```
SELECT FirstName + ' ' + LastName AS "Employee Name",

IDENTITYCOL AS "Employee ID",

HomePhone,

Region

FROM Northwind.dbo.Employees

ORDER BY LastName, FirstName ASC
```

Trong ví dụ trên ta select cột "Employee Name" là sản phẩm ghép lại của cột FirstName và LastName được cách nhau bằng một khoảng trắng. Một giá trị thuộc loại identity để làm cột "Employee ID". Kết quả sẽ được sắp theo thứ tự từ nhỏ tới lớn (ASC) (còn DESC là từ lớn tới nhỏ) trong đó cột LastName được sắp trước rồi mới tới cột FirstName.

The INTO Clause

INTO Clause cho phép ta select data từ một hay nhiều table sau đó kết quả sẽ được insert vào một table mới. Table này được tạo ra do kết quả của câu lệnh SELECT INTO. Ví dụ:

SELECT FirstName, LastName

INTO EmployeeNames

FROM Employers

Câu lệnh tên sẽ tạo ra một table mới có tên là EmployeeNames với 2 cột là FirstName và LastName sau đó kết quả select được từ table Employers sẽ được insert vào table mới này. Nếu table EmployeeNames tồn tại SQL Server sẽ báo lỗi. Câu lệnh này thường hay được sử dụng để select một lượng data lớn từ nhiều table khác nhau vào một table mới (thường dùng cho mục đích tạm thời (temporary table)) mà khỏi phải thực thi câu lệnh Insert nhiều lần.

Một cách khác cũng select data từ một hay nhiều table và insert vào một table khác là dùng "**Insert Into...Select...**". Nhưng câu lệnh này không tạo ra một table mới. Nghĩa là ta table đó phải tồn tại trước. Ví dụ:

INSERT INTO EmployeeNames

SELECT FirstName, LastName

FROM Employers

Chú ý là không có chữ "**Value**" trong câu Insert này.

The GROUP BY and HAVING Clauses

GROUP BY dùng để tạo ra các giá trị tổng (aggregate values) cho từng hàng trong kết quả select được. Chỉ có một hàng cho từng giá trị riêng biệt (distinct) của từng cột. Các cột được select đều phải nằm trong GROUP BY Clause. Hãy xem ví dụ phức tạp sau:

SELECT OrdD1.OrderID AS OrderID,

SUM(OrdD1.Quantity) AS "Units Sold",

SUM(OrdD1.UnitPrice * OrdD1.Quantity) AS Revenue

FROM [Order Details] AS OrdD1

```
WHERE OrdD1.OrderID in (SELECT DISTINCT OrdD2.OrderID

FROM [Order Details] AS OrdD2

WHERE OrdD2.UnitPrice > $100)

GROUP BY OrdD1.OrderID

HAVING SUM(OrdD1.Quantity) > 100
```

Trong ví dụ trên đầu tiên ta select những order riêng biệt (distinct) từ Order Details table với giá > 100. Sau đó tiếp tục select OrderID, "Units Sold", Revenue từ kết quả trên trong đó "Units Sold" và Revenue là những aggregate columns (cho giá trị tổng một cột của những hàng có cùng OrderID). HAVING Clause đóng vai trò như một filter dùng để lọc lại các giá trị cần select mà thôi. HAVING Clause thường đi chung với GROUP BY mặc dù có thể xuất hiện riêng lẻ.

UNION

Uninon keyword có nhiệm vụ ghép nối kết quả của 2 hay nhiều queries lại thành một kết quả.

Ví dụ:

Giả sử có table1(ColumnA varchar(10), ColumnB int) và table2(ColumnC varchar(10), ColumnD int). Ta muốn select data từ table1 và ghép với data từ table2 để tạo thành một kết quả duy nhất ta làm như sau:

```
SELECT * FROM Table1

UNION ALL

SELECT * FROM Table2
```

Nếu không có keyword ALL thì những hàng giống nhau từ 2 table sẽ chỉ xuất hiện một lần trong kết quả. Còn khi dùng ALL thì các hàng trong 2 table đều có trong kết quả bất chấp việc lập lại.

Khi Dùng Union phải chú ý hai chuyện: số cột select ở 2 queries phải bằng nhau và data type của các cột tương ứng phải compatible (tương thích).