

NỘI DUNG

Các thành phần của SQL

Ngôn ngữ định nghĩa dữ liệu

Ngôn ngữ thao tác dữ liệu

Hàm sử dụng trong T-SQL

Truy vấn dữ liệu

Gộp, xoay dữ liệu

Một số vấn đề khác: biến toàn cục, bảng tạm, null,...

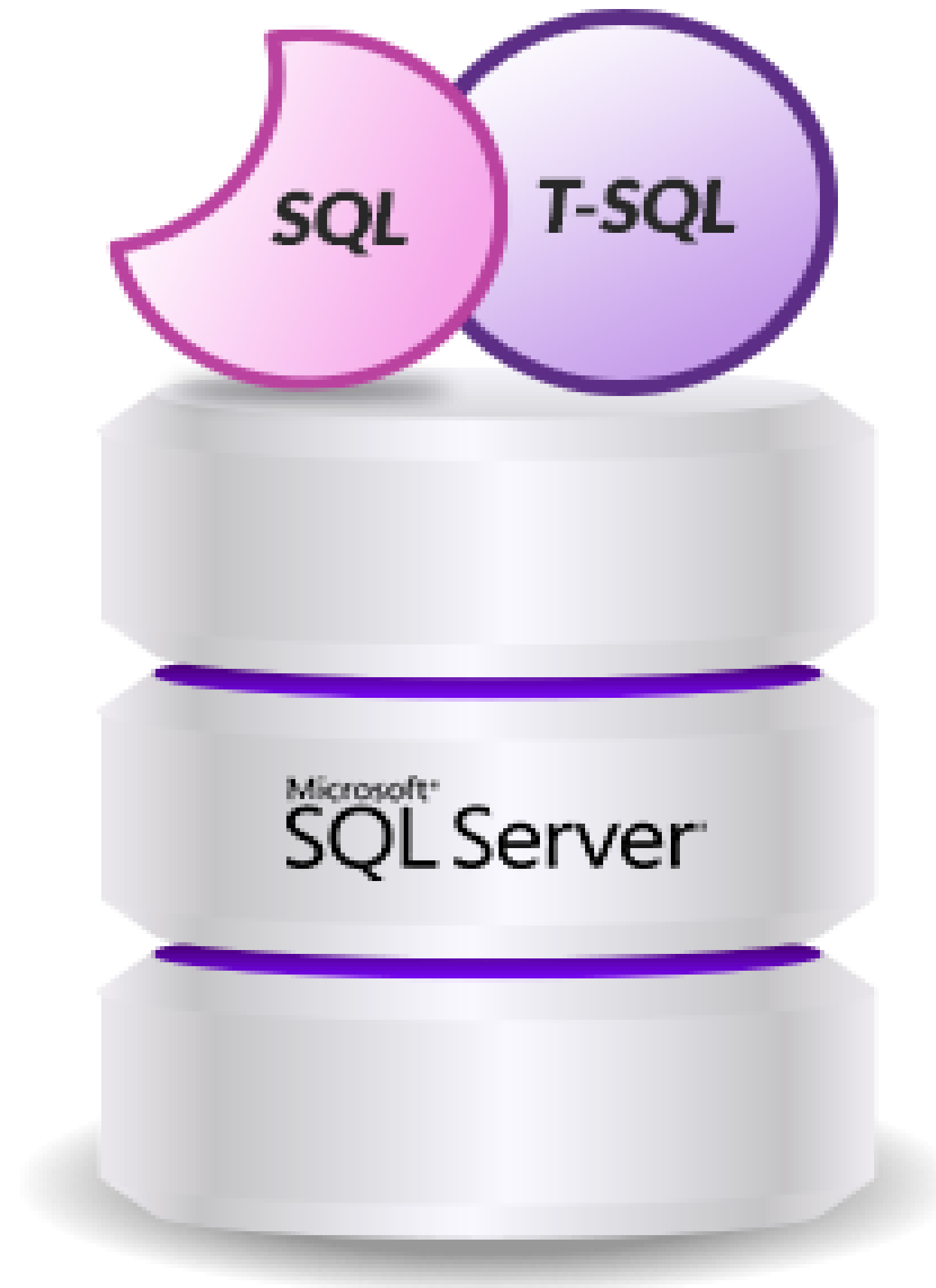
Giới thiệu về T-SQL (Transact-SQL)



- T-SQL là ngôn ngữ SQL mở rộng dựa trên SQL chuẩn được sử dụng trong SQL Server

- ✓ **Ngôn ngữ định nghĩa dữ liệu (DDL):** tạo, thay đổi và xóa các cấu trúc dữ liệu như bảng, chỉ mục, view.
- ✓ **Ngôn ngữ thao tác dữ liệu (DML):** truy vấn và cập nhật dữ liệu trong bảng.
- ✓ **Ngôn ngữ điều khiển dữ liệu (DCL):** phân quyền truy cập dữ liệu.
- ✓ **Ngôn ngữ kiểm soát giao dịch (TCL):** quản lý các giao dịch trong CSDL.

KHÁC NHAU GIỮA SQL VÀ T-SQL



CÂU TRÚC KHỞI LỆNH T-SQL

Begin

-- Khai báo biến;

-- Các câu lệnh T-SQL

End

Quy ước đặt tên:

- ✓ **Tên bảng:** Sử dụng danh từ số ít, viết hoa từng chữ cái đầu.
- ✓ **Tên cột:** Dùng danh từ mô tả, có thể viết hoa từng chữ cái đầu hoặc dùng dấu gạch dưới
- ✓ **Tên khóa, chỉ mục:** Bắt đầu bằng tiền tố PK_, FK_, IX_

Quy ước cú pháp:

- ✓ **Viết hoa từ khóa SQL:** Phân biệt với tên bảng, cột
- ✓ **Canh lề và xuống dòng:** Để dễ đọc và bảo trì
- ✓ **Comment:** Sử dụng -- cho dòng đơn, /* ... */ cho nhiều dòng để giải thích

Stored Procedures:

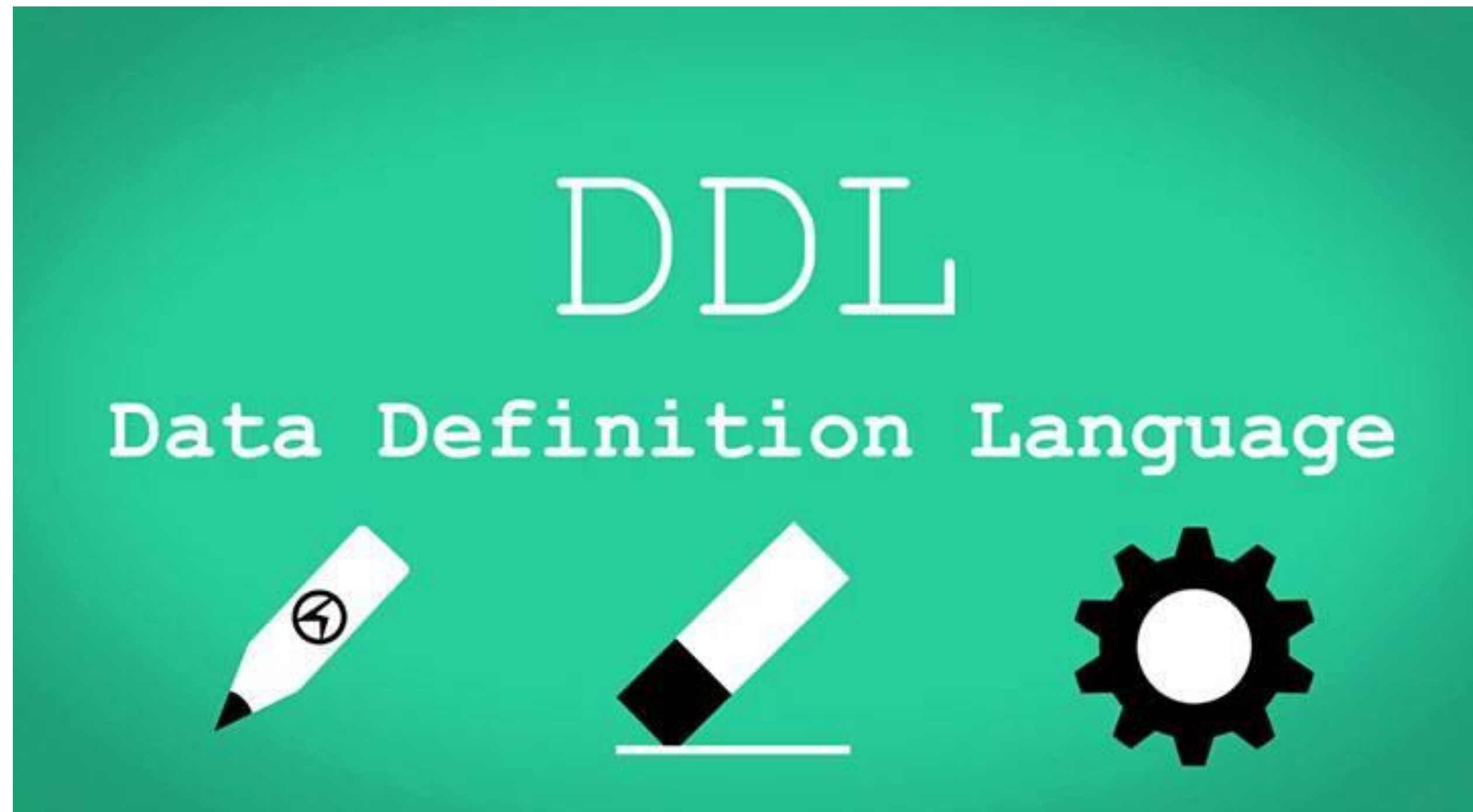
- **Tên Stored Procedure:** Bắt đầu với sp_ hoặc usp_
 - Ví dụ: usp_GetCustomerOrders
- **Biến đầu vào:** Sử dụng ký tự @ với tên rõ ràng (VD: @CustomerID)

Functions:

- **Tên Function:** Bắt đầu bằng tiền tố fn_
 - Ví dụ: fn_CalculateDiscount
- **Comment:** Đặt ngay trên function để giải thích chức năng

Xử lý lỗi: Sử dụng TRY...CATCH để tránh dừng đột ngột

Ngôn ngữ định nghĩa dữ liệu (DDL)



- DDL cho phép người dùng **tạo (CREATE)**, **chỉnh sửa (ALTER)** và **xóa (DROP)** các đối tượng như bảng, chỉ mục, các ràng buộc dữ liệu.

Ngôn ngữ định nghĩa dữ liệu (DDL)

□ Các lệnh chính trong DDL

- **CREATE**: lệnh được sử dụng để tạo CSDL hay tạo bảng
- **ALTER**: thay đổi nội dung trong bảng
- **DROP**: xóa 1 số nội dung trong CSDL hoặc bảng
- **TRUNCATE**: xóa tất cả nội dung trong bảng

Ngôn ngữ định nghĩa dữ liệu (DDL)

1. Cú pháp tạo bảng

```
CREATE      TABLE      <table-name>
(
    column1      data-type [RBTV],
    [column2      data-type [RBTV],
    ...
    [columnnn      data-type [RBTV]]
    [RBTV]
)
```

| TT | Thuộc Tính | Kiểu Dữ Liệu | Ràng Buộc | Ghi Chú |
|----|------------|--------------|-------------|-------------|
| 1 | sMaBP | varchar(10) | Primary Key | Mã bộ phận |
| 2 | sTenBoPhan | nvarchar(25) | | Tên bộ phận |

```
CREATE      TABLE      tblBoPhan
(
    sMaBP      varchar(10) PRIMARY KEY,
    sTenBP      nvarchar(25)
)
```

Các ràng buộc khi tạo bảng

- Là thực hiện cài đặt các yêu cầu về việc nhập giá trị cho các thuộc tính và tính đúng đắn cho thuộc tính
- Theo 2 cách:
 - Đặt ràng buộc ngay tại vị trí khai báo của cột \Leftrightarrow ***ràng buộc trực tiếp cho thuộc tính***
 - Đặt ràng buộc xuống phần cuối cùng của câu lệnh sau khi đã liệt kê hết các cột của bảng \Leftrightarrow ***thường dùng với ràng buộc gộp cho nhiều thuộc tính***

Primary
Key

Check

Default

Foreign
Key

Unique

Not Null

Create Database <Tên CSDL>;

USE <tên CSDL>;

CREATE TABLE <*table-name*>

(

column1 data-type [**RBTV**],

[*column2* data-type [**RBTV**],

...

[*columnn* data-type [**RBTV**]

[**RBTV**]

)

PRIMARY KEY

Hoặc

PRIMARY KEY ([d/s cột])

CHECK (*điều kiện*)

UNIQUE

hoặc

UNIQUE (DS cột)

NOT NULL

DEFAULT (giá-trị)

Hoặc

DEFAULT (giá-trị) **FOR** tên-cột

IDENTITY [(*start, step*)]

REFERENCES *bảng-tham-chiếu(cột tham chiếu)*

hoặc

FOREIGN KEY ([cột])

REFERENCES *bảng-tham-chiếu(cột tham chiếu)*

Primary Key

Check

Unique

Not Null

Default

Identity

Foreign Key

2. Sửa cấu trúc bảng

- Sử dụng câu lệnh **ALTER TABLE**.
- Thực hiện được các thao tác sau:
 - *Bổ sung một cột vào bảng.*
 - *Xoá một cột khỏi bảng.*
 - *Thay đổi định nghĩa kiểu của một cột trong bảng.*
 - *Xoá bỏ hoặc bổ sung các ràng buộc cho bảng*

3. Thêm cột mới vào trong bảng

- Cú pháp:

ALTER TABLE Tên_bảng

ADD Tên_cột Kiểu_dữ_liệu [RBTv] [...]

- *Luôn thêm cột mới vào cuối bảng*

- Ví dụ:

ALTER TABLE VATTU

ADD Soluong int

4. Hủy bỏ một cột trong bảng

- ❑ Cú pháp

ALTER TABLE Tên_bảng

DROP COLUMN Tên_cột [...]

ROW Tên_hàng

- ❑ Ví dụ:

ALTER TABLE VATTU

DROP COLUMN TenVtu

5. Sửa đổi kiểu dữ liệu của cột

□ Cú pháp:

ALTER TABLE Tên_bảng

ALTER COLUMN Tên_cột Kiểu_dữ_liệu_mới

□ Ví dụ:

ALTER TABLE VATTU

ALTER COLUMN TenVtu **Nvarchar(30)**

➤ Lưu ý: *Kiểu dữ liệu mới phải lớn hơn kiểu dữ liệu cũ đã có*

6. Thêm ràng buộc cho cột

□ Cú pháp:

ALTER TABLE Tên_bảng

ADD CONSTRAINT Tên_ràng_buộc Loại_ràng_buộc

□ Ví dụ:

ALTER TABLE VATTU

ADD CONSTRAINT CK_NgayNhap

CHECK (Ngaynhap <= GetDate())

7. Hủy ràng buộc đã đặt

- ❑ Cú pháp

ALTER TABLE Tên_bảng

DROP CONSTRAINT Tên_ràng_buộc

- ❑ Ví dụ:

ALTER TABLE VATTU

DROP CONSTRAINT CK_NgayNhap

8. Bật tắt các ràng buộc

❑ Cú pháp:

ALTER TABLE Tên_bảng

NOCHECK CONSTRAINT ALL | Tên_constraint [...]

ALTER TABLE Tên_bảng

CHECK CONSTRAINT ALL | Tên_constraint [...]

9. Đổi tên cột

❑ Cú pháp:

EXEC SP_Rename '*Tên_bảng.Tên_cột*', '*Tên_mới*', '**COLUMN**'

❑ Ví dụ:

EXEC SP_RENAME 'VATTU.MAVTU', 'MAVATTU', '**COLUMN**'

10. Đổi tên bảng

- ❑ Cú pháp

EXEC sp_rename 'Tên_bảng', 'Tên_mới'

- ❑ Ví dụ:

EXEC SP_RENAME 'VATTU', 'VT'

11. Xóa bảng

- Để xóa một bảng khỏi CSDL
- Cú pháp:

DROP TABLE Danh_sách_tên_các_bảng

- Lưu ý:
 - Câu lệnh Drop Table không thể thực hiện nếu bảng cần xóa được tham chiếu bởi một **Foreign Key**
 - Các ràng buộc, chỉ mục, trigger,.. đều bị xóa, nếu tạo lại bảng thì cũng phải tạo lại các đối tượng này
 - Sau khi xóa không thể khôi phục lại bảng và dữ liệu bảng

Ngôn ngữ thao tác dữ liệu (DML)

- Là một phần của SQL dùng để thao tác và quản lý dữ liệu trong CSDL.
- DML cho phép người dùng thực hiện các thao tác **INSERT, UPDATE, DELETE** và **SELECT** dữ liệu từ các bảng.



1. Thêm dữ liệu vào bảng

- Cho phép bổ sung thêm dữ liệu vào bảng
- Thêm từng dòng

```
INSERT INTO table-name [([col1, col2, col3, ...])]  
VALUES(value1, value2, value3,...);
```

Ngôn ngữ thao tác dữ liệu (DML)

tblNhanVien(iMaNV, sTenNV, dNgaysinh, sGioitinh, fLuong, sMaP)

***iMaNV có kiểu Autonumber**

INSERT INTO tblNhanVien(sTenNV, dNgaysinh, sGioitinh, fLuong, sMaP)

VALUES (N 'Hoàng Hải' , '13/04/1975' , N'Nam' , 1000000, CNTT),
(N 'Nguyễn Hà' , '13/04/1979' , N 'Nữ' , 1000000, Luat),
(N 'Phạm Hải' , '08/04/1976' , N 'Nữ' , 2000000, Luat);

| sMaNV | sTenNV | dNgaySinh | sGioiTinh | fLuong | sMaP |
|-------|-----------|------------|-----------|---------|------|
| 1 | Hoàng Hải | 13/04/1975 | Nam | 1000000 | CNTT |
| 2 | Nguyễn Hà | 13/04/1979 | Nữ | 1000000 | Luat |
| 3 | Phạm Hải | 08/04/1976 | Nữ | 2000000 | Luat |

2. Cập nhật dữ liệu trong bảng

- Được sử dụng để cập nhật giá trị dữ liệu đã có trong các bảng
- Cú pháp:

UPDATE *tên_bảng*

SET *tên_cột = biểu_thức*

[, ..., *tên_cột_k = biểu_thức_k*]

[**FROM** *danh_sách_bảng*]

[**WHERE** *điều_kiện*]

Ví dụ

UPDATE tblNhanVien

SET dNgaySinh = '15/08/1980'

WHERE MaNv = 2;

| sMaNV | sTenNV | dNgaySinh | sGioiTinh | fLuong | sMaP |
|-------|-----------|------------|-----------|---------|------|
| 1 | Hoàng Hải | 13/04/1975 | Nam | 1000000 | CNTT |
| 2 | Nguyễn Hà | 13/04/1979 | Nữ | 1000000 | Luat |
| 3 | Phạm Hải | 08/04/1976 | Nữ | 2000000 | Luat |

| sMaNV | sTenNV | dNgaySinh | sGioiTinh | fLuong | sMaP |
|-------|-----------|------------|-----------|---------|------|
| 1 | Hoàng Hải | 13/04/1975 | Nam | 1000000 | CNTT |
| 2 | Nguyễn Hà | 15/08/1980 | Nữ | 1000000 | Luat |
| 3 | Phạm Hải | 08/04/1976 | Nữ | 2000000 | Luat |

Ngôn ngữ thao tác dữ liệu (DML)

tblNhanVien(sMaNV, sTenNV, dNgaysinh, sGioitinh, fLuong, sMaP)

***Cập nhật tăng thêm lương 200.000 cho các Nhân viên nữ**

UPDATE tblNhanVien

SET fLuong = fLuong + 200.000

WHERE sGioiTinh = N'Nữ'

| sMaNV | sTenNV | dNgaySinh | sGioiTinh | fLuong | sMaP |
|-------|-----------|------------|-----------|-----------|------|
| 1 | Hoàng Hải | 13/04/1975 | Nam | 1.000.000 | CNTT |
| 2 | Nguyễn Hà | 15/08/1980 | Nữ | 1.000.000 | Luat |
| 3 | Phạm Hải | 08/04/1976 | Nữ | 2.000.000 | Luat |

| sMaNV | sTenNV | dNgaySinh | sGioiTinh | fLuong | sMaP |
|-------|-----------|------------|-----------|-----------|------|
| 1 | Hoàng Hải | 13/04/1975 | Nam | 1.000.000 | CNTT |
| 2 | Nguyễn Hà | 15/08/1980 | Nữ | 1.200.000 | Luat |
| 3 | Phạm Hải | 08/04/1976 | Nữ | 2.200.000 | Luat |

3. Xóa bỏ dữ liệu

- Cho phép xóa các dòng trong một bảng thỏa mãn điều kiện nào đó
- Cú pháp:

Delete from <Tên-bảng>

[From <ds-bảng>]

[Where <điều kiện>]

tblNhanVien(sMaNV, sTenNV, dNgaysinh, sGioitinh, fLuong, sMaP)

***Xóa nhân viên Nữ**

DELETE FROM tblNhanVien

WHERE sGioiTinh = N'Nữ'

| sMaNV | sTenNV | dNgaySinh | sGioiTinh | fLuong | sMaP |
|-------|-----------|------------|-----------|---------|------|
| 1 | Hoàng Hải | 13/04/1975 | Nam | 1000000 | CNTT |
| 2 | Nguyễn Hà | 15/08/1980 | Nữ | 1000000 | Luat |
| 3 | Phạm Hải | 08/04/1976 | Nữ | 2000000 | Luat |

| sMaNV | sTenNV | dNgaySinh | sGioiTinh | fLuong | sMaP |
|-------|-----------|------------|-----------|---------|------|
| 1 | Hoàng Hải | 13/04/1975 | Nam | 1000000 | CNTT |

tblPhongBan(sMaP, sTenPhong, sDiaChi, sDienThoai) tblNhanVien(sMaNV,
sTenNV, dNgaysinh, sGioitinh, fHSL, fPC, sMaP)

- **Xóa một các Nhân viên có mã phòng là ‘PTC’**

DELETE FROM tblNhanVien

WHERE sMaP = ‘PTC’;

- **Xóa tất cả bảng Nhân Viên**

DELETE FROM tblNhanVien;

- **Xóa các nhân viên của phòng ‘Hành Chính’**

DELETE FROM tblNhanVien

FROM tblPhongBan

WHERE sTenPhong = N’Hành Chính’ **and**

tblNhanVien.sMaP = tblPhongBan.sMaP

- **Hàm chuỗi:** LEN(), UPPER(), LOWER(), SUBSTRING().

```
SELECT UPPER(Ten) FROM KhachHang;
```

- **Hàm số học:** ABS(), ROUND(), CEILING(), FLOOR().

```
SELECT ROUND(Diem, 1) FROM SinhVien;
```

- **Hàm ngày và giờ:** GETDATE(), DATEADD(), DATEDIFF().

```
SELECT DATEADD(YEAR, 1, NgaySinh) FROM KhachHang;
```

- **Hàm tổng hợp:** SUM(), AVG(), MIN(), MAX(), COUNT().

```
SELECT AVG(Diem) FROM SinhVien;
```


❑ **SELECT:** Lệnh truy vấn để lấy dữ liệu từ các bảng.

SELECT FirstName, LastName FROM Employees;

❑ **WHERE:** để lọc dữ liệu dựa trên điều kiện nhất định.

SELECT * FROM Employees WHERE Salary > 5000;

❑ **ORDER BY:** để sắp xếp kết quả theo thứ tự tăng hoặc giảm.

SELECT * FROM Employees ORDER BY LastName ASC;

❑ **Kết hợp bảng (JOIN):** Dùng để kết hợp dữ liệu từ hai hoặc nhiều bảng (**INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN**)

GROUP BY: Dùng để nhóm dữ liệu theo một hoặc nhiều cột.

HAVING: Dùng để lọc các nhóm sau khi đã áp dụng Group by.

```
SELECT DepartmentID, COUNT(*) AS EmployeeCount  
FROM Employees GROUP BY DepartmentID  
HAVING COUNT(*) > 5;
```

- ✓ **PIVOT:** Xoay dữ liệu từ dạng hàng thành dạng cột để thuận tiện cho việc phân tích.

```
SELECT *  
FROM (SELECT Year, Region, Sales FROM SalesData) AS  
SourceTable  
PIVOT (  
    SUM(Sales)  
    FOR Region IN ([North], [South], [East], [West])  
) AS PivotTable;
```

- ✓ **Lợi ích:** Tạo báo cáo tóm tắt, dễ dàng so sánh dữ liệu theo các chiều khác nhau.

MỘT SỐ VẤN ĐỀ KHÁC TRONG SQL

- Biến toàn cục, biến cục bộ
- Bảng tạm
- Giá trị Null

- **Biến:** là một đối tượng lưu trữ một giá trị dữ liệu. Dữ liệu có thể được truyền đến câu lệnh SQL bằng cách sử dụng tên biến.
- Biến có thể phân thành 2 loại:
 - Biến cục bộ
 - Biến toàn cục

- Trong Transact-SQL, biến cục bộ được khai báo và sử dụng tạm thời khi thực thi câu lệnh SQL
- Cú pháp:

Declare *@Tên_Biến Kiểu_Dữ_Liệu*

- Trong đó:
 - @tên_biến: đặt theo quy tắc đặt tên, phải bắt đầu bằng @
 - Kiểu_dữ_liệu: kiểu dữ liệu hệ thống hoặc kiểu dữ liệu người dùng
- Ví dụ: **declare** @manv int

Ví dụ

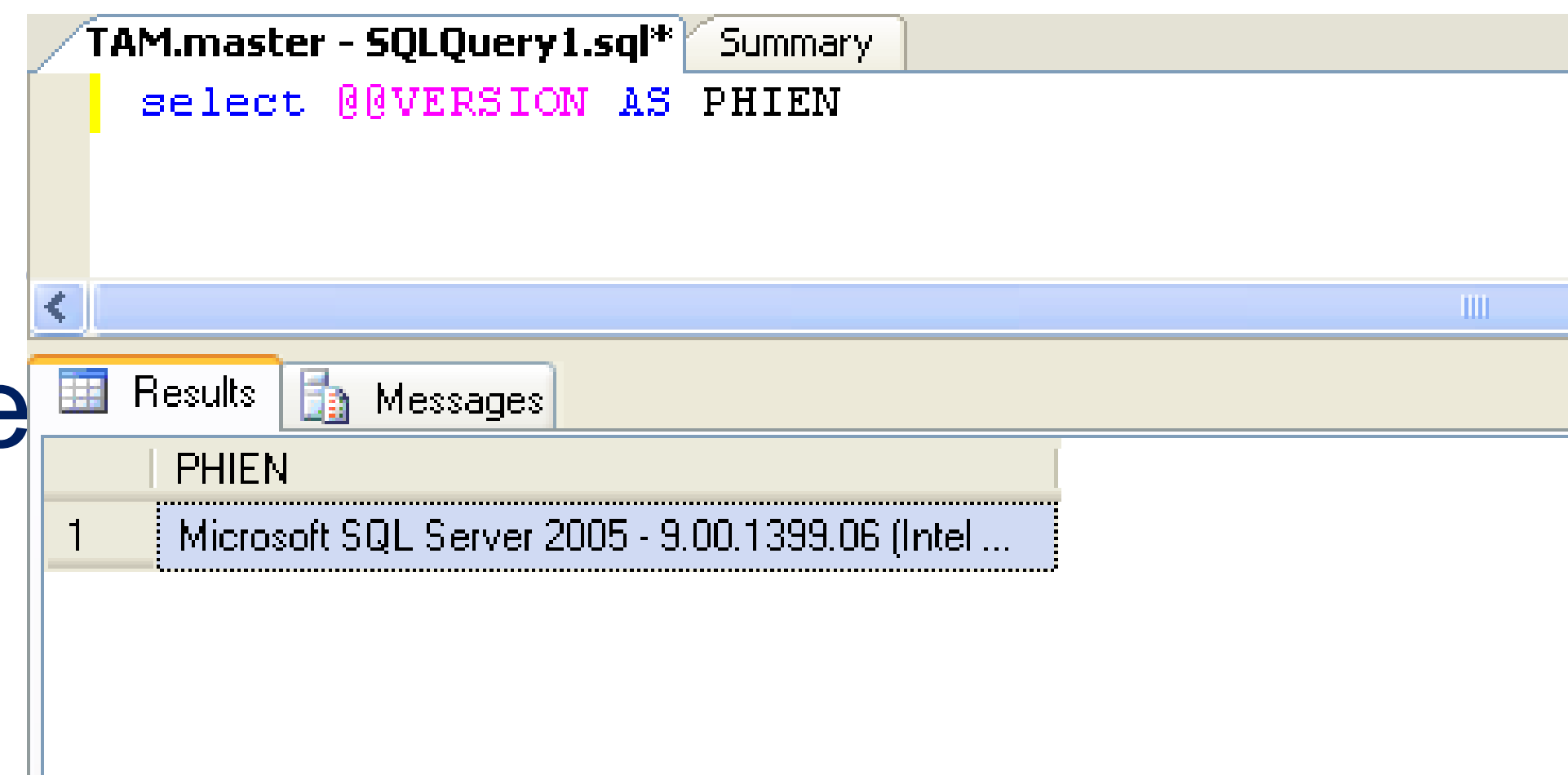
- Gán giá trị cho biến
 - Cú pháp:
Set *@tên_biến = giá trị/hàm*
hoặc
Select *@tên_biến = Giá trị/hàm*
 - Ví dụ: sử dụng biến cục bộ để truy vấn dữ liệu
Declare @manv int
Set @manv = 1
Select hoten, diachi
From Nhanvien
Where manv=@manv

- Biến toàn cục là biến có sẵn và hệ thống quản lý
- Biến toàn cục trong SQL Server được đặt tên bắt đầu bởi 2 ký hiệu @
- Cú pháp:

`SELECT @@LANGUAGE.kiểu dữ liệu`

Ví dụ: biến toàn cục được sử dụng xem thông tin phiên bản SQL Server

`SELECT @@VERSION AS PHIEN`



Bảng Tạm (Temporary Table): Là bảng được lưu tạm thời trong phiên làm việc và sẽ tự động xóa khi phiên kết thúc.

Ứng dụng: Hữu ích cho việc xử lý dữ liệu tạm thời hoặc phân tích dữ liệu trung gian mà không cần lưu trữ lâu dài.

MỘT SỐ VẤN ĐỀ KHÁC - GIÁ TRỊ NULL

Null:

- ✓ Biểu thị giá trị không xác định hoặc không có dữ liệu.
- ✓ **IS NULL/IS NOT NULL**: Kiểm tra giá trị NULL trong câu lệnh WHERE

```
SELECT * FROM Employees WHERE ManagerID IS NULL;
```

Hàm Xử Lý Null:

- ✓ **ISNULL()**: Thay thế giá trị NULL bằng một giá trị khác

Ứng dụng: để xử lý trường hợp dữ liệu thiếu hoặc không có giá trị, giúp đảm bảo tính toàn vẹn của kết quả truy vấn.

THỰC HÀNH



- ✓ Câu hỏi trắc nghiệm
- ✓ Bài tập

