

Bài 1

Tổng quan hệ quản trị SQL Server

1. Các thành phần của SQL Server

- **SQL(*Structure Query Language*) Server** là *hệ quản trị CSDL Client/Server* gồm các tính năng:

- + Định nghĩa CSDL, chỉnh sửa CSDL, lược đồ CSDL, quan hệ giữa các lược đồ.
- + Thêm, sửa, xóa thông tin.
- + Hỗ trợ các tính năng bảo mật, sao lưu phục hồi, cấp quyền truy nhập.

- **Các phiên bản của SQL Server:**

- + Enterprise: Full;
- + Standard: Limit Advanced feature;
- + Personal: PC;
- + Developer: Limit connect;
- + Express: Not User Interface;
- + Trial: Same Enterprise Editor, Limit time.

1. Các thành phần của SQL Server

1.1 Cơ sở dữ liệu - Database (Lưu CSDL)

1.2 Bảng - Tables (Lưu các bảng đã thiết kế): là đối tượng lưu trữ dữ liệu thực. Mỗi bảng được định nghĩa nhiều trường(cột), mỗi trường (*field còn gọi là column name*) ứng với một loại kiểu dữ liệu.

1.3 Chỉ mục – Indexs: là đối tượng chỉ tồn tại trong bảng hay khung nhìn (*views*), ảnh hưởng rất lớn đến tốc độ truy cập dữ liệu, tìm kiếm thông tin trên bảng.

1.4 Bẫy lỗi – Trigger: là đối tượng chỉ tồn tại trong bảng, cụ thể là một đoạn mã tự động thực thi khi có một hành động nào đó xảy ra đối với dữ liệu trong bảng như insert, update, delete.

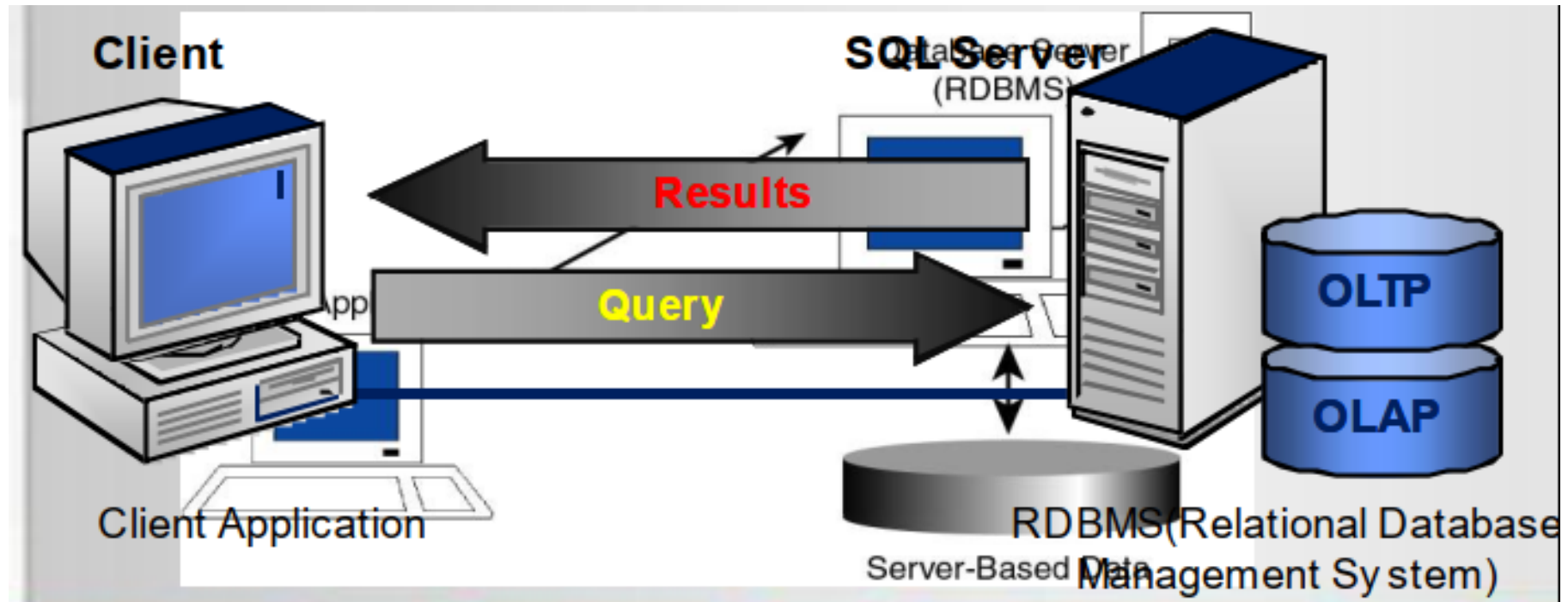
1.5 Lược đồ quan hệ - Database Diagrams: dùng để ràng buộc dữ liệu giữa các bảng.

1.6 Khung nhìn – Views: giống như bảng nhưng view không chứa dữ liệu, view có thể tạo thêm trường mới dựa vào câu lệnh SQL (view còn được hiểu là câu lệnh SQL được biên dịch sẵn).

1.7 Thủ tục nội – Stored Procedure: hay còn gọi là spocs. Cho phép khai báo biến, nhận tham số, thực thi các phát biểu có điều kiện trên CSDL, giúp tiết kiệm thời gian thực thi trên dữ liệu.

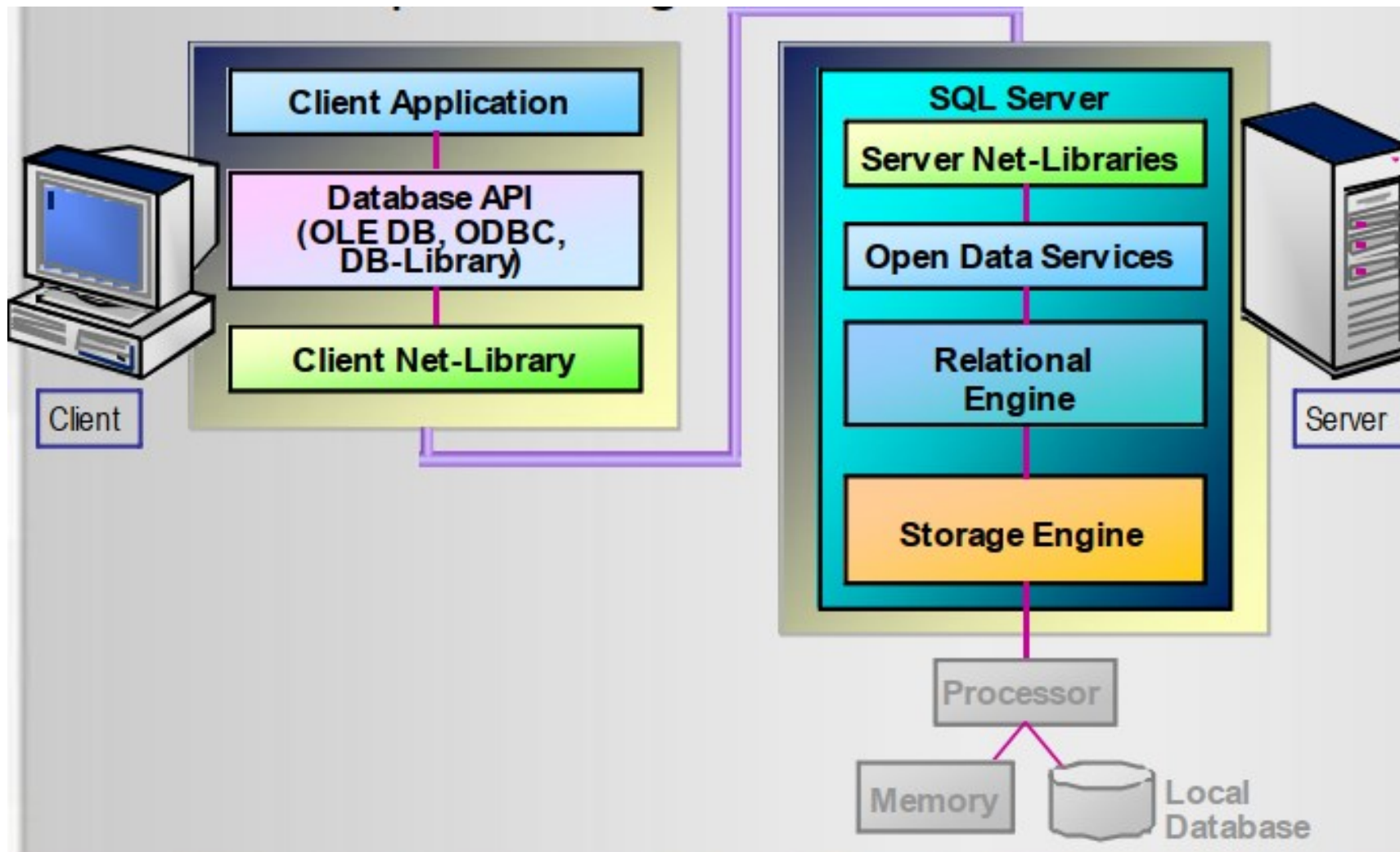
2. Kiến trúc cơ sở dữ liệu SQL Server

- + Hệ thống phía Server: xử lý yêu cầu và phục vụ
- + Hệ thống phía Client: nơi yêu cầu và nhận dữ liệu
- + Hệ thống giao tiếp giữa Client và Server (Network)



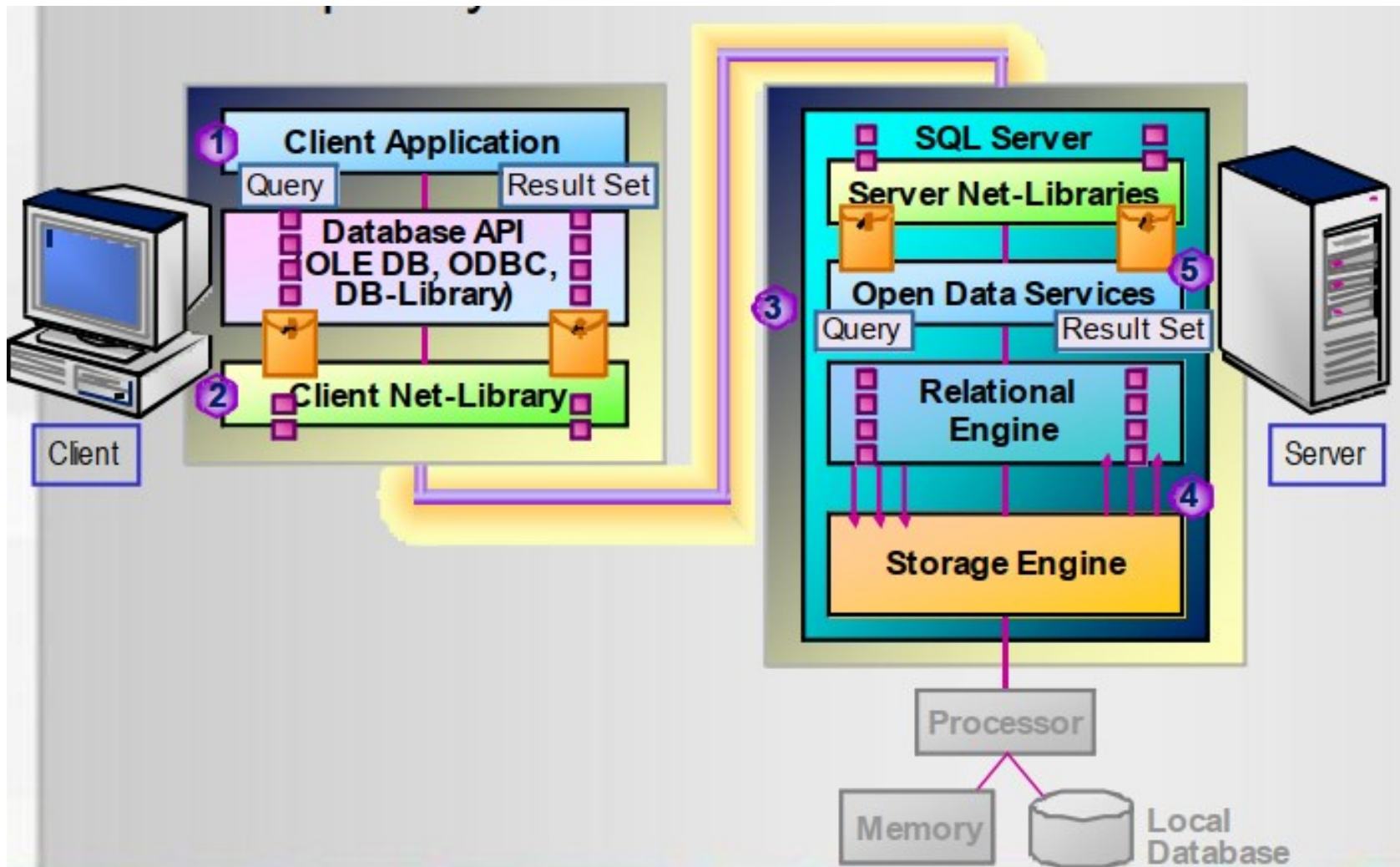
2. Kiến trúc cơ sở dữ liệu SQL Server

Các thành phần trong Client/Server:



2. Kiến trúc cơ sở dữ liệu SQL Server

Giao tiếp xử lý client/server:



3. Phân loại các loại CSDL của SQL Server

3.1 System Databases(cơ sở dữ liệu của hệ thống SQL server)

- + Master: lưu trữ dữ liệu hệ thống của SQL Sever
- + Model: lưu trữ tất cả template (mẫu) để tạo cơ sở dữ liệu mới
- + MSDB: lưu trữ tất cả tác vụ xảy ra trong SQL Server (Khi người dùng tạo một lịch trình cho backup dữ liệu hay lịch trình để thực thi stored procedure, tất cả tác vụ này được lưu trữ vào msdb).
- + Tempdb: lưu trữ những tác vụ tạm của SQL Server

Lưu ý: System Databases không được xóa.

3.2 Databases khác:

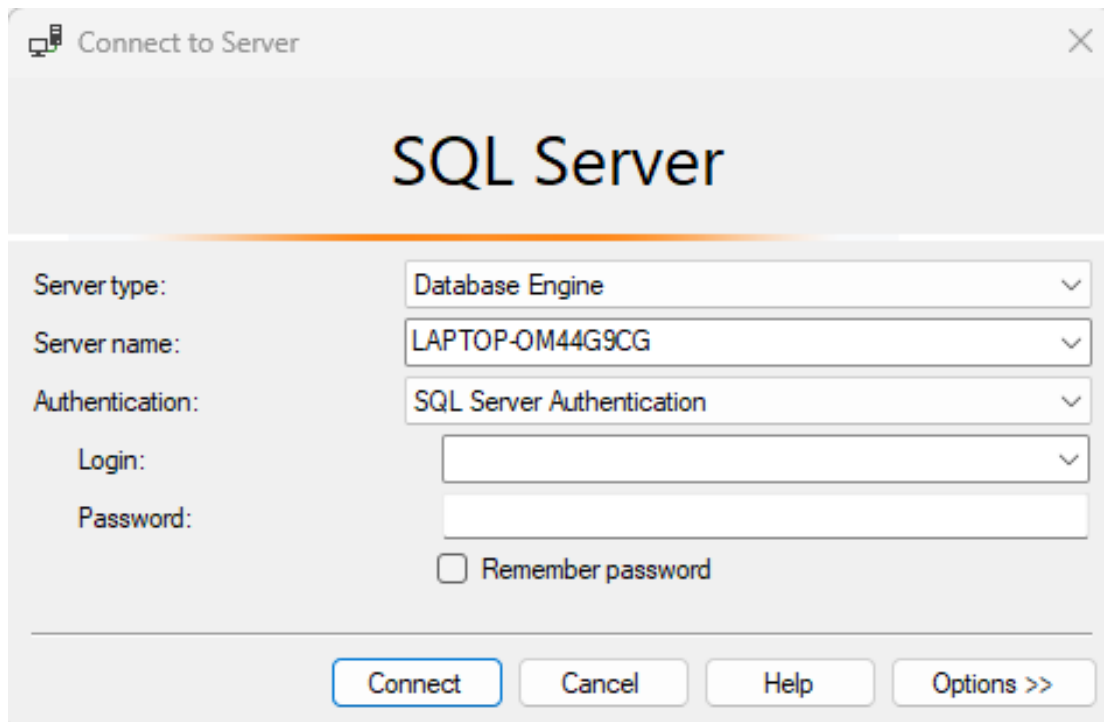
Do người dùng tự thiết kế

4. Cơ chế đăng nhập SQL Server

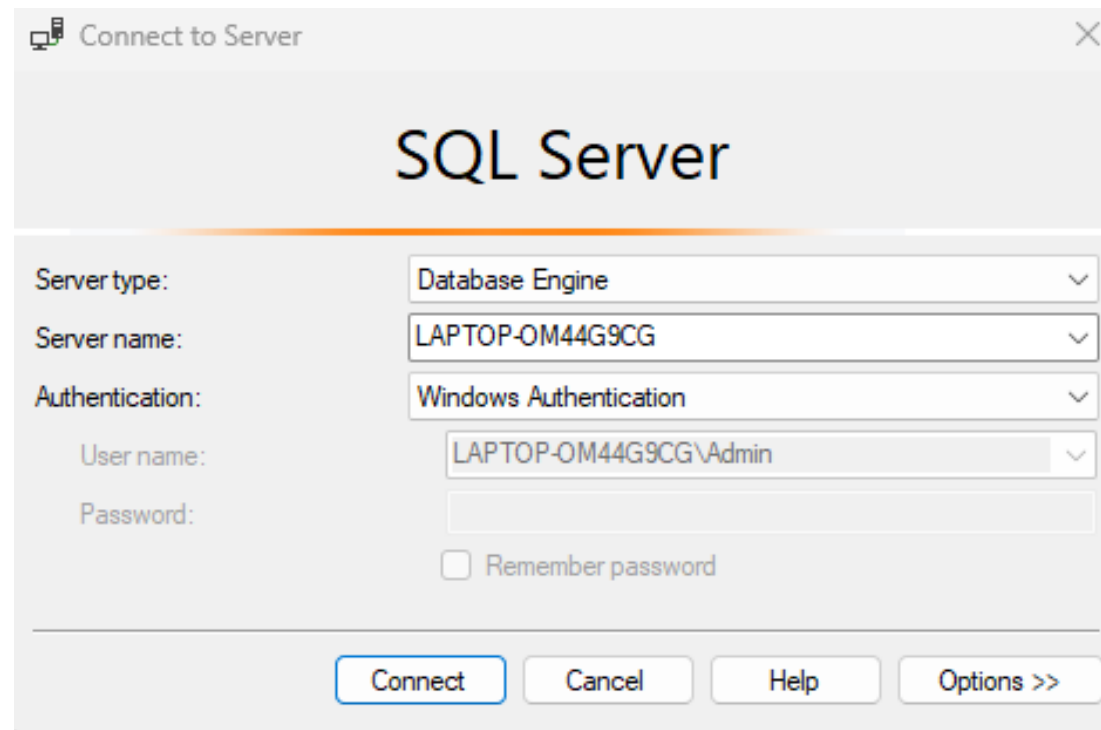
SQL Server xác thực những đăng nhập với 2 cơ chế:

- + SQL Server authentication
- + Windows authentication

Lưu ý: Khi đăng nhập hệ thống SQL server, SQL Server sẽ đưa ra yêu cầu bạn phải chọn 1 trong 2 cơ chế xác thực ở trên.



The screenshot shows the 'Connect to Server' dialog box. The title bar says 'Connect to Server'. The main heading is 'SQL Server'. Below this, there are several fields: 'Server type:' with a dropdown menu showing 'Database Engine'; 'Server name:' with a dropdown menu showing 'LAPTOP-OM44G9CG'; 'Authentication:' with a dropdown menu showing 'SQL Server Authentication'; 'Login:' with a dropdown menu; and 'Password:' with a text box. There is also a checkbox labeled 'Remember password' which is unchecked. At the bottom, there are four buttons: 'Connect', 'Cancel', 'Help', and 'Options >>'.



The screenshot shows the 'Connect to Server' dialog box. The title bar says 'Connect to Server'. The main heading is 'SQL Server'. Below this, there are several fields: 'Server type:' with a dropdown menu showing 'Database Engine'; 'Server name:' with a dropdown menu showing 'LAPTOP-OM44G9CG'; 'Authentication:' with a dropdown menu showing 'Windows Authentication'; 'User name:' with a dropdown menu showing 'LAPTOP-OM44G9CG\Admin'; and 'Password:' with a text box. There is also a checkbox labeled 'Remember password' which is unchecked. At the bottom, there are four buttons: 'Connect', 'Cancel', 'Help', and 'Options >>'.

5. Nhập/Xuất/Xóa CSDL trong SQL Server

5.1 Nhập(import database) CSDL trong SQL server

Chúng ta có một CSDL, muốn gửi CSDL này cho một người khác sử dụng hoặc muốn sao chép sang một máy tính khác sử dụng SQL Server. Chúng ta sẽ dùng cách import database vào sql server mà chúng ta muốn sử dụng.

5.2 Xuất(Export database) CSDL trong SQL server

Trong quá trình thao tác với SQL Server, chắc hẳn rằng sẽ có đôi lúc chúng ta sẽ cần xuất cơ sở dữ liệu ra tập tin .sql. Việc chuyển đổi này có nhiều mục đích khác nhau, có thể là sao lưu hoặc cũng có thể là dùng sql đó để cài trên một SQL Server khác. Chúng ta sẽ dùng cách **Export database** ra một file sql và lưu lại file đó với nhu cầu sử dụng.

5.3 Xóa>Delete database) CSDL trong SQL server

Nếu chúng ta xác định không còn sử dụng Datsbase đã tạo ra trước đó và muốn nó được xóa ra khỏi sql server. Chúng ta dùng cách Delete database, nếu chúng ta đã xóa CSDL ra khỏi sql server thì sẽ không phục hồi lại được.

Bài thực hành 1

1. Khởi động công cụ SQL Server Management Studio và đăng nhập vào SQL Server theo 2 cơ chế xác thực
2. Nạp(import database) CSDL từ file bên ngoài vào trong SQL server
3. *Xuất(Export database) CSDL trong SQL server ra file để lưu trữ*
4. *Xóa>Delete database) file CSDL đã có sẵn ra khỏi SQL server*
5. Thoát *khỏi SQL server*

Câu hỏi ôn tập

1. SQL Server gồm các tính năng nào?
2. SQL Server có bao nhiêu phiên bản?
3. Kể tên các thành phần của SQL Server? Trình bày chức năng của các thành phần trên?
4. Cơ sở dữ liệu SQL Server có kiến trúc như thế nào?
5. *System Databases dùng để lưu trữ gì?*
6. *Databases khác dùng để lưu trữ gì?*
7. SQL Server có mấy cơ chế đăng nhập? Kể tên các cơ chế đăng nhập?

Bài 2

ĐỊNH NGHĨA DỮ LIỆU VỚI T-SQL

* **Khái niệm T-SQL**

- Transact-SQL(T-SQL) là ngôn ngữ SQL mở rộng dựa trên SQL chuẩn của tổ chức ISO (International Organization for Standardization) và ANSI (American National Standards Institute) được sử dụng trong SQL Server.

1. Cơ sở dữ liệu (Database)

1.1 Tạo mới CSDL

- **Cú pháp:**

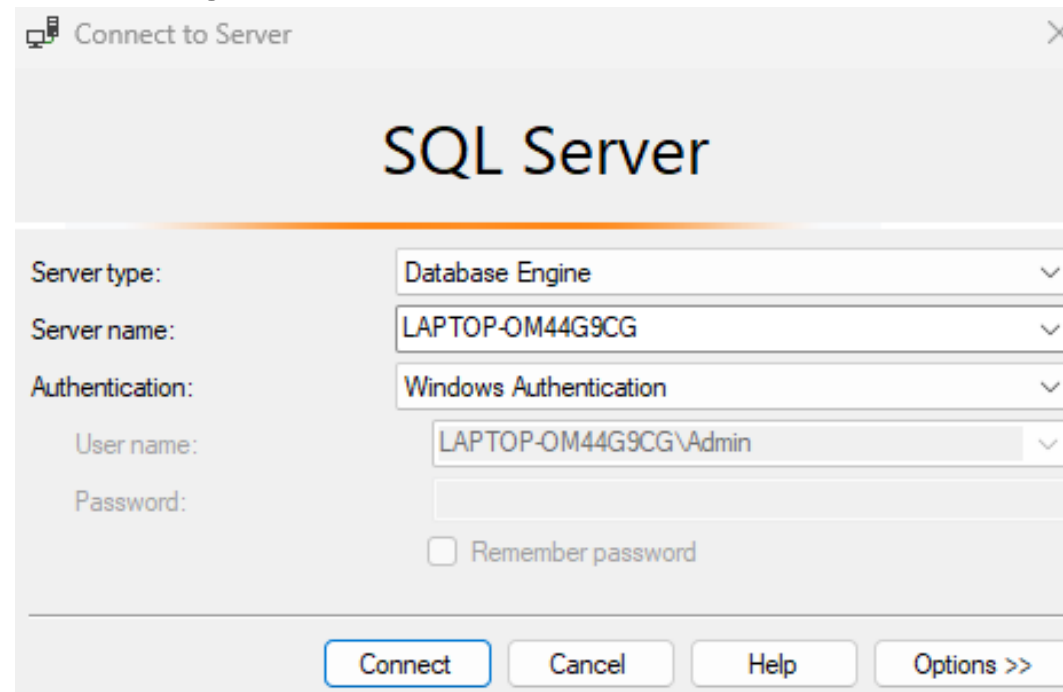
CREAT DATABASE <Ten_database>

- Ví dụ: tạo mới CSDL tên Quanlyhocsinh

Các bước thực hiện:

B1. Chọn công cụ **SQL server management studio** để mở SQL server.

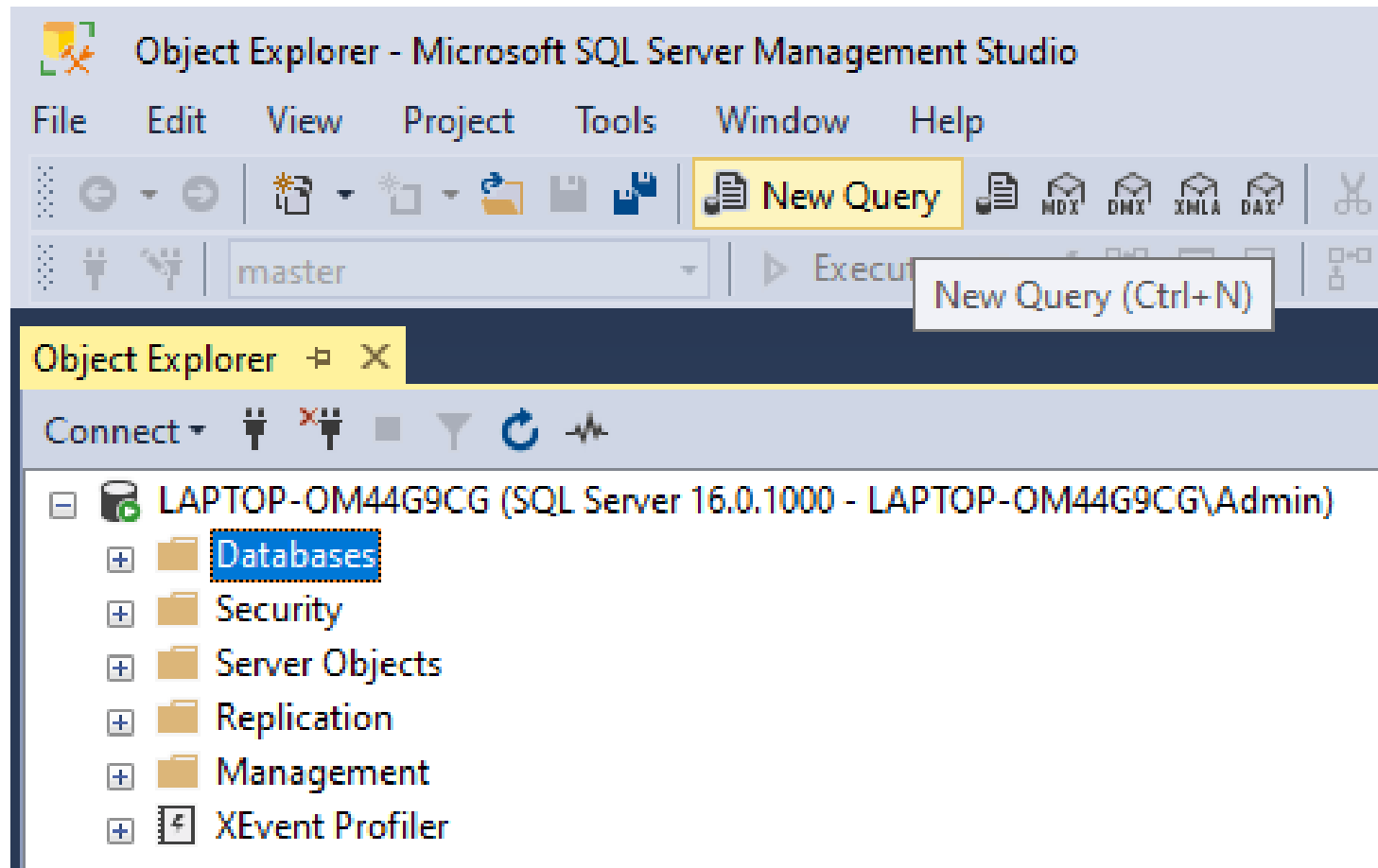
B2. Đăng nhập theo cơ chế chứng thực của **windows**, chọn **connect** để kết với SQL server.



1. Cơ sở dữ liệu (Database)

1.1 Tạo mới CSDL

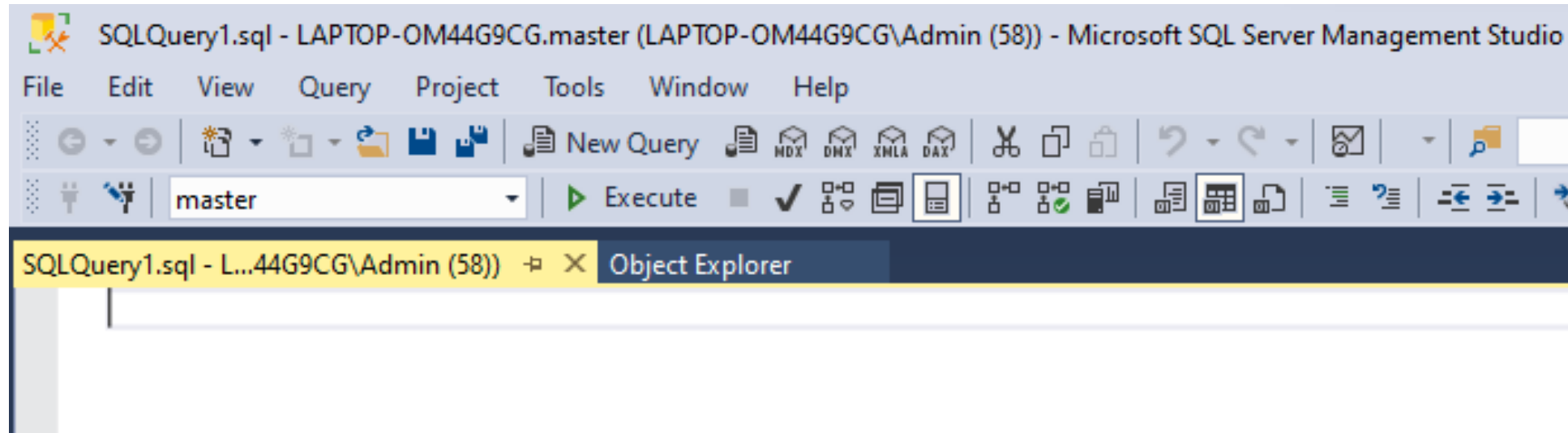
B3. Trong màn hình chính của SQL server, ta chọn mục New Query để vào khung viết lệnh SQL.



1. Cơ sở dữ liệu (Database)

1.1 Tạo mới CSDL

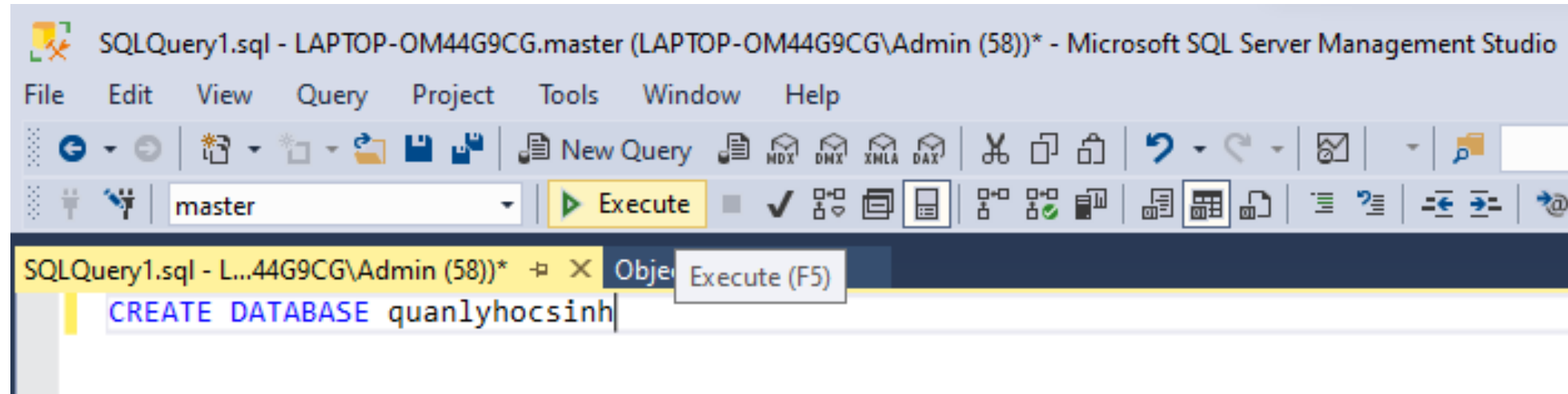
B4. Tại cửa sổ SQL Query ta viết lệnh để tạo mới CSDL.



1. Cơ sở dữ liệu (Database)

1.1 Tạo mới CSDL

B5. Viết dòng lệnh như sau: CREATE DATABASE quanlyhocsinh.

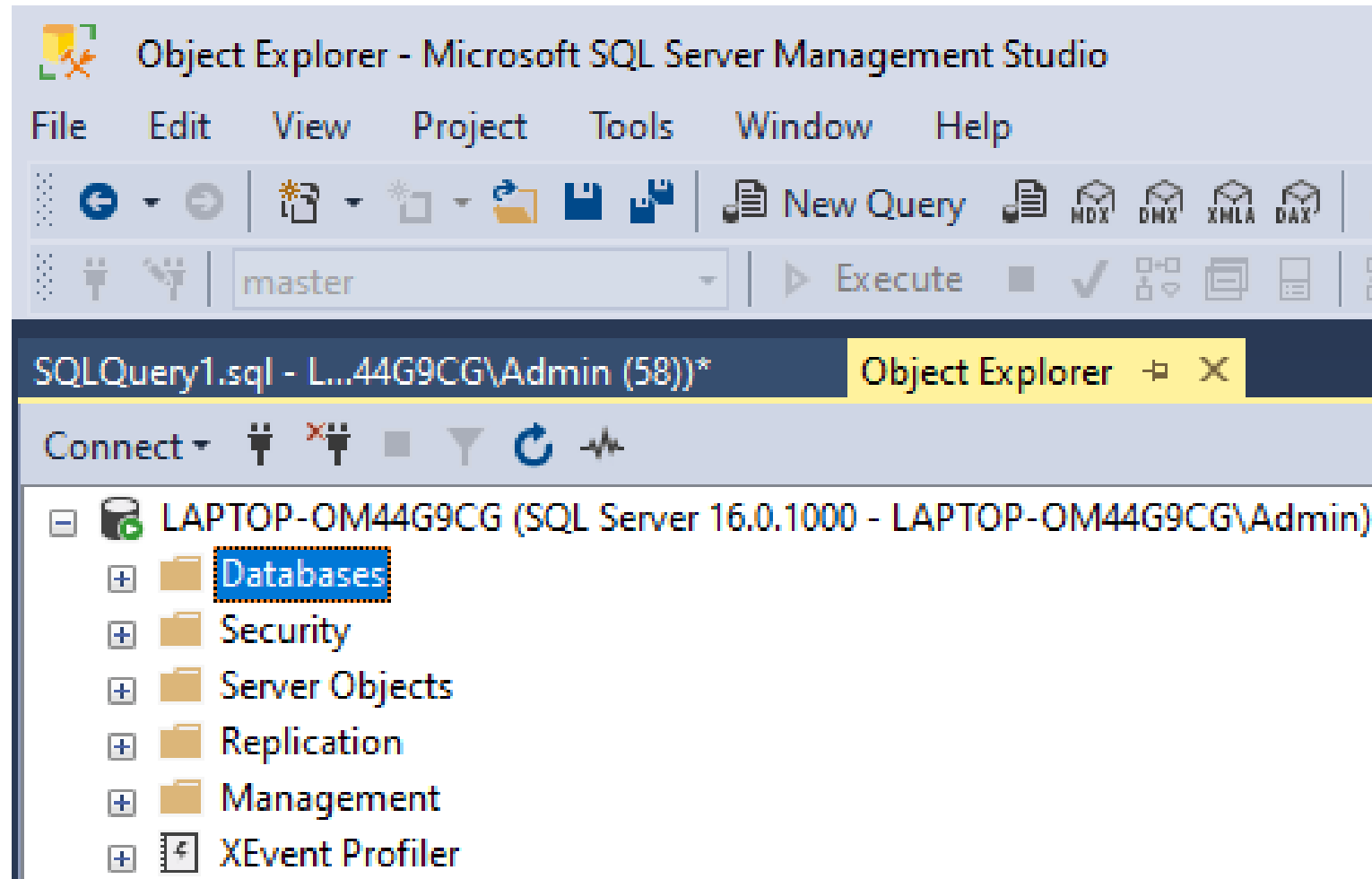


Gõ lệnh xong, bôi đen dòng lệnh, nhấn nút execute trên thanh công cụ hoặc dùng chuột bôi đen dòng lệnh và nhấn phím F5.

1. Cơ sở dữ liệu (Database)

1.1 Tạo mới CSDL

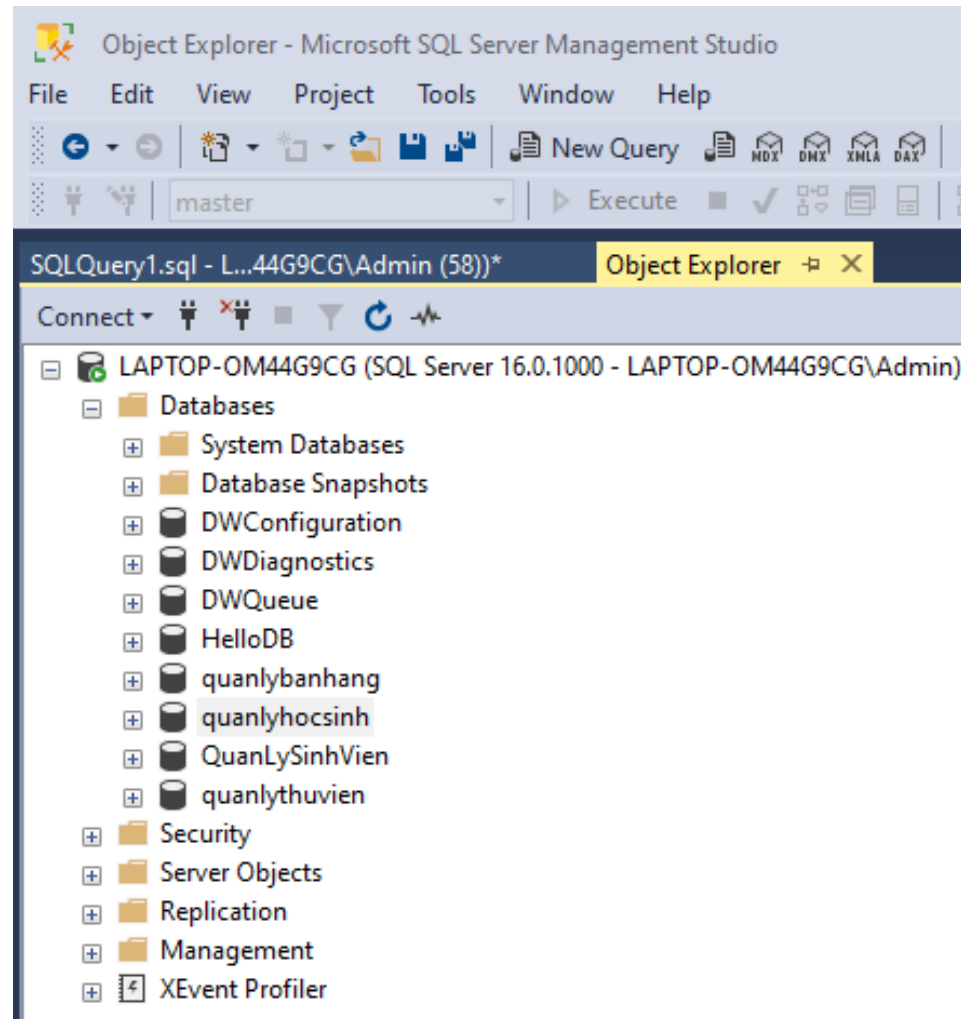
B6. Mở cửa sổ Object Explore để kiểm tra, ta chọn dấu “+” trước mục Database.



1. Cơ sở dữ liệu (Database)

1.1 Tạo mới CSDL

Quan sát ta thấy CSDL vừa tạo mới có tên quanlyhocsinh.



1. Cơ sở dữ liệu (Database)

1.2 Mở CSDL để thao tác

+ Cú pháp:

USE <TEN_DATABASE>

+ Ví dụ: Mở CSDL quanlyhocsinh

B1. Tại cửa sổ SQL Query ta gõ lệnh và thực thi như sau:

USE quanlyhocsinh

B2. Bôi đen tất cả dòng lệnh và nhấn F5/nhấn nút Execute trên thanh công cụ.

B3. Ta có thể thao tác trên CSDL được rồi.

1. Cơ sở dữ liệu (Database)

1.3 Sửa một CSDL

+ *Cú pháp:*

ALTER DATABASE <Tên_DB_cũ> MODIFYNAME = <Tên_DB_mới>

+ *Ví dụ:* Sửa tên database quanlyhocsinh thành quanlyhs.

B1. Tại cửa sổ SQL Query ta gõ lệnh và thực thi như sau:

USE quanlyhocsinh

ALTER DATABASE quanlyhocsinh MODIFY NAME = quanlyhs

B2. Bôi đen tất cả dòng lệnh và nhấn F5/nhấn nút Execute trên thanh công cụ.

B3. Mở cửa sổ Object Explore để vào kiểm tra ta thấy không còn CSDL quanlyhocsinh, mà có thêm mới CSDL vừa đổi tên là quanlyhs.

1. Cơ sở dữ liệu (Database)

1.4 Xóa một CSDL

+ **Cú pháp:**

DROP DATABASE <TênCSDL>

+ *Ví dụ:* Xóa CSDL quanlyhs ra khỏi SQL server

B1. Tại cửa sổ SQL Query ta gõ lệnh và thực thi như sau:

DROP DATABASE quanlyhs

B2. Bôi đen tất cả dòng lệnh và nhấn F5/nhấn nút Execute trên thanh công cụ.

B3. Mở cửa sổ Object Explore để vào kiểm tra ta thấy không còn CSDL quanlyhs.

2. Các kiểu dữ liệu của SQL server

STT	Tên kiểu	Ý nghĩa
1	char	Chiều dài tối đa 8.000 ký tự (không phải là ký tự Unicode)
2	varchar	Tối đa 8.000 ký tự. (không phải dạng Unicode).
3	varchar (tối đa)	Chiều dài tối đa 231 ký tự(chỉ có trong SQL Server 2005).
4	text	Chiều dài tối đa là 2,147,483,647 ký tự.(không phải dạng Unicode)
5	nchar	Chiều dài tối đa 4.000 ký tự. (Chiều dài cố định Unicode)
6	nvarchar	Chiều dài tối đa 4.000 ký tự. (Chiều dài biến Unicode)
7	nvarchar (tối đa)	Chiều dài tối đa 231 ký tự (chỉ SQL Server 2005).
8	image	Chiều dài tối đa là 2,147,483,647 byte.
9	date	Lưu trữ một ngày như ngày 30/6/1991
10	time	Lưu trữ thời gian như 12:30 PM
11	float	-1.79E + 308 - 1,79E + 308
12	int	-2.147.483.648 -> 2.147.483.647
13	numeric	-10 ^ 38 +1 , 10 ^ 38 -1

3. Các loại ràng buộc dữ liệu của SQL server

3.1 Ràng buộc CHECK:

+ Ràng buộc CHECK được sử dụng nhằm kiểm tra điều kiện hợp lệ đối với dữ liệu.

+ **Cú pháp:**

[CONSTRAINT tên_ràng_buộc] CHECK (điều_kiện)

+ *Ví dụ:* Tạo ràng buộc CHECK cho cột Diem của bảng tbl_KETQUA trong CSDL Quanlysinhvien với điều kiện (Diem \geq 0 And Diem \leq 10).

Ta viết đoạn lệnh sau:

CONSTRAINT kiểmtra_nhaphiem CHECK (Diem \geq 0 And Diem \leq 10)

3. Các loại ràng buộc dữ liệu của SQL server

3.2 Ràng buộc *PRIMARY*(KHÓA CHÍNH)

+ Ràng buộc PRIMARY KEY được sử dụng để định nghĩa CỘT khoá chính của bảng.

+ Cú pháp:

[CONSTRAINT tên_ràng_buộc] PRIMARY KEY [(danh_sách_cột)]

+ Ví dụ: Tạo ràng buộc *PRIMARY KEY* cho cột mã môn học của bảng tbl_MONHOC trong CSDL quanlysinhvien.

Ta viết đoạn lệnh sau:

CONSTRAINT khoa_chinh PRIMARY KEY (Ma_mon_hoc)

3. Các loại ràng buộc dữ liệu của SQL server

3.3 Ràng buộc FOREIGN KEY(KHÓA NGOẠI)

+ Khóa ngoại có thể tham chiếu vào khóa chính hay cột có ràng buộc duy nhất của một bảng khác. Khóa ngoại có thể chứa giá trị NULL.

+ Cú pháp:

```
[CONSTRAINT tên_ràng_buộc] FOREIGN KEY [(danh_sách_cột)]  
REFERENCES tên_bảng_tham_chiếu(danh_sách_cột_tham_chiếu)
```

+ Ví dụ: Tạo khóa ngoại cho cột nghề học của bảng tbl_SINHVIEN tham chiếu đến cột mã nghề của bảng tbl_NGHEHOC.

Ta viết đoạn lệnh sau:

```
CONSTRAINT khoa_ngoai FOREIGN KEY (nghe_hoc)  
REFERENCES tbl_NGHEHOC(Ma_nghe)
```

3. Các loại ràng buộc dữ liệu của SQL server

3.4 Ràng buộc *UNIQUE*

+ Ràng buộc UNIQUE được tạo cho cột để đảm bảo tất cả các giá trị trong một cột là duy nhất.

+ Cú pháp:

[CONSTRAINT tên_ràng_buộc UNIQUE (danh_sách_cột)]

3.5 Ràng buộc *DEFAULT*

+ Ràng buộc DEFAULT cung cấp giá trị mặc định cho cột.

+ Cú pháp: tên_cột_ràng_buộc DEFAULT giá_trị_mặc_định

+ Ví dụ: Tạo giá trị mặc định cho cột nơi sinh của bảng tbl_SINHVIEN là TP.HCM.

Ta viết đoạn lệnh sau:

Noi_Sinh DEFAULT 'TP.HCM'

4. Bảng (Table)

4.1 Lệnh Tạo cấu trúc một bảng (Table)

+ **Cú pháp:**

```
CREATE TABLE <tên_bảng> (  
    <tên_cột> <kiểu_dữ_liệu> [<ràng_buộc_dữ_liệu>],  
    <tên_cột> <kiểu_dữ_liệu> [<ràng_buộc_dữ_liệu>],  
    ....) ;
```

+ **Ví dụ:** Tạo bảng nghề học và đặt tên cho bảng là **tbl_NGHEHOC**, có các cột như sau:

Tên cột	Kiểu dữ liệu	Ràng buộc dữ liệu
MaNghe	nvarchar(50)	Primary key
TenNghe	nvarchar(150)	

Ta viết đoạn lệnh sau:

```
CREATE TABLE tbl_NGHEHOC (  
    MaNghe nvarchar(50) CONSTRAINT khoa_chinh PRIMARY KEY ,  
    TenNghe nvarchar(150)  
);
```

4. Bảng (Table)

4.2 *Lệnh Sửa cấu trúc một bảng*

+ Cú pháp: ALTER TABLE <tên_bảng>

+ **Ví dụ 1:** Ta tạo thêm một cột có tên GhiChu vào trong bảng tbl_NGHEHOC.

Ta viết đoạn lệnh sau:

```
ALTER TABLE tbl_NGHEHOC ADD GhiChu nvarchar(255);
```

+ **Ví dụ 2:** Ta muốn thay đổi độ rộng kiểu dữ liệu của cột TenNghe từ 150 lên 255.

Ta viết đoạn lệnh sau:

```
ALTER TABLE tbl_NGHEHOC ALTER COLUMN TenNghe nvarchar(255);
```

4. Bảng (Table)

4.3 *Lệnh Xóa một bảng*

+ **Cú pháp:**

DROP TABLE <tên_bảng>

+ **Ví dụ:** Xóa bảng tbl_NGHEHOC

Ta viết đoạn lệnh sau:

DROP TABLE tbl_NGHEHOC;

Bài thực hành 1

Yêu cầu:

1-Tạo file cơ sở dữ liệu có tên, **quanlysinhvien**

2-Tạo 4 bảng sau:

+ Bảng Danh mục khoa lưu tên:

tbl_DMKHOA

Column Name	Data Type	Allow Nulls
<u>MaKhoa</u>	nvarchar(10)	no
TenKhoa	nvarchar(50)	

* **MaKhoa** là khóa chính

Bài thực hành 1

+ Bảng SinhVien lưu tên:

tbl_SINHVIEN

Column Name	Data Type	Allow Nulls
<u>MaSV</u>	nvarchar(10)	no
HoSV	nvarchar(50)	
TenSV	nvarchar(50)	
GioiTinh	nvarchar(10)	
NgaySinh	Date	
NoiSinh	nvarchar(50)	
DiaChi	nvarchar(250)	
Quan	nvarchar(50)	
MaKhoa	nvarchar(10)	
HocBong	int	

* **MaSV** là khóa chính

Bài thực hành 1

+ Bảng Môn học lưu tên:

tbl_MONHOC

Column Name	Data Type	Allow Nulls
<u>MaMH</u>	nvarchar(10)	no
TenMH	nvarchar(50)	
SoTiet	Int	

* **MaMH** là khóa chính

+ Bảng Kết quả lưu tên:

tbl_KETQUA

Column Name	Data Type	Allow Nulls
<u>MaSV</u>	nvarchar(10)	no
<u>MaMH</u>	nvarchar(10)	no
DiemTB	float	no

* Có 2 khóa chính là: **MaSV** và **MaMH**

Bài thực hành 1

+ Bảng Kết quả lưu tên:

tbl_KETQUA

Column Name	Data Type	Allow Nulls
<u>MaSV</u>	nvarchar(10)	no
<u>MaMH</u>	nvarchar(10)	no
DiemTB	float	no

* Có 2 khóa chính là: **MaSV** và **MaMH**

Create table tbl_ketqua (Masv nvarchar(10) not null, Mamh nvarchar(10) not null, Diemtb float, **constraint khoa_key primary key (Masv, Mamh))**

Bài 3.

Thao tác dữ liệu với T-SQL

1 Thêm, cập nhật và xóa dữ liệu

1.1 Thêm dữ liệu vào bảng

Cú pháp:

INSERT INTO **tên_bảng** [(**danh_sách_cột**)] **VALUES** (**danh_sách_trị**)

Ví dụ: - Thêm mới dữ liệu vào bảng Danh mục khoa (**tbl_DMKHOA**) của **quanlysinhvien**

INSERT INTO **tbl_DMKHOA** **VALUES** ('NN',N'Ngoại Ngữ')

Bảng : tbl_DMKHOA

Makhoa	Tenkhoa
NN	Ngoại ngữ
TH	Tin Học
KT	Kế toán
DCN	Điện công nghiệp
XD	Xây dựng
CK	CƠ KHÍ

1 Thêm, cập nhật và xóa dữ liệu

1.1 Thêm dữ liệu vào bảng

Bảng: **tbl_SINHVIENT**

Ma SV	Ho SV	Ten SV	Gioi Tinh	Ngay Sinh	Noi Sinh	DiaChi	Quan	Ma Khoa	Hoc Bong
TH01	Nguyễn thị	Hải	Nữ	23/02/97	TP.HCM	12 Võ văn Tần	Q3	TH	
TH02	Trần văn	Chinh	Nam	24/12/97	TP.HCM	34 Nguyễn bình khiêm	Q1	TH	80000
TH03	Lê Bạch	Yến	Nữ	21/02/96	Tiền Giang	765 PasTeur	Q3	TH	
NN01	Trần Anh	Tuấn	Nam	20/11/97	Long An	12 Điện biên phủ	BT	NN	80000
NN02	Trần Thanh	Triều	Nam	13/02/97	Nghệ An	3 Nguyễn thiện thuật	Q3	NN	
NN03	Nguyễn văn	Chính	Nam	31/12/95	TP.HCM	12 Lê Lợi	Q5	NN	
KT01	Trần thanh	Mai	Nữ	12/08/97	Bến tre	32 Lê lai	Q1	KT	
KT02	Trần Thu	Thùy	Nữ	01/01/97	Bình Dương	45 Nguyễn thái sơn	BC	KT	80000
KT03	Lê thị	Thanh	Nữ	01/01/95	Tây Ninh	345 Nguyễn kiệm	Q1	KT	
DCN01	Trần Trọng	Kim	Nam	20/12/77	Đồng Nai	123 Lê văn sỹ	Q1	DCN	
DCN02	Tào Hồng	Thiên	Nam	01/07/78	TP.Huế	56 Hoàng Diệu	TB	DCN	80000
XD01	Hoàng Minh	Châu	Nam	03/05/76	Tây Ninh	102 Kỳ Đồng	TB	XD	
XD02	Phan quỳnh	Hương	Nữ	05/05/79	Bình Dương	234 Phạm văn hai	GV	XD	80000 ₄₀

1 Thêm, cập nhật và xóa dữ liệu

1.1 Thêm dữ liệu vào bảng

Bảng : tbl_MONHOC

MaMH	TenMH	SoTiet
01	Cơ sở Dữ liệu	30
02	Mạng máy tính	30
03	Cấu trúc máy tính	30
04	Tiếng Anh	30
05	Dịch thuật	90
06	Tiếng Trung	90
07	Mạch điện	90
08	Đo lường điện	45
09	Máy điện 1	45
10	Lý thuyết thống kê	45
11	Lý thuyết kế toán	60
12	Tin học kế toán	60
13	Vật liệu xây dựng	15
14	Vẽ kỹ thuật và đọc bản vẽ	45
15	Xây gạch	180

1 Thêm, cập nhật và xóa dữ liệu

1.1 Thêm dữ liệu vào bảng

Bảng: **tbl_KETQUA**

MaSV	MaMH	Diem
TH01	01	2
TH01	02	7
TH01	03	5
TH02	01	10
TH02	02	9
TH02	03	6.5
TH03	01	5
TH03	02	3
TH03	03	10
NN01	04	9
NN01	05	8
NN01	06	7
NN02	04	5
NN02	05	8
NN02	06	9
NN03	04	10
NN03	05	9
NN03	06	8

DCN01	07	10
DCN01	08	9
DCN01	09	8
DCN02	07	7
DCN02	08	6
DCN02	09	5
KT01	10	6
KT01	11	8
KT01	12	7
KT02	10	9
KT02	11	7
KT02	12	6
KT03	10	5
KT03	11	8
KT03	12	9
XD01	13	4
XD01	14	5
XD01	15	6
XD02	13	4
XD02	14	5
XD02	15	10

1 Thêm, cập nhật và xóa dữ liệu

1.2 Cập nhật dữ liệu

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: - Cập nhật mới dữ liệu vào bảng Môn học (**tbl_MONHOC**) của **quanlysinhvien**

Cú pháp:

```
UPDATE tbl_MONHOC SET Sotiet = 150 WHERE Mamh = '15'
```


1 Thêm, cập nhật và xóa dữ liệu

1.3 Xóa dữ liệu

+ Xóa dữ liệu 1 dòng theo điều kiện

Cú pháp:

DELETE FROM tên_bảng [FROM danh_sách_bảng] [WHERE điều_kiện]

Ví dụ: - Xóa dữ liệu trong bảng Danh mục khoa (tbl_DMKHOA) có mã KHOA là 'CK' của quanlysinhvien

Cú pháp:

DELETE FROM tbl_DMKHOA WHERE Makhoa = 'CK'

1 Thêm, cập nhật và xóa dữ liệu

1.3 Xóa dữ liệu

+ Xóa tất cả dữ liệu của bảng (Table)

Cú pháp:

TRUNCATE TABLE <tên_bảng>

Ví dụ: - Xóa toàn bộ dữ liệu trong bảng tbl_tam

TRUNCATE TABLE tbl_tam;

2 Câu lệnh Select truy xuất dữ liệu

Cú pháp:

```
SELECT [ALL | DISTINCT][TOP n] danh_sách_ch_ọn  
[INTO tên_bảng_mới]  
FROM danh_sách_bảng/khung_nhìn  
[WHERE điều_kiện]  
[GROUP BY danh_sách_cột]  
[HAVING điều_kiện]  
[ORDER BY cột_sắp_xếp]  
[COMPUTE danh_sách_hàm_gộp [BY danh_sách_cột]]
```

2. Câu lệnh Select truy xuất dữ liệu

2.1 Chọn tất cả các cột:

- Cú pháp:

SELECT * FROM <tên_bảng>

- Ví dụ: Liệt kê tất cả các thông tin trong bảng sinh viên (tbl_SINHVIEN).

Ta viết lệnh: SELECT * FROM tbl_SINHVIEN;

2.2 Chọn một số cột cụ thể:

- Cú pháp:

SELECT <danh_sách_tên_cột> FROM <tên_bảng>

- Ví dụ: Liệt kê các thông tin sinh viên gồm: mã sinh viên, họ sinh viên, tên sinh viên, ngày sinh, giới tính.

Ta viết lệnh: SELECT masv, hosv, tensv, ngaysinh, gioitinh FROM tbl_sinhvien;

2. Câu lệnh Select truy xuất dữ liệu

2.3 Thay đổi tiêu đề cột:

- Cú pháp:

```
SELECT <Tiêu_đề_cột = tên_field_của_bảng |  
      Tên_field_của_bảng AS Tiêu_đề_cột |  
      Tên_field_của_bảng Tiêu_đề_cột > FROM <tên_bảng>;
```

- **Ví dụ:** Thay đổi tên tiêu đề có dấu tiếng việt như sau: hosv= Họ sinh viên, tensv=Tên sinh viên, ngaysinh=Ngày sinh.

Ta viết lệnh:

```
SELECT [Họ sinh viên]=hosv, tensv AS [Tên sinh viên], ngaysinh [ngày sinh], gioitinh  
FROM tbl_sinhvien;
```

2. Câu lệnh Select truy xuất dữ liệu

2.4 Loại bỏ các dòng dữ liệu trùng nhau:

- Cú pháp:

```
SELECT distinct <danhsach_tencot> FROM <tenbang>
```

- Ví dụ: Loại bỏ các tên sinh viên trùng nhau trong danh sách sinh viên.

Ta viết lệnh:

```
SELECT distinct hosv, tensv FROM tbl_sinhvien;
```

2.5 Lựa chọn một số lượng giới hạn các dòng:

- Cú pháp:

```
SELECT Top <sodong_hien_thi> <danhsach_tencot> FROM <tenbang>;
```

- Ví dụ: Chỉ liệt kê 3 sinh viên trong danh sách nhiều sinh viên.

```
SELECT Top 3 hosv, tensv FROM tbl_sinhvien;
```

2. Câu lệnh Select truy xuất dữ liệu

2.6 Mệnh đề FROM:

- Trong mệnh đề FROM có thể sử dụng bí danh (alias) nhằm làm cho câu truy vấn dễ nhìn hơn.
- Ví dụ: *Select * from tbl_sinhvien* **hs**;

2. Câu lệnh Select truy xuất dữ liệu

2.7 Mệnh đề WHERE:

Trong mệnh đề Where – điều kiện đối với việc truy xuất dữ liệu.

+ Cú pháp:

Select < danh sách tên cột > from < tên_bảng > Where < danh sách điều kiện >;

+ Ví dụ: Lọc danh sách sinh viên có năm sinh lớn hơn năm 2006.

Ta viết lệnh:

Select * from tbl_sinhvien WHERE *ngaysinh* > 01/01/2006;

2. Câu lệnh Select truy xuất dữ liệu

2.7 Mệnh đề WHERE:

Trong mệnh đề WHERE thường sử dụng:

- Các toán tử kết hợp điều kiện (AND, OR)
- Các toán tử so sánh
- Kiểm tra giới hạn của dữ liệu (BETWEEN/ NOT BETWEEN)
- Tập hợp
- Kiểm tra khuôn dạng dữ liệu.
- Các giá trị NULL
- Các toán tử so sánh:

Toán tử	Ý nghĩa
=	Bằng
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
<>	Khác
!>	Không lớn hơn
!<	Không nhỏ hơn

2. Câu lệnh Select truy xuất dữ liệu

2.7 Mệnh đề WHERE:

Trong mệnh đề WHERE thường sử dụng:

- Toán tử LIKE/ NOT LIKE và ký tự đại diện (WildCard)

Ký tự đại diện	Ý nghĩa
%	Chuỗi ký tự bất kỳ gồm không hoặc nhiều ký tự
_	Một ký tự bất kì
[]	Một ký tự nằm trong giới hạn được chỉ định. Ví dụ:[a-f] hàm ý chỉ một trong các ký tự: a, b, c, d, e, f.
[^]	Một ký tự không nằm trong giới hạn được chỉ định. Ví dụ:[^a-f] hàm ý chỉ một ký tự khác tất cả các ký tự: a, b, c, d, e, f.

Ví dụ: Lọc danh sách sinh viên có họ là ‘ Nguyễn ‘.

*Select * from tbl_sinhvien WHERE hosv LIKE N'Nguyễn %';*

Lưu ý: Để so sánh font chữ có dấu trong mệnh đề ta thêm chữ N phí trước 1 chuỗi.

2. Câu lệnh **Select** truy xuất dữ liệu

2.8 Mệnh đề **GROUP BY**:

Mệnh đề **GROUP BY** sử dụng trong câu lệnh **SELECT** nhằm gom nhóm các dữ liệu, và trên mỗi nhóm dữ liệu thực hiện tính toán các giá trị thống kê như: tính tổng, tính trung bình,...

*Các hàm gộp sử dụng trong mệnh đề **GROUP BY**:*

- * Hàm **Sum**(biểu_thức) – tính tổng.
- * Hàm **Avg**(biểu_thức) – tính trung bình.
- * Hàm **Count**(biểu_thức) – đếm số giá trị.
- * Hàm **Count**(*) – đếm số các dòng được chọn.
- * Hàm **Max**(biểu_thức) – tính giá trị lớn nhất.
- * Hàm **Min**(biểu_thức) – tính giá trị nhỏ nhất.

2. Câu lệnh Select truy xuất dữ liệu

2.8 Mệnh đề GROUP BY:

- **Ví dụ 1:** Tính tuổi trung bình, tuổi nhỏ nhất và lớn nhất của sinh viên.

Ta viết lệnh:

```
select min(year(getdate())-year(ngaysinh)) as tuoi_nhonhat, max(year(getdate())-  
year(ngaysinh)) as tuoi_lonnhat, avg(year(getdate())-year(ngaysinh)) as tuoi_trungbinh from  
tbl_sinhvien;
```

- **Ví dụ 2:** Từ bảng tbl_sinhvien thống kê số lượng học sinh theo từng khoa.

Ta viết lệnh:

```
Select Makhoa, count(masv) as TongSoHS from tbl_sinhvien GROUP BY Makhoa;
```

2. Câu lệnh Select truy xuất dữ liệu

2.9 Mệnh đề HAVING:

Mệnh đề HAVING được sử dụng nhằm chỉ định điều kiện đối với các giá trị thống kê được sản sinh từ các hàm gộp. *Mệnh đề HAVING thường không thực sự có nghĩa nếu như không sử dụng kết hợp với mệnh đề GROUP BY.*

+ Ví dụ: Từ bảng tbl_sinhvien ta liệt kê các nghề có hơn 10 sinh viên đăng ký học.

Ta viết lệnh:

```
Select makhoa, count(masv) as TongSoHS from tbl_sinhvien GROUP BY makhoa  
HAVING count(masv) > 10;
```

Bài tập thực hành

Bài tập : Sử dụng CSDL **quanlysinhvien** đã tạo ở bài học trước để tạo các truy vấn theo yêu cầu sau :

A. Truy vấn (select...)

1. Hiển thị danh sách sinh viên gồm: MaSV, HoSV, TenSV, NgaySinh, GioiTinh.
2. Hiển thị danh sách sinh viên như câu 1 nhưng sắp xếp theo MaSV tăng dần.
3. Hiển thị danh sách sinh viên gồm: Masv, HoSV, TenSV, NgaySinh, GioiTinh, Makhoa, HocBong (sắp xếp Makhoa tăng dần, MaSV giảm dần).
4. Hiển thị bảng điểm sinh viên gồm: MaSV, HoSV, TenSV, MaMH, Diem.
5. Hiển thị thông tin sinh viên gồm: MaSV, HoSV, TenSV, TenMH, SoTiet
6. Hiển thị danh sách sinh viên có tên sinh viên là “Chinh”.
7. Hiển thị danh sách sinh viên có nơi sinh là “TP.HCM”.
8. Hiển thị danh sách sinh viên có nơi sinh là “TP.HCM” hoặc “Hà Nội”.
9. Hiển thị danh sách sinh viên có ơ Sinh là “Hà Nội” và học Khoa “tin học”.
10. Hiển thị danh sách sinh viên có Họ là “Trần” và học Khoa “ngoại ngữ”.
11. Hiển thị danh sách sinh viên có Mã số bắt đầu là B và học Khoa “ngoại ngữ”.
12. Hiển thị danh sách sinh viên có GioiTinh là “Nữ” và nơi sinh là “Tây Ninh”.
13. Hiển thị danh sách sinh viên có ngày sinh lớn hơn ngày 01/01/1997.

Bài tập thực hành

B. Truy vấn với các cột tính toán tự tạo ra thông qua các hàm gộp:

1. Hiển thị danh sách sinh viên gồm: MaKhoa, MaSV, HoSV, TenSV, ngaysinh, Tuoi và sắp xếp tuổi theo thứ tự tăng dần. Tính tuổi theo công thức: **Tuoi = (year(getdate())-year(ngaysinh))**.
2. Hiển thị danh sách sinh viên gồm: MaSV, HoTenSV, GioiTinh, MaKhoa, TenKhoa. Biết rằng **HoTenSV= [HoSV] & [TenSV]**.
3. Hiển thị danh sách sinh viên gồm: MaSV, HoTenSV, Ngaysinh, NoiSinh. Nhưng chỉ lọc các sinh viên sinh vào tháng 12.
4. Hiển thị danh sách sinh viên gồm: MaSV, HoTenSV, Ngaysinh, NoiSinh. Nhưng chỉ lọc Sinh viên sinh vào tháng 11 và năm 1997.
5. Hiển thị danh sách sinh viên gồm: MaSV, HoTenSV, HocBong, MaKhoa, TenKhoa, HocBongTang. Biết rằng **HocBongTang =(10%*[HocBong] +[HocBong])**.
6. Hiển thị danh sách sinh viên gồm các thông tin sau: MaSV, HoTenSV, TenMH, Diem, DanhGia. Biết rằng: Nếu điểm < 5 là Rớt, ngược lại là Đậu.

Bài tập thực hành

C. Truy vấn tính tổng, thống kê

1. Thống kê số sinh viên của từng Quận, gồm: Quận, Tổng số SV
2. Thống kê số sinh viên của từng Khoa, gồm: Mã khoa, Tên khoa, Tổng số SV, chỉ liệt kê khoa “Tin học”.
3. Thống kê số sinh viên của từng môn học, gồm: Mã môn học, Tên môn học, Tổng số SV.
4. Thống kê số sinh Viên nữ của từng Khoa, gồm: Mã khoa, Tên khoa, Tổng số sinh viên nữ.
5. Cho biết **tuổi trung bình** của Sinh viên của từng Khoa, gồm: Mã khoa, Tên khoa, Tuổi trung bình.
6. Hiện thị bảng điểm tổng hợp của sinh viên gồm: Mã sinh viên, Họ Tên SV, Điểm TB, Tổng số môn thi. Điểm TB là trung bình điểm thi của các môn đã học, Tổng số môn thi là tổng số các môn thi của sinh viên.

Bài 4.

Định nghĩa Thủ tục với T-SQL

1. Khái niệm thủ tục

Thủ tục (Stored Procedure) là một đối tượng trong CSDL, bao gồm nhiều câu lệnh SQL được tập hợp lại với nhau thành một nhóm, và tất cả các lệnh này sẽ được thực thi khi thủ tục được gọi.

2. Định nghĩa thủ tục

- **Cú pháp:**

CREATE PROCEDURE <tên_thủ_tục>

[< danh_sách_tham_số >]

[WITH RECOMPILE | ENCRYPTION | RECOMPILE, ENCRYPTION]

AS

Begin

[< các_câu_lệnh_của_thủ_tục >]

End;

2. Định nghĩa thủ tục

Trong đó:

+ **WITH RECOMPILE** : yêu cầu SQL Server biên dịch lại thủ tục lưu trữ mỗi khi được gọi.

+ **WITH ENCRYPTION** : yêu cầu SQL Server mã hóa thủ tục lưu trữ

+ **< các_câu_lệnh_của_thủ_tục >** : Các câu lệnh SQL. Các lệnh này có thể nằm trong cặp BEGIN...END hoặc không.

2. Định nghĩa thủ tục

* Khai báo biến sử dụng trong Thủ tục (khai báo tham số)

+ Trong thủ tục lưu trữ có thể có các biến nhằm lưu các kết quả tính toán hay truy xuất từ CSDL. Các biến trong thủ tục được khai báo bằng từ khóa DECLARE.

+ Cú pháp:

```
DECLARE @tên_biến kiểu_dữ_liệu
```

+ Ví dụ:

```
Declare @diemthi float
```

+ Lưu ý: Nếu khai báo biến không lưu trữ để tính toán thì chúng ta có thể bỏ từ khóa DECLARE lúc khai báo biến.

```
@diem float
```

2. Định nghĩa thủ tục

- Ví dụ 1: Viết thủ tục xem danh sách sinh viên.

- Mở cửa sổ soạn code SQL, gõ đoạn code sau:

```
create procedure SP_xemDSSV
```

```
AS
```

```
Begin
```

```
    select * from tbl_sinhvien
```

```
End;
```

- Quét chọn đoạn code + nhấn F5 để tạo thủ tục.

3. Gọi thủ tục thực thi chương trình

+ Cú pháp:

EXEC Tên_thủ_tục_lưu_trữ [danh_sách_tham_số]

+ Gọi Thủ tục *SP_xemDSSV*:

- Gõ code: *exec SP_xemDSSV*
- Quét chọn + nhấn F5 để gọi thủ tục thực thi.

+ Ví dụ 2: Viết thủ tục có dùng tham số, lọc danh sách sinh viên của khoa “Tin học”

* Ta soạn code và nhấn F5 để tạo thủ tục sau:

```
create procedure sp_TimSVtheomakhoa
```

```
@ma_khoa
```

```
AS
```

```
Begin
```

```
select * from tbl_sinhvien where makhoa = (@ma_khoa)
```

```
End;
```

* Ta gọi thủ có truyền tham số:

```
Exec sp_TimSVtheomakhoa 'TH'
```

4. Khai báo tham số với giá trị mặc định trong thủ tục

Các tham số được khai báo trong thủ tục có thể nhận các giá trị mặc định. Giá trị mặc định sẽ được gán cho tham số trong trường hợp không truyền đối số cho tham số khi có lời gọi đến thủ tục.

+ **Cú pháp:**

@tên_tham_số kiểu_dữ_liệu = giá_trị_mặc_định

+ Ví dụ: Lộc thông tin sinh viên có mã **TH01**.

```
Create procedure sp_loc_sinhvien  
@masinhvien nvarchar(10) = 'TH01'  
As  
Begin  
    select * from tbl_sinhvien where masv = (@masinhvien)  
End;
```

+ Ta gọi thủ tục: exec **sp_loc_sinhvien**

5. Giá trị trả về trong thủ tục

Trong trường hợp cần phải giữ lại giá trị của đối số sau khi kết thúc thủ tục, ta phải khai báo tham số của thủ tục.

+ Cú pháp:

`@tên_tham_số kiểu_dữ_liệu OUTPUT`

+ Ví dụ: Viết thủ tục tính tổng của 2 số `a + b` như sau:

```
create procedure sp_tong
  @a int, @b int,
  @kq int output
As
Begin
    select @kq = @a + @b
End;
```

Thực thi thủ tục:

```
Declare @tong int /* khai báo biến
set @tong = 0 /* khởi tạo biến
exec sp_tong 10, 5, @tong output /* gọi thủ tục
select @tong /* xem kết quả
```

=> Kết quả là 15.

6. Sửa/Xóa thủ tục

* Sửa thủ tục:

- Cú pháp:

```
ALTER PROCEDURE tên_thủ_tục [< danh_sách_tham_số >]  
    [WITH RECOMPILE | ENCRYPTION | RECOMPILE, ENCRYPTION]  
    AS  
    Begin  
        < các_câu_lệnh_của_thủ_tục >  
    End;
```

Lưu ý: Câu lệnh thực hiện tương tự như câu lệnh CREATE PROCEDURE.

* Xóa thủ tục

- Cú pháp:

```
DROP PROCEDURE < tên_thủ_tục >
```

7. Câu lệnh có cấu trúc điều khiển trong thủ tục

7.1 Cấu trúc While (cấu trúc lặp)

- Cú pháp:

```
While <Biểu_thức_logic>  
Begin  
    <Các_lệnh>  
End
```

- Ví dụ: In ra màn hình 10 số nguyên dương từ 1 đến 10.

Ta viết như sau:

```
Declare @a int  
set @a = 1  
Print 'Các số nguyên từ 1 đến 10:'  
While @a <= 10  
Begin  
    Print convert(char(3), @a)  
    Set @a = @a + 1  
End
```

7. Câu lệnh có cấu trúc điều khiển trong thủ tục

7.2 Cấu trúc IF (cấu trúc rẽ nhánh)

+ Cú pháp:

```
If <biểu_thức_logic>  
    Câu_lệnh 1/Khởi_lệnh 1  
else  
    Câu_lệnh 2/Khởi_lệnh 2
```

+ Ví dụ: Ta viết lệnh in ra màn hình số a với các trường hợp như sau:

Nếu $a > 0$, in ra a là số dương. Nếu $a < 0$, in ra a là số âm. Các trường hợp còn lại, in ra a là 'số 0' và kết thúc.

Ta viết lệnh như sau:

```
Declare @a int  
set @a = - 5  
if @a > 0  
    print convert(char(5),@a) + ' là số dương'  
else if @a < 0  
    print convert(char(5),@a) + ' là số âm'  
else print 'Số 0'
```

→ Kết quả: -5 là số âm

7. Câu lệnh có cấu trúc điều khiển trong thủ tục

7.3 Cấu trúc Case...then... (cấu trúc lựa chọn)

+ Cú pháp:

*Case Biểu_thức When Giá_trị_1 then kết_quả_1 [When Giá_trị_2 then kết_quả_2...]
[Else kết_quả_N+1] End*

+ Ví dụ: Hiển thị danh sách các môn học có trong bảng danh mục môn học theo từng loại môn học.

Ta viết lệnh như sau: (Câu lệnh select được viết bên trong Thủ tục)

```
Select Loai = case left(Mamh,3) when 'LTC' then 'Lập trình C' when 'MMT' then 'Mạng máy tính'  
when 'SQL' then 'Hệ quản trị CSDL SQL server' when 'WSR' then 'HĐH Windows server' else  
'Chưa phân loại' End, Mamonhoc, TenMonHoc, SoTiet From tbl_MonHoc
```

Ôn tập – Thực hành

1. *Viết thủ tục nhập mới sinh viên.*
2. *Viết thủ tục liệt kê sinh viên khoa tin học.*
3. *Viết thủ tục xóa tên sinh viên theo mã sinh viên*
4. *Viết ba thủ tục tính hiệu($a-b$), tích($a*b$), thương(a/b) của 2 số nguyên a và b .*

Bài 5.

Định nghĩa Hàm với T-SQL

1. Khái niệm Hàm

- Hàm là một chương trình con (Sub Program).
- Hàm dùng để thực hiện một xử lý tính toán và trả về kết quả. Ví dụ: tính tuổi, tính điểm trung bình, tính tổng nhập, tổng xuất của vật tư...
- Hàm chỉ cho phép loại tham số đầu vào (INPUT).

Phân loại:

- Hàm đơn trị (Scalar Function).
- Hàm đọc bảng (Inline Table-Valued Function).
- Hàm tạo bảng (Multi-Statement Table-Valued Function).

Lệnh tạo Hàm:

CREATE FUNCTION dùng để tạo hàm (UDF: User-Defined Function) trong CSDL.

2. Phân loại Hàm

2.1 Hàm đơn trị (Scalar Function)

Cú pháp:

CREATE FUNCTION Tên_hàm(Khai_báo_tham_số)

RETURNS Kiểu_dữ_liệu_trả_về

AS

BEGIN

Lệnh...

RETURN Giá_trị_trả_về

END

GO

2. Phân loại Hàm

2.1 Hàm đơn trị (Scalar Function)

Ví dụ 1: Hàm tính diện tích hình tròn khi biết bán kính

```
CREATE FUNCTION dbo.Dien_tich_hinh_tron (@Ban_kinh float = 1.0)
    RETURNS float
    WITH RETURNS NULL ON NULL INPUT
AS
BEGIN
    RETURN PI() * POWER(@Ban_kinh, 2)
END
```

GO

* Gọi thực hiện

+ Cách 1

```
PRINT dbo.Dien_tich_hinh_tron(3.5)
```

GO

+ Cách 2

```
DECLARE @r float = 3.5
```

```
DECLARE @s float = dbo.Dien_tich_hinh_tron(@r)
```

```
PRINT @s
```

GO

2. Phân loại Hàm

2.2 Hàm đọc bảng (Inline Table-Valued Function).

Cú pháp:

```
CREATE FUNCTION Tên_hàm(Khai_báo_tham_số)
RETURNS TABLE
AS
RETURN
(
    Câu_truy_vấn
)
GO
```

2. Phân loại Hàm

2.2 Hàm đọc bảng (Inline Table-Valued Function).

Ví dụ 2: Tạo hàm đọc danh sách sinh viên theo mã khoa truyền vào

```
CREATE FUNCTION Danh_sach_sinh_vien(@mymakhoa nvarchar(10))  
RETURNS TABLE  
AS  
RETURN  
(  
    SELECT *  
    FROM tbl_SINHVIEN  
    WHERE makhoa = @mymakhoa OR @mymakhoa = 0  
)  
GO
```

*** Gọi hàm**

```
SELECT * FROM Danh_sach_sinh_vien('TH')
```

2. Phân loại Hàm

2.3 Hàm tạo bảng (Multi-Statement Table-Valued Function).

Cú pháp:

```
CREATE FUNCTION Tên_hàm(Khai_báo_tham_số)
RETURNS Tên_biến TABLE(Khai_báo_cấu_trúc_bảng)
AS
BEGIN
    INSERT | UPDATE | DELETE
    RETURN
END
GO
```

2. Phân loại Hàm

2.3 Hàm tạo bảng (Multi-Statement Table-Valued Function).

Ví dụ 3: Tạo hàm thống kê sinh viên theo khoa

```
CREATE FUNCTION Thong_ke_sinh_vien_theo_khoa()  
RETURNS @bang TABLE  
(  
    Makhoa nvarchar(10) , Tenkhoa nvarchar(30) , Sosv tinyint  
)  
AS  
BEGIN  
    INSERT INTO @bang  
    SELECT sv.makhoa, k.tenkhoa, COUNT(*)  
    FROM tbl_SINHVIEN sv JOIN tbl_DMKHOA k ON sv.makhoa = k.makhoa  
    GROUP BY sv.makhoa, k.tenkhoa  
    RETURN  
END
```

*** Gọi hàm**

```
SELECT * FROM Thong_ke_sinh_vien_theo_khoa()
```

Bài tập – Thực hành

1. Tạo 1 Hàm đơn trị (Scalar Function)

Viết hàm tính diện tích hình chữ nhật khi biết khoảng cách của chiều dài và chiều rộng.

2. Tạo 1 Hàm đọc bảng (Inline Table-Valued Function).

Viết hàm đọc danh sách sinh viên theo Quận, truyền vào tham số Quận.

3. Tạo 1 Hàm tạo bảng (Multi-Statement Table-Valued Function).

Viết hàm thống kê sinh viên theo độ tuổi

Bài 6.

Định nghĩa Trigger

1. Giới thiệu Trigger

- * Trigger là một thủ tục nội tại (Stored Procedure) đặc biệt, được thực thi tự động mỗi khi có một câu lệnh INSERT/ UPDATE/ DELETE làm thay đổi dữ liệu trên bảng.
 - Phải được liên kết với một bảng/ bảng ảo.
 - Không thể gọi mà được thực hiện tự động.
 - Không có tham số.
- * Lệnh cập nhật dữ liệu trên bảng > xảy ra biến cố (event) > Trigger sẽ được tự động gọi thực hiện.
- * Lệnh **CREATE TRIGGER** dùng để tạo Trigger trong CSDL.

2. Phân loại Trigger

- DDL Trigger Là Trigger xảy ra với các biến cố của Database hoặc Server
- DML Trigger:
 - + Trigger xảy ra với các hành động INSERT/ UPDATE/ DELETE trên bảng.
 - + Dùng để kiểm tra các ràng buộc toàn vẹn phức tạp.
 - + Dùng để xử lý tính toán và cập nhật tự động.

3. Data manipulation language triggers (DML triggers)

3.1 AFTER (FOR) Trigger

- Cú pháp:

```
CREATE TRIGGER Tên_Trigger  
ON Tên_bảng  
  AFTER | FOR  
  INSERT | UPDATE | DELETE  
AS  
  Các_lệnh  
GO
```

- Dùng cho việc cập nhật bảng.
- Chạy sau các hành động kiểm tra dữ liệu của các Constraint.
- Dữ liệu đã bị tạm thời thay đổi trong bảng.
- Chỉ áp dụng cho bảng.

3. Data manipulation language triggers (DML triggers)

3.1 AFTER (FOR) Trigger

- Ví dụ 1: --Tạo Trigger không cho người dùng thêm/ sửa/ xóa Điểm vào thứ 7 hoặc cn

```
USE quanlysinhvien
GO
CREATE TRIGGER tg_khong_themsuaxoa_diem_t7_cn
ON tbl_KETQUA
AFTER INSERT, UPDATE, DELETE
AS
    IF DATEPART(dw, GETDATE()) = 7 OR DATEPART(dw, GETDATE()) = 1
    BEGIN
        ROLLBACK
        RAISERROR(N'Không được phép cập nhật dữ liệu!', 16, 1)
        RETURN
    END
GO
--Kiểm tra
INSERT tbl_KETQUA(Masv, Mamh, Diem) VALUES('TH01', '04', 10)
UPDATE tbl_KETQUA SET Diem = 9 WHERE Masv = 'TH01' AND Mamh = '04'
DELETE tbl_KETQUA WHERE Masv = 'TH01' AND Mamh = '04'
```

3. Data manipulation language triggers (DML triggers)

3.2 INSTEAD OF Trigger

- Cú pháp:

```
CREATE TRIGGER Tên_Trigger  
ON Tên_bảng_ảo | Tên_bảng  
INSTEAD OF  
INSERT | UPDATE | DELETE  
AS  
    Các_lệnh  
GO
```

- Dùng cho việc cập nhật bảng hoặc bảng ảo.
- Chạy trước các hành động kiểm tra dữ liệu.
- Dữ liệu chưa bị thay đổi.
- Có thể thay thế hành động cập nhật dữ liệu bằng các hành động khác.
- Có thể áp dụng cho bảng hoặc bảng ảo.
- Thường dùng cho việc cập nhật bảng ảo.

3. Data manipulation language triggers (DML triggers)

3.2 INSTEAD OF Trigger

- Ví dụ 2:

```
USE quanlysinhvien
GO
--Tạo bảng ảo
CREATE VIEW mymonhoc
AS
    SELECT *
    FROM tbl_MONHOC mh JOIN tbl_KETQUA kq ON mh.Mamh = kq.Mamh
GO
SELECT * FROM mymonhoc
--Kiểm tra
DELETE mymonhoc WHERE mamh = '01'
```

3. Data manipulation language triggers (DML triggers)

3.2 INSTEAD OF Trigger

- Ví dụ 2:

--1.

CREATE TRIGGER tg_mymonhoc_D1

ON mymomhoc

INSTEAD OF DELETE

AS

PRINT N'Thông báo!'

GO

--Kiểm tra

DELETE mymonhoc **WHERE** Mamh = '01'

GO

3. Data manipulation language triggers (DML triggers)

3.2 INSTEAD OF Trigger

- Ví dụ 2: --2.

```
CREATE TRIGGER tg_mymonhoc_D2
ON mymonhoc
INSTEAD OF DELETE
AS
    SET NOCOUNT ON
    IF EXISTS(SELECT 1 FROM deleted WHERE monhoc IN (SELECT Mamh FROM
tbl_KETQUA))
    BEGIN
        RAISERROR(N'Không xóa vì đã có điểm thi!', 16, 1)
        RETURN
    END
    DELETE tbl_MONHOC WHERE Mamh IN (SELECT Mamh FROM deleted)
GO
--Kiểm tra
DELETE mymonhoc WHERE Mamh IN('01','14')
```


3. Data manipulation language triggers (DML triggers)

3.3 Sửa Trigger

Lệnh **ALTER TRIGGER** dùng để sửa Trigger trong CSDL

- Cú pháp:

* **AFTER (FOR) Trigger**

```
ALTER TRIGGER Tên_Trigger  
ON Tên_bảng  
    AFTER | FOR  
    INSERT | UPDATE | DELETE  
AS  
    Các_lệnh  
GO
```

* **INSTEAD OF Trigger**

```
ALTER TRIGGER Tên_Trigger  
ON Tên_bảng_ảo | Tên_bảng  
    INSTEAD OF  
    INSERT | UPDATE | DELETE  
AS  
    Các_lệnh  
GO
```

3. Data manipulation language triggers (DML triggers)

3.4 Xóa Trigger

Lệnh **DROP TRIGGER** dùng để xóa Trigger trong CSDL

- Cú pháp:

DROP TRIGGER **Ten_Trigger**

- Ví dụ:

DROP TRIGGER **tg_khong_themsuaxoa_diem_t7_cn**

Bài tập – Thực hành