

# Quản Trị Dữ Liệu Với Microsoft SQL Server

## Chương: 5

### Transact-SQL



- Giải thích ngôn ngữ giao tác Transact-SQL
- Liệt kê các nhóm lệnh Transact-SQL khác nhau
- Giải thích các kiểu dữ liệu khác nhau được hỗ trợ bởi Transact-SQL
- Giải thích các thành phần ngôn ngữ Transact-SQL
- Giải thích luận lý vị từ (predicate logic)
- Mô tả thứ tự logic của các toán tử trong câu lệnh SELECT.

- SQL là ngôn ngữ phổ biến được sử dụng trong thế giới csdl.
- Hầu hết các sản phẩm hệ quản trị csdl hiện đại sử dụng một số kiểu SQL địa phương(dialect) như là ngôn ngữ truy vấn chính của chúng.
- SQL có thể dùng để tạo, hủy các đối tượng, như là các bảng, trên server csdl và thực hiện một số việc như nhập dữ liệu vào chúng hoặc truy vấn để lấy dữ liệu.
- Ngôn ngữ giao tác Transact-SQL là một sự bổ sung(implementation ) ngôn ngữ chuẩn SQL của hãng Microsoft.
- T-SQL, ngôn ngữ này cung cấp một cách thức chuẩn hóa để giao tiếp với csdl.
- Ngôn ngữ Transact-SQL là sự cải tiến của SQL. SQL là tên thường gọi của ngôn ngữ csdl quan hệ chuẩn của Học viện quốc gia Hoa kỳ: ANSI-SQL (American National Standards Institute).
- Nó cung cấp một ngôn ngữ bao hàm toàn diện (comprehensive language), hỗ trợ việc định nghĩa bảng, chèn, xóa, cập nhật, và truy cập dữ liệu trong bảng.

# Transact-SQL 1-2

Transact-SQL là một ngôn ngữ mạnh cung cấp các tính năng như kiểu dữ liệu, các đối tượng tạm, và các thủ tục lưu mở rộng.

Các con trỏ có khả năng cuộn (Scrollable cursors), xử lý điều kiện, điều khiển giao tác (transaction control), ngoại lệ và quản lý lỗi cũng là một vài tính năng được hỗ trợ trong bởi Transact-SQL.

Ngôn ngữ Transact-SQL trong SQL Server 2012 cung cấp hiệu suất được cải tiến, chức năng tăng cường và các tính năng được cải tiến.

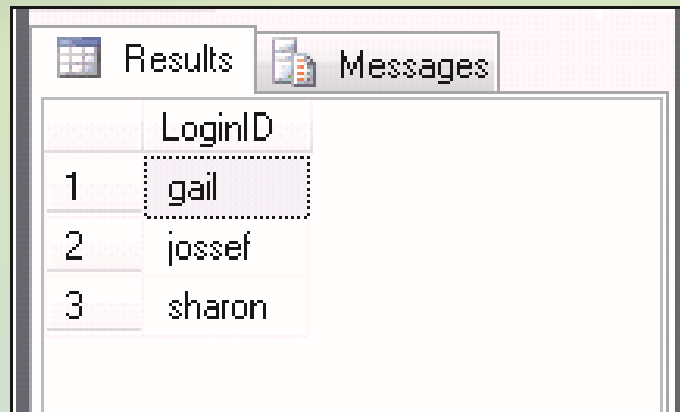
Các cải tiến gồm có các hàm tính toán (scalar functions), phân trang, sequences, khám phá meta-data, và hỗ trợ quản lý lỗi tốt hơn.

# Transact-SQL 2-2

Đoạn mã dưới đây minh họa câu lệnh Transact-SQL, câu lệnh SELECT, nó được sử dụng để lấy tất cả các bản ghi của nhân viên có JobTitle là 'Design Engineer' từ bảng Employee.

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

➤ Hình dưới đây trình bày kết quả của câu lệnh SELECT:



	LoginID
1	gail
2	jossef
3	sharon

- Transact-SQL có nhiều thành phần cú pháp được dùng bởi hầu hết các câu lệnh.
- Những thành phần này bao gồm kiểu dữ liệu, vị từ (predicates/các biểu thức điều kiện), các hàm, các biểu thức, điều khiển luồng, chú thích và phân lô (batch separators).

# Data Definition Language (DDL)

DDL(Ngôn ngữ định nghĩa dữ liệu) được sử dụng để định nghĩa và quản lý tất cả các thuộc tính(attributes) và đặc tính (properties) của csdl, bao gồm các bố cục dòng (row layouts), các định nghĩa cột, các cột khóa, các vị trí tập tin, và chính sách lưu trữ (storage strategy).

Các câu lệnh DDL được sử dụng để xây dựng và chỉnh sửa cấu trúc các bảng(table) và các đối tượng khác như views, triggers, stored procedures, vv...

Với mỗi đối tượng, thường có các câu lệnh CREATE, ALTER, và DROP (ví dụ như, CREATE TABLE, ALTER TABLE, và DROP TABLE).

Hầu hết các câu lệnh DDL đều có dạng như sau:



**CREATE**  
object\_name

**ALTER**  
object\_name

**DROP**  
object\_name

Trong câu lệnh DDL, object\_name có thể là một bảng(table), view, trigger, stored procedure, vv...

# Data Manipulation Language (DML)

DML(Ngôn ngữ thao tác dữ liệu) được sử dụng để lấy (select), chèn (insert), cập nhật (update), hoặc xóa (delete) dữ liệu trong các đối tượng được định nghĩa bằng DDL.

Tất cả các csdl người dùng có thể sử dụng những dòng lệnh này trong các thao tác thường hay thực hiện trên csdl.

Dưới đây là các câu lệnh DML khác nhau:

Câu lệnh  
**SELECT**

Câu lệnh  
**INSERT**

Câu lệnh  
**UPDATE**

Câu lệnh  
**DELETE**



# Data Control Language (DCL)

Dữ liệu là phần quan trọng nhất trong csdl, do vậy các bước nên được kiểm tra phù hợp để ko có người dùng bất hợp lệ truy cập dữ liệu.

DCL(Ngôn ngữ quan lý dữ liệu) dùng để quản lý các quyền hạn trên các đối tượng csdl

Các quyền hạn được quản lý bằng việc dùng các câu lệnh GRANT, REVOKE, and DENY.

Các câu lệnh DCL cũng được dùng cho việc bảo mật csdl. Dưới đây là ba câu lệnh DCL cơ bản:

**Câu lệnh  
GRANT**

**Câu lệnh  
REVOKE**

**Câu lệnh  
DENY**



# Kiểu dữ liệu 1-6

- Một kiểu dữ liệu là một thuộc tính định nghĩa loại dữ liệu mà đối tượng có thể chứa.
- Kiểu dữ liệu phải được sử dụng cho các đối tượng như các cột, các tham số, các biến và các hàm trả về giá trị và các thủ tục lưu có mã số trả về.
- Ngôn ngữ Transact-SQL có một số các kiểu dữ liệu cơ bản, như là varchar, text, và int
- Tất cả dữ liệu lưu trữ trong SQL Server phải tương thích với một trong các kiểu cơ bản.
- Dưới đây là các đối tượng sử dụng kiểu dữ liệu:

Các cột có trong bảng và trong view

Các tham số trong của thủ tục lưu

Các biến

Các hàm Transact-SQL trả về giá trị dữ liệu

Các thủ tục lưu (Stored procedures) trả về một mã số kiểu nguyên

- Ngoài ra trong SQL Server 2012 còn có các cột, các biến, và biểu thức được gán kiểu dữ liệu.

# Kiểu dữ liệu 2-6

- SQL Server 2012 hỗ trợ ba loại kiểu dữ liệu:

## Các kiểu dữ liệu hệ thống

- Những kiểu dữ liệu này được cung cấp bởi SQL Server 2012.
- Bảng dưới đây trình bày các kiểu dữ liệu hệ thống của SQL Server 2012.

Category	Data Type	Description
Exact Numerics	int	A column of this type occupies 4 bytes of memory space. Is typically used to hold integer values. Can hold integer data from $-2^{31}$ (-2,147,483,648) to $2^{31}-1$ (2,147,483,647).
	smallint	A column of this type occupies 2 bytes of memory space. Can hold integer data from -32,768 to 32,767.
	tinyint	A column of this type occupies 1 byte of memory space. Can hold integer data from 0 to 255.
	bigint	A column of this type occupies 8 bytes of memory space. Can hold data in the range $-2^{63}$ (-9,223,372,036,854,775,808) to $2^{63}-1$ (9,223,372,036,854,775,807).
	numeric	A column of this type has fixed precision and scale.
	money	A column of this type occupies 8 bytes of memory space. Represents monetary data values ranging from $-2^{63}/10000$ (-922,337,203,685,477.5808) to $2^{63}-1$ (922,337,203,685,477.5807).

# Kiểu dữ liệu 3-6

Category	Data Type	Description
Approximate Numerics	float	A column of this type occupies 8 bytes of memory space. Represents floating point number ranging from $-1.79E+308$ through $1.79E+308$ .
	real	A column of this type occupies 4 bytes of memory space. Represents floating precision number ranging from $-3.40E+38$ through $3.40E+38$ .
Date and Time	datetime	Represents date and time. Stored as two 4-byte integers.
	smalldatetime	Represents date and time.
Character String	char	Stores character data that is fixed-length and non-Unicode.
	varchar	Stores character data that is variable-length and non-Unicode with a maximum of 8,000 characters.
	text	Stores character data that is variable-length and non-Unicode with a maximum length of $2^{31} - 1$ (2,147,483,647) characters.
Unicode Types	nchar	Stores Unicode character data of fixed-length.
	nvarchar	Stores variable-length Unicode character data.

# Kiểu dữ liệu 4-6

Category	Data Type	Description
Other Data Types	timestamp	A column of this type occupies 8 bytes of memory space. Can hold automatically generated, unique binary numbers that are generated for a database.
	binary(n)	Stores fixed-length binary data with a maximum length of 8000 bytes.
	varbinary(n)	Stores variable-length binary data with a maximum length of 8000 bytes.
	image	Stores variable-length binary data with a maximum length of $2^{30}-1$ (1,073,741,823) bytes.
	uniqueidentifier	A column of this type occupies 16 bytes of memory space. Also, stores a globally unique identifier (GUID).

## Kiểu dữ liệu bí danh (Alias Data Types)

- Được xây dựng dựa trên các kiểu dữ liệu hệ thống.
- Được dùng khi có nhiều bảng có các cột cùng kiểu dữ liệu, có đặc tính tương tự nhau như cùng độ dài, không được phép null.
- Có thể tạo ra khi muốn sử dụng nó chung cho tất cả những bảng này.

## Kiểu dữ liệu 5-6

- Kiểu dữ liệu ánh xạ, có thể được tạo bằng lệnh CREATE TYPE.
- Cú pháp 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 lược đồ(schema) mà kiểu dữ liệu bí danh sẽ được tạo trong đó. Một lược đồ là một tập hợp các đối tượng như các bảng, các view, và vv... trong csdl.
- type\_name: xác định tên của kiểu dữ liệu bí danh đang được tạo.
- base\_type: xác định tên của kiểu dữ liệu hệ thống mà kiểu bí danh sẽ được tạo dựa trên nó.
- precision and scale: chỉ ra độ chính xác và tỷ lệ cho dữ liệu dạng số.
- NULL | NOT NULL: chỉ ra kiểu dữ liệu có hay không lưu trữ giá trị null không.

## Data Types 6-6

- Đoạn mã sau đây trình bày cách tạo kiểu dữ liệu bí danh bằng câu lệnh `CREATE TYPE`.

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

- Trong đoạn mã, kiểu dữ liệu dựng sẵn `varchar` được lưu trữ trong kiểu dữ liệu mới có tên `usertype` bằng câu lệnh `CREATE TYPE`.

### Kiểu dữ liệu do người dùng định nghĩa (User-defined Types)

- Được tạo bằng các ngôn ngữ lập trình được hỗ trợ bởi .NET Framework.

# Các thành phần ngôn ngữ Transact-SQL

- Được sử dụng trong SQL Server 2012 để làm việc với dữ liệu được nhập vào csdl SQL Server.
- Gồm có các thành phần sau:

Vị từ (Predicates)

Toán tử

Hàm

Biến

Biểu thức

Điều khiển luồng

Lỗi

Giao tác

Chú thích

Phân lô (Batch Separators)



# Các vị từ và các toán tử 1-3

- Các vị từ (Predicates) được sử dụng để định giá xem biểu thức là TRUE, FALSE, hoặc UNKNOWN(không xác định) . Một số vị từ có sẵn trong Transact-SQL:

IN	<ul style="list-style-type: none"> <li>• Xác định xem giá trị chỉ ra có khớp(match) với bất kỳ giá trị nào trong một truy vấn con hoặc một danh sách không.</li> </ul>
BETWEEN	<ul style="list-style-type: none"> <li>• Chỉ ra một miền giá trị để kiểm tra.</li> </ul>
LIKE	<ul style="list-style-type: none"> <li>• Được sử dụng để so khớp các kí tự với một mẫu được chỉ ra.</li> </ul>
CONTAINS	<ul style="list-style-type: none"> <li>• Tìm kiếm chính xác(precise) hoặc tương đối (less precise) khớp với một từ và đoạn văn, từ bên trong một khoảng.</li> </ul>

- Một số ví dụ sử dụng vị từ:

Vị từ	Ví dụ
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);

# Các vị từ và các toán tử 2-3

- Các toán tử được sử dụng để thực hiện tính toán số học, so sánh, ghép nối, hoặc gán giá trị.
- SQL Server có 7 nhóm toán tử. Bảng sau đây mô tả các toán tử khác nhau được hỗ trợ trong SQL Server 2012.

Toán tử	Mô tả	Ký hiệu
So sánh	So sánh một giá trị với một giá trị khác hoặc một biểu thức	=, <, >, >=, <=, !=, !>
Luận lý	Kiểm tra tính đúng đắn của một điều kiện	AND, OR, NOT
Toán học	Thực hiện hiện tính toán số học như cộng, trừ, nhân, chia	+, -, *, /, %
Ghép nối	Ghép hai chuỗi thành một chuỗi	+
Gán	Gán giá trị cho một biến	=

- Bảng dưới đây trình bày độ ưu tiên của predicates và toán tử :

Order	Operators
1	() Parentheses
2	*, /, %
3	+, -
4	=, <, >, >=, <=, !=, !>

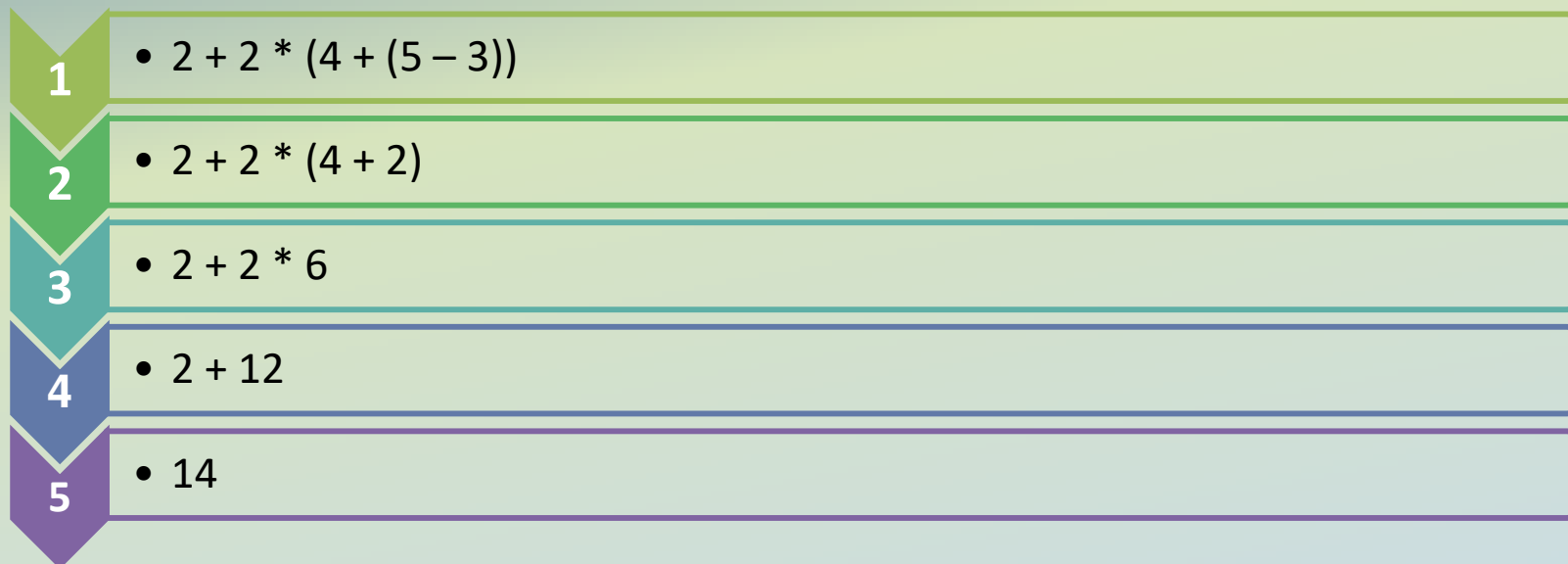
Order	Operators
5	NOT
6	AND
7	BETWEEN, IN, CONTAINS, LIKE, OR
8	=

## Các vị từ và các toán tử 3-3

- Đoạn mã sau đây trình bày thực thi toán tử phù hợp với độ ưu tiên:

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

- Sau đây là các bước đi đến kết quả:



- Đoạn mã sẽ hiển thị 14.

# Hàm 1-2

- Là một bộ các lệnh Transact-SQL được dùng để thực hiện một số bài toán.
- SQL Server 2012 có bốn loại hàm như sau:

## Các hàm trả về tập các dòng (Rowset Functions)

- Được dùng trả về một đối tượng có thể được sử dụng thay thế tham chiếu bảng.
- Ví dụ, OPENDATASOURCE, OPENQUERY, OPENROWSET, và OPENXML.

## Các hàm tổng hợp (Aggregate Functions)

- Cung cấp các hàm tổng hợp để trợ giúp thống kê khối lượng dữ liệu lớn.
- Ví dụ, SUM, MIN, MAX, AVG, COUNT, COUNTBIG, vv...

## Các hàm xếp hạng (Ranking Functions)

- Có thể thực hiện nhiều bài toán như tạo mảng, phát sinh một dãy số, tìm thứ hạng, vv một cách dễ dàng và nhanh hơn khi sử dụng các hàm xếp hạng.
- Ví dụ, RANK, DENSE\_RANK, NTILE, and ROW\_NUMBER.

## Các hàm tính toán (Scalar functions)

- Đầu vào là một giá trị và đầu ra cũng nhận được một giá trị.

## Hàm 2-2

- Bảng sau đây trình bày các hàm tính toán được sử dụng phổ biến trong SQL:

Hàm	Mô tả	Ví dụ
Hàm chuyển đổi	Hàm chuyển đổi được sử dụng để chuyển đổi kiểu giá trị từ kiểu này sang kiểu khác. Ngoài ra, nó có thể sử dụng để đặt được các kiểu định dạng ngày tháng đặc biệt.	CONVERT
Hàm ngày tháng và thời gian	Các hàm ngày tháng và thời gian được sử dụng để thao tác trên các giá trị ngày tháng và thời gian. Chúng hữu ích để thực hiện 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 thao tác đại số trên các giá trị số.	RAND, ROUND, POWER, ABS, CEILING, FLOOR
Hàm hệ thống	SQL Server cung cấp các hàm hệ thống, trả về metadata hoặc các thiết lập cấu hình.	HOST_ID, HOST_NAME, ISNULL
Hàm xử lý chuỗi	Các hàm xử lý chuỗi được sử dụng cho các chuỗi đầu vào như char và varchar. Kết quả trả về có thể là một chuỗi hoặc một số.	SUBSTRING, LEFT, RIGHT, LEN, DATALENGTH, REPLACE, REPLICATE, UPPER, LOWER, RTRIM, LTRIM

- Còn có các hàm tính toán khác như hàm con trỏ (cursor functions), các hàm logic, các hàm metadata, các hàm bảo mật, vv có sẵn trong SQL Server 2012.

Biến là một đối tượng để lưu trữ một giá trị dữ liệu. Biến có thể phân thành 2 loại là biến cục bộ và biến toàn cục.

Trong Transact-SQL, biến cục bộ được tạo và sử dụng để lưu trữ tạm thời trong khi các câu lệnh SQL được thực thi.

Dữ liệu có thể truyền tới các câu lệnh SQL bằng biến cục bộ. Tên biến cục bộ phải bắt đầu với kí hiệu '@'.

Biến toàn cục là các biến xây dựng sẵn, được định nghĩa và duy trì bởi hệ thống. Biến hệ thống trong SQL Server được bắt đầu bằng hai kí hiệu '@'.

Có thể lấy được giá trị của biến bất kỳ bằng câu truy vấn SELECT đơn giản.

## Biểu thức 1-2

Một biểu thức là sự kết hợp của các định danh, các giá trị và các toán tử mà SQL Server có thể định giá theo thứ tự để đạt được kết quả.

Biểu thức có thể sử dụng ở nhiều nơi khác nhau khi truy xuất hoặc thay đổi dữ liệu.

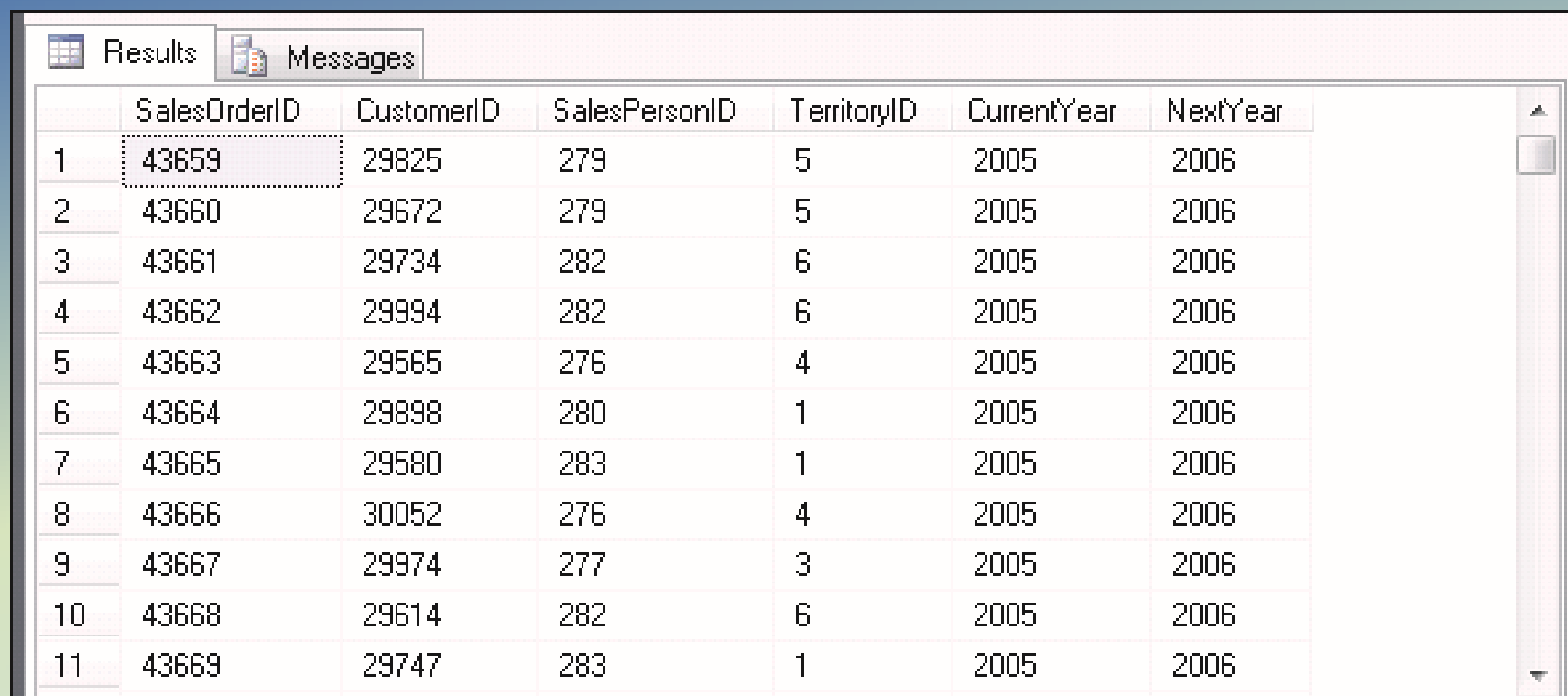
Đoạn mã sau đây minh họa một biểu thức thao tác trên cột, cộng một số nguyên với kết quả của hàm YEAR trên cột `datetime`:

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



## Biểu thức2-2

- Hình sau đây trình bày kết quả của biểu thức :



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with 11 rows and 7 columns. The columns are SalesOrderID, CustomerID, SalesPersonID, TerritoryID, CurrentYear, and NextYear. The first row is highlighted with a dotted border.

	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

# Điều khiển luồng, Lỗi, và Giao tác

- Mặc dù Transact-SQL là ngôn ngữ chủ yếu để lấy dữ liệu, nó còn hỗ trợ các câu lệnh điều khiển luồng thực thi và tìm lỗi.
- Điều khiển luồng xác định luồng thực thi các câu lệnh Transact-SQL, các khối lệnh, các hàm do người dùng định nghĩa, và các thủ tục lưu (stored procedures).
- Bảng sau đây trình bày một số câu lệnh điều khiển luồng được sử dụng phổ biến trong Transact-SQL:

Câu lệnh điều khiển luồng	Mô tả
IF...ELSE	Cung cấp điều khiển rẽ nhánh dựa trên kiểm tra luận lý(logical test).
WHILE	Lặp câu lệnh hoặc khối lệnh cho tới khi điều kiện là đúng(true)
BEGIN...END	Định nghĩa phạm vi của một khối các lệnh Transact-SQL.
TRY...CATCH	Định nghĩa cấu trúc cho việc thực thi và quản lý lỗi.
BEGIN TRANSACTION	Đánh dấu khối các câu lệnh là một phần của một giao tác tường minh.

## Chú thích 1-2

- Chú thích là một chuỗi văn bản có tính mô tả, còn được biết như các ghi chú, trong mã chương trình, nó được bỏ qua bởi trình biên dịch.
- Chú thích được đặt bên trong mã nguồn của câu lệnh đơn, một lô (batch), hoặc một thủ tục lưu.
- Chú thích giải thích mục đích của chương trình, các điều kiện thực đặc biệt, và cung cấp xem lại thông tin lịch sử.
- Microsoft SQL Server hỗ trợ hai kiểu chú thích:

### - - (Hai dấu trừ)

- Toàn bộ dòng mã(code) hoặc một phần đoạn code có thể được đánh dấu chú thích nếu hai dấu trừ (- -) được đặt ở phần đầu.
- Phần còn lại của dòng trở thành chú thích.
- Đoạn mã dưới đây minh họa cách sử dụng kiểu chú thích này:

```
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
```

## Chú thích 2-2

```
/* ... */
```

- Những kí tự chú thích này có thể lẫn trong cùng dòng mã lệnh, trên nhiều dòng hoặc thậm chí bên trong đoạn mã thực thi.
- Mọi thứ nằm giữa cặp chú thích /\* và \*/ được xem là một phần của chú thích.
- Có thể sử dụng để chú thích trên nhiều dòng.
- Dưới đây là đoạn code minh họa cách sử dụng kiểu chú thích này:

```
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
```

# Phân lô (Batch Separators)

Một lô là một tập hợp có một hoặc nhiều câu lệnh Transact-SQL được gửi một lần từ một ứng dụng tới SQL Server để thực thi.

Các câu lệnh Transact-SQL trong lô (batch) được biên dịch thành một khối duy nhất, được gọi là kế hoạch thực thi (execution plan).

Các câu lệnh trong kế hoạch thực thi sau đó được thực thi một lần tại một thời điểm.

Tiến trình mà trong đó một tập các dòng lệnh được xử lý một lần tại một thời điểm từ một lô dòng lệnh được gọi là xử lý theo lô (batch processing).

Một phân lô được quản lý bởi công cụ SQL Server client như là SSMS để thực thi các dòng lệnh. Ví dụ, bạn cần chỉ ra từ khóa GO như là một sự phân chia lô trong SSMS.

- Một ví dụ của câu lệnh phân lô được cho trong đoạn mã dưới đây:

```
USE AdventureWorks2012
SELECT * FROM HumanResources.Employee
GO
```

- Ở đây, hai câu lệnh sẽ được nhóm thành một kế hoạch thực thi (execution plan) nhưng một lệnh được thực thi tại một thời điểm.
- Từ khóa GO ký hiệu kết thúc một lô.

# Lý thuyết tập hợp (Set Theory)

Lý thuyết tập hợp là một cơ sở toán học được sử dụng trong mô hình csdl quan hệ

Một tập hợp là một tập các đối tượng duy nhất(distinct)

Ví dụ, tất cả các nhân viên trong bảng Employee có thể được coi như là một tập.

- Bảng sau đây trình bày các ứng dụng khác trong lý thuyết tập hợp và ứng dụng phù hợp của chúng vào các truy vấn SQL Server.

Ứng dụng lý thuyết tập hợp	Ứng dụng trong các truy vấn SQL Server
Tác động đến toàn bộ tập cùng một lúc.	Truy vấn toàn bộ bảng cùng một lúc.
Sử dụng các khai báo, xử lý dựa trên tập hợp.	Dùng các thuộc tính trong SQL Server để lấy dữ liệu được chỉ ra.
Các phần tử trong tập phải duy nhất.	Định nghĩa các khóa duy nhất trong bảng.
Không có chỉ thị sắp xếp	Các kết quả của truy vấn không được lấy theo thứ tự bất kỳ nào.

# Luận lý vị từ (Predicate logic)

- Logic vị từ(hay luận lý vị từ) là một framework toán học bao gồm các bài kiểm tra logic cho ra một kết quả. Kết quả luôn được hiển thị như là đúng hoặc sai.
- Trong Transact-SQL, các biểu thức như WHERE và CASE là các biểu thức dựa trên logic vị từ.
- Logic vị từ cũng sử dụng trong một số tình huống khác trong Transact-SQL. Sau đây là một số ứng dụng của logic vị từ trong Transact-SQL:

Đảm bảo toàn vẹn dữ liệu bằng CHECK constraint

Điều khiển luồng bằng câu lệnh IF

Ghép (Joining) các bảng bằng bộ lọc ON

Lọc dữ liệu trong truy vấn bằng mệnh đề WHERE và HAVING

Cung cấp điều kiện logic cho biểu thức CASE

Định nghĩa các truy vấn con



# Thứ tự logic của các toán tử trong câu lệnh SELECT 1-5

- Cùng với cú pháp của các thành phần SQL Server khác nhau, người dùng SQL Server cũng cần phải biết việc xử lý toàn bộ câu truy vấn được thi hành như thế nào.
- Xử lý này là một xử lý logic ngắt(break) truy vấn và thực thi truy vấn phù hợp với chuỗi định nghĩa trước trong SQL Server 2012.
- Câu lệnh SELECT là một truy vấn sẽ được dùng để giải thích xử lý logic của việc thực thi truy vấn.
- Cú pháp câu lệnh SELECT như sau:

## Cú pháp:

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

# Thứ tự logic của các toán tử trong câu lệnh SELECT 2-5

- Bảng giải thích các thành phần của câu lệnh SELECT:

Thành phần	Mô tả
SELECT <select list>	Định nghĩa các cột được trả về
FROM <table source>	Định nghĩa các bảng được truy vấn
WHERE <search condition>	Lọc các dòng bằng cách dùng các biểu thức
GROUP BY <group by list>	Dàn (arranges) xếp các dòng bằng cách nhóm
HAVING <search condition>	Sử dụng biểu thức để lọc các nhóm
ORDER BY <order by list>	Sắp xếp kết quả

- Đoạn mã sau đây minh họa câu lệnh SELECT:

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

# Thứ tự logic của các toán tử trong câu lệnh SELECT 3-5

➤ Trong ví dụ trên, thứ tự mà SQL Server sẽ thực thi câu lệnh SELECT như sau:

1

- Thứ nhất, mệnh đề `FROM` được định giá để xác định các bảng nguồn sẽ được truy vấn.

2

- Tiếp theo, mệnh đề `WHERE` được đánh giá để lọc cá dòng trong bảng nguồn.
- Việc lọc này được xác định bởi biểu thức được đề cập trong mệnh đề `WHERE`.

3

- Tiếp theo đến, mệnh đề `GROUP BY` được định giá.
- Mệnh đề này dàn xếp các giá trị đã được lọc nhận được từ mệnh đề `WHERE`.

4

- Kế tiếp, mệnh đề `HAVING` được định giá dựa trên biểu thức điều kiện được cung cấp.

5

- Tiếp theo, mệnh đề `SELECT` được thi hành để xác định xem các cột nào sẽ xuất hiện trong kết quả truy vấn.

6

- Cuối cùng, câu lệnh `ORDER BY` được định giá để hiển thị kết quả đầu ra.

# Thứ tự logic của các toán tử trong câu lệnh SELECT 3-5

1

- Thứ nhất, mệnh đề `FROM` được định giá để xác định các bảng nguồn sẽ được truy vấn.

2

- Tiếp theo, mệnh đề `WHERE` được đánh giá để lọc cá dòng trong bảng nguồn.
- Việc lọc này được xác định bởi biểu thức được đề cập trong mệnh đề `WHERE`.

3

- Tiếp theo đến, mệnh đề `GROUP BY` được định giá.
- Mệnh đề này dàn xếp các giá trị đã được lọc nhận được từ mệnh đề `WHERE`.

4

- Kế tiếp, mệnh đề `HAVING` được định giá dựa trên biểu thức điều kiện được cung cấp.

5

- Tiếp theo, mệnh đề `SELECT` được thi hành để xác định xem các cột nào sẽ xuất hiện trong kết quả truy vấn.

6

- Cuối cùng, câu lệnh `ORDER BY` được định giá để hiển thị kết quả đầu ra.

# Thứ tự logic của các toán tử trong câu lệnh SELECT 4-5

➤ Thứ tự thực thi câu lệnh SELECT trong đoạn code sẽ như dưới đây:

1

- `SELECT SalesPersonID, YEAR(OrderDate) AS OrderYear`

2

- `FROM SalesOrderHeader`

3

- `WHERE CustomerID = 30084`

4

- `GROUP BY SalesPersonID, YEAR(OrderDate)`

5

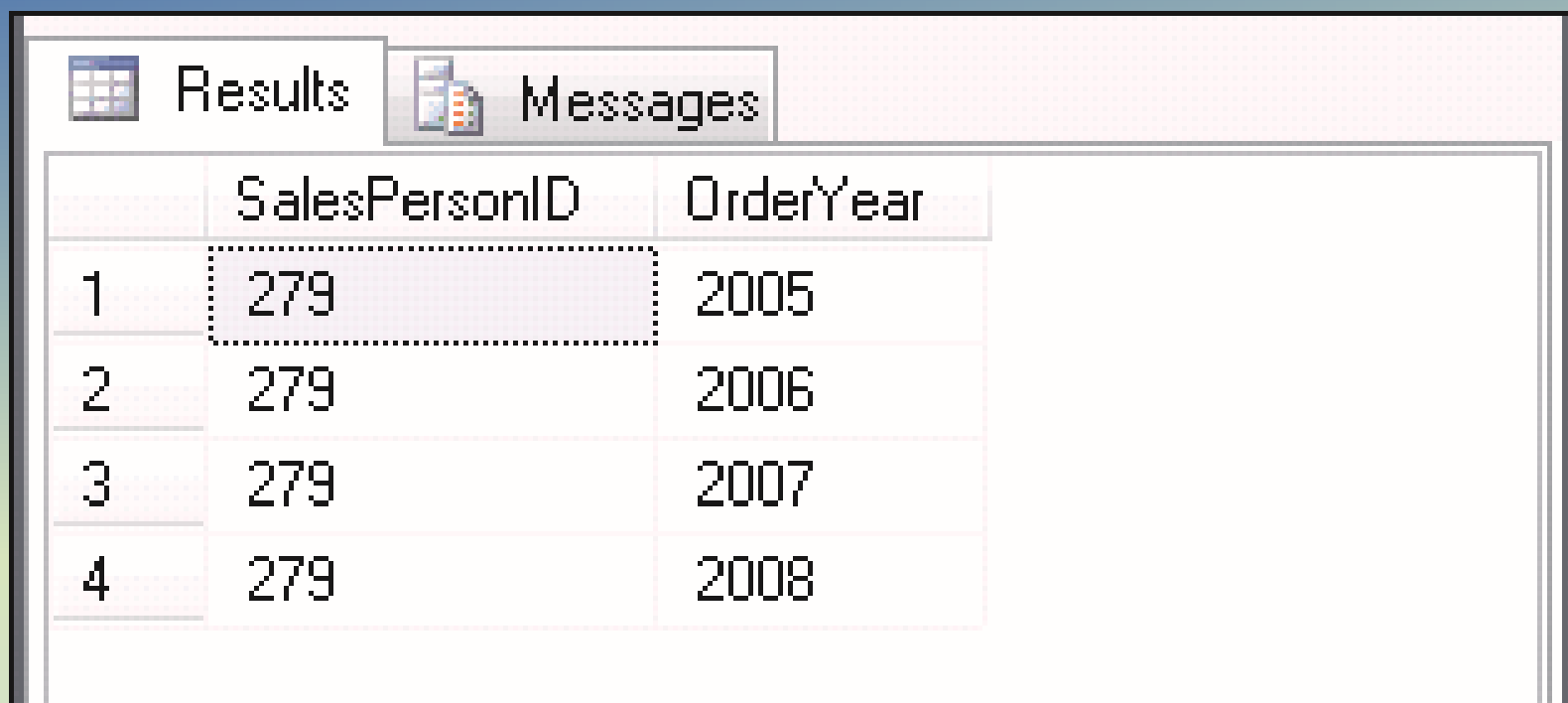
- `HAVING COUNT(*) > 1`

6

- `ORDER BY SalesPersonID, OrderYear;`

# Thứ tự logic của các toán tử trong câu lệnh SELECT 5-5

- Hình dưới đây trình bày kết quả của câu lệnh SELECT:



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

# Tóm tắt

- Transact-SQL là một ngôn ngữ mạnh mẽ mà cung cấp các tính năng như các kiểu dữ liệu, các đối tượng tạm thời, và các thủ tục lưu trữ mở rộng.
- SQL Server hỗ trợ ba nhóm lệnh Transact-SQL, cụ thể là, DDL, DML, và DCL.
- Một kiểu dữ liệu là một thuộc tính xác định các kiểu dữ liệu mà một đối tượng có thể chứa.
- Các thành phần của ngôn ngữ Transact-SQL bao gồm các vị từ(biểu thức điều kiện), các toán tử, các hàm, các biến, các biểu thức, điều khiển luồng, lỗi, và các giao dịch, chú thích, và phân chia lô(batch).
- Tập hợp và logic vị từ là hai cơ sở toán học được sử dụng trong SQL Server 2012.
- Lý thuyết tập hợp là một nền tảng toán học được sử dụng trong mô hình cơ sở dữ liệu quan hệ, trong đó một tập là một bộ sưu tập của các đối tượng duy nhất coi như một toàn thể.
- Logic vị từ là một framework toán học bao gồm các kiểm tra logic cho ra một kết quả.