



# SQL SERVER

Lương Trần Hy Hiến

# NỘI DUNG

2

- CSDL
  - ▣ Định nghĩa
  - ▣ Tạo, xóa, sao lưu, phục hồi
- Bảng
  - ▣ Ngôn ngữ định nghĩa
    - Tạo, xóa, thay đổi, thiết lập ràng buộc
  - ▣ Ngôn ngữ thao tác
    - Chèn, xóa, sửa
    - Truy vấn
- Các hàm thường dùng

# CÁC KHÁI NIỆM VỀ CƠ SỞ DỮ LIỆU

- Khái niệm thực thể
- Khái niệm mối quan hệ thực thể
- Khái niệm CSDL
- Thao tác dữ liệu

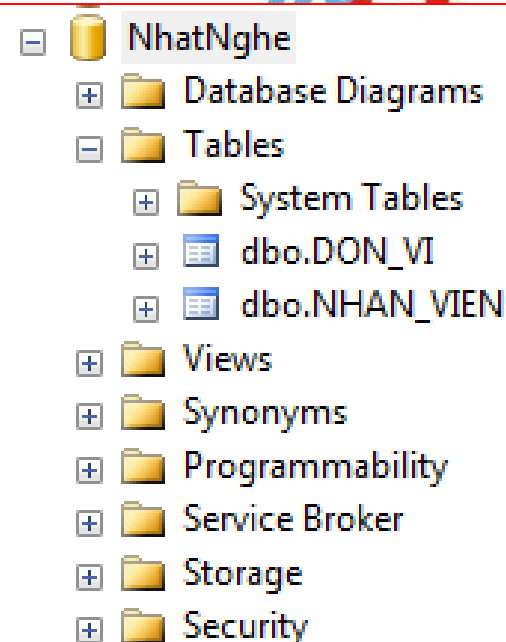
# CSDL LÀ GÌ

## DON\_VI

	MA_DV	TEN_DV
1	11111	Kinh doanh
2	22222	Nhân sự
3	33333	Hành chính
4	44444	Điện toán

## NHAN\_VIEN

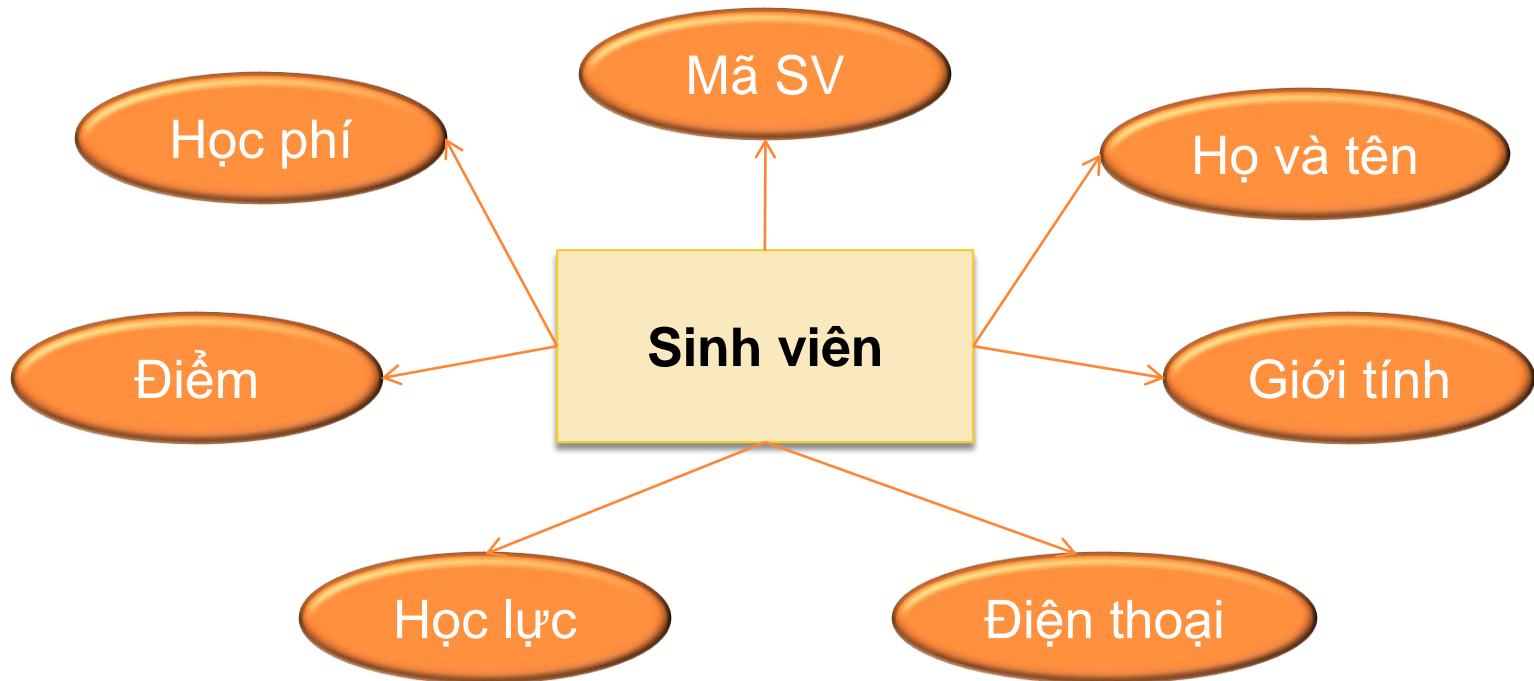
	MA_NV	HO_TEN	GIOI_TINH	NGAY_SINH	CMND	LUONG	MA_DV
1	1	Nguyễn Minh Huy	0	2001-12-31 00:00:00.000	123456789	800000	44444
2	2	Lê Hương Thảo An	0	2002-01-22 00:00:00.000	123456780	900000	11111
3	3	Nguyễn Đình Thúc	0	2000-09-30 00:00:00.000	123456781	2800000	44444
4	4	Nguyễn Thanh Bình	0	2000-09-30 00:00:00.000	123456782	2800000	44444



# THỰC THỂ LÀ GÌ ?

- **Thực thể**: các vật thể, hình thức có thực liên quan đến bài toán hiện tại.
  - Sinh viên, giáo viên, nhân viên...
  - Lớp học, phòng ban, nhà xưởng, kho chứa, thư viện...
  - Điện thoại, máy tính, quần áo, bánh kẹo...
  - Dịch vụ sửa chữa, dịch vụ biểu diễn...
  - Đơn đặt hàng, phiếu bán hàng, phiếu chuyển hàng...
- **Thuộc tính**: được sử dụng để mô tả thực thể
  - Sinh viên: họ tên, giới tính, điểm, chuyên ngành...
  - Sản phẩm: tên, giá, ngày sản xuất...

# BIỂU DIỄN THỰC THỂ ?



MÃ LỚP	TÊN LỚP	NGÀNH
L001	Thiết kế web	Tin học
L002	Thiết kế mẫu	Tin học
L003	Lập trình web	Tin học
L004	Marketing	Kinh tế
L005	Quản trị kinh doanh	Kinh tế

Mỗi hàng chứa  
1 thực thể

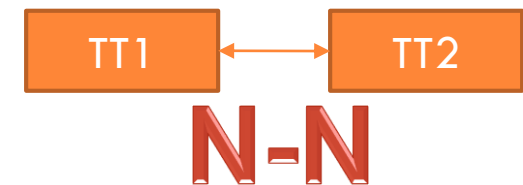
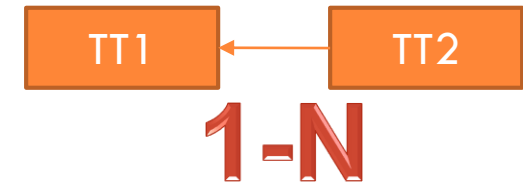
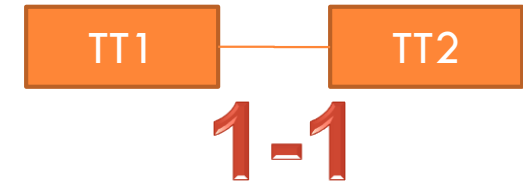
TRƯỜNG ĐẠI HỌC  
CÔNG NGHIỆP  
HÀ NỘI

MÃ SV	HỌ VÀ TÊN	ĐIỂM	LỚP
SV001	Phạm Anh Tuấn	5.5	Thiết kế web
SV002	Nguyễn Thị Anh Thư	7.5	Thiết kế mẫu
SV003	Tôn Nữ Hoàng Trâm	9.5	Thiết kế web
SV004	Trịnh Ngọc Sang	8.0	Marketing
SV005	Hoàng Tuấn Kiệt	6.5	Marketing

TRƯỜNG ĐẠI HỌC  
CÔNG NGHIỆP  
HÀ NỘI

# QUAN HỆ GIỮA CÁC THỰC THỂ

- Sinh viên & Lớp học
- Nhân viên & Phòng ban
- Sinh viên & Môn học
- Sinh viên & Giáo viên
- Tài xế & Xe
- Khách hàng & Chính sách giảm giá
- Tour & Du khách
- Sản phẩm & Danh mục
- Đơn hàng & Khách hàng
- Đơn hàng & Mặt hàng



**Mũi tên phía nào thì khóa ở phía đó**



MÃ LỚP	TÊN LỚP	NGÀNH
L001	Thiết kế web	Tin học
L002	Thiết kế mẫu	Tin học
L003	Lập trình web	Tin học
L004	Marketing	Kinh tế
L005	Quản trị kinh doanh	Kinh tế

MÃ SV	HỌ VÀ TÊN	ĐIỂM	LỚP
SV001	Phạm Anh Tuấn	5.5	Thiết kế web
SV002	Nguyễn Thị Anh Thư	7.5	Thiết kế mẫu
SV003	Tôn Nữ Hoàng Trâm	9.5	Thiết kế web
SV004	Trịnh Ngọc Sang	8.0	Marketing
SV005	Hoàng Tuấn Kiệt	6.5	Anh ngữ



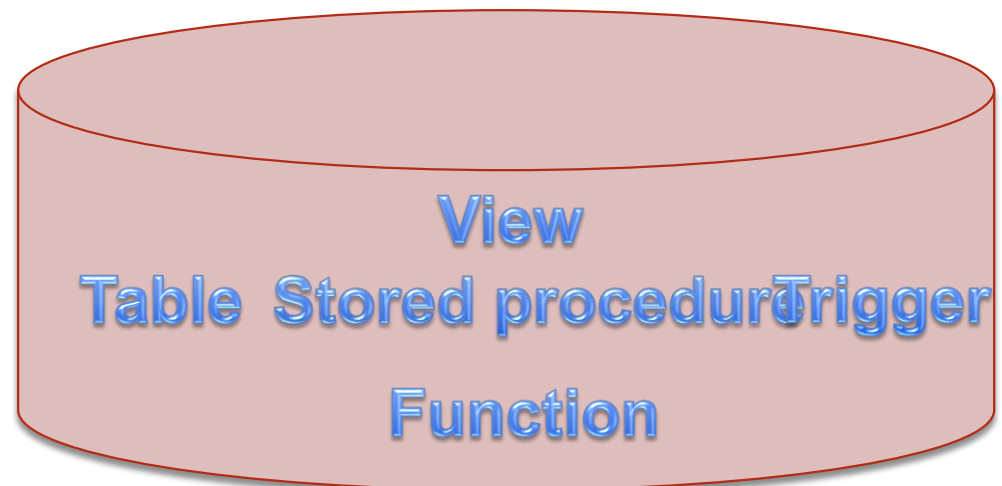
# CSDL LÀ GÌ ?

10

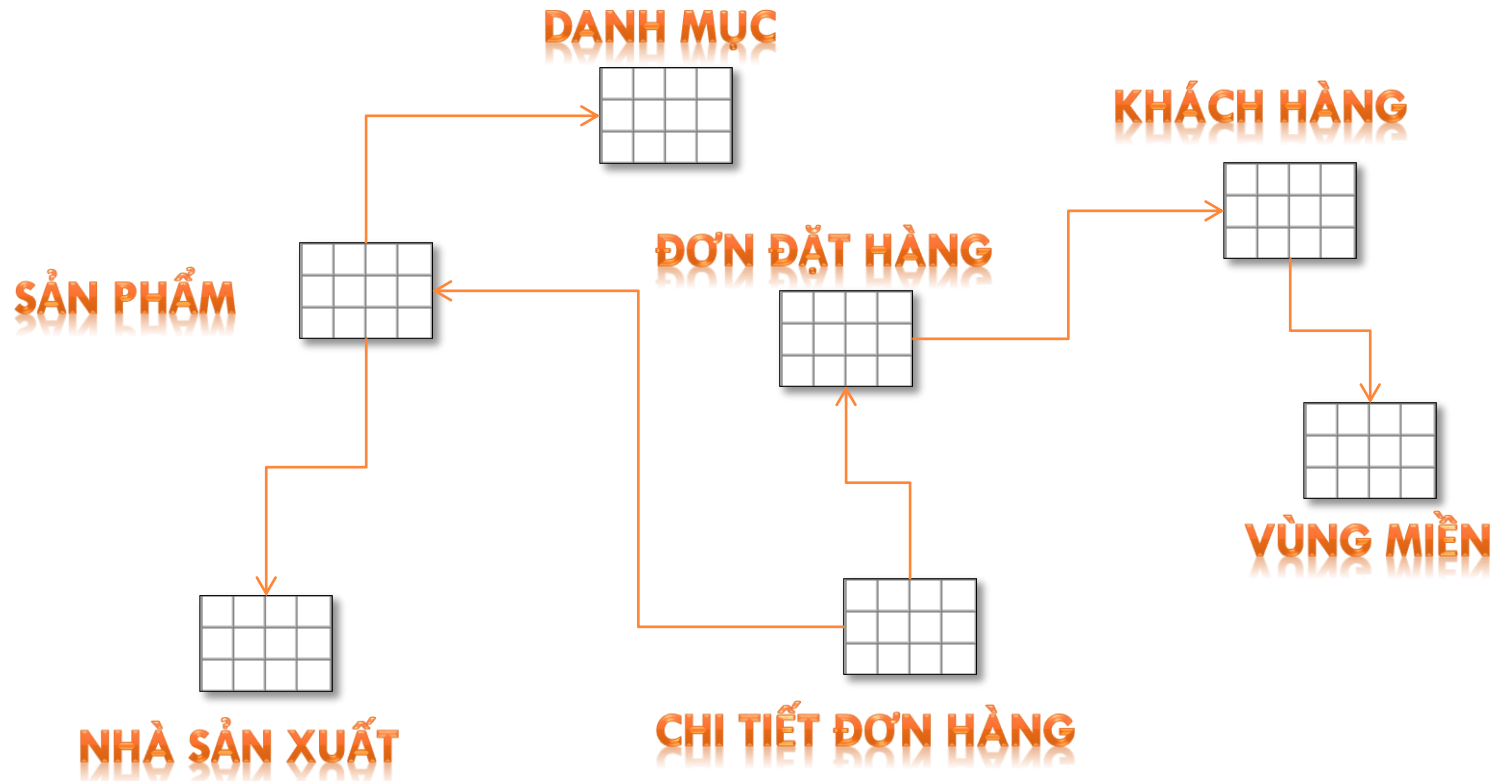
- Gồm nhiều bảng và mối quan hệ giữa chúng
- Mỗi bảng chứa nhiều thực thể cùng loại (*thể hiện*)
- Mỗi thực thể có nhiều thuộc tính
- Mỗi thuộc tính chịu nhiều ràng buộc
  - **Khóa chính** (có giá trị duy nhất)
  - **Khóa ngoại** (có giá trị của cột khóa ở bảng khác)
  - **Null** (cho phép để trống)
  - **Miền giá trị** (giá trị phải thuộc vào)
- Mỗi bảng có thể có quan hệ với nhiều bảng khác
- Có thể có nhiều cột tham gia vào khóa chính

# CSDL là gì

- Là cơ sở lưu trữ dữ liệu và chức năng thao tác, truy vấn dữ liệu mà 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 như
  - ▣ SQL Server
  - ▣ MySQL Server
  - ▣ Oracle
  - ▣ Sysbase

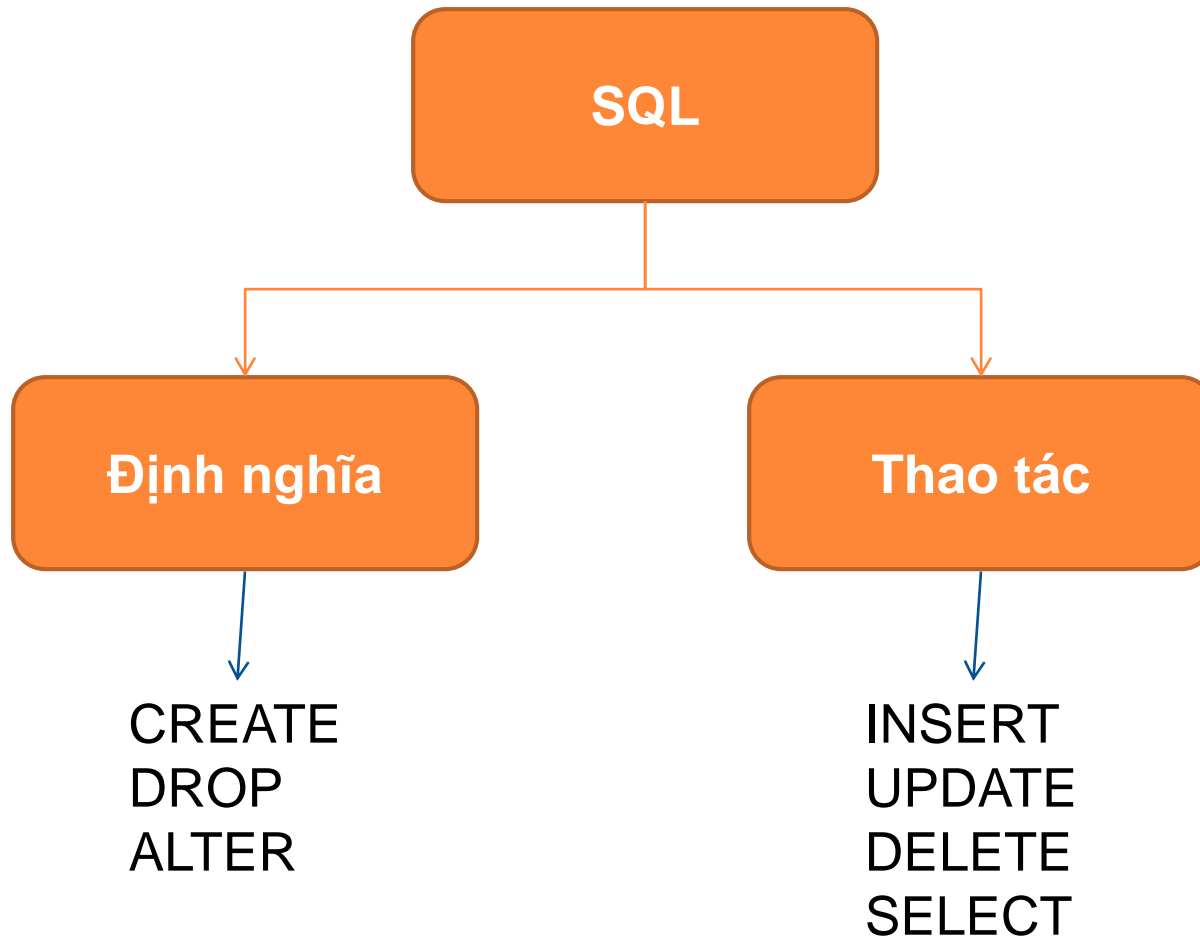


# CSDL LÀ GÌ ?



# SQL là gì?

- **SQL** (Structure Query Language) là một ngôn ngữ được sử dụng để giao tiếp với hệ quản trị cơ sở dữ liệu quan hệ.
- SQL được chia làm 2 nhóm chính là ĐỊNH NGHĨA (DDL=Data Definition Language) và THAO TÁC (DML=Data Manipulation Language) dữ liệu
- **DDL** (Ngôn ngữ định nghĩa dữ liệu) giúp
  - ▣ Tạo, xóa, sao lưu, phục hồi CSDL
  - ▣ Tạo, xóa, sửa bảng, view, stored procedure, function, trigger
- **DML** (Ngôn ngữ thao tác dữ liệu) giúp
  - ▣ Chèn thêm, xóa bớt, cập nhật, truy vấn, thống kê số liệu
  - ▣ Sử dụng view, stored procedure, function, transaction...



# Ngôn ngữ định nghĩa (DDL)

- Đối với CSDL
  - ▣ **CREATE**: Tạo
  - ▣ **DROP**: Xóa
  - ▣ **BACKUP**: Sao lưu
  - ▣ **RESTORE**: Phục hồi
- Đối với đối tượng trong CSDL (Table, View, Stored Procedure, Trigger, Function)
  - ▣ **CREATE**: Tạo
  - ▣ **DROP**: Xóa
  - ▣ **ALTER**: Chỉnh sửa

# CSDL-TẠO CSDL

16

- Tạo, xóa
  - ▣ **CREATE DATABASE** NhatNghe;
  - ▣ **USE** NhatNghe;
  - ▣ **DROP DATABASE** NhatNghe;
- Sao lưu và phục hồi (dùng công cụ)
  - ▣ Online (Backup/Restore)
  - ▣ Offline (Detach/Attach)



# BẢNG

17

- Định nghĩa
- Các khái niệm
  - ▣ Cột
  - ▣ Kiểu dữ liệu
  - ▣ Ràng buộc
- Các thao tác bảng
  - ▣ Tạo (CREATE TABLE)
  - ▣ Xóa (DROP TABLE)
  - ▣ Chỉnh sửa (ALTER TABLE)

# Bảng và khóa chính

- 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**

MAMONHOC	TENMONHOC	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

# NGÔN NGỮ ĐỊNH NGHĨA

- Tạo bảng
  - ▣ **CREATE TABLE**
- Xóa bảng
  - ▣ **DROP TABLE**
- Sửa đổi cấu trúc
  - ▣ **ALTER TABLE**
    - ALTER COLUMN
    - ADD COLUMN
    - DROP COLUMN



# LỆNH TẠO BẢNG

20

```
CREATE TABLE DON_VI  
(  
    MA_DV NCHAR(5) NOT NULL,  
    TEN_DV NVARCHAR(50) NOT NULL,  
    CONSTRAINT PK_DV PRIMARY KEY (MA_DV)  
)
```

Tên bảng

Tên cột, kiểu dữ liệu, ràng buộc

Khóa chính

# TẠO BẢNG

MaSV	HoTen	NgaySinh	GioiTinh	DiemTB

CREATE TABLE SinhVien

(

MaSV NVARCHAR(20) NOT NULL,

HoTen NVARCHAR(50) NOT NULL,

NgaySinh DATETIME NULL,

GioiTinh INT DEFAULT 1,

DiemTB FLOAT NOT NULL,

CONSTRAINT SV\_PK PRIMARY KEY(MaSV)

);

# Kiểu dữ liệu

- Kiểu chuỗi ([N]-unicode)
  - ▣ **[N]CHAR, [N]VARCHAR, [N]TEXT**
- Số thực
  - ▣ **FLOAT, DOUBLE, MONEY**
- Ngày, giờ
  - ▣ **DATE, TIME, DATETIME, TIMESTAMP**
- Số nguyên
  - ▣ **BIT, SMALLINT, INT, BIGINT**
- Nhị phân
  - ▣ **BINARY, VARBINARY, IMAGE**

# Kiểu dữ liệu


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

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)



## SinhVien

	Column Name	Data Type	Allow Nulls
	MaSV	varchar(50)	<input type="checkbox"/>
	HoTen	varchar(50)	<input type="checkbox"/>
	Diem	float	<input type="checkbox"/>
	NgaySinh	datetime	<input checked="" type="checkbox"/>
	GioiTinh	int	<input checked="" type="checkbox"/>
	MaLH	varchar(50)	<input type="checkbox"/>
	GhiChi	varchar(255)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


Cho phép  
để trống

Quan hệ  
khóa ngoại

Cột khóa Ngoại

Khóa  
chính

## LopHoc

	Column Name	Data Type	Allow Nulls
	MaLH	varchar(50)	<input type="checkbox"/>
	TenLop	varchar(50)	<input checked="" type="checkbox"/>
	NgayKG	datetime	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

MaLH	TenLop	NgayKG
LTW01	LAP TRINH WEB VOI PHP	1/1/2009 12:00:00 AM
LTW02	LAP TRINH WEB VOI ASP	1/1/2010 12:00:00 AM
TKW03	THIET KE WEB CHUYEN NGHIEP	1/1/2011 12:00:00 AM

MaSV	HoTen	Diem	NgaySinh	GioiTinh	MaLH	GhiChu
SV001	Nguyen Nghiem	9	1/1/2000 12:00:00 AM	1	LTW01	NULL
SV002	Le Thi Huong Thao	8	2/2/2000 12:00:00 AM	0	TKW03	NULL
SV003	Nguyen Dinh Thien Long	10	3/3/2000 12:00:00 AM	1	LTW01	NULL
SV004	Nguyen Dinh Hoang Long	10	4/4/2000 12:00:00 AM	1	LTW02	NULL
SV005	Nguyen Thi Minh Khai	7	5/5/2000 12:00:00 AM	0	TKW03	NULL
SV006	Ly Thai To	8	6/6/2000 12:00:00 AM	1	LTW02	NULL

# Thiết lập Quan hệ

- ON DELETE
- ON UPDATE
  - ▣ Cascade
  - ▣ Set Null
  - ▣ Set Default
  - ▣ No Action

Categories					
	Column Name	Condensed Type	Nullable	Identity	Description
🔑	Id	char(3)	No	<input type="checkbox"/>	Mã chủng loại
	Name	nvarchar(50)	No	<input type="checkbox"/>	Tên chủng loại
				<input type="checkbox"/>	



Products					
	Column Name	Condensed Type	Nullable	Identity	Description
🔑	Id	int	No	<input checked="" type="checkbox"/>	Mã sản phẩm
	Name	nvarchar(50)	No	<input type="checkbox"/>	Tên sản phẩm
	UnitPrice	float	No	<input type="checkbox"/>	Đơn giá
	ProductDate	datetime	No	<input type="checkbox"/>	Ngày sản xuất
	Image	nvarchar(50)	No	<input type="checkbox"/>	Hình ảnh
	CategoryId	char(3)	No	<input type="checkbox"/>	Mã chủng loại
	Available	bit	No	<input type="checkbox"/>	Còn hay hết hàng
				<input type="checkbox"/>	

# RÀNG BUỘC

- **PRIMARY KEY** : Khóa chính
  - **CONSTRAINT** <tên> **PRIMARY KEY** (<cột>)
- **FOREIGN KEY** : Khóa ngoại
  - **CONSTRAINT** <tên> **FOREIGN KEY** (<cột>)  
**REFERENCES** <bảng> (<cột>) [**ON DELETE CASCADE**] [**ON UPDATE CASCADE**]
- **CHECK**: Kiểm tra
  - **CONSTRAINT** <tên> **CHECK** (<điều kiện>)
- **NOT NULL** : Không cho để trống
- **DEFAULT** : Mặc định
- **UNIQUE**: Duy nhất

# ON DELETE|UPDATE [X]

- **X=CASCADE**
  - ▣ Xóa (cập nhật) tất cả các record ở bảng con có khóa ngoại bị xóa (cập nhật) ở bảng cha
- **X=SET NULL**
  - ▣ Thiết lập giá trị null cho khóa ngoại của các record bị xóa (cập nhật) ở bản cha
- **X=SET DEFAULT**
  - ▣ Thiết lập giá trị mặc định cho khóa ngoại của các record bị xóa (cập nhật) ở bản cha
- **X=NO ACTION**
  - ▣ Không cho xóa (cập nhật) các record trong bảng cha có liên quan đến record trong bảng con

# LỆNH TẠO BẢNG



```
CREATE TABLE NHAN_VIEN↓
```

```
(↓
```

```
MA_NV INT IDENTITY(1, 1) NOT NULL, ↓
```

```
HO_TEN NVARCHAR(50) NOT NULL, ↓
```

```
GIOI_TINH BIT DEFAULT 0, ↓
```

```
NGAY_SINH DATETIME NULL, ↓
```

```
CMND FLOAT NOT NULL, ↓
```

```
LUONG INT NULL, ↓
```

```
MA_DV NCHAR(5) NULL, ↓
```

```
CONSTRAINT PK_NV PRIMARY KEY (MA_NV), ↓
```

```
CONSTRAINT UN_CMND UNIQUE (CMND), ↓
```

```
CONSTRAINT CK_LUONG CHECK (LUONG > 0), ↓
```

```
CONSTRAINT FK_DV FOREIGN KEY (MA_DV) | ↓
```

```
REFERENCES DON_VI (MA_DV) ↓
```

```
ON DELETE CASCADE ON UPDATE CASCADE↓
```

```
) ↓
```

Xem kiểu dữ liệu  
và ràng buộc ở  
giáo trình

An orange starburst-shaped callout box is positioned to the right of the SQL code, pointing towards the 'CMND' and 'LUONG' fields. It contains the text 'Xem kiểu dữ liệu và ràng buộc ở giáo trình' (Check data types and constraints in the textbook).

# XÓA, CHỈNH SỬA BẢNG

31

```
--Xóa bảng nhân viên↓  
DROP TABLE NHAN_VIEN↓  
↓  
--Xóa bảng DON_VI↓  
DROP TABLE DON_VI↓  
↓  
--Thêm cột PHOTO vào bảng NHAN_VIEN↓  
ALTER TABLE NHAN_VIEN↓  
    ADD PHOTO NVARCHAR(50) ; ↓  
↓  
--Chỉnh sửa kiểu và ràng buộc cho cột PHOTO↓  
ALTER TABLE NHAN_VIEN↓  
    ALTER COLUMN PHOTO NVARCHAR(100) NOT NULL; ↓  
↓  
--Xóa cột PHOTO khỏi bảng NHAN_VIEN ↓  
ALTER TABLE NHAN_VIEN↓  
    DROP COLUMN PHOTO; ↓
```

# XÓA, CHỈNH SỬA BẢNG (2)

--Thay đổi giá trị mặc định cho cột GIOI\_TINH↓

ALTER TABLE NHAN\_VIEN ↓

ADD DEFAULT 0 FOR GIOI\_TINH;↓

↓

--Thêm ràng buộc khóa chính ↓

ALTER TABLE DON\_VI↓

ADD CONSTRAINT FK\_DV PRIMARY KEY(MA\_DV);↓

↓

--Thêm ràng buộc khóa ngoại↓

ALTER TABLE NHAN\_VIEN↓

ADD CONSTRAINT FK\_DV FOREIGN KEY(MA\_DV) REFERENCES DON\_VI (MA\_DV);↓  
ON UPDATE CASCADE↓  
ON DELETE SET NULL↓

↓

--Xóa ràng buộc↓

ALTER TABLE NHAN\_VIEN↓

DROP CONSTRAINT FK\_DV;↓



# NGÔN NGỮ THAO TÁC DỮ LIỆU

# Thao tác dữ liệu

- Chèn dữ liệu (INSERT INTO)

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

- Cập nhật dữ liệu (UPDATE)

**UPDATE** <tên bảng> **SET** <tên cột 1>=<giá trị 1>,  
<tên cột 2>=<giá trị 2>... **WHERE** <điều kiện>

- Xóa dữ liệu (DELETE)

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

# THAO TÁC DỮ LIỆU

35

- INSERT INTO: Chèn dữ liệu

```
INSERT INTO SanPham(MaSP, TenSP, DonGia, NgaySX)  
VALUES (1, '8610', 100, '12/31/2000');
```

- UPDATE: Cập nhật dữ liệu

```
UPDATE SanPham SET DonGia=123, NgaySX='12/31/2009'  
WHERE MaSP > 100;
```

- DELETE: Xóa dữ liệu

```
DELETE FROM SanPham WHERE MaSP=123;
```

# THAO TÁC DỮ LIỆU ĐƠN GIẢN

## NHẤT NGHỆ

```
INSERT INTO DON_VI (MA_DV, TEN_DV) VALUES ('4444', N'Điện toán');↓
```

↓

```
INSERT INTO NHAN_VIEN (HO_TEN, NGAY_SINH, LUONG, CMND, MA_DV) ↓  
VALUES (N'Nguyễn Nghiệm', '12/31/2001', 800000, 123456789, 4444);↓
```

↓

```
UPDATE NHAN_VIEN SET HO_TEN=UPPER (HO_TEN) ↓
```

```
UPDATE NHAN_VIEN SET LUONG=LUONG*1.5 WHERE YEAR (NGAY_SINH) < 1970 ↓
```

↓

```
DELETE FROM NHAN_VIEN ↓
```

```
DELETE NHAN_VIEN WHERE LEN (CMND) < 9 ↓
```

↓

```
SELECT * FROM NHAN_VIEN ↓
```

```
SELECT HO_TEN, ISNULL (LUONG, 'THAO THUAN') AS LUONG FROM NHAN_VIEN ↓
```

```
SELECT ↓
```

```
HO_TEN AS 'HỌ VÀ TÊN', ↓
```

```
CASE ↓
```

```
WHEN YEAR (GETDATE ()) - YEAR (NGAY_SINH) < 30 THEN 'Thanh niên' ↓
```

```
WHEN YEAR (GETDATE ()) - YEAR (NGAY_SINH) < 50 THEN 'Trung niên' ↓
```

```
ELSE 'Cao niên' ↓
```

```
END AS 'ĐỘ TUỔI' ↓
```

```
FROM NHAN_VIEN ←
```

# TRUY VẤN DỮ LIỆU

- Liệt kê các cột **C1, C2, C3**
  - ▣ **SELECT C1, C2, C3 FROM T**
- Đặt bí danh cho cột **C2** là **NAME**
  - ▣ **SELECT C1, C2 AS 'NAME', C3 FROM T**
- Sử dụng **DISTINCT** để loại bỏ các hàng trùng dữ liệu
  - ▣ **SELECT DISTINCT C1, C2, C3 FROM T**
- Sử dụng **TOP** để hạn chế số lượng bản ghi
  - ▣ **SELECT TOP (10) C1, C2, C3 FROM T**
- Sử dụng dấu sao (\*) để liệt kê tất cả các cột
  - ▣ **SELECT \* FROM T**
- Sắp xếp
  - ▣ **SELECT \* FROM T ORDER BY C2 DESC**
  - ▣ **SELECT \* FROM T ORDER BY C2 ASC**

# LỌC DỮ LIỆU

38

- Sử dụng WHERE và điều kiện để chỉ ra các bản ghi cần thực hiện cho các câu lệnh UPDATE, DELETE và SELECT
  - SELECT \* FROM T **WHERE <điều kiện>**
  - DELETE FROM T **WHERE <điều kiện>**
  - UPDATE T SET C2=3, C3='12/31/2010' **WHERE <điều kiện>**
- Sau đây là một số ví dụ về biểu thức lọc WHERE
  - Các bản ghi có giá trị của C1 là 'ABC'
    - ... WHERE **C1='ABC'**
  - Các bản ghi có giá trị của C2 lớn hơn 70
    - ... WHERE **C2 > 70**
  - Các bản ghi có giá trị của C2 là 'X' và C3 trước 31-03-2010
    - ... WHERE **C2='X' AND C3 <='01/31/1995'**

# TOÁN TỬ

39

- Nhóm toán tử **số học**
  - +, -, \*, /, %
  - ...WHERE **4+C2<=300**
- Nhóm toán tử **lôgic**
  - AND, OR, NOT
  - ...WHERE **NOT (C2>=10 AND C2<=20)**
- Nhóm toán tử **so sánh**
  - >, <, =, >=, <=, <>, !=
  - ...WHERE **(C2<>10 OR C3<='1995-01-01')**
- Nhóm toán tử **đặc biệt**
  - **LIKE, BETWEEN, IN, IS NULL, IS NOT NULL**

# TRUY VẤN NÂNG CAO

40

- Toán tử đặc biệt
  - ▣ **LIKE** và ký tự đại diện (% , \_ , [], [^])
  - ▣ **IN** (tập hợp hoặc truy vấn con)
  - ▣ **BETWEEN** <ngày/số> **AND** <ngày/số>
- Ví dụ
  - ▣ WHERE C **LIKE** 'Ph[ạa][nm] %T\_n%'
  - ▣ WHERE C **IN** (234, 789, 2, 5)
  - ▣ WHERE C **IN** (SELECT TOP 10 C FROM T2)
  - ▣ WHERE C **BETWEEN** 20 AND 3000



# TOÁN TỬ LIKE

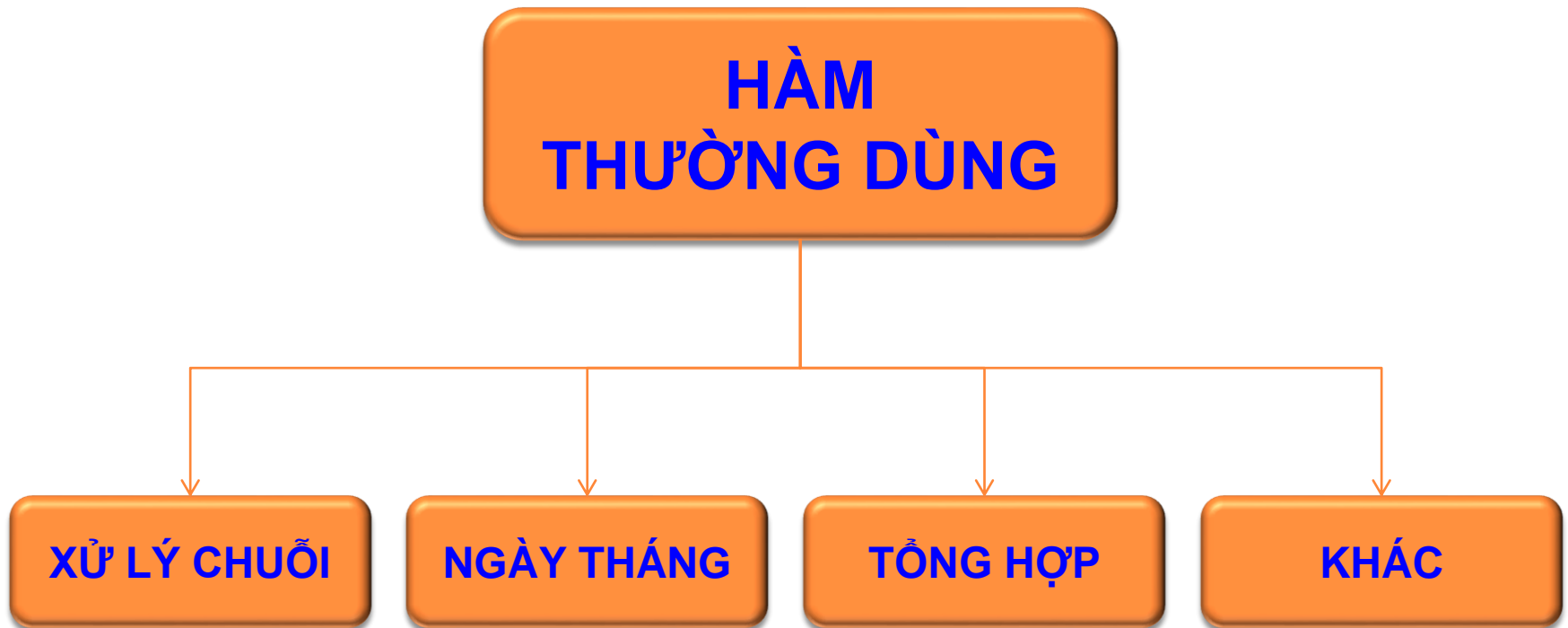
41

- Cột C1 có **chứa** chuỗi '**ABC**' ?
  - ▣ ...WHERE C1 **LIKE** '%ABC%'
- Cột C1 có **kết thúc** bởi chuỗi '**ABC**' ?
  - ▣ ...WHERE C1 **LIKE** '%ABC'
- Cột C1 có **chứa** một trong các ký tự '**A**', '**B**' hay '**C**' ?
  - ▣ ...WHERE C1 **LIKE** '%[ABC]%'
- Cột C1 có **chứa** chuỗi '**ABxC**', với **x** là ký tự bất kỳ ?
  - ▣ ...WHERE C1 **LIKE** '%AB\_C%'

# TOÁN TỬ ĐẶC BIỆT

42

- Kiểm tra phạm vi số hoặc ngày
  - ...WHERE C3 **BETWEEN** '12/31/2000' **AND** '12/31/2010'
- Kiểm tra tập hợp bất kỳ (số, ngày, chuỗi,...)
  - ...WHERE C1 **IN** ('SV001', 'SV009', 'SV075')
  - ...WHERE C2 **NOT IN** (100, 55, 65, 18, 22, 54)
  - ...WHERE C3 **IN** ('12/31/2000', '12/31/2005', '12/31/2010')
- Kiểm tra giá trị null
  - ...WHERE C3 **IS NULL**
- Kiểm tra giá trị không null
  - ...WHERE C1 **IS NOT NULL**



# HÀM XỬ LÝ CHUỖI

44

- **UPPER**(chuỗi) đổi sang chữ hoa
  - SELECT **UPPER(C1)** FROM T
- **LOWER**(chuỗi) đổi sang chữ thường
  - SELECT C3 FROM T WHERE **LOWER(C1)**='xyz'
- **TRIM**(chuỗi) cắt bỏ ký tự trắng 2 đầu chuỗi
  - SELECT **TRIM(C1)** FROM T
- **CONCAT**('chuỗi 1', 'chuỗi 2') ghép 2 chuỗi
  - SELECT **CONCAT('SV:', C1)** AS MaSV FROM T

# HÀM XỬ LÝ NGÀY

45

- **GETDATE()** lấy ngày hiện tại
  - *SELECT \* FROM T WHERE C3 < **GetDate()***
- Trích một phần của ngày-giờ
  - **DATEPART**(part, NgayGio) lấy năm-tháng-ngày
- Các hàm lấy ngày, tháng, năm, giờ, phút, giây
  - **YEAR**(ngày) lấy năm
  - **MONTH**(ngày) lấy tháng
  - **HOURL**(ngày) lấy giờ
  - **MINUTE**(ngày) lấy phút
  - **SECOND**(ngày) lấy giây
  - *SELECT SUM(C2) FROM T WHERE **Year(C3)**=2011*

# HÀM TỔNG HỢP

46

- **SUM**(biểu thức) dùng để tính tổng
  - SELECT **SUM(C2)** AS 'Tong' FROM T
- **MIN**(biểu thức) dùng để lấy giá trị nhỏ nhất
  - SELECT **MIN(C2)** AS 'NhoNhat' FROM T
- **MAX**(biểu thức) dùng để lấy giá trị lớn nhất
  - SELECT **MAX(C2)** AS 'LonNhat' FROM T
- **AVG**(biểu thức) dùng để tính giá trị trung bình
  - SELECT **AVG(C2)** AS 'TrungBinh' FROM T
- **COUNT**(cột hoặc \*) dùng để đếm số bản ghi
  - SELECT **COUNT(\*)** AS 'SoLuong' FROM T

# VÍ DỤ TỔNG HỢP

SELECT

NgaySinh AS 'NGÀY SINH',

Diem AS 'ĐIỂM',

CASE WHEN Diem < 5 THEN 'Rớt' ELSE 'Đậu' END AS 'KẾT QUẢ'

FROM SinhVien;

SELECT TOP (30)

CONCAT(MaSV, ' (', MaLH, ') ') AS 'MÃ SV',

UPPER(HoTen) AS 'HỌ VÀ TÊN',

Diem AS 'ĐIỂM',

CASE WHEN Diem < 5 THEN 'Yếu' WHEN Diem < 7.5 THEN 'Trung Bình'  
ELSE 'Giỏi' END AS 'XẾP LOẠI',

NgaySinh AS 'NGÀY SINH'

FROM SinhVien

ORDER BY Diem DESC

# THỐNG KÊ – GROUP BY

48

- Sử dụng GROUP BY và các hàm tổng hợp để thống kê số liệu (tổng, số lượng, trung bình, nhỏ nhất, lớn nhất) 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
```

Cột tự do  
(không sử  
dụng hàm tổng  
hợp) phải là  
cột nhóm (phải  
ở trong  
GROUP BY)



# GROUP BY...HAVING

```
SELECT
    MaLH AS "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
```

**HAVING <ĐK>**  
là điều kiện nhóm  
trong khi đó  
**WHERE <ĐK>** là  
điều kiện chung  
(toàn bộ)

Chỉ hiện các nhóm thỏa điều kiện này

# THỐNG KÊ – GROUP BY

50

```
SELECT YEAR(NGAY_SINH) NAM, COUNT(MA_NV) AS SOLUONG  
FROM NHAN_VIEN  
WHERE HO_TEN LIKE N'Nguyễn %'  
GROUP BY YEAR(NGAY_SINH)  
HAVING COUNT(MA_NV) > 3
```

SUM(),  
COUNT(),  
MIN(), MAX(),  
AVG()

# CÁC HÀM VÀ LỆNH KHÁC

51

- **ISNULL**(<CỘT>, <GIÁ TRỊ KHI CỘT NULL>)
  - ▣ SELECT C1, **ISNULL**(C2, 3) AS CC2 FROM T
  
- **CASE**
  - ▣ **WHEN** <ĐK> **THEN** <GIA TRỊ>
  - ▣ **WHEN** <ĐK> **THEN** <GIA TRỊ>
  - ▣ **ELSE** <GIA TRỊ>
- **END**

# TRUY VẤN CON

52

- **SELECT \* FROM SinhVien WHERE MaLH IN (SELECT MaLH FROM LopHoc WHERE Year(NgayKG)=2000)**
- **SELECT \*, (SELECT COUNT(\*) FROM SinhVien sv WHERE lh.MaLH=sv.MaLH) AS SoHV FROM LopHoc lh**
- **SELECT \* FROM SinhVien WHERE DiemTB > (SELECT AVG(DiemTB) FROM SinhVien)**
- **SELECT \*, (SELECT COUNT(MaSV) FROM SinhVien sv WHERE sv.MaLH=lh.MaLH) AS SoLuong FROM LopHoc lh**

# KẾT NỐI 2 BẢNG

```
SELECT T.*, P.* FROM Trai T XJOIN Phai P ON T.CX=P.CY
```

## [INNER] JOIN

Lấy các bản ghi của cả 2 bảng (**Trai** và **Phai**) có chung giá trị của cột kết nối (**CX** và **CY**)

## LEFT [OUTER] JOIN

Lấy tất cả các bản ghi của bảng **Trai** nối với các bản ghi của bảng Phai. NULL được điền vào các bản ghi của bảng Phai nếu không có giá trị để nối.

## RIGHT [OUTER] JOIN

Lấy tất cả các bản ghi của bảng Phai nối với các bản ghi của bảng Trai. NULL được điền vào các bản ghi của bảng Trai nếu không có giá trị để nối.

# Các loại 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 nối với các bản ghi của bảng Phải. Nếu không có bản ghi nào của bảng Phải thỏa mãn cơ sở kết nối thì giá trị NULL sẽ được điền vào các bản ghi tương ứng của tập kết quả.
- **RIGHT [OUTER] JOIN**: kết nối ngoài, bên phải
  - ▣ Lấy tất cả các bản ghi của bảng Phải kết nối với các bản ghi của bảng Trái. Nếu không có bản ghi nào của bảng Trái thỏa mãn cơ sở kết nối thì giá trị NULL sẽ được điền vào các bản ghi tương ứng của tập kết quả.
- **FULL [OUTER] JOIN**: kết nối ngoài, cả 2 bên
  - ▣ Kết hợp của LEFT JOIN và RIGHT JOIN

## Ghe

GheSo	BanSo
1	1
2	1
3	1
4	2
5	2
6	NULL
7	NULL
8	3

Cột dùng để  
kết nối

## Ban

BanSo	Ten
1	Ban an
2	Ban uong tra
3	Ban tho
4	Ban trang diem

Bàn không ghế

Ghế không bàn

SELECT \* FROM Ban

SELECT \* FROM Ghe

# INNER JOIN

Ban		Ghe	
BanSo	Ten	GheSo	BanSo
1	Ban an	1	1
1	Ban an	2	1
1	Ban an	3	1
2	Ban uong tra	4	2
2	Ban uong tra	5	2
3	Ban tho	8	3

Cột kết nối của cả 2 bảng có chung giá trị

```
SELECT b.*, g.* FROM Ban b
INNER JOIN Ghe g ON b.BanSo=g.BanSo
```



# LEFT OUTER JOIN

Tất cả các bản (tất cả các  
bản ghi của bảng bên trái)

BanSo	Ten	GheSo	BanSo
1	Ban an	1	1
1	Ban an	2	1
1	Ban an	3	1
2	Ban uong tra	4	2
2	Ban uong tra	5	2
3	Ban tho	8	3
4	Ban trang diem	NULL	NULL

Bản số 4 không có ghế

```
SELECT b.*, g.* FROM Ban b
LEFT OUTER JOIN Ghe g ON b.BanSo=g.BanSo
```

# RIGHT OUTER JOIN

Ghế số 6 và 7 không có bàn

BanSo	Ten	GheSo	BanSo
1	Ban an	1	1
1	Ban an	2	1
1	Ban an	3	1
2	Ban uong tra	4	2
2	Ban uong tra	5	2
NULL	NULL	6	NULL
NULL	NULL	7	NULL
3	Ban tho	8	3

Tất cả các ghế (tất cả các bàn ghi của bàn bên phải)

```
SELECT b.*, g.* FROM Ban b
RIGHT OUTER JOIN Ghe g ON b.BanSo=g.BanSo
```

# LAB GUIDE

59

- ❑ SQL.LAB1: TẠO, SAO LƯU, PHỤC HỒI, XÓA CSDL
- ❑ SQL.LAB2: QUẢN LÝ BẢNG
- ❑ SQL.LAB3: THIẾT LẬP MỐI QUAN HỆ
- ❑ SQL.LAB4: THAO TÁC DỮ LIỆU BẰNG LỆNH
- ❑ SQL.LAB5: LUYỆN TẬP
- ❑ SQL.LAB6: VIEW, SP, FUNCTION, TRIGGER