

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

Bài 05

Transact-SQL

Tóm tắt

- Giới thiệu ngôn ngữ Transact-SQL.
- Các nhóm lệnh
- Kiểu dữ liệu
- Các thành phần ngôn ngữ T-SQL
- Câu lệnh SELECT

Giới thiệu

- **SQL** là ngôn ngữ phổ biến sử dụng trong thế giới csdl.
- Mỗi sản phẩm csdl của các nhà cung cấp khác nhau sử dụng SQL như là **ngôn ngữ truy vấn chính** (có đôi chút khác biệt).
- Hãng Microsoft tạo nên **Transact-SQL (T-SQL)** dựa trên ngôn ngữ gốc SQL dành riêng cho sản phẩm **SQL Server**.



Giới thiệu

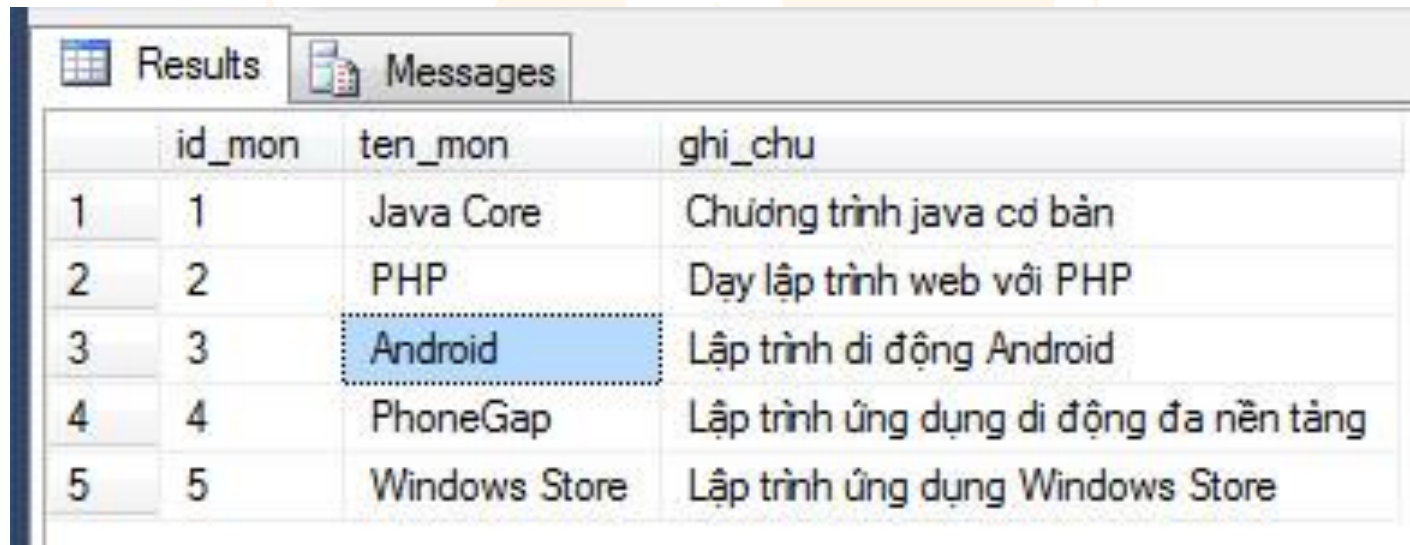
- **T-SQL** là **ngôn ngữ mạnh mẽ** cung cấp các tính năng như kiểu dữ liệu, đối tượng tạm, các thủ tục lưu trữ mở rộng.
- **T-SQL** hỗ trợ bắt **ngoại lệ, quản lý lỗi**.
- **T-SQL** trong SQL Server 2014 có **hiệu suất cải tiến**, chức năng tăng cường và nhiều tính năng được nâng cấp.
- **T-SQL** có hỗ trợ các hàm tính toán (scalar functions), phân trang, sequence, meta-data, và nhiều tính năng khác.

Giới thiệu

Đoạn mã sau minh họa câu lệnh SELECT trong T-SQL dùng để truy vấn dữ liệu bảng Monhoc:

```
SELECT * FROM MonHoc
```

Kết quả:



The screenshot shows a SQL Server interface with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with 5 rows and 4 columns: 'id_mon', 'ten_mon', and 'ghi_chu'. The third row, containing 'Android', is highlighted with a blue selection box. The background of the slide features a faint, stylized map of the world.

| | id_mon | ten_mon | ghi_chu |
|---|--------|---------------|--|
| 1 | 1 | Java Core | Chương trình java cơ bản |
| 2 | 2 | PHP | Dạy lập trình web với PHP |
| 3 | 3 | Android | Lập trình di động Android |
| 4 | 4 | PhoneGap | Lập trình ứng dụng di động đa nền tảng |
| 5 | 5 | Windows Store | Lập trình ứng dụng Windows Store |

Nhóm lệnh

Ngôn ngữ T-SQL có thể chia làm 3 nhóm lệnh:

- Nhóm **định nghĩa dữ liệu (DDL)**: Gồm các lệnh tạo, thay đổi các bảng dữ liệu (**Create, Drop, Alter, ...**)
- Nhóm **thao tác dữ liệu (DML)**: Gồm các lệnh làm thay đổi dữ liệu (**Insert, Delete, Update,...**) lưu trong các bảng
- Nhóm **điều khiển dữ liệu (DCL)**: Gồm các lệnh quản lý quyền truy nhập vào dữ liệu và các bảng (**Grant, Revoke, ...**)

Nhóm lệnh

Data Definition Language (DDL):

- 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).

CREATE
object_name

ALTER
object_name

DROP
object_name

Nhóm lệnh

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.

Câu lệnh
SELECT

Câu lệnh
INSERT

Câu lệnh
UPDATE

Câu lệnh
DELETE

Nhóm lệnh

Data Control Language (DCL):

- Dữ liệu là phần quan trọng nhất trong csdl, do vậy các bước nên được kiểm tra phù hợp để không có người dùng **bất hợp lệ truy cập dữ liệu**.
- DCL(Ngôn ngữ quản 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**.

Câu lệnh
GRANT

Câu lệnh
REVOKE

Câu lệnh
DENY

Kiểu dữ liệu

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

Kiểu dữ liệu

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

Kiểu dữ liệu

| 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

| 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

| 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). |

Thành phần ngôn ngữ

T-SQL là ngôn ngữ gồm các thành phần:

Vị từ (Predicates)

Toán tử

Hàm

Biến

Biểu thức

Điều khiển luồng

Lỗi

Giao tác

Chú thích

Phân lô (Batch Separators)

Thành phần ngôn ngữ

Vị từ:

IN

- 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.

BETWEEN

- Chỉ ra một miền giá trị để kiểm tra.

LIKE

- Được sử dụng để so khớp các kí tự với một mẫu được chỉ ra.

CONTAINS

- Tìm kiếm chính xác(precise) hoặc tương đối (less precise) khớp với một từ và đoạn văn, từ bên trong một khoảng.

| 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); |

Thành phần ngôn ngữ

Toán tử:

| 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 | = |

Độ ưu tiên:

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

Thành phần ngôn ngữ

Ví dụ thực thi toán tử:

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

1

• $2 + 2 * (4 + (5 - 3))$

2

• $2 + 2 * (4 + 2)$

3

• $2 + 2 * 6$

4

• $2 + 12$

5

• **14**

Thành phần ngôn ngữ

Hàm:

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.

Thành phần ngôn ngữ

Biến:

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.

Thành phần ngôn ngữ

Biểu thức:

Một biểu thức là sự kết hợp của các định danh, các giá trị và cá 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 ma_lop, ten_lop, he_dao_tao, khai_giang,  
YEAR(khai_giang) + 2 AS N'Năm tốt nghiệp'  
FROM LOPHOC
```

| Results | | Messages | | | |
|---------|--------|------------------------|------------|------------|----------------|
| | ma_lop | ten_lop | he_dao_tao | khai_giang | Năm tốt nghiệp |
| 1 | C1507G | Lập trình viên cao cấp | ACCP 2013 | 2015-07-09 | 2017 |

Thành phần ngôn ngữ

Điều khiển luồng, lỗi, giao dịch:

| 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. |

Thành phần ngôn ngữ

Chú thích:

-- (Hai dấu trừ)

```
USE AdventureWorks2014
-- HumanResources.Employee table contains the details of an employee.
-- This statement retrieves all the rows of the table
-- HumanResources.Employee.
SELECT * FROM HumanResources.Employee
```

/* ... */

```
USE AdventureWorks2014
/* HumanResources.Employee table contains the details of an
employee.
This statement retrieves all the rows of the table
HumanResources.Employee. */
SELECT * FROM HumanResources.Employee
```

Thành phần ngôn ngữ

Phân lô với lệnh GO:

```
USE AdventureWorks2014  
SELECT * FROM HumanResources.Employee  
GO
```


Lệnh SELECT

Cú pháp lệnh SELECT thông thường như sau:

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

| 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 |

Lệnh SELECT

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.

Tóm tắt bài học

- **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).

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

Thank for watching!

