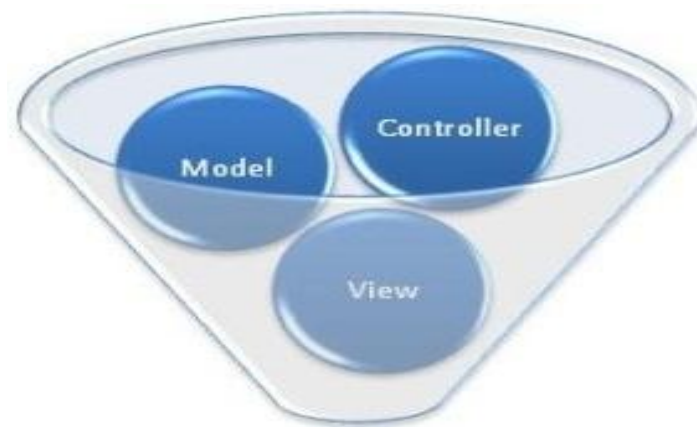


Microsoft
ASP.net MVC 5

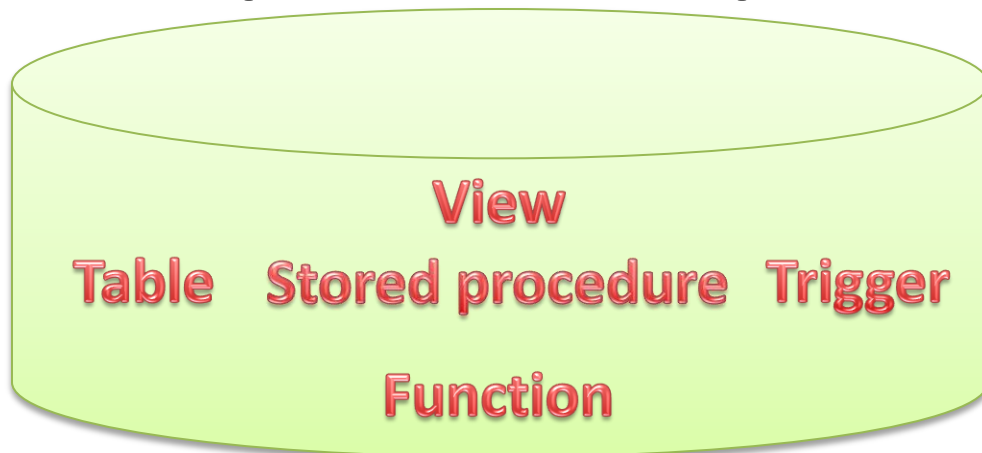


DATABASE

ThS. Nguyễn Nghiệm
0913.745.789
nghiemn@fpt.edu.vn
songlong2k@gmail.com

- ☐ Tạo, xóa, sao lưu và phục hồi dữ liệu
- ☐ Tạo, xóa, sửa đổi bảng
- ☐ Hiểu rõ EShopV10
- ☐ Thao tác và truy vấn dữ liệu trên EshopV10
- ☐ Truy vấn có kết nối nhiều bảng
- ☐ Truy vấn thông tin tổng hợp và thống kê

- ❑ Cơ sở lưu trữ và các tính năng quản lý dữ liệu như thao tác, truy vấn... được các ứng dụng (viết bằng Java, C++, PHP, C#...) sử dụng.
- ❑ Các thành phần cơ bản trong CSDL gồm bảng (**table**), khung nhìn (**view**), thủ tục lưu (**stored procedure**), hàm (**function**), **trigger**.
- ❑ CSDL được quản lý bởi **hệ quản trị CSDL quan hệ**
 - ✗ SQL Server
 - ✗ MySQL Server
 - ✗ Oracle
 - ✗ Sysbase



Bảng KHOA

MAKHOA	TENKHOA	DIENTHOAI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054822462
DHT04	Khoa Hoá học	
...	...	

Bảng LOP

MALOP	TENLOP	KHOA	HEDAOTAO	NAMNHAPHOC	SISO	MAKHOA
C24101	Toán K24	24	Chính quy	2000	5	DHT01
C24102	Tin K24	24	Chính quy	2000	8	DHT02
C24103	Lý K24	24	Chính quy	2000	7	DHT03
C24301	Sinh K24	24	Chính quy	2000	5	DHT05

Bảng SINHVIEN

MASV	HODEM	TEN	NGAYSINH	GIOTINH	NOISINH	MALOP
0241010001	Ngô Thị Nhật	Anh	Nov 27 1982	0	Quảng Ninh, Quảng Bình	C24101
0241010002	Nguyễn Thị Ngọc	Anh	Mar 21 1983	0	Tân Kỳ, Nghệ An	C24101
0241010003	Ngô Việt	Bắc	May 11 1982	1	Yên Khánh, Ninh Bình	C24101
0241010004	Nguyễn Đình	Bình	Oct 6 1982	1	Huế	C24101
0241010005	Hồ Đăng	Chiến	Jan 20 1982	1	Phong Điền, TTHuế	C24101
0241020001	Nguyễn Tuấn	Anh	Jul 15 1979	1	Đo Linh, Quảng Trị	C24102
0241020002	Trần Thị Kim	Anh	Nov 4 1982	0	Phong Điền, TTHuế	C24102
0241020003	Võ Đức	Ân	May 24 1982	1	Huế	C24102
0241020004	Nguyễn Công	Bình	Jun 6 1979	1	Thăng Bình, Quảng Nam	C24102
0241020005	Nguyễn Thanh	Bình	Apr 24 1982	1	Huế	C24102
...

- ❑ Bảng (**table**) là đơn vị chứa dữ liệu duy nhất trong CSDL, được định nghĩa gồm nhiều cột (**column**), mỗi cột có kiểu dữ liệu (**data type**) và ràng buộc (**constraints**) riêng.
- ❑ Dữ liệu lưu trong bảng gồm nhiều hàng (**row/record**), mỗi hàng lưu thông tin của mỗi thực thể (**entity**)
- ❑ Khóa chính (**primary key**) là một hoặc nhiều cột dùng để **xác định duy nhất** mỗi thực thể trong bảng

Bảng:

MonHoc

Khóa chính:

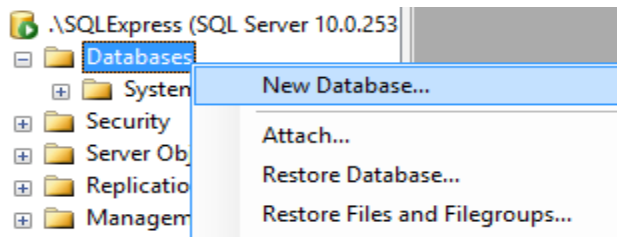
MaMonHoc

MA MONHOC	TEN MONHOC	SODVHT
HO-001	Hoá đại cương	3
TI-001	Tin học đại cương	4
TI-002	Ngôn ngữ C	5
TI-003	Lý thuyết hệ điều hành	4
TI-004	Cấu trúc dữ liệu và giải thuật	4
TO-001	Đại số tuyến tính	4
TO-002	Giải tích 1	4
TO-003	Bài tập Đại số	2
TO-004	Bài tập Giải tích 1	2
VL-001	Vật lý đại cương	3

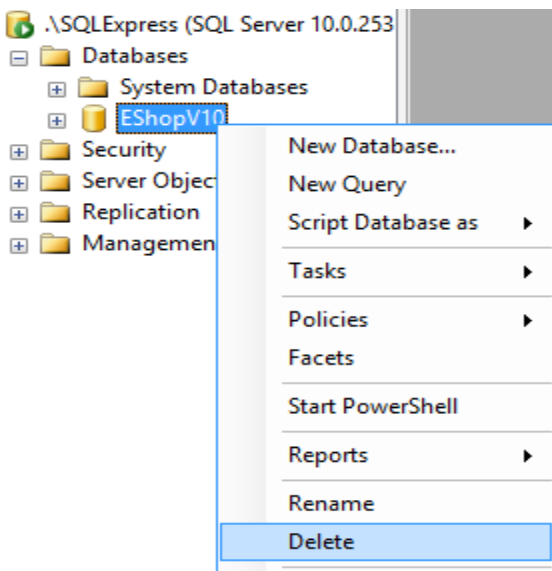
MAKHOA	TENKHOA	DIENTHOAI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054823462
...

MALOP	TENLOP	KHOA	HEDAOTAO	NAMNHAPHOC	SISO	MAKHOA
C24101	Toán K24	24	Chính quy	2000	5	DHT01
C25101	Toán K25	25	Chính quy	2001	5	DHT01
C25102	Tin K25	25	Chính quy	2001	6	DHT02
C24102	Tin K24	24	Chính quy	2000	8	DHT02
...

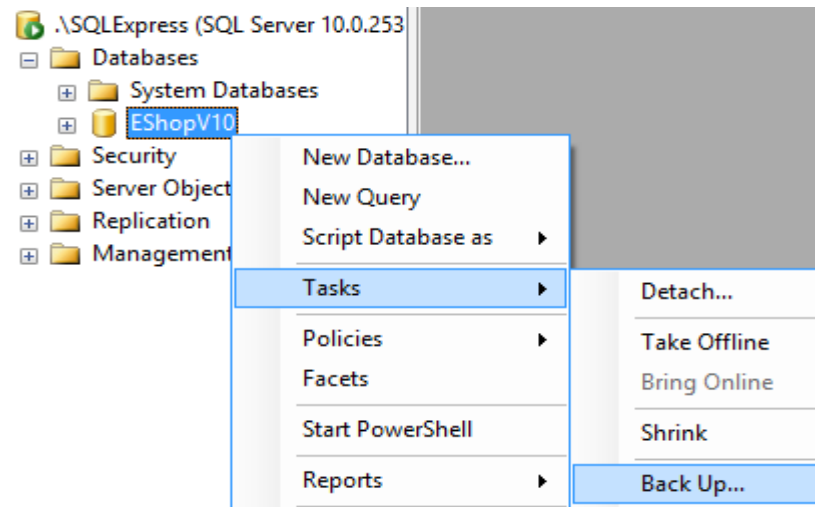
Bảng LOP



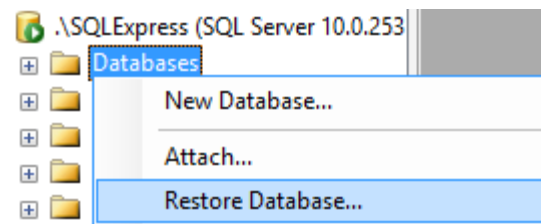
CREATE DATABASE EShopV10



DROP DATABASE EShopV10



BACKUP DATABASE EShopV10 TO DISK='c:/dbs/EShopV10.bak'



RESTORE DATABASE EShopV10 FROM DISK='c:/dbs/EShopV10.bak'

❑ Backup/Restore

- ✍ Phương pháp online (có thể thực hiện lúc CSDL đang chạy)

❑ Xuất scripts/Nhập từ scripts

- ✍ Trong suốt phiên bản

❑ Attach/Detach

- ✍ Detach và chép
- ✍ Attach các file dữ liệu đã chép lúc detach


```
CREATE TABLE <tên bảng>
```

```
(  
    <tên cột> <kiểu dữ liệu> <ràng buộc>,  
    <tên cột> <kiểu dữ liệu> <ràng buộc>,  
    ...  
    <các ràng buộc khác>  
)
```

Kiểu dữ liệu?
Ràng buộc?

```
CREATE TABLE Categories
```

```
(  
    Id int IDENTITY(1,1) NOT NULL, --cột tự tăng  
    Name nvarchar(50) NOT NULL,  
    NameVN nvarchar(50) NOT NULL,  
    PRIMARY KEY(Id) --khóa chính  
)
```

Nhóm	Kiểu	Mô tả
Chuỗi	[N] CHAR(n) , [N] VARCHAR(n) , [N] TEXT	Chuỗi có độ dài cố định, biến đổi và cực lớn. [N] chỉ định lưu unicode, (n) chỉ định số ký tự tối đa.
Số	BIT	Số nguyên 1 bit (dùng cho kiểu logic)
	SMALLINT, INT, BIGINT	Số nguyên cỡ nhỏ, vừa và lớn
	FLOAT, NUMERIC, DECIMAL	Số thực
	MONEY	Số cực lớn, lưu tiền tệ
Ngày	DATETIME, DATE, TIME	Ngày và giờ, Ngày, giờ
Nhị phân	BINARY(n) , VARBINARY(n) , IMAGE	Nhị phân số byte cố định, biến đổi và cực lớn

Ràng buộc	Ý nghĩa	Ví dụ
NULL	Cho phép null	NgaySinh NULL
NOT NULL	Bắt buộc phải nhập	Email NOT NULL
DEFAULT	Giá trị mặc định	Diem DEFAULT 0
UNIQUE	Giá trị của cột này là duy nhất	CMND UNIQUE
CHECK	Kiểm tra	CHECK(Diem >=0 AND Diem<=10)
PRIMARY KEY	Khóa chính	PRIMARY KEY (MaNV)
FOREIGN KEY	Khóa ngoại	FOREIGN KEY(MaPB) REFERENCES PhongBan(MaPB)

```
CREATE TABLE Products
```

```
(
```

```
    Id int IDENTITY(1,1) NOT NULL, --Mã tự tăng
```

```
    Name nvarchar(60) NOT NULL, --Tên hàng hóa
```

```
    UnitPrice float NOT NULL, --Đơn giá
```

```
    Image nvarchar(50) NOT NULL, --Hình ảnh
```

```
    ProductDate date NOT NULL, --Ngày sản xuất
```

```
    Available bit NOT NULL DEFAULT 0, --Tình trạng
```

```
    CategoryId int NOT NULL, --Mã loại hàng
```

```
    Description nvarchar(max) NULL, --Mô tả hàng
```

```
    PRIMARY KEY(Id), --Khóa chính
```

```
    FOREIGN KEY(CategoryId) REFERENCES Categories(Id)
```

```
        ON DELETE CASCADE, --Khóa ngoại, xóa dây chuyền
```

```
    UNIQUE(Name), --Tên duy nhất
```

```
    CHECK(UnitPrice >= 0) --Giá phải dương
```

```
)
```

```
ALTER TABLE <tên bảng>  
  ADD <định nghĩa cột> |  
  ALTER COLUMN <tên cột> <kiểu dữ liệu> [NULL | NOT NULL] |  
  DROP COLUMN <tên cột> |  
  ADD CONSTRAINT <tên ràng buộc> <định nghĩa ràng buộc> |  
  DROP CONSTRAINT <tên ràng buộc>
```

Ví dụ 1: Thêm cột Quantity vào bảng Products

```
ALTER TABLE Products ADD Quantity INT DEFAULT 0
```

Ví dụ 2: Xóa cột Description khỏi bảng Products

```
ALTER TABLE Products DROP COLUMN Description
```

Ví dụ 3: Thay đổi kiểu dữ liệu cột Description của bảng Products

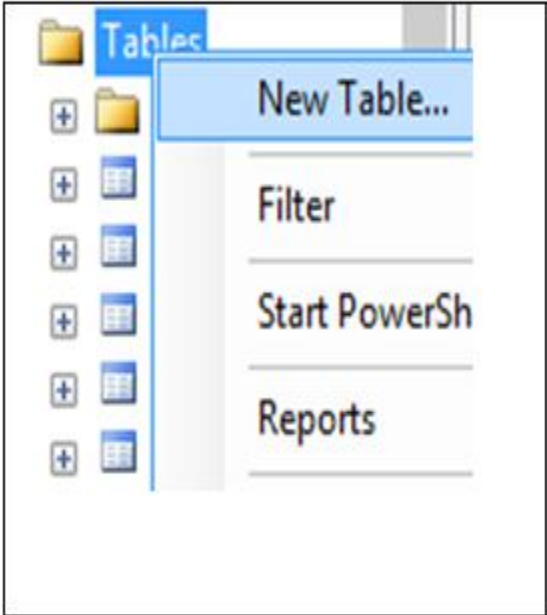
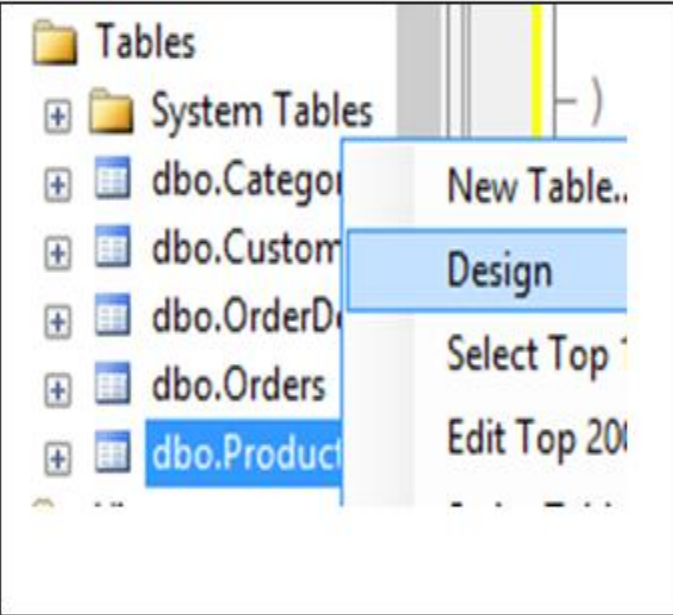
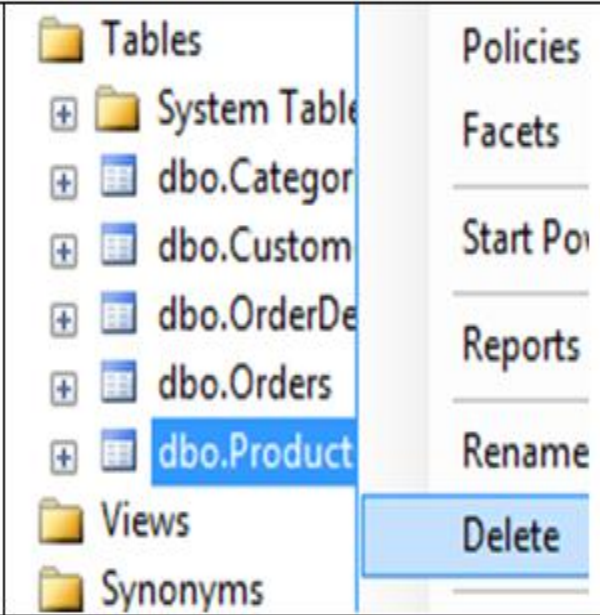
```
ALTER TABLE Products ALTER COLUMN Description NVARCHAR(1000)
```

Ví dụ 4: Thêm ràng buộc mặc định cho cột UnitPrice của bảng Products

```
ALTER TABLE Products ADD DEFAULT 0 FOR UnitPrice
```

Ví dụ 5: Thêm khóa ngoại cho bảng Products

```
ALTER TABLE Products  
  ADD FOREIGN KEY (CategoryId) REFERENCES Categories (Id)  
  ON DELETE CASCADE
```

		
Tạo bảng	Sửa bảng	Xóa bảng

CREATE TABLE

ALTER TABLE

DROP TABLE

SONGLONG **dbo.Products**

	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Name	nvarchar(60)	<input type="checkbox"/>
UnitPrice	float	<input type="checkbox"/>
Image	nvarchar(50)	<input type="checkbox"/>
ProductDate	date	<input type="checkbox"/>
Available	bit	<input type="checkbox"/>
CategoryId	int	<input type="checkbox"/>
Description	nvarchar(MAX)	<input checked="" type="checkbox"/>

Column Properties

Table Designer

Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	Mã hàng hóa
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscription	No
Identity Specification	Yes

Properties

[Tbl] dbo.Products

(Identity)

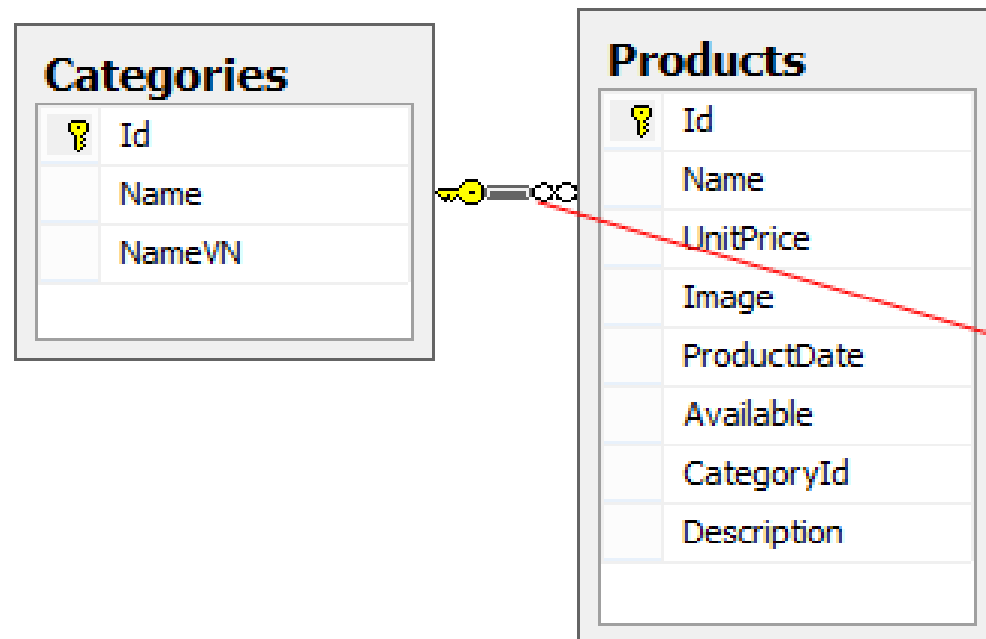
(Name)	Products
Database Name	EShopV10
Description	
Schema	
Server Name	nglong

Table Designer

Identity Column	Id
Indexable	Yes
Lock Escalation	Table
Regular Data Space Space	PRIMARY
Replicated	No
Row GUID Column	
Text/Image Filegroup	PRIMARY

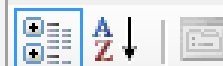
/10 - Diagram_0*

SONGLONG.EShopV10 - dbo.Products



Properties

[Rel] FK_HangHoa_Loai1



(General)

Check Existing Data On Yes

Tables And Columns S

Database Designer

Enforce For Replication Yes

Enforce Foreign Key C Yes

INSERT And UPDATE S

Delete Rule Cascade

Update Rule Cascade

Identity

No Action

(Name)

Cascade

Description

Set Null

Set Default

FOREIGN KEY(CategoryId) **REFERENCES** Categories (Id)

ON DELETE CASCADE, --Khóa ngoại, xóa dây chuyền

SONGLONG.EShopV10 - Diagram_0* SONGLONG.EShopV

	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
▶	Name	nvarchar(60)	<input type="checkbox"/>
	UnitPrice		<input type="checkbox"/>
	Image		<input type="checkbox"/>
	ProductDa		<input type="checkbox"/>
	Available		<input type="checkbox"/>
	CategoryI		<input type="checkbox"/>
	Description		<input checked="" type="checkbox"/>

🔑 Set Primary Key

➕ Insert Column

➖ Delete Column

🔗 Relationships...

🔑 Indexes/Keys...

Indexes/Keys

Selected Primary/Unique Key or Index:

IX_Products*
PK_Products

Editing properties for new unique key or index.

(General)

Columns	Name (ASC)
Is Unique	Yes
Type	Index

Identity



(Name)	IX_Products
Description	









Table Designer

Create As Clustered	No
Data Space Specification	PRIMARY
Fill Specification	

Add Delete Close

SONGLONG.EShopV10 - Diagram_0* SONGLONG.EShopV1

Column Name	Data Type	Allow Nulls
 Id	int	<input type="checkbox"/>
Name	nvarchar(60)	<input type="checkbox"/>
 UnitPrice	float	<input type="checkbox"/>
Image		
ProductDate		
Available		
CategoryId		
Description		

 Set Primary Key
 Insert Column
 Delete Column
 Relationships...
 Indexes/Keys...
 Fulltext Index...
 XML Indexes...
 Check Constraints.

?

×

Check Constraints

Selected Check Constraint:

CK_Products*

Editing properties for new check constraint. The 'Expression' property needs to be filled in before the new check constraint will be accepted.

(General)

Expression

UnitPrice >= 0

Identity

(Name)

CK_Products

Description

Đơn giá phải là số dương

...

Table Designer

Check Existing Data On Creatio

Yes

Enforce For INSERTs And UPDA

Yes

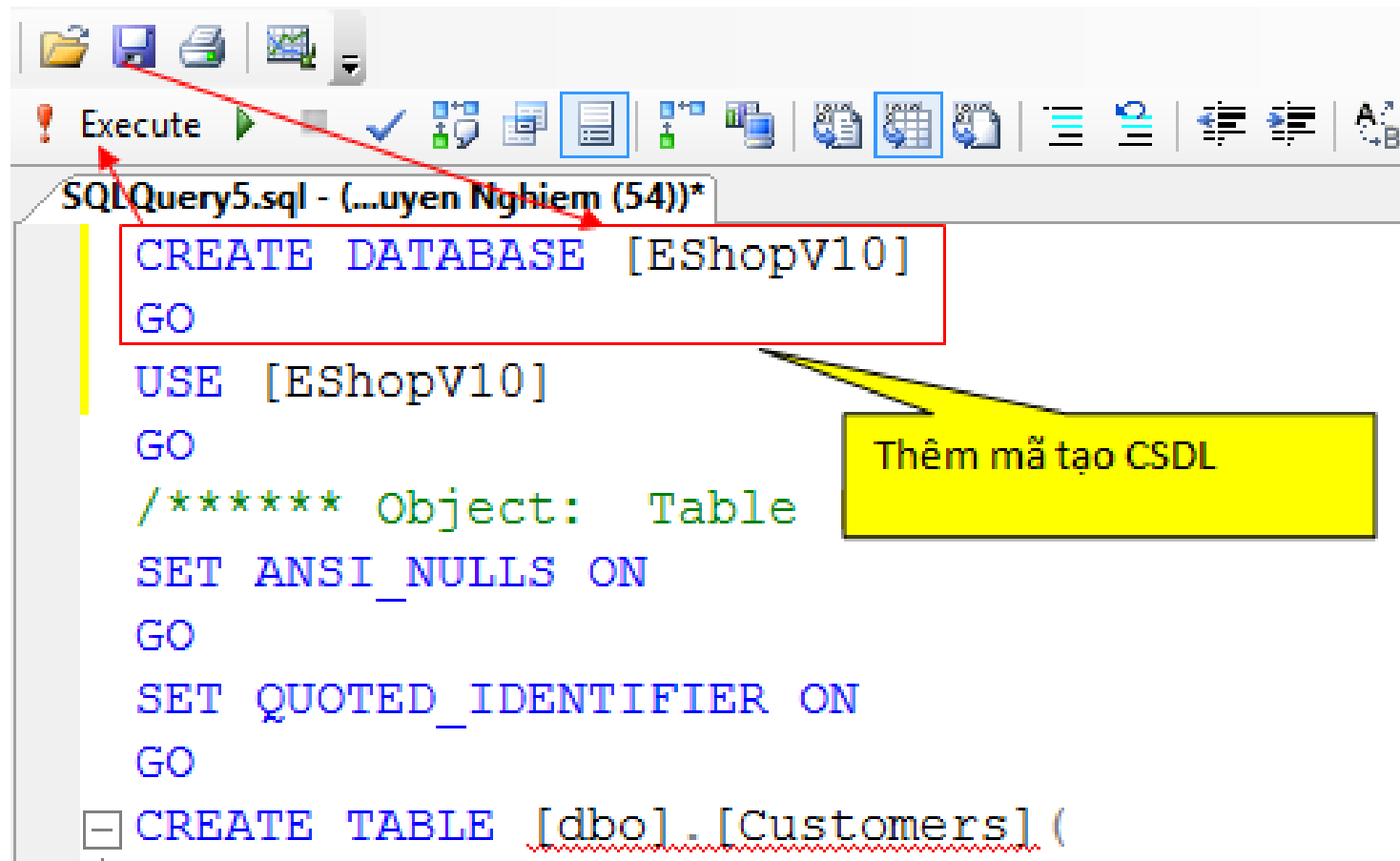
Enforce For Replication

Yes

Add

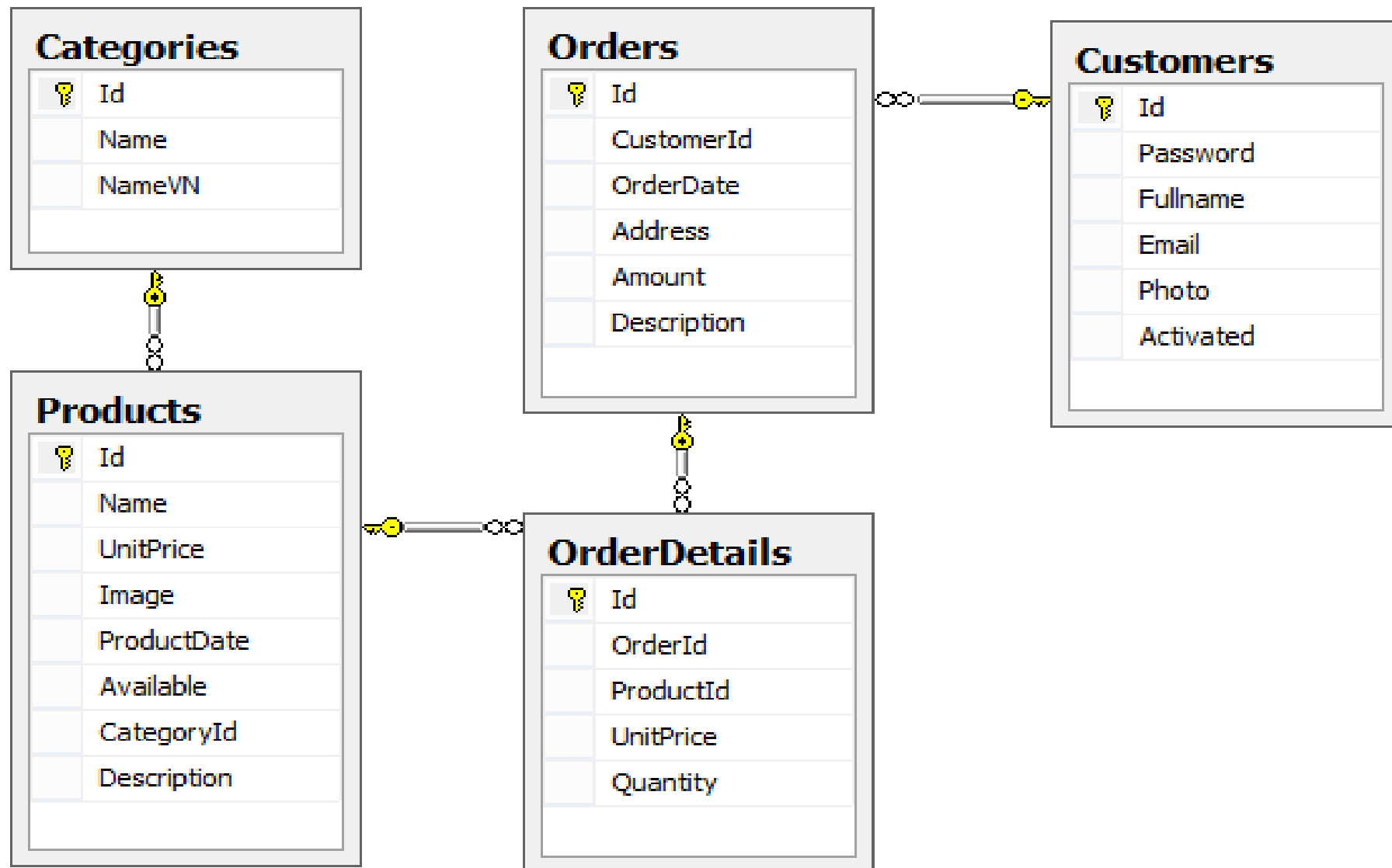
Delete

Close




The screenshot shows the SQL Server Enterprise Manager interface. The title bar of the query window reads "SQLQuery5.sql - (...uyen Nghiem (54))*". The toolbar includes an "Execute" button with a green play icon. A red rectangular box highlights the first two lines of the SQL script: "CREATE DATABASE [EShopV10]" and "GO". A yellow callout box with a black border points to the "GO" statement, containing the text "Thêm mã tạo CSDL". The SQL script is as follows:


```
CREATE DATABASE [EShopV10]
GO
USE [EShopV10]
GO
/***** Object:  Table
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Customers](
```




Categories

Column Name	Condensed Type	Nullable	Default Value	Description
 Id	int	No		Mã loại
Name	nvarchar(50)	No		Tên loại hàng
NameVN	nvarchar(50)	No		Tên tiếng Việt


OrderDetails

Column Name	Condensed Type	Nullable	Default Value	Description
 Id	int	No		Mã chi tiết
OrderId	int	No		Mã hóa đơn
ProductId	int	No		Mã hàng hóa
UnitPrice	float	No	((0))	Đơn giá bán
Quantity	int	No	((1))	Số lượng mua


Customers

Column Name	Condensed Type	Nullable	Default Value	Description
 Id	nvarchar(20)	No		Mã khách hàng
Password	nvarchar(50)	No		Mật khẩu đăng r
Fullname	nvarchar(50)	No		Họ và tên
Email	nvarchar(50)	No		Email
Photo	nvarchar(50)	No	(N'Photo.gif')	Hình ảnh
Activated	bit	No	((0))	Đã kích hoạt chưa?

Orders

Column Name	Condensed Type	Nullable	Default Value	Description
 Id	int	No		Mã hóa đơn
CustomerId	nvarchar(20)	No		Mã khách hàng
OrderDate	datetime	No	(getdate())	Ngày đặt hàng
Address	nvarchar(60)	No		Địa chỉ nhận
Amount	float	No	((0))	Tổng số tiền phải trả
Description	nvarchar(1000)	Yes		Ghi chú hóa đơn

Products

Column Name	Condensed Type	Nullable	Default Value	Description
 Id	int	No		Mã hàng hóa
Name	nvarchar(60)	No		Tên hàng hóa
UnitPrice	float	No	((0))	Đơn giá
Image	nvarchar(50)	No	(N'Product.gif')	Hình ảnh
ProductDate	date	No	(getdate())	Ngày sản xuất
Available	bit	No	((1))	Đang kinh doanh ?
CategoryId	int	No		Mã loại, FK
Description	nvarchar(MAX)	Yes		Mô tả hàng hóa

```
INSERT INTO <tên bảng> (<danh sách cột>) VALUES (<danh sách giá trị>)
```

```
INSERT INTO Categories (Name, NameVN) VALUES (N'Fan', N'Quạt máy')
```

```
INSERT INTO Products (Name, UnitPrice, Image, ProductDate, Available,  
CategoryId, Description) VALUES (N'Nokia 2015', 1000, N'NK15.png', '2015-  
01-01', 1, 1002, N'Điện thoại hiện đại nhất thế giới hiện nay')
```

```
UPDATE <tên bảng> SET <cột1> = <giá trị 1>, ... WHERE <điều kiện>
```

```
UPDATE Categories SET Name = N'Cup', NameVN = N'Tách trà' WHERE Id = 1008
```

```
UPDATE Products SET UnitPrice=UnitPrice*1.2, Available=1 WHERE  
YEAR(ProductDate) = 2015
```

```
DELETE FROM <tên bảng> WHERE <điều kiện>
```

```
DELETE FROM Categories WHERE Id IN (1001, 1005, 1009)
```

```
DELETE FROM Products WHERE Name LIKE '%temp%'
```

```

SELECT <danh sách cột>
FROM <tên bảng>
    JOIN <tên bảng 2> ON <điều kiện kết nối>,
    JOIN <tên bảng 3> ON <điều kiện kết nối>,
WHERE <điều kiện lọc bản ghi>
GROUP BY <biểu thức nhóm>
HAVING <điều kiện nhóm>
ORDER BY <biểu thức sắp xếp>
    
```

```
SELECT * FROM Products
```

Results Messages							
Id	Name	UnitPrice	Image	ProductDate	Available	CategoryId	Description
1019	Teatime Chocolate ...	9.2	1019.jpg	2005-02-02	1	1002	EmEditor us
1020	Sir Rodney's Mama...	81	1020.jpg	2007-11-01	1	1002	EmEditor us
1021	Sir Rodney's Scones	10	1021.jpg	2010-07-29	1	1002	EmEditor us
1022	Gustaf flower	21	1022.jpg	2008-12-01	1	1004	EmEditor us
1023	Tunnbr Korea	9	1023.jpg	2011-08-31	1	1004	EmEditor us
1024	Guaranaj Fantajstica	4.5	1024.jpg	2008-03-13	1	1000	EmEditor us

```
SELECT * FROM Orders WHERE Id=11156
SELECT * FROM OrderDetails WHERE OrderId=11156
```

Id	CustomerId	OrderDate	Address	Amount	Description
11156	nnghiem	2013-04-17 00:00:00.000	205G CC/KCN ...	377.5	Anh yêu em n

Id	OrderId	ProductId	UnitPrice	Quantity
102339	11156	1001	190	1
102340	11156	1003	10	2
102341	11156	1002	19	3
102342	11156	1004	22	4
102343	11156	1024	4.5	5

3

```
SELECT * FROM Categories WHERE Id=1004
SELECT * FROM Products WHERE CategoryId=1004
```

Id	Name	NameVN
1004	Perfumes	Nước hoa

2

```
SELECT Name, UnitPrice, YEAR(ProductDate) [Year]
FROM Products
WHERE UnitPrice BETWEEN 45 AND 50
```

Name	UnitPrice	Year
Russle Sauerkraut	45.6	2011
Ipoh Coffee	46	1980
Tarte au sucre	49.3	2008

1

Id	Name	UnitPrice	Image	ProductDate	Avail...	CategoryId	Description
1022	Gustaf flower	21	1022.jpg	2008-12-01	1	1004	EmEditor uses
1023	T...	9	1023.jpg	2011-08-31	1	1004	EmEditor uses
1042	an H...	14	1042.jpg	1973-11-21	1	1004	EmEditor uses
1052		7	1052.jpg	2001-05-20	1	1004	EmEditor uses
1056	li no...	38	1056.jpg	2007-05-18	1	1004	EmEditor uses
1057	gelo	19.5	1057.jpg	2010-02-16	1	1004	EmEditor uses
1064	gute ...	33.25	1064.jpg	2009-05-15	1	1004	EmEditor uses

❑ Chỉ định các cột cần truy vấn

~~✍~~ `SELECT Id, Name, UnitPrice FROM Products`

❑ Truy vấn tất cả các cột

~~✍~~ `SELECT * FROM Products`

❑ Đặt bí danh cho cột

~~✍~~ `SELECT Name AS 'Tên HH' FROM Products`

❑ Loại bỏ các hàng giống nhau

~~✍~~ `SELECT DISTINCT Year(ProductDate) FROM
Products`

❑ Hạn chế số lượng bản ghi truy vấn

~~✍~~ `SELECT TOP Name, UnitPrice FROM Products`

❑ Sắp xếp tăng (ASC-mặc định)

~~✍~~ SELECT * FROM Products ORDER BY Id ASC

~~✍~~ SELECT * FROM Products ORDER BY UnitPrice

❑ Sắp xếp giảm (DESC)

~~✍~~ SELECT * FROM Products ORDER BY UnitPrice
DESC

❑ Sắp xếp nhiều cột

~~✍~~ SELECT * FROM Products ORDER BY UnitPrice
ASC, Name DESC

- ❑ Lọc dữ liệu kết quả truy vấn theo điều kiện

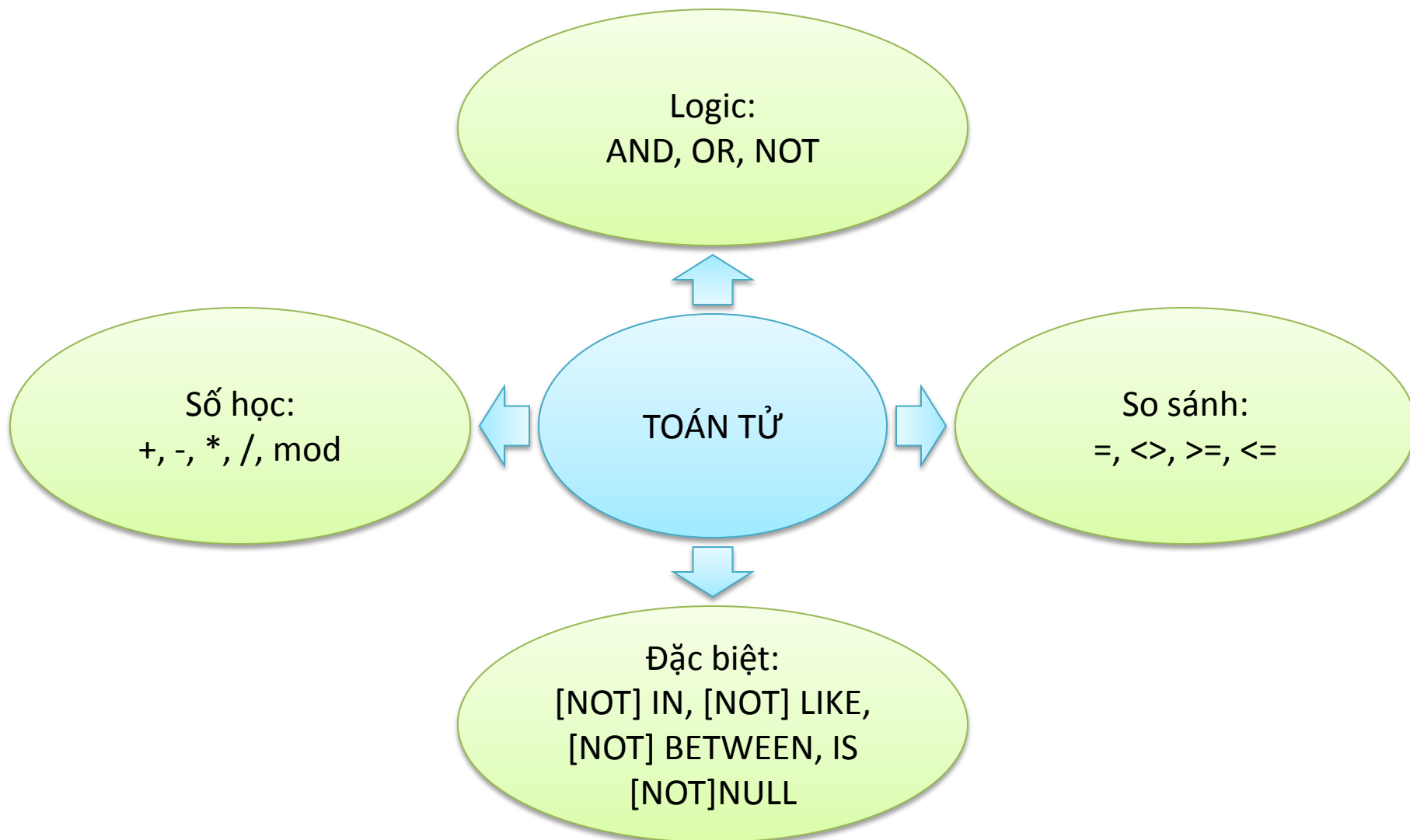
 `SELECT * FROM Products WHERE <điều kiện>`

- ❑ Các tình huống sử dụng mệnh đề WHERE

 `... WHERE Name='ABC'`

 `... WHERE UnitPrice > 70`

 `... WHERE ProductDate <= '01/31/1995'`



❑ Nhóm toán tử số học:

~~✍~~...**WHERE** 4 + UnitPrice <= 300

❑ Nhóm toán tử logic:

~~✍~~...**WHERE NOT** (UnitPrice >= 10 **AND** UnitPrice <= 20)

❑ Nhóm toán tử so sánh:

~~✍~~...**WHERE** (UnitPrice <> 10 **OR** ProductDate <= '12/31/2011')

❑ Nhóm toán tử đặc biệt:

~~✎~~ **LIKE** <chuỗi mẫu>

~~✎~~ **BETWEEN** <min> **AND** <max>

~~✎~~ **IN** (tập hợp)

~~✎~~ **IS NULL**

~~✎~~ **IS NOT NULL**

❑ Ví dụ:

~~✎~~ **SELECT * FROM Products WHERE Name **LIKE****
'%Nokia%'

❑ **[NOT] LIKE** <mẫu tìm kiếm> dùng để tìm kiếm tương tự. Các ký tự đại diện sử dụng cùng với **LIKE** gồm

~~✎~~ **%**: đại diện nhóm ký tự bất kỳ

~~✎~~ **_**: đại diện 1 ký tự bất kỳ

~~✎~~ **[xyz]**: một ký tự thuộc nhóm x,y hoặc z

~~✎~~ **[^xyz]**: một ký tự không thuộc nhóm x, y và z

❑ Các ví dụ khác

✓ ...**WHERE** Name **LIKE** '%ABC%'

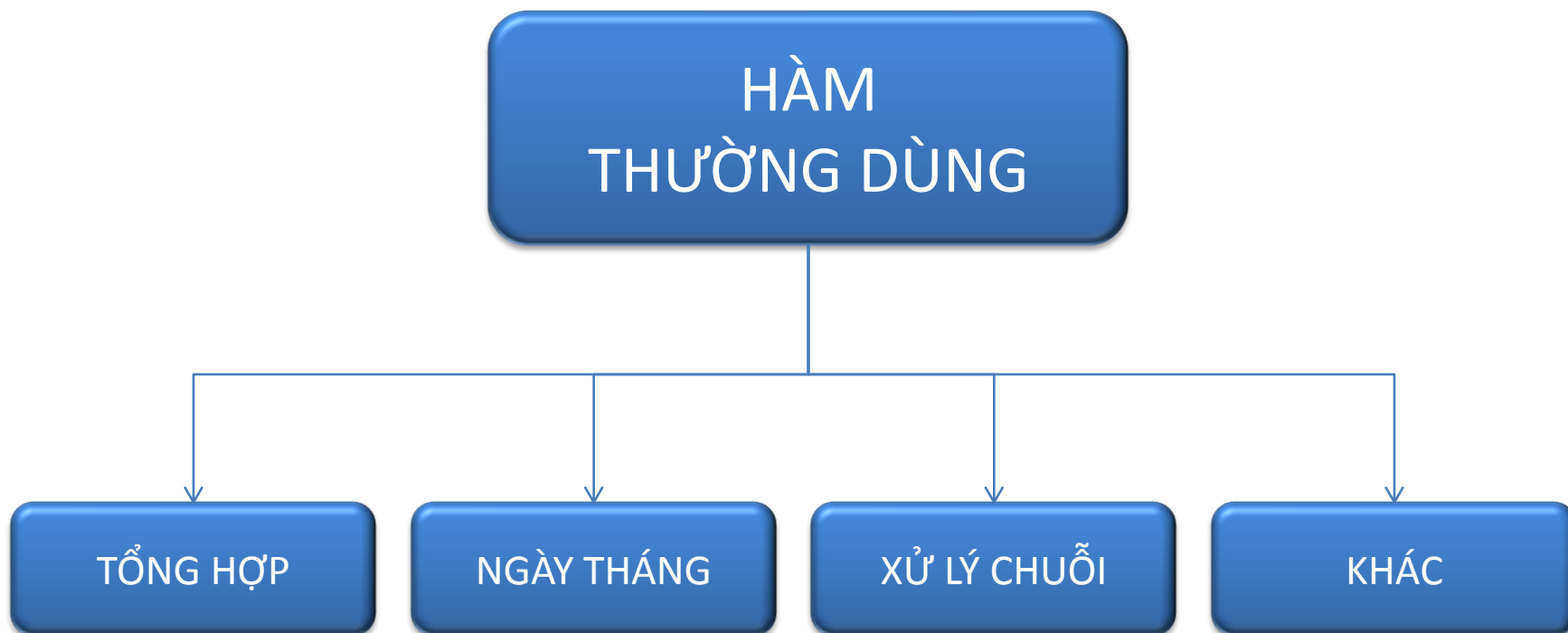
✓ ...**WHERE** Name **LIKE** '%ABC'

✓ ...**WHERE** Name **LIKE** '%[ABC]%'

✓ ...**WHERE** Name **LIKE** '%AB_C%'

- ❑ **[NOT] IN** (tập hợp hoặc truy vấn con) dùng để tìm kiếm trong tập hợp.
 - ✍ ...WHERE CategoryId **IN** (1001, 1007, 1005)
 - ✍ ...WHERE Quantity **NOT IN** (10, 100, 1000)
 - ✍ ...WHERE Category **IN** (SELECT Id FROM Categories)
- ❑ **[NOT] BETWEEN** <ngày/số> **AND** <ngày/số> dùng để tìm theo phạm vi giá trị.
 - ✍ ...WHERE Year(ProductDate) **BETWEEN** 2000 **AND** 2014
 - ✍ ...WHERE UnitPrice **NOT BETWEEN** 20 **AND** 3000
- ❑ **IS [NOT] NULL** dùng để kiểm tra giá trị null
 - ✍ ...WHERE Description **IS NULL**
 - ✍ ...WHERE Description **IS NOT NULL**

- ❑ Các câu lệnh SQL có thể lồng nhau. Câu lệnh SQL bên trong câu lệnh SQL khác được gọi là truy vấn con.
- ❑ Một số trường hợp thường gặp
 - ✂ SELECT * FROM Products WHERE CategoryId IN *(SELECT Id FROM Categories WHERE Name LIKE '%Nokia%')*
 - ✂ SELECT Name, *(SELECT COUNT(*) FROM Products p WHERE p.CategoryId=c.Id)* AS SoHV FROM Category c
 - ✂ SELECT * FROM Products WHERE UnitPrice > *(SELECT AVG(UnitPrice) FROM Products)*



❑ **UPPER** (chuỗi) đổi sang chữ hoa

 **SELECT UPPER** (Name) **FROM** Products

❑ **LOWER** (chuỗi) đổi sang chữ thường

 **SELECT * FROM** Products **WHERE LOWER**(Name)='x'

❑ **TRIM** (chuỗi) cắt bỏ ký tự trắng 2 đầu chuỗi

 **SELECT TRIM** (Name) **FROM** Products

❑ **LEN**('chuỗi 1') độ dài chuỗi

 **SELECT LEN**(Name) **FROM** Products

- ❑ **SUM** (biểu thức) dùng để tính tổng
 - ~~✎~~ **SELECT SUM** (Quantity) **FROM** Products
- ❑ **MIN** (biểu thức) dùng để lấy giá trị nhỏ nhất
 - ~~✎~~ **SELECT MIN** (UnitPrice) **FROM** Products
- ❑ **MAX** (biểu thức) dùng để lấy giá trị lớn nhất
 - ~~✎~~ **SELECT MAX** (ProductDate) **FROM** Products
- ❑ **AVG** (biểu thức) dùng để tính giá trị trung bình
 - ~~✎~~ **SELECT AVG** (UnitPrice) **FROM** Products
- ❑ **COUNT** (cột hoặc *) dùng để đếm số bản ghi
 - ~~✎~~ **SELECT COUNT** (*) **FROM** Products

❑ **GETDATE ()** lấy ngày hiện tại

~~✍~~ **SELECT * FROM** Products **WHERE** ProductDate < **GetDate()**

❑ Trích một phần của ngày-giờ

~~✍~~ **DATEPART** (part, ProductDate) lấy năm-tháng-ngày

❑ Các hàm lấy ngày, tháng, năm, giờ, phút, giây

~~✍~~ **YEAR** (ProductDate) lấy năm

~~✍~~ **MONTH** (ProductDate) lấy tháng

~~✍~~ **HOURL** (ProductDate) lấy giờ

~~✍~~ **MINUTE** (ProductDate) lấy phút

~~✍~~ **SECOND** (ProductDate) lấy giây

~~✍~~ **SELECT SUM** (UnitPrice) **FROM** Products **WHERE** Year
(ProductDate)=2010

- ❑ **ISNULL** (<biểu thức>, <giá trị thay thế>) dùng để chuyển giá trị null sang một giá trị khác.

~~✍~~ **SELECT** Id, **ISNULL** (Description, 'Không') **FROM**
Orders

- ❑ Lệnh **CASE** dùng để tạo giá trị phù hợp với các điều kiện đặt ra.

~~✍~~ **CASE**

✓ **WHEN** UnitPrice > 100 **THEN** 'Giá cao'

✓ **WHEN** UnitPrice < 10 **THEN** 'Giá thấp'

✓ **ELSE** 'Giá bình thường'

~~✍~~ **END**

```
SELECT
    c.NameVN [Loại],
    p.Name [Sản Phẩm]
FROM Products p
    JOIN Categories c ON c.Id=p.CategoryId
ORDER BY c.NameVN
```

Results



Messages

Loại	Sản Phẩm
Điện thoại	Queso Cabrales
Điện thoại	Queso Manchego La Pastora
Điện thoại	Gorgonzola Telino
Điện thoại	Mascarpone Fabioli
Điện thoại	Geitost
Điện thoại	Raclette Courdavault
Điện thoại	Camembert Pierrot
Điện thoại	Gudbrandsdalsost
Điện thoại	Flotemysost
Điện thoại	Mozzarella di Giovanni
Đồng hồ đeo tay	Aniseed Syrup
Đồng hồ đeo tay	Change
Đồng hồ đeo tay	Guaraná Fantástica
Đồng hồ đeo tay	Sasquatch Ale

❑ Yêu cầu:

- ✍ Hiển thị danh sách tên sinh viên và tên lớp.

❑ Giải pháp:

- ✍ Kết nối bảng Lop và bảng SinhVien lại với nhau sau đó lấy lấy Lop.TenLop và SinhVien.Ten

❑ Ví dụ:

- ✍ `SELECT SinhVien.Ten, Lop.TenLop FROM SinhVien
INNER JOIN Lop ON Lop.MaLop=SinhVien.MaLop`

❑ Khái niệm

- ✍ `INNER JOIN` là loại kết nối
- ✍ SinhVien gọi là **bảng trái**
- ✍ Lop gọi là **bảng phải**
- ✍ `Lop.MaLop=SinhVien.MaLop` là cơ sở kết nối

❑ [INNER] JOIN: kết nối trong

- ✎ Lấy các bản ghi của cả 2 bảng (Trái và Phải) có chung giá trị của cột kết nối.

❑ LEFT [OUTER] JOIN: kết nối ngoài, bên trái

- ✎ Lấy tất cả các bản ghi của bảng trái kết.

❑ RIGHT [OUTER] JOIN: kết nối ngoài, bên phải

- ✎ Lấy tất cả các bản ghi của bảng phải.

❑ FULL [OUTER] JOIN: kết nối ngoài, cả 2 bên

- ✎ Kết hợp của LEFT JOIN và RIGHT JOIN

--Các lớp có sinh viên và các sinh viên có lớp

```
SELECT Lop.TenLop, SinhVien.Ten FROM Lop  
JOIN SinhVien ON Lop.MaLop=SinhVien.MaLop
```

--Tất cả các lớp có hoặc không sinh viên của lớp

```
SELECT Lop.TenLop, SinhVien.Ten FROM Lop  
LEFT JOIN SinhVien ON Lop.MaLop=SinhVien.MaLop
```

--Tất cả các lớp có hoặc không sinh viên của lớp

```
SELECT Lop.TenLop, SinhVien.Ten FROM SinhVien  
RIGHT JOIN Lop ON Lop.MaLop=SinhVien.MaLop
```

--Tất cả các lớp, tất cả các sinh viên

```
SELECT Lop.TenLop, SinhVien.Ten FROM SinhVien  
FULL JOIN Lop ON Lop.MaLop=SinhVien.MaLop
```


❑ UNION được sử dụng để kết nối các tập dữ liệu cùng cấu trúc.

--Hợp tập các nhân viên với tập khách hàng

```
SELECT MaNhanVien MA, Ten TEN FROM NhanVien  
UNION
```

```
SELECT MaKhachHang, TenCongTy FROM KhachHang
```

❑ Thống kê là tổng hợp số liệu theo từng nhóm bản ghi.

✎ Nhóm

✎ Thông tin tổng hợp

❑ Ví dụ:

✎ Đếm sinh viên theo lớp.

✎ Điểm thấp nhất, cao nhất, trung bình của mỗi lớp

✎ Doanh số bán hàng của các quý

✎ Doanh số bán hàng của từng nhà cung cấp

- ❑ Sử dụng **GROUP BY** kết hợp với các hàm tổng hợp số liệu (SUM, COUNT, MIN, MAX, AVG) để thống kê theo nhóm

SELECT

MaLH AS "MÃ LỚP",

COUNT (MaSV) AS "SỐ LƯỢNG",

MAX (Diem) AS "ĐIỂM CAO NHẤT",

AVG (Diem) AS "ĐIỂM TRUNG BÌNH",

MAX (NgaySinh) AS "SV TRẺ NHẤT"

FROM SinhVien

GROUP BY MaLH

Tổng hợp số liệu
theo nhóm

Nhóm

❑ Kết hợp với HAVING để lọc nhóm

SELECT

MaLHAS "MÃ LỚP",

COUNT(MaSV) "SỐ LƯỢNG",

MAX(Diem) "ĐIỂM CAO NHẤT",

AVG(Diem) "ĐIỂM TRUNG BÌNH",

MAX(NgaySinh) "SV TRẺ NHẤT"

FROM SinhVien

WHERE MONTH(NgaySinh) **IN** (1, 3, 5)

GROUP BY MaLH

HAVING **AVG**(Diem) > 5.0

Điều kiện này dùng để lọc các bản ghi trước khi phân nhóm

Điều kiện này dùng để lọc nhóm

```
SELECT
    c.NameVN [Loại],
    COUNT(*) [Số SP],
    AVG(p.UnitPrice) [Giá TB],
    MIN(p.UnitPrice) [Giá Thấp Nhất]
FROM Products p
    JOIN Categories c ON c.Id=p.CategoryId
GROUP BY c.NameVN
```

Results Messages

Loại	Số SP	Giá TB	Giá Thấp Nhất
Điện thoại	10	28.73	2.5
Đồng hồ đeo tay	15	51.05	4.5
Giày thời trang	5	32.37	10
Máy ảnh	15	23.962	9.2
Máy tính xách tay	11	23.2181818181818	10
Nữ trang	7	60.1485714285714	7.45
Nước hoa	7	20.25	7
Túi xách du lịch	12	20.6825	6

```
SELECT
    p.Name [Sản Phẩm],
    SUM(d.UnitPrice*d.Quantity) [Doanh Số],
    SUM(d.Quantity) [Số Lượng Bán],
    COUNT(*) [Số Lần Bán],
    MIN(d.UnitPrice) [Giá Thấp Nhất]
FROM OrderDetails d
    JOIN Products p ON p.Id=d.ProductId
GROUP BY p.Name
ORDER BY [Doanh Số] DESC
```

Results Messages

Sản Phẩm	Doanh Số	Số Lượng Bán	Số Lần Bán	Giá Thấp Nhất
Cate de Blaye	150511.2	625	25	210.8
Tharinger Rostbratwurst	87736.4	746	32	99
Raclette Courdavault	76461	1499	56	44
Camembert Pierrot	50286	1577	51	27.2
Tarte au sucre	49827.9	1083	48	39.4
Gnocchi di nonna Alice	45273.2	1267	51	30.4
Manjimup Dried Apples	44742.6	886	39	42.4
Alice Mutton	35755.2	985	39	31.2
Camarvon Tigers	31987.5	539	27	50

```
SELECT
    c.NameVN [Loại Hàng],
    SUM(d.UnitPrice*d.Quantity) [Doanh Số],
    SUM(d.Quantity) [Số Lượng Bán],
    COUNT(*) [Số Lần Bán],
    MIN(d.UnitPrice) [Giá Thấp Nhất]
FROM OrderDetails d
    JOIN Products p ON p.Id=d.ProductId
    JOIN Categories c ON c.Id=p.CategoryId
    JOIN Orders o ON o.Id=d.OrderId
WHERE o.OrderDate BETWEEN '1996-01-01' AND '1996-12-31'
GROUP BY c.NameVN
ORDER BY [Doanh Số] DESC
```

Loại Hàng	Doanh Số	Số Lượng Bán	Số Lần Bán	Giá Thấp
Đồng hồ đeo tay	53879.2	1842	78	3.6
Điện thoại	35538.4	1716	63	2
Máy ảnh	31511.6	1357	59	7.3
Nữ trang	31042.2	955	38	5.9
Túi xách du lịch	21589.6	1286	56	4.8
Máy tính xách tay	17265.3	833	35	8
Giày thời trang	14966.8	540	25	8
Nước hoa	9417.2	512	24	5.6

```
SELECT
    DATEPART(QUARTER, o.OrderDate) [Quý],
    SUM(d.UnitPrice*d.Quantity) [Doanh Số],
    SUM(d.Quantity) [Số Lượng Bán],
    COUNT(*) [Số Lần Bán],
    MIN(d.UnitPrice) [Giá Thấp Nhất]
FROM OrderDetails d
    JOIN Orders o ON o.Id=d.OrderId
WHERE o.OrderDate BETWEEN '1995-01-01' AND '2000-12-31'
GROUP BY DATEPART(QUARTER, o.OrderDate)
ORDER BY [Doanh Số] DESC
```

Quý	Doanh Số	Số Lượng Bán	Số Lần Bán	Giá Thấp Nhất
1	448521.12	16202	659	2
4	322762.92	12323	503	2
2	292770.51	10668	463	2.5
3	241635.29	9613	414	2