

Session - 5

Transact-SQL

Welcome to the Session, **Transact-SQL**.

This session explains Transact-SQL and the different categories of Transact-SQL statements. It also explains the various data types and elements supported by Transact-SQL. Finally, the session explains set theory, predicate logic, and the logical order of operators in the SELECT statement.

In this Session, you will learn to:

- Explain Transact-SQL
- List the different categories of Transact-SQL statements
- Explain the various data types supported by Transact-SQL
- Explain Transact-SQL language elements
- Explain sets and predicate logic
- Describe the logical order of operators in the SELECT statement



5.1 Giới thiệu

SQL là ngôn ngữ phổ quát được sử dụng trong thế giới cơ sở dữ liệu. Hầu hết các sản phẩm RDBMS hiện đại sử dụng một số loại phương ngữ SQL làm ngôn ngữ truy vấn chính. Có thể sử dụng SQL để tạo ra hoặc hủy bỏ các đối tượng, như là bảng, trên máy chủ cơ sở dữ liệu và làm nhiều việc với các đối tượng này, như là đưa dữ liệu vào trong đó hoặc truy vấn dữ liệu. Transact-SQL là thực hiện SQL tiêu chuẩn của Microsoft.

Thường được gọi là T-SQL, ngôn ngữ này thực hiện một cách chuẩn hóa để giao tiếp với cơ sở dữ liệu. Ngôn ngữ Transact-SQL là bản nâng cao của SQL, ngôn ngữ cơ sở dữ liệu quan hệ tiêu chuẩn của Viện Tiêu chuẩn Quốc gia Mỹ (ANSI). Họ cung cấp một ngôn ngữ toàn diện hỗ trợ định nghĩa các bảng định, chèn, xóa, cập nhật và truy cập dữ liệu trong bảng.

5.2 Transact-SQL

Transact-SQL là một ngôn ngữ mạnh mẽ đem lại các tính năng như là các kiểu dữ liệu, các đối tượng tạm thời, và các thủ tục đã lưu trữ mở rộng. Con trỏ cuộn, xử lý có điều kiện, kiểm soát giao tác, và ngoại lệ và xử lý lỗi cũng có một số tính năng được Transact-SQL hỗ trợ.

Ngôn ngữ Transact-SQL trong SQL Server 2012 cung cấp hiệu suất nâng cao, chức năng tăng lên và các đặc điểm tăng cường. Các cải tiến bao gồm các hàm vô hướng, phân trang, trình tự, khám phá siêu dữ liệu, và hỗ trợ xử lý lỗi tốt hơn.

Code Snippet 1 trình bày câu lệnh Transact-SQL , SELECT, được sử dụng để lấy tất cả các bản ghi của nhân viên với 'Design Engineer' là JobTitle từ bảng Employee.

Code Snippet 1:

```
SELECT LoginID
FROM Employee
WHERE JobTitle = 'Design Engineer'
```

Hình 5.1 trình bày kết quả lấy tất cả bản ghi của nhân viên với 'Design Engineer' là JobTitle từ bảng Employee.



	LoginID	JobTitle
1	gail	Design Engineer
2	jossef	Design Engineer
3	sharon	Design Engineer

Hình 5.1: Output of SELECT Statement

Transact-SQL đưa vào nhiều thành phần cú pháp được dùng bởi hoặc ảnh hưởng đến hầu hết các câu lệnh. Những phần tử này bao gồm các kiểu dữ liệu, các xác nhận, các hàm, các biến, các biểu thức, điều khiển dòng chảy, chú thích, và dấu tách khối lệnh.

5.3 Các thể loại câu lệnh Transact-SQL khác nhau

SQL Server hỗ trợ ba loại câu lệnh Transact-SQL, cụ thể là: DDL, DML và DCL.

5.4 Ngôn ngữ định nghĩa dữ liệu (DDL)

DDL, thường là một phần của DBMS, được sử dụng để định nghĩa và quản lý tất cả các thuộc tính và tính chất của một cơ sở dữ liệu, bao gồm bố trí hàng, định nghĩa cột, cột khóa, địa điểm tập tin, và chiến lược lưu trữ. Các câu lệnh DDL được sử dụng để xây dựng và sửa đổi cấu trúc của bảng và các đối tượng khác như dạng xem, khởi phát, thủ tục đã lưu trữ, và vân vân. Đối với từng đối tượng, thường có các câu lệnh CREATE, ALTER, và DROP (như là CREATE TABLE, ALTER TABLE, và DROP TABLE). Hầu hết các câu lệnh DDL lấy dạng sau đây :

- CREATE object _ name
- ALTER object _ name
- DROP object _ name

Trong các câu lệnh DDL, object_name có thể là bảng, dạng xem, khởi phát, thủ tục đã lưu trữ, và vân vân.

5.5 Ngôn ngữ thao tác dữ liệu (DML)

DML được sử dụng để chọn, chèn, cập nhật, hoặc xóa dữ liệu trong các đối tượng được định nghĩa với DDL. Tất cả người dùng cơ sở dữ liệu có thể sử dụng những câu lệnh này trong các phép tính thông thường trên một cơ sở dữ liệu. Các câu lệnh DML khác nhau như sau:

- SELECT statement
- INSERT statement
- UPDATE statement
- DELETE statement

5.6 Ngôn ngữ kiểm soát dữ liệu (DCL)

Dữ liệu là một phần quan trọng của cơ sở dữ liệu, do vậy nên có những bước thích hợp để kiểm tra xem không có người dùng không hợp lệ nào truy cập vào dữ liệu. Ngôn ngữ kiểm soát dữ liệu được dùng để kiểm soát sự cho phép đối với các đối tượng cơ sở dữ liệu. Quyền được kiểm soát bằng cách sử dụng câu lệnh GRANT, REVOKE, và DENY. Các câu lệnh DCL cũng được sử dụng để bảo mật cơ sở dữ liệu. Ba câu lệnh DCL cơ bản như sau:

- GRANT statement
- REVOKE statement
- DENY statement

5.7 Các kiểu dữ liệu

Kiểu dữ liệu là một thuộc tính định nghĩa kiểu dữ liệu mà một đối tượng có thể có. Kiểu dữ liệu phải được cung cấp cho các cột, tham số, biến, và hàm trả về các giá trị dữ liệu, và các thủ tục lưu trữ có một mã trả lại. Transact-SQL đưa vào một số kiểu dữ liệu cơ bản, như là varchar, text, và int. Tất cả các dữ liệu được lưu trữ trong SQL Server phải tương thích với một trong các kiểu dữ liệu cơ sở.

Các đối tượng sau đây có kiểu dữ liệu :

- Cột có trong bảng và dạng xem
- Các tham số trong các thủ tục được lưu
- Các biến số
- Các hàm Transact-SQL mà trả lại một hoặc nhiều giá trị dữ liệu thuộc một kiểu dữ liệu cụ thể
- Thủ tục lưu trữ có một mã trả lại thuộc kiểu dữ liệu số nguyên

Các mục khác nhau trong SQL Server 2012 như cột, các biến, và biểu thức được gán các kiểu dữ liệu. SQL Server 2012 hỗ trợ ba kiểu dữ liệu:

➤ Kiểu dữ liệu do hệ thống định nghĩa

Những kiểu dữ liệu được cung cấp bởi SQL Server 2012. Bảng 5.1 trình bày các kiểu dữ liệu do hệ thống định nghĩa thường được sử dụng của SQL Server 2012.

Loại	Kiểu Dữ liệu	Mô tả
Các số chính xác	int	Cột của loại này chiếm giữ 4 byte không gian bộ nhớ. Thường được dùng để giữ các giá trị số nguyên. Có thể giữ dữ liệu số nguyên từ -2 ³¹ (-2.147.483.648) đến 2 ³¹ -1 (2.147.483.647).
	smallint	Cột của loại này chiếm giữ 2 byte không gian bộ nhớ. Có thể giữ dữ liệu số nguyên từ -32.768 đến 32.767.
	tinyint	Cột thuộc loại này chiếm giữ 1 byte không gian bộ nhớ. Có thể giữ dữ liệu số nguyên từ 0 đến 255.
	bigint	Cột của loại này chiếm giữ 8 byte không gian bộ nhớ. Có thể chứa dữ liệu trong khoảng -2 ⁶³ (-9.223.372.036.854.775.808) đến 2 ⁶³ -1 (9.223.372.036.854.775.807).
	numeric	Cột thuộc loại này có độ chính xác và tỉ lệ cố định.
	money	Cột của loại này chiếm giữ 8 byte không gian bộ nhớ. Trình bày các giá trị dữ liệu tiền tệ trong khoảng từ -2 ⁶³ /10000 (-922,337,203,685,477.5808) to 2 ⁶³ -1 (922,337,203,685,477.5807).

Loại	Kiểu Dữ liệu	Mô tả
Các số gần đúng	float	Cột của loại này chiếm giữ 8 byte không gian bộ nhớ. Biểu diễn số có dấu thập phân thay đổi dao động từ -1.79E
	real	Cột của loại này chiếm giữ 4 byte không gian bộ nhớ. Biểu diễn số có độ chính xác thay đổi dao động từ -3.40E+38 through 3.40E+38.
Date and Time	datetime	Biểu diễn ngày và giờ. Được lưu như hai số nguyên 4
	smalldatetime	Biểu diễn ngày và giờ.
Character String	char	Lưu trữ dữ liệu ký tự có chiều dài cố định và không phải là Unicode.
	varchar	Lưu trữ dữ liệu ký tự có chiều dài thay đổi và Chuỗi ký tự không phải là Unicode với tối đa 8.000 ký tự.
	text	Lưu trữ dữ liệu ký tự có chiều dài thay đổi và không phải là Unicode với chiều dài tối đa $2^{31} - 1$ (2.147.483.647) ký tự.
Unicode Types	nchar	Lưu trữ dữ liệu ký tự Unicode với độ dài cố định.
	nvarchar	Lưu trữ dữ liệu ký tự Unicode có độ dài biến đổi.
Other Data Types	timestamp	Cột của loại này chiếm giữ 8 byte không gian bộ nhớ. Có thể giữ tự động các số nhị phân duy nhất, được tạo ra mà được tạo ra cho một cơ sở dữ liệu.
	binary(n)	Stores fixed-length binary data with a maximum length of 8000 bytes.
	varbinary(n)	Lưu trữ dữ liệu nhị phân có chiều dài cố định với chiều dài tối đa là 8000 byte.
	image	Lưu trữ dữ liệu nhị phân với độ dài biến đổi với độ dài tối đa là of $2^{30}-1$ (1,073,741,823) bytes.
	uniqueidentifier	Cột của loại này chiếm giữ 16 byte không gian bộ nhớ. Ngoài ra, lưu trữ mã định danh duy nhất toàn cầu (GUID

Table 5.1: System-Defined Data Types

➤ Kiểu dữ liệu alias

Những cái này dựa trên các kiểu dữ liệu do hệ thống cung cấp. Các kiểu dữ liệu bí danh được sử dụng khi nhiều hơn một bảng lưu trữ cùng một kiểu dữ liệu trong một cột và có đặc điểm tương tự như là chiều dài, khả năng null, và loại. Trong các trường hợp như vậy, có thể tạo ra kiểu dữ liệu bí danh để có thể được dùng chung cho các bảng này.

Các kiểu dữ liệu bí danh có thể được tạo ra bằng cách sử dụng câu lệnh CREATE TYPE. Cú pháp cho câu lệnh CREATE TYPE như sau:

Cú pháp

```
CREATE TYPE [ schema_name. ] type_name { FROM base_type [ ( precision [ , scale ] ) ] [ NULL | NOT NULL ] } [ ; ]
```

trong đó:

schema_name: xác định tên của sơ đồ trong đó kiểu dữ liệu bí danh đang được tạo ra.

Sơ đồ là một tập hợp các đối tượng như bảng, dạng xem, và vân vân trong một cơ sở dữ liệu.

type_name: xác định tên của loại bí danh đang được tạo ra.

base_type: xác định tên của kiểu dữ liệu do hệ thống định nghĩa dựa trên đó kiểu dữ liệu bí danh đang được tạo ra.

precision và scale: xác định độ chính xác và tỉ lệ cho dữ liệu số.

NULL | NOT NULL: xác định xem kiểu dữ liệu có thể chứa giá trị null hay không.

Code Snippet 2 trình bày cách để tạo ra một kiểu dữ liệu bí danh sử dụng câu lệnh CREATE TYPE.

Code Snippet 2:

```
CREATE TYPE usertype FROM varchar(20) NOT NULL
```

Trong đoạn mã, kiểu dữ liệu dựng sẵn varchar được lưu giữ như một kiểu dữ liệu mới có tên là **usertype** bằng cách sử dụng câu lệnh CREATE TYPE.

➤ Các loại do người dùng định nghĩa

Những loại này được tạo ra bằng cách sử dụng các ngôn ngữ lập trình được .NET Framework hỗ trợ.

5.5 Các phần tử ngôn ngữ Transact-SQL

Những phần tử ngôn ngữ Transact-SQL được sử dụng trong SQL Server 2012 để làm việc trên các dữ liệu được nhập vào cơ sở dữ liệu SQL Server. Những phần tử ngôn ngữ Transact-SQL bao gồm các xác nhận, toán tử, hàm, biến, biểu thức, điều khiển dòng chảy, lỗi, và giao tác, chú thích, và dấu tách khối lệnh.

5.6 Các xác nhận và toán tử

Các xác nhận được sử dụng để đánh giá liệu một biểu thức là TRUE, FALSE, hay UNKNOWN. Một số xác nhận có sẵn trong Transact-SQL như sau:

- **IN** - Xác định xem một giá trị đã chỉ định so khớp với bất kỳ giá trị trong một truy vấn con hoặc một danh sách.
- **BETWEEN** - Chỉ ra một loạt các giá trị để kiểm tra.
- **LIKE** - Được sử dụng để so khớp các ký tự so với một mô hình cụ thể.
- **CONTAINS** - Tìm kiếm các trùng khớp chính xác hoặc chưa chính xác đối với những từ đơn và cụm từ đơn lẻ, các từ trong một khoảng cách nhất định với nhau, hoặc các trùng khớp có trọng số.

Table 5.2 trình bày một số ví dụ về xác nhận.

Predicate	Example
IN	SELECT UserID, FirstName, LastName, Salary FROM Employee WHERE Salary IN(5000,20000);
BETWEEN	Select UserID, FirstName, LastName, Salary FROM Employee WHERE Salary BETWEEN 5000 and 20000;
LIKE	Select UserID, FirstName, LastName, Salary FROM Employee WHERE FirstName LIKE '%h%'
CONTAINS	SELECT UserID, FirstName, LastName, Salary FROM Employee WHERE Salary CONTAINS(5000);

Table 5.2: Predicate Examples

Các toán tử được dùng để thực hiện tính toán số học, so sánh, ghép hoặc gán các giá trị. Ví dụ, dữ liệu có thể được kiểm tra để xác minh rằng cột **COUNTRY** cho dữ liệu khách hàng được tập kết (hoặc có giá trị NOT NULL). Trong các truy vấn, bất cứ ai có thể xem dữ liệu trong bảng yêu cầu toán tử cũng có thể thực hiện các phép tính. Yêu cầu có sự cho phép thích hợp trước khi có thể thay đổi thành công dữ liệu. SQL Server có bảy loại toán tử. Bảng 5.3 mô tả các toán tử khác nhau được hỗ trợ trong SQL Server 2012.

Toán tử	Mô tả	Ví dụ
So sánh	So sánh một giá trị đối với giá trị khác hoặc một biểu thức	=, <, >, >=, <=, !=, !>
Logic	Thử về sự đúng sai của một điều kiện	AND,OR,NOT
Số học	Thực hiện các phép tính số học như cộng, trừ, nhân và chia	+, -, *, /, %
Sự ghép nối	Kết hợp hai chuỗi thành một chuỗi	+
Gán	Gán một giá trị cho một biến số	=

Table 5.3: Operators

Table 5.4 trình bày thứ tự ưu tiên của các xác nhận và toán tử.

Order	Operators
1	() Parentheses
2	*, /, %
3	+, -
4	=, <, >, >=, <=, !=, !>
5	NOT
6	AND
7	BETWEEN, IN, CONTAINS, LIKE, OR
8	=

Table 5.4: Precedence of Predicates and Operators

Code Snippet 3 trình bày thực thi các toán tử theo thứ tự ưu tiên.

Code Snippet 3:

```
DECLARE @Number int;  
SET @Number = 2 + 2 * (4 + (5 - 3))  
SELECT @Number
```

Ở đây, các bước để đi đến kết quả như sau :

1. $2 + 2 * (4 + (5 - 3))$
2. $2 + 2 * (4 + 2)$
3. $2 + 2 * 6$
4. $2 + 12$
5. 14

Do đó, đoạn mã sẽ hiển thị 14.

5.5.2 Các hàm

Hàm là tập hợp các câu lệnh Transact-SQL được dùng để thực hiện tác vụ nào đó. Transact-SQL bao gồm một số lượng lớn các hàm. Các hàm này có thể hữu dụng khi dữ liệu được tính toán hoặc thao tác. Trong SQL, các hàm làm việc với dữ liệu, hoặc nhóm dữ liệu, để trả lại một giá trị yêu cầu. Chúng có thể được sử dụng trong danh sách SELECT, hoặc bất cứ nơi nào trong một biểu thức. Bốn loại hàm trong SQL Server 2012 như sau:

- **Rowset functions (Các hàm tập hàng)** - Trong Transact-SQL, hàm tập hàng được sử dụng để trả về một đối tượng có thể được sử dụng thay cho tham khảo bảng. Ví dụ, **OPENDATASOURCE**, **OPENQUERY**, **OPENROWSET**, and **OPENXML** là các hàm tập hàng.
- **Aggregate functions (Các hàm tổng hợp)** - Transact-SQL cung cấp các hàm tổng hợp để hỗ trợ việc tổng kết số lượng lớn dữ liệu. Ví dụ, **SUM**, **MIN**, **MAX**, **AVG**, **COUNT**, **COUNTBIG**, và vân vân là các hàm tổng hợp.
- **Ranking functions (Các hàm xếp hạng)** - Nhiều nhiệm vụ, chẳng hạn như tạo ra các mảng, tạo ra số thứ tự, tìm kiếm các hạng, và vân vân có thể được thực hiện một cách dễ dàng hơn và nhanh hơn bằng cách sử dụng các hàm xếp hạng. Ví dụ, **RANK**, **DENSE _ RANK**, **NTILE**, và **ROW _ NUMBER** là các hàm xếp hạng.
- **Scalar functions (Các hàm vô hướng)** - Trong các hàm vô hướng, đầu vào là một giá trị đơn và đầu ra nhận được cũng là một giá trị đơn

Table 5.5 trình bày các hàm vô hướng thường được sử dụng trong SQL.

Tên hàm	Mô tả	Ví dụ
Hàm chuyển đổi	Hàm chuyển đổi được sử dụng để chuyển đổi giá trị của một kiểu dữ liệu sang loại khác. Ngoài ra, nó có thể được dùng để có được nhiều các định dạng đặc biệt của ngày.	CONVERT
Hàm date và time	Hàm ngày và thời gian được sử dụng để thao tác các giá trị ngày và thời gian. Chúng rất hữu ích để thực hiện các tính toán dựa trên thời gian và ngày tháng.	GETDATE, SYSDATETIME, GETUTCDATE, DATEADD, DATEDIFF, YEAR, MONTH, DAY
Hàm toán học	Các hàm toán học thực hiện các phép tính đại số đối với các giá trị số.	RAND, ROUND, POWER, ABS, CEILING, FLOOR
Hàm hệ thống	SQL Server cung cấp các hàm hệ thống để trả về các thiết lập siêu dữ liệu hoặc cấu hình.	HOST_ID, HOST_NAME, ISNULL
Hàm chuỗi	Hàm chuỗi được sử dụng cho đầu vào chuỗi như char và varchar. Đầu ra có thể là một chuỗi hoặc một giá trị số.	SUBSTRING, LEFT, RIGHT, LEN, DATALENGTH, REPLACE, REPLICATE, UPPER, LOWER, RTRIM, LTRIM

Table 5.5: Các hàm vô hướng

Ngoài ra còn có các hàm vô hướng khác như hàm con trỏ, hàm logic, hàm siêu dữ liệu, hàm bảo mật, và...vv có sẵn trong SQL Server 2012.

5.5.3 Các biến

Biến là một đối tượng mà có thể giữ một giá trị dữ liệu. Trong Transact-SQL, các biến có thể được phân thành các biến cục bộ và toàn cầu.

Trong Transact-SQL, các biến số nội bộ được tạo ra và sử dụng để lưu trữ tạm thời trong khi các câu lệnh SQL được thực thi. Có thể truyền dữ liệu cho các câu lệnh SQL bằng cách sử dụng các biến số nội bộ. Tên của biến cục bộ phải được bắt đầu bằng dấu '@'.

Các biến số toàn cục là các biến số được sẵn được hệ thống định nghĩa và duy trì. Biến toàn cầu trong SQL Server được bắt đầu với hai dấu '@@'. Giá trị của bất kỳ biến nào cũng có thể được lấy ra với một truy vấn SELECT đơn giản.

5.5.4 Các biểu thức

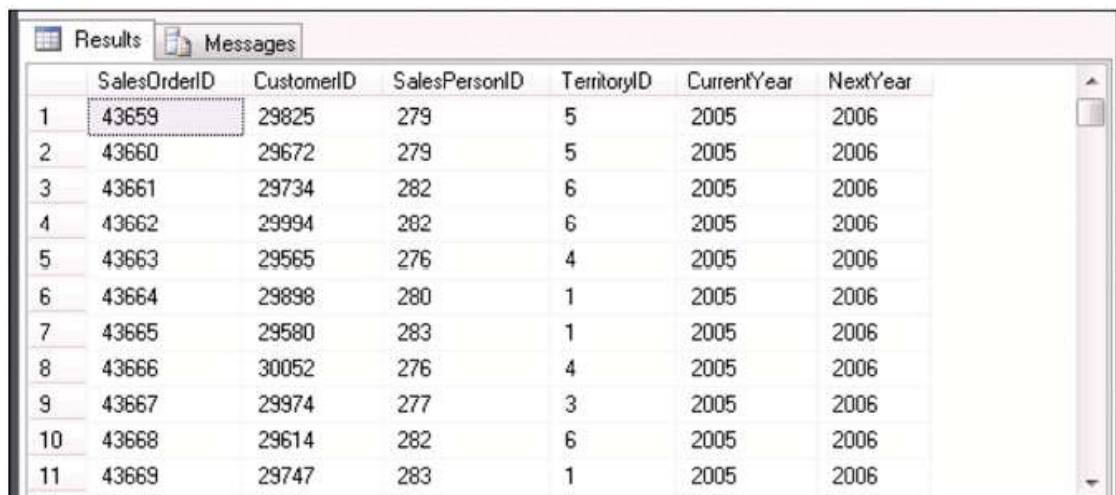
Biểu thức là một tổ hợp các ký hiệu nhận dạng, giá trị và toán tử mà SQL Server có thể đánh giá nhằm có được kết quả. Có thể sử dụng biểu thức ở một số nơi khác nhau khi truy cập hay thay đổi dữ liệu.

Code Snippet 4 trình bày một biểu thức hoạt động trên một cột để thêm một số nguyên vào kết quả của hàm YEAR trên cột datetime.

Code Snippet 4:

```
SELECT SalesOrderID, CustomerID, SalesPersonID, TerritoryID, YEAR (OrderDate)
AS CurrentYear, YEAR (OrderDate) + 1 AS NextYear
FROM Sales.SalesOrderHeader
```

Hình 5.2 trình bày các kết quả của biểu thức.



	SalesOrderID	CustomerID	SalesPersonID	TerritoryID	CurrentYear	NextYear
1	43659	29825	279	5	2005	2006
2	43660	29672	279	5	2005	2006
3	43661	29734	282	6	2005	2006
4	43662	29994	282	6	2005	2006
5	43663	29565	276	4	2005	2006
6	43664	29898	280	1	2005	2006
7	43665	29580	283	1	2005	2006
8	43666	30052	276	4	2005	2006
9	43667	29974	277	3	2005	2006
10	43668	29614	282	6	2005	2006
11	43669	29747	283	1	2005	2006

Hình 5.2: Expression Result

5.5.5 Kiểm soát dòng chảy, lỗi, và giao tác

Mặc dù Transact-SQL chủ yếu là một ngôn ngữ truy vấn dữ liệu, nó hỗ trợ các câu lệnh kiểm soát dòng chảy để thực thi và tìm lỗi. Ngôn ngữ kiểm soát dòng chảy xác định lưu lượng thực thi các câu lệnh Transact-SQL, khối câu lệnh, hàm do người dùng định nghĩa, và các thủ tục lưu trữ.

Table 5.6 trình bày một số câu lệnh kiểm soát dòng chảy thường được sử dụng trong Transact-SQL.

Control-of-Flow Statement	Description
IF...ELSE	Cung cấp phân nhánh kiểm soát dựa trên một thử
WHILE	Lặp lại một câu lệnh hoặc một khối lệnh miễn là điều kiện vẫn còn đúng.
BEGIN...END	Định nghĩa phạm vi của khối lệnh Transact-SQL.
TRY...CATCH	Định nghĩa cấu trúc cho ngoại lệ và xử lý lỗi.
BEGIN TRANSACTION	Đánh dấu một khối lệnh như là một phần của giao tác rõ ràng.

Table 5.6: Control-of-Flow Statements

5.5.6 Chú thích

Chú thích là chuỗi văn bản mô tả, còn được gọi là nhận xét, trong mã chương trình sẽ được trình biên dịch bỏ qua. Chú thích có thể được đưa vào trong mã nguồn của một câu lệnh đơn lẻ, một lô, hoặc một thủ tục lưu trữ. Chú thích giải thích mục đích của chương trình, điều kiện thực thi đặc biệt, và cung cấp thông tin lịch sử sửa đổi. Microsoft SQL Server hỗ trợ hai loại kiểu dạng nhận xét:

➤ -- (hai dấu nối)

Có thể đánh dấu dòng mã đầy đủ hoặc một phần của mã thành một nhận xét, nếu hai dấu nối (- -) được đặt ở đoạn bắt đầu. Phần còn lại của dòng trở thành nhận xét.

Code Snippet 5 hiển thị việc sử dụng kiểu chú thích này.

Code Snippet 5:

```
USE AdventureWorks2012
-- HumanResources.Employee table contains the details of an employee.
-- This statement retrieves all the rows of the table
-- HumanResources.Employee.
SELECT * FROM HumanResources.Employee
```

➤ /* ... */ (cặp ký tự dấu gạch chéo-dấu sao)

Có thể sử dụng các ký tự nhận xét này trên cùng hàng như mã được thực thi, trên các dòng của chính chúng, hoặc thậm chí trong mã có thể thực thi. Mọi thứ trong dòng bắt đầu từ cặp chú thích mở (/*) đến cặp chú thích đóng (*/) được coi là một phần của chú thích. Đối với chú thích nhiều dòng, cặp ký tự chú thích mở mở comment (/*) phải bắt đầu chú thích, và cặp ký tự nhận xét đóng (*/) phải kết thúc chú thích.

Code Snippet 6 hiển thị việc sử dụng kiểu chú thích này.

Code Snippet 6:

```
USE AdventureWorks2012

/* HumanResources.Employee table contains the details of an employee.
This statement retrieves all the rows of the table
HumanResources.Employee. */
SELECT * FROM HumanResources.Employee
```

5.5.7 Dấu tách khối lệnh

Khối lệnh là tập hợp một hoặc nhiều câu lệnh Transact-SQL được gửi một lúc từ một ứng dụng vào SQL Server để thực thi. Các câu lệnh Transact-SQL trong một khối lệnh được biên dịch thành một đơn vị thực thi đơn lẻ, được gọi là kế hoạch thực thi. Các câu lệnh trong kế hoạch thực thi sau đó được thực thi từng cái một. Quá trình trong đó tập hợp các lệnh được xử lý từng cái một từ một khối lệnh được gọi là việc xử lý khối lệnh.

Dấu tách khối lệnh được xử lý bằng các công cụ máy khách SQL Server như SSMS để thực thi các lệnh. Ví dụ, bạn cần phải chỉ ra GO như một dấu tách khối lệnh trong SSMS.

Một ví dụ về câu lệnh khối lệnh được đưa ra trong Code Snippet 7.

Code Snippet 7:

```
USE AdventureWorks2012

SELECT * FROM HumanResources.Employee

GO
```

Trong Code Snippet 7, hai câu lệnh sẽ được nhóm lại thành một kế hoạch thực thi, nhưng được thực hiện mỗi lần một câu lệnh. Từ khóa GO báo hiệu sự kết thúc của một khối lệnh.

5.6 Các tập hợp, và xác nhận logic

Các tập hợp và logic xác nhận là hai nguyên tắc cơ bản toán học được sử dụng trong SQL Server 2012. Cả hai lý thuyết này được sử dụng trong truy vấn dữ liệu trong SQL Server 2012.

5.6.1 Lý thuyết tập hợp

Lý thuyết tập hợp là nền tảng toán học được sử dụng trong mô hình cơ sở dữ liệu quan hệ. Tập hợp là một bộ sưu tập các đối tượng riêng biệt được coi như là toàn bộ. Ví dụ, tất cả các nhân viên dưới bảng **Employee** có thể được coi là một tập hợp. Nhân viên là những đối tượng khác nhau tạo thành một phần của tập hợp trong bảng **Employee**.

Table 5.7 cho thấy các ứng dụng khác nhau trong lý thuyết tập hợp và ứng dụng tương ứng của chúng trong các truy vấn SQL Server.

Các ứng dụng lý thuyết tập hợp	Ứng dụng trong các truy vấn SQL Server
Hành động trên toàn bộ tập hợp cùng	Truy vấn toàn bộ bảng cùng một lúc.
Sử dụng khai báo, xử lý dựa trên tập hợp	Sử dụng các thuộc tính trong SQL Server để lấy dữ liệu cụ thể.
Các phần tử trong tập hợp phải là duy	Định nghĩa các khóa duy nhất trong bảng.
Không có hướng dẫn phân loại.	Kết quả truy vấn không được lấy theo bất kỳ thứ tự nào

Table 5.7: Ứng dụng trong Lý thuyết tập hợp với các truy vấn SQL Server

5.6.2 Logic xác nhận

Logic xác nhận là một khuôn khổ toán học bao gồm các bài kiểm tra logic cung cấp kết quả. Kết quả luôn được hiển thị như đúng hoặc sai. Trong Transact-SQL, các biểu thức như biểu thức WHERE và CASE được dựa trên logic xác nhận. Logic xác nhận cũng được sử dụng trong các tình huống khác trong Transact-SQL. Một số ứng dụng logic xác nhận trong Transact-SQL như sau :

- Thực thi toàn vẹn dữ liệu sử dụng ràng buộc CHECK
- Kiểm soát dòng chảy sử dụng câu lệnh IF
- Nối các bảng sử dụng bộ lọc ON
- Dữ liệu lọc trong các truy vấn sử dụng mệnh đề WHERE và HAVING
- Cung cấp logic có điều kiện cho các biểu thức CASE
- Định nghĩa các truy vấn con

5.7 Thứ tự logic của các toán tử trong câu lệnh SELECT

Cùng với cú pháp của các phần tử SQL Server khác nhau, người dùng SQL Server cũng phải biết quy trình toàn bộ truy vấn được thực thi như thế nào. Quy trình này là một quá trình logic chia nhỏ truy vấn và thực hiện truy vấn đó theo một trình tự được định nghĩa trước trong SQL Server 2012. Câu lệnh SELECT là một truy vấn sẽ được sử dụng để giải thích quy trình logic về thực hiện truy vấn.

Sau đây là cú pháp của câu lệnh SELECT.

Cú pháp:

```
SELECT <select list>
FROM <table source>
WHERE <search condition>
GROUP BY <group by list>
HAVING <search condition>
ORDER BY <order by list>
```

Table 5.8 giải thích những phần tử của câu lệnh SELECT.

Phần tử	Mô tả
SELECT <select list>	Định nghĩa các cột được trả về
FROM <table source>	Định nghĩa bảng được truy vấn
WHERE <search condition>	Lọc các hàng bằng cách sử dụng các xác nhận
GROUP BY <group by list>	Sắp xếp các hàng theo nhóm
HAVING <search condition>	Lọc các nhóm sử dụng các xác nhận
ORDER BY <order by list>	Sắp xếp đầu ra

Table 5.8: Elements of SELECT Statement

Code Snippet 8 trình bày một câu lệnh Select.

Code Snippet 8:

```
SELECT SalesPersonID, YEAR (OrderDate) AS OrderYear
FROM Sales.SalesOrderHeader
WHERE CustomerID = 30084
GROUP BY SalesPersonID, YEAR (OrderDate)
HAVING COUNT (*) > 1
ORDER BY SalesPersonID, OrderYear;
```

Trong ví dụ này, thứ tự theo đó SQL Server sẽ thực hiện câu lệnh SELECT như sau:

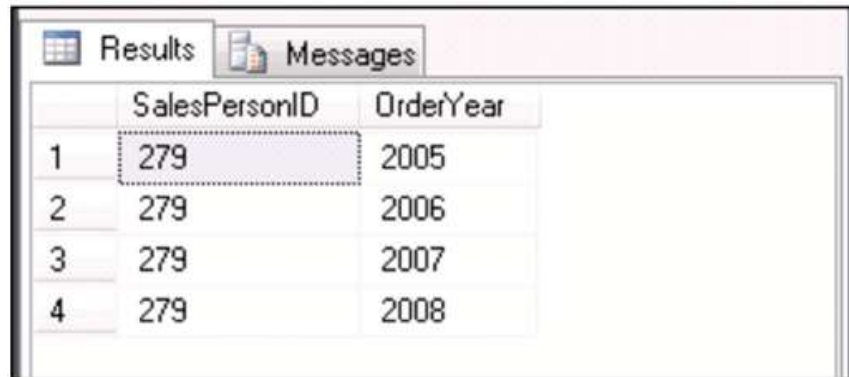
1. Đầu tiên, mệnh đề FROM được đánh giá để định nghĩa bảng nguồn sẽ được truy vấn.
2. Tiếp theo, mệnh đề WHERE được đánh giá để lọc các hàng trong bảng nguồn. Việc lọc này được định nghĩa bằng xác nhận đã đề cập trong mệnh đề WHERE.
3. Sau đó, mệnh đề GROUP BY được đánh giá. Mệnh đề này sắp xếp các giá trị đã lọc nhận được từ mệnh đề WHERE.

4. Tiếp theo, mệnh đề HAVING được đánh giá dựa trên xác nhận được cung cấp.
5. Tiếp theo, mệnh đề SELECT được thực hiện để xác định các cột sẽ xuất hiện trong kết quả truy vấn.
6. Cuối cùng, câu lệnh ORDER BY được đánh giá để hiển thị đầu ra.

Trình tự thực hiện cho câu lệnh SELECT trong Code Snippet 8 sẽ như sau:

5. **SELECT SalesPersonID, YEAR(OrderDate) AS OrderYear**
1. **FROM SalesOrderHeader**
2. **WHERE CustomerID = 30084**
3. **GROUP BY SalesPersonID, YEAR(OrderDate)**
4. **HAVING COUNT(*) > 1**
6. **ORDER BY SalesPersonID, OrderYear;**

Hình 5.3 hiển thị kết quả của câu lệnh SELECT.



	SalesPersonID	OrderYear
1	279	2005
2	279	2006
3	279	2007
4	279	2008

Hình 5.3: SELECT Statement Result

5.8 Kiểm tra tiến độ của bạn

1. Điều nào sau đây được sử dụng để định nghĩa và quản lý tất cả các thuộc tính và tính chất của cơ sở dữ liệu, bao gồm bố trí hàng, định nghĩa cột, cột khóa, địa điểm tập tin, và chiến lược lưu trữ?

(A)	DDL	(C)	DCL
(B)	DML	(D)	DML

2. Điều nào sau đây không được sử dụng trong DCL?

(A)	GRANT statement	(C)	UPDATE statement
(B)	REVOKE statement	(D)	DENY statement

3. Điều nào sau đây chỉ ra phạm vi giá trị để kiểm tra?

(A)	IN	(C)	LIKE
(B)	BETWEEN	(D)	CONTAINS

4. So khớp nội dung sau.

Control-of-Flow Statement		Mô tả	
a.	IF... ELSE	1.	Đánh dấu một khối lệnh như là một phần của giao tác rõ ràng.
b.	WHILE	2.	Định nghĩa cấu trúc cho ngoại lệ và xử lý lỗi.
c.	BEGIN ... END	3.	Lặp lại một câu lệnh hoặc một khối lệnh khi điều kiện vẫn còn đúng.
d.	TRY...CATCH	4.	Định nghĩa phạm vi của khối lệnh Transact-SQL
e.	BEGIN TRANSACTION	5.	Cung cấp phân nhánh kiểm soát dựa trên một thử nghiệm hợp lý

(A)	a-4, b-2, c-3, d-1, e-5	(C)	a-1,b-4, c-5,d-3, e-2
(B)	a-1, b-2, c-4, d-3, e-5	(D)	a-5,b-3,c-4,d-2,e-1

5. Điều nào sau đây là hai nguyên tắc cơ bản toán học được sử dụng trong SQL Server 2012?

(A)	Phân số và tập hợp	(C)	Logic xác nhận và phân số
(B)	Tập hợp và logic xác nhận	(D)	Xác suất và phân số

6. Điều nào sau đây sẽ là kết quả của đoạn mã?

SET @Number = 2 * (4 + 5) + 2 * (4 + (5 - 3))

(A)	120	(C)	42
(B)	62	(D)	26

5.8.1 Đáp án

1.	A
2.	C
3.	B
4.	D
5.	B
6.	C

Tóm tắt

- Transact-SQL là một ngôn ngữ mạnh mẽ đem lại các tính năng như là các kiểu dữ liệu, các đối tượng tạm thời, và các thủ tục đã lưu trữ mở rộng.
- SQL Server hỗ trợ ba loại câu lệnh Transact-SQL, cụ thể là: DDL, DML và DCL.
- Kiểu dữ liệu là một thuộc tính định nghĩa kiểu dữ liệu mà một đối tượng có thể có.
- Những phần tử ngôn ngữ Transact-SQL bao gồm các xác nhận, toán tử, hàm, biến, biểu thức, điều khiển dòng chảy, lỗi, và giao tác, chú thích, và dấu tách khối lệnh.
- Các tập hợp và logic xác nhận là hai nguyên tắc cơ bản toán học được sử dụng trong SQL Server 2012.
- Lý thuyết tập hợp là nền tảng toán học được sử dụng trong mô hình cơ sở dữ liệu quan hệ, nơi tập hợp là một bộ sưu tập các đối tượng riêng biệt được coi như toàn thể.
- Logic xác nhận là một khuôn khổ toán học bao gồm các bài kiểm tra logic cung cấp kết quả.



1. Sử dụng Query Editor để thực thi một truy vấn. Bảo đảm là việc kết nối vào một đối tượng định hình máy chủ mới được thiết lập. Sau đó, trong cơ sở dữ liệu AdventureWorks2012, thực hiện một truy vấn để chọn các cột cụ thể là ProductID, Name, và ProductNumber từ bảng Production.Product, và Product ID và ModifiedDate từ bảng Production.ProductDocument.