

# SQL

Biên soạn: TS. Nguyễn Quốc Tuấn  
Bm. Mạng và Các HTTT

# Nội dung

- Tổng quan về SQL
- Giới thiệu về SQL Server
- T-SQL

# SQL

- SQL – Structured Query Language
- Chức năng của SQL
  - Định nghĩa dữ liệu
  - Truy xuất và thao tác với dữ liệu
  - Điều khiển truy cập
  - Đảm bảo toàn vẹn dữ liệu
- Vai trò của SQL
  - Là ngôn ngữ hỏi có tính tương tác
  - Là ngôn ngữ lập trình CSDL
  - Là ngôn ngữ quản trị CSDL
  - Là ngôn ngữ cho hệ thống Client/Server
  - Là ngôn ngữ truy cập dữ liệu trên Web

# Câu lệnh SQL

- SQL chuẩn bao gồm khoảng 40 câu lệnh
- Cú pháp chi tiết một số câu lệnh có thể thay đổi tùy vào HQTCSDL
- Ví dụ:
  - Định nghĩa dữ liệu
    - CREATE TABLE
    - DROP TABLE
    - ALTER TABLE
  - Thao tác dữ liệu:
    - SELECT
    - INSERT
    - UPDATE
    - DELETE

# Câu lệnh SQL

- Điều khiển truy cập
  - GRANT
  - REVOKE
  - DENY
- Lập trình
  - DECLARE
  - OPEN
  - FETCH
  - EXECUTE
  - CLOSE
  - DELETE

# Kiểu dữ liệu trong SQL

Tên kiểu	Mô tả
CHAR ( <i>n</i> )	Kiểu chuỗi với độ dài cố định
NCHAR ( <i>n</i> )	Kiểu chuỗi với độ dài cố định hỗ trợ UNICODE
VARCHAR ( <i>n</i> )	Kiểu chuỗi với độ dài chính xác
NVARCHAR ( <i>n</i> )	Kiểu chuỗi với độ dài chính xác hỗ trợ UNICODE
INTEGER	Số nguyên có giá trị từ $-2^{31}$ đến $2^{31} - 1$
INT	Như kiểu Integer
TINYINT	Số nguyên có giá trị từ 0 đến 255.
SMALLINT	Số nguyên có giá trị từ $-2^{15}$ đến $2^{15} - 1$
BIGINT	Số nguyên có giá trị từ $-2^{63}$ đến $2^{63} - 1$
NUMERIC ( <i>p,s</i> )	Kiểu số với độ chính xác cố định.
DECIMAL ( <i>p,s</i> )	Tương tự kiểu Numeric

# Kiểu dữ liệu trong SQL

Tên kiểu	Mô tả
FLOAT	Số thực có giá trị từ -1.79E+308 đến 1.79E+308
REAL	Số thực có giá trị từ -3.40E + 38 đến 3.40E + 38
MONEY	Kiểu tiền tệ
BIT	Kiểu bit (có giá trị 0 hoặc 1)
DATETIME	Kiểu ngày giờ (chính xác đến phần trăm của giây)
SMALLDATETIME	Kiểu ngày giờ (chính xác đến phút)
TIMESTAMP	
BINARY	Dữ liệu nhị phân với độ dài cố định (tối đa 8000 bytes)
VARBINARY	Dữ liệu nhị phân với độ dài chính xác (tối đa 8000 bytes)
IMAGE	Dữ liệu nhị phân với độ dài chính xác (tối đa 2,147,483,647 bytes)
TEXT	Dữ liệu kiểu chuỗi với độ dài lớn (tối đa 2,147,483,647 ký tự)
NTEXT	Dữ liệu kiểu chuỗi với độ dài lớn và hỗ trợ UNICODE (tối đa 1,073,741,823 ký tự)

# Nội dung

- Tổng quan về SQL
- Giới thiệu về SQL Server
- T-SQL

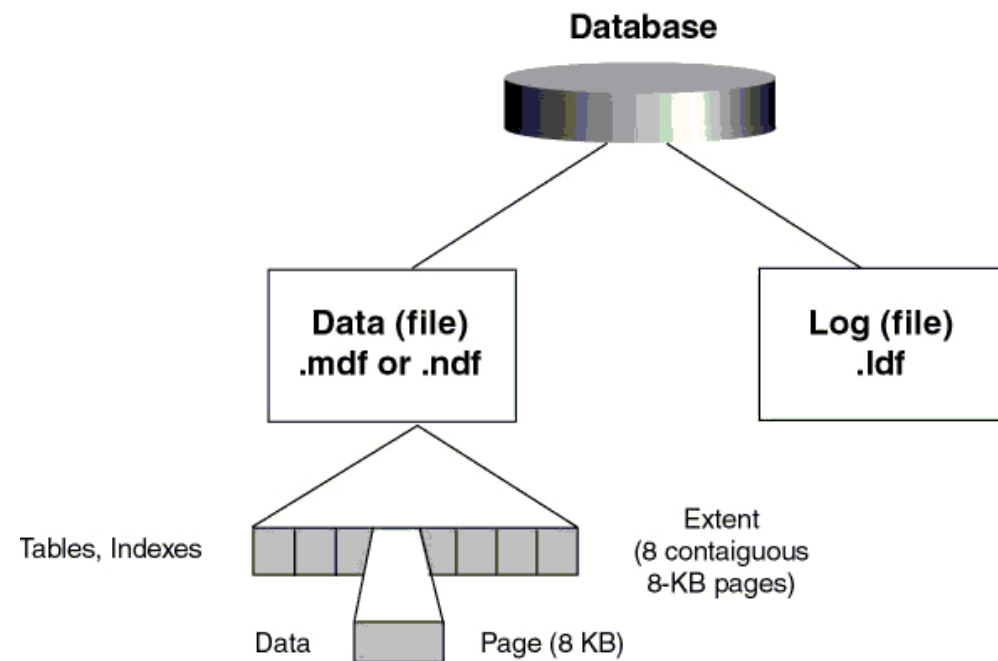


# Giới thiệu SQL Server

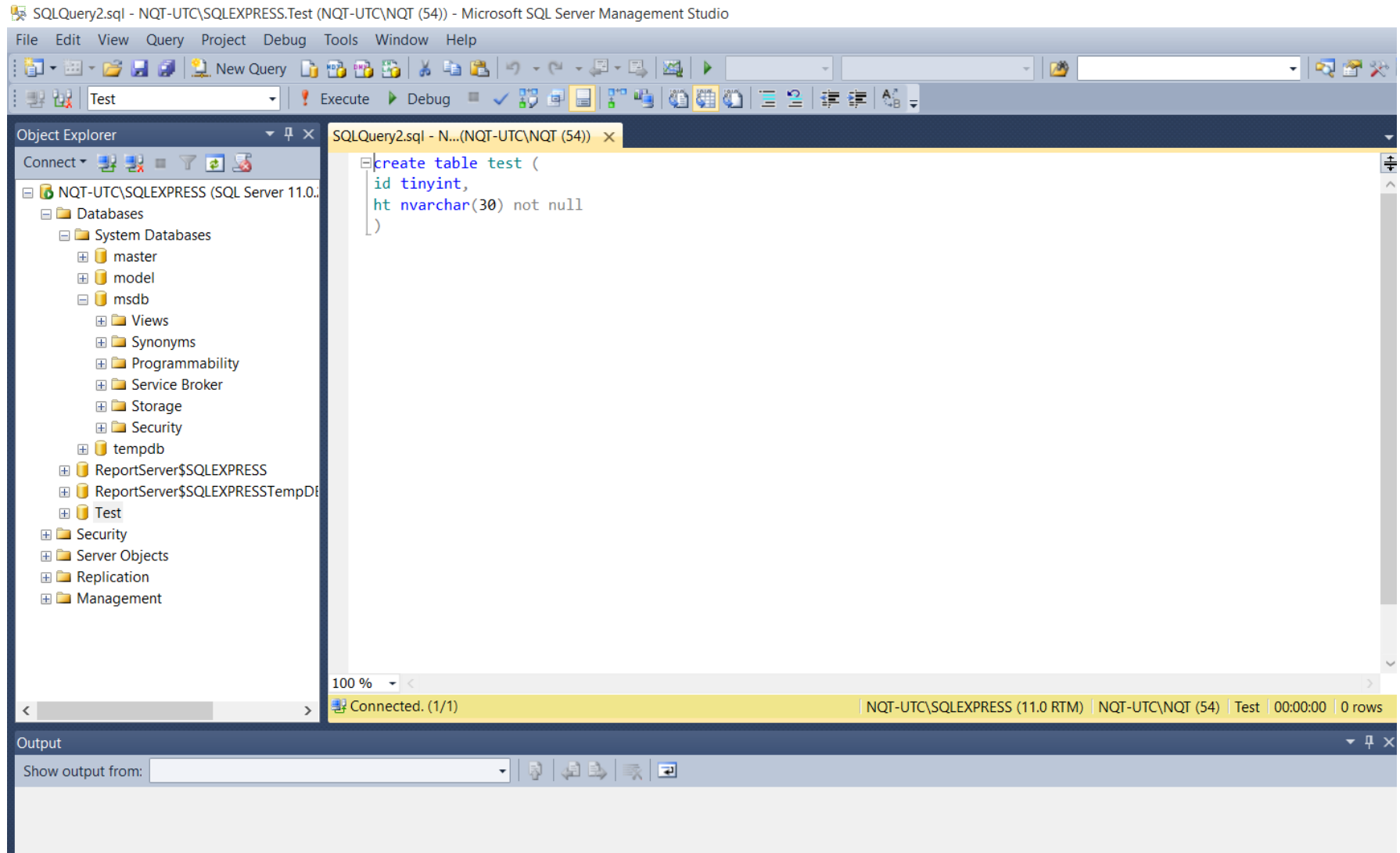
- SQL Server 7.0, 2000, 2003, 2005, 2007, 2008, 2012, 2014, 2016, 2017, 2019
- SQL Server 2014:
  - Là một HQT CSDL quan hệ
  - Sử dụng Transact-SQL để trao đổi dữ liệu Client/Server
  - Các phiên bản:
    - Express
    - Standard
    - Business Intelligence
    - Enterprise

# Giới thiệu SQL Server

- Cấu trúc của SQL Server
  - Multi-Instance: Default Instance
  - Mỗi Instance gồm 4 system databases và các users database
    - Master, Tempdb, Model, Msdb
- Cấu trúc vật lý của 1 SQL Server
  - Primary data file (.mdf)
  - Secondary data file (.ndf)
  - Transaction log file (.ldf)



# Giới thiệu SQL Server



# Nội dung

- Tổng quan về SQL
- Giới thiệu về SQL Server
- T-SQL

# T-SQL

- Giới thiệu T-SQL
- Ngôn ngữ định nghĩa dữ liệu (DDL)
- Ngôn ngữ thao tác dữ liệu

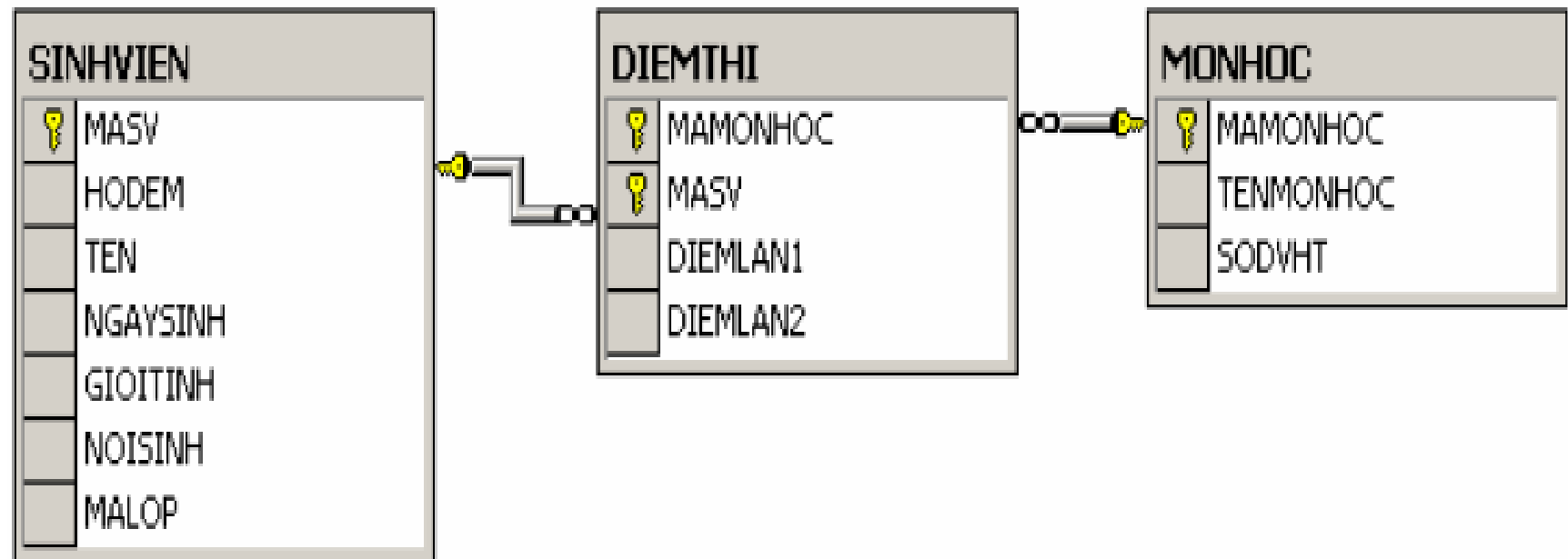
# T-SQL

- Transact-SQL
- Là ngôn ngữ mở rộng của SQL chuẩn của ISO và ANSI
- Khác với P-SQL(Procedural - SQL) dùng trong Oracle

# Ngôn ngữ định nghĩa dữ liệu

- DDL – Data Definition Language
- Gồm các lệnh định nghĩa các đối tượng trong CSDL
  - CREATE object\_Name
  - ALTER object\_Name
  - DROP object\_Name

*object\_Name: table, view, stored procedure, indexes*





# Tạo, xóa CSDL

- *Tạo Cơ sở dữ liệu*

- *Cú pháp*

- ```
CREATE DATABASE <Tên cơ sở dữ liệu> [;]
```

- *Ví dụ*

- ```
CREATE DATABASE QuanlyDiem
```

- *Sử dụng Cơ sở dữ liệu*

- *Cú pháp*

- ```
USE <Tên cơ sở dữ liệu>
```

- *Ví dụ*

- ```
USE QuanlyDiem
```

- *Xóa Cơ sở dữ liệu*

- *Cú pháp*

- ```
DROP DATABASE <Tên cơ sở dữ liệu>
```

- *Ví dụ*

- ```
DROP DATABASE QuanlyDiem
```

# Tạo bảng - CREATE TABLE

- *Cú pháp*

```
CREATE TABLE tên_bảng
(
    tên_cột    thuộc_tính_cột    các_ràng_buộc
    [...
    ,tên_cột_n thuộc_tính_cột_n    các_ràng_buộc_cột_n]
    [,các_ràng_buộc_trên_bảng]
)
```

- *Ví dụ*

```
CREATE TABLE nhanvien
(
    manv        NVARCHAR(10)    NOT NULL,
    hoten        NVARCHAR(50)    NOT NULL,
    ngaysinh     DATETIME        NULL,
    dienthoai    NVARCHAR(10)    NULL,
    hsluong      DECIMAL(3,2)    DEFAULT (1.92)
)
```

# CREATE TABLE

## ***Ràng buộc PRIMARY KEY***

- *Cú pháp*

```
[CONSTRAINT tên_ràng_buộc]  
PRIMARY KEY [ (danh_sách_cột) ]
```

- *Ví dụ*

```
CREATE TABLE sinhvien  
(  
    masv          NVARCHAR(10)  
        CONSTRAINT pk_sinhvien_masv PRIMARY KEY,  
    hodem         NVARCHAR(25)    NOT NULL ,  
    ten           NVARCHAR(10)    NOT NULL ,  
    ngaysinh      DATETIME,  
    gioitinh      BIT,  
    noisinh       NVARCHAR(255),  
    malop         NVARCHAR(10)  
)
```

```
CREATE TABLE diemthi  
(  
    mamonhoc      NVARCHAR(10)    NOT NULL ,  
    masv          NVARCHAR(10)    NOT NULL ,  
    diemlan1      NUMERIC(4, 2),  
    diemlan2      NUMERIC(4, 2),  
    CONSTRAINT pk_diemthi PRIMARY KEY(mamonhoc,masv)  
)
```

# CREATE TABLE

## ***Ràng buộc FOREIGN KEY***

- *Cú pháp*

```
[CONSTRAINT tên_ràng_buộc]
FOREIGN KEY [(danh_sách_cột)]
REFERENCES tên_bảng_tham_chiếu(danh_sách_cột_tham_chiếu)
[ON DELETE CASCADE | NO ACTION | SET NULL | SET DEFAULT]
[ON UPDATE CASCADE | NO ACTION | SET NULL | SET DEFAULT]
```

- *Ví dụ: Tạo bảng Điểm thi*

# CREATE TABLE

## ***Ràng buộc FOREIGN KEY***

```
CREATE TABLE diemthi
(
    mamonhoc NVARCHAR(10) NOT NULL ,
    masv      NVARCHAR(10) NOT NULL ,
    diemlan1  NUMERIC(4, 2),
    diemlan2  NUMERIC(4, 2),
    CONSTRAINT pk_diemthi PRIMARY KEY(mamonhoc,masv),
    CONSTRAINT fk_diemthi_mamonhoc
        FOREIGN KEY(mamonhoc)
        REFERENCES monhoc(mamonhoc)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT fk_diemthi_masv
        FOREIGN KEY(masv)
        REFERENCES sinhvien(masv)
        ON DELETE CASCADE
        ON UPDATE CASCADE
)
```

# CREATE TABLE

## ***Ràng buộc UNIQUE***

- *Cú pháp*

```
[CONSTRAINT tên_ràng_buộc]  
UNIQUE [(danh_sách_cột)]
```

- *Ví dụ*

```
CREATE TABLE lop  
(  
    malop          NVARCHAR(10)          NOT NULL,  
    tenlop          NVARCHAR(30)          NOT NULL,  
    khoa            SMALLINT              NULL,  
    hedaotao        NVARCHAR(25)          NULL,  
    namnhaphoc      INT                   NULL,  
    makhoa          NVARCHAR(5),  
    CONSTRAINT pk_lop PRIMARY KEY (malop),  
    CONSTRAINT unique_lop_tenlop UNIQUE(tenlop)  
)
```

# CREATE TABLE

## ***Ràng buộc DEFAULT***

- *Cú pháp*

```
[CONSTRAINT tên_ràng buộc]  
DEFAULT(giá trị) FOR Tên_trường
```

- *Ví dụ*

```
ALTER TABLE chitietdathang  
ADD  
CONSTRAINT df_chitietdathang_soluong  
DEFAULT(1) FOR soluong,
```

# CREATE TABLE

## ***Ràng buộc CHECK***

- *Cú pháp*

[CONSTRAINT tên\_ràng\_buộc]  
CHECK (điều kiện)

- *Ví dụ*

```
ALTER TABLE nhanvien  
ADD  
    CONSTRAINT chk_nhanvien_ngaylamviec  
    CHECK (datediff(yy,ngaysinh,ngaylamviec)  
           BETWEEN 18 AND 60)
```



# Sửa nội dung bảng- ALTER TABLE

- *Bổ sung một cột vào bảng*
- *Xóa một cột khỏi bảng*
- *Thay đổi định nghĩa của một cột trong bảng*
- *Xóa hoặc bổ sung ràng buộc cho bảng*

# ALTER TABLE

```
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
```

# ALTER TABLE

- Ví dụ: Cho định nghĩa 2 bảng NHANVIEN và DONVI như sau

```
CREATE TABLE nhanvien
(
manv          NVARCHAR(10)      NOT NULL,
hoten         NVARCHAR(30)      NOT NULL,
ngaysinh      DATETIME,
diachi        CHAR(30)          NOT NULL
)
```

```
CREATE TABLE donvi
(
madv          INT               NOT NULL   PRIMARY KEY,
tendv         NVARCHAR(30)      NOT NULL
)
```

# ALTER TABLE

- Bài tập
  - Bổ sung thêm cột MADV cho bảng NHANVIEN
  - Định nghĩa lại kiểu dữ liệu của cột DIACHI trong bảng NHANVIEN cho phép nhận giá trị NULL
  - Xóa cột ngày sinh khỏi bảng NHANVIEN
  - Định nghĩa khóa chính cho bảng NHANVIEN là trường MANV
  - Định nghĩa khóa ngoại của bảng NHANVIEN trên cột MADV của bảng DONVI

# ALTER TABLE

```
ALTER TABLE nhanvien
ADD
    dienthoai NVARCHAR(6)
    CONSTRAINT chk_nhanvien_dienthoai
    CHECK (dienthoai LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]')
```

```
ALTER TABLE nhanvien
ADD
    madv INT NULL
```

```
ALTER TABLE nhanvien
ALTER COLUMN diachi NVARCHAR(100) NULL
```

# ALTER TABLE

```
ALTER TABLE nhanvien
```

```
ADD
```

```
    CONSTRAINT pk_nhanvien PRIMARY KEY (manv)
```

```
ALTER TABLE nhanvien
```

```
ADD
```

```
    CONSTRAINT fk_nhanvien_madv
```

```
    FOREIGN KEY (madv) REFERENCES donvi (madv)
```

```
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE
```

# Xóa bảng - DROP TABLE

- **Cú pháp**

```
DROP TABLE tên_bảng
```

- **Ghi chú:**

- Câu lệnh này sẽ xóa các ràng buộc, chỉ mục, trigger liên quan đến bảng cần xóa
- Khi xóa bằng lệnh DROP không thể khôi phục lại được
- Không thể thực hiện được nếu vẫn còn ràng buộc về khóa ngoại

- Ví dụ: Xóa bảng DONVI

```
ALTER TABLE nhanvien
```

```
DROP CONSTRAINT fk_nhanvien_madv
```

```
DROP TABLE donvi
```

# T-SQL

- Giới thiệu T-SQL
- Ngôn ngữ định nghĩa dữ liệu (DDL)
- Ngôn ngữ thao tác dữ liệu



# Ngôn ngữ thao tác dữ liệu

## DML – Data Manipulation Language

- SELECT
- INSERT
- UPDATE
- DELETE

# SELECT

- Chức năng
  - Truy xuất dữ liệu từ các dòng và các cột của một hoặc nhiều bảng
  - Phép chiếu
  - Phép nối
- Cú pháp

```
SELECT [ALL | DISTINCT][TOP n] danh_sách_chọn  
[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]]
```

# SELECT

- Ví dụ

```
SELECT manv, hoten, ngaysinh  
FROM nhanvien
```

# SELECT

- Truy vấn thường
- Truy vấn có điều kiện
- Tạo mới bảng dữ liệu từ kết quả của SELECT
- Sắp xếp kết quả truy vấn
- Phép nối
- Thống kê dữ liệu với GROUP BY
- Truy vấn con

# SELECT – Truy vấn thường

- Chọn tất cả các trường: \*
- Chọn một số cột

***Tenbang.Tentruong***

- Thay đổi tiêu đề các cột

tiêu\_đề\_cột = tên\_trường

tên\_trường AS tiêu\_đề\_cột

tên\_trường    tiêu\_đề\_cột

- Ví dụ:

```
SELECT 'Mã lớp'= malop,tenlop 'Tên lớp',khoa AS 'Khoá'
FROM lop
```

# SELECT

- Truy vấn thông thường
- **Truy vấn có điều kiện**
- Tạo mới bảng dữ liệu từ kết quả của SELECT
- Sắp xếp kết quả truy vấn
- Phép nối
- Thống kê dữ liệu với GROUP BY
- Truy vấn con

# SELECT – Truy vấn có điều kiện

- Sử dụng mệnh đề WHERE
- Sau mệnh đề WHERE là một biểu thức logic để lọc các kết quả thỏa mãn
- Các toán tử so sánh
- Toán tử BETWEEN
- Danh sách (IN và NOT IN)
- Toán tử LIKE

# Các toán tử so sánh

Toán tử	ý nghĩa
$=$	Bằng
$>$	Lớn hơn
$<$	Nhỏ hơn
$>=$	Lớn hơn hoặc bằng
$<=$	Nhỏ hơn hoặc bằng
$\neq$	Khác
$!>$	Không lớn hơn
$!<$	Không nhỏ hơn



# BETWEEN

- Kiểm tra dữ liệu trong phạm vi nào đó

Cách sử dụng	Ý nghĩa
giá_trị BETWEEN a AND b	$a \leq \text{giá\_trị} \leq b$
giá_trị NOT BETWEEN a AND b	$(\text{giá\_trị} < a) \text{ AND } (\text{giá\_trị} > b)$

- Ví dụ

```
SELECT hodem,ten,year(getdate())-year(ngaysinh) AS tuoi
FROM sinhvien
WHERE ten='Bình' AND
      YEAR(GETDATE())-YEAR(ngaysinh) BETWEEN 20 AND 22
```

# Từ khóa LIKE

Ký tự đại diện	ý nghĩa
%	Chuỗi ký tự bất kỳ gồm không hoặc nhiều ký tự
_	Ký tự đơn bất kỳ
[]	Ký tự đơn bất kỳ trong giới hạn được chỉ định (ví dụ [a-f]) hay một tập (ví dụ [abcdef])
[^]	Ký tự đơn bất kỳ không nằm trong giới hạn được chỉ định ( ví dụ [^a-f] hay một tập (ví dụ [^abcdef])).

```
SELECT hodem,ten FROM sinhvien  
WHERE hodem LIKE 'Lê%'
```

# SELECT

- Truy vấn thông thường
- Truy vấn có điều kiện
- Tạo mới bảng dữ liệu từ kết quả của SELECT
- Sắp xếp kết quả truy vấn
- Phép nối
- Thống kê dữ liệu với GROUP BY
- Truy vấn con

# Tạo bảng mới từ kết quả của SELECT

- Câu lệnh SELECT ... INTO
  - Tạo bảng mới có cấu trúc và dữ liệu được xác định từ kết quả của truy vấn
  - Bảng mới có số cột bằng số cột chỉ định và số dòng là số dòng kết quả truy vấn
- Ví dụ

```
SELECT hodem,ten, YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi  
INTO tuoisv  
FROM sinhvien
```

# SELECT

- Truy vấn thông thường
- Truy vấn có điều kiện
- Tạo mới bảng dữ liệu từ kết quả của SELECT
- Sắp xếp kết quả truy vấn
- Phép nối
- Thống kê dữ liệu với GROUP BY
- Truy vấn con

# Sắp xếp kết quả truy vấn

- Sử dụng ORDER BY
- Tối đa 16 cột

```
SELECT * FROM monhoc  
ORDER BY sodvht DESC
```

```
SELECT hodem,ten,gioitinh,  
       YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi  
FROM sinhvien  
WHERE ten='Bình'  
ORDER BY gioitinh,tuoi
```

# SELECT

- Truy vấn thông thường
- Truy vấn có điều kiện
- Tạo mới bảng dữ liệu từ kết quả của SELECT
- Sắp xếp kết quả truy vấn
- **Phép nối**
- Thống kê dữ liệu với GROUP BY
- Truy vấn con

# Phép nối

- Phép nối bằng (equi-joint) và phép nối tự nhiên (natural-join)
- Phép nối ngoài (Outer-join)
  - Phép nối ngoài trái ( $\ast=$ )
  - Phép nối ngoài phải ( $=\ast$ )
  - Phép nối ngoài đầy đủ
- Phép nối nhiều bảng



# Phép nối

- Phép nối trong

tên\_bảng\_1 [INNER] JOIN tên\_bảng\_2 ON điều\_kiện\_nối

```
SELECT hodem,ten,ngaysinh  
FROM sinhvien,lop  
WHERE tenlop='Tin K24' AND  
       sinhvien.malop=lop.malop
```

```
SELECT hodem,ten,ngaysinh  
FROM sinhvien INNER JOIN lop  
       ON sinhvien.malop=lop.malop  
WHERE tenlop='Tin K24'
```

# Phép nối ngoài

- Phép nối ngoài trái (LEFT OUTER JOIN)
- Phép nối ngoài phải (RIGHT OUTER JOIN)
- Phép nối ngoài đầy đủ (FULL OUTER JOIN)

```
tên_bảng_1 LEFT|RIGHT|FULL [OUTER] JOIN tên_bảng_2  
ON điều_kiện_nối
```

# Phép nối ngoài

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
4	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

```
SELECT *  
FROM nhanvien LEFT OUTER JOIN donvi  
ON nhanvien.madv=donvi.madv
```

# Phép nối ngoài (LEFT OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
4	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

HOTEN	MADV	MADV	TENDV
Thanh	1	1	Doi ngoai
Hoa	2	2	Hanh chinh
Nam	2	2	Hanh chinh
Vinh	1	1	Doi ngoai
Hung	5	NULL	NULL
Phuong	NULL	NULL	NULL

# Phép nối ngoài (RIGHT OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
4	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

```
SELECT *  
FROM nhanvien RIGHT OUTER JOIN donvi  
ON nhanvien.madv=donvi.madv
```

# Phép nối ngoài (RIGHT OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
4	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

HOTEN	MADV	MADV	TENDV
Thanh	1	1	Doi ngoai
Vinh	1	1	Doi ngoai
Hoa	2	2	Hanh chinh
Nam	2	2	Hanh chinh
NULL	NULL	3	Ke toan
NULL	NULL	4	Kinh doanh

# Phép nối ngoài (FULL OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
4	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

```
SELECT *  
FROM nhanvien FULL OUTER JOIN donvi  
ON nhanvien.madv=donvi.madv
```

# Phép nối ngoài (FULL OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
4	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

HOTEN	MADV	MADV	TENDV
Thanh	1	1	Doi ngoai
Hoa	2	2	Hanh chinh
Nam	2	2	Hanh chinh
Vinh	1	1	Doi ngoai
Hung	5	NULL	NULL
Phuong	NULL	NULL	NULL
NULL	NULL	4	Kinh doanh
NULL	NULL	3	Ke toan



# Phép nối nhiều bảng

- Cho phép thực hiện phép nối nhiều bảng một cách rõ ràng
- Phép nối thực hiện theo thứ tự định nghĩa
- Ví dụ:

```
SELECT hodem,ten,ngaysinh
FROM (sinhvien INNER JOIN lop
      ON sinhvien.malop=lop.malop)
     INNER JOIN khoa ON lop.makhoa=khoa.makhoa
WHERE tenkhoa= 'Khoa công nghệ thông tin'
```

# SELECT

- Truy vấn thông thường
- Truy vấn có điều kiện
- Tạo mới bảng dữ liệu từ kết quả của SELECT
- Sắp xếp kết quả truy vấn
- Phép nối
- **Thống kê dữ liệu với GROUP BY**
- Truy vấn con

# Thống kê dữ liệu với GROUP BY

- Mệnh đề GROUP BY cho phép phân hoạch các dòng dữ liệu thành các nhóm dữ liệu và thực hiện các phép toán trên các nhóm dữ liệu đó
- Các hàm gộp được sử dụng để tính toán trên toàn bảng, hoặc trên mỗi nhóm dữ liệu
- Các hàm gộp được sử dụng như là các cột trong danh sách các cột trong SELECT hoặc HAVING, không được xuất hiện sau WHERE

# Thống kê dữ liệu với GROUP BY

## Hàm gộp

SUM([ALL | DISTINCT] *biểu\_thức*)

AVG([ALL | DISTINCT] *biểu\_thức*)

COUNT([ALL | DISTINCT] *biểu\_thức*)

COUNT(\*)

MAX(*biểu\_thức*)

MIN(*biểu\_thức*)

## Chức năng

Tính tổng các giá trị.

Tính trung bình của các giá trị

Đếm số các giá trị trong biểu thức.

Đếm số các dòng được chọn.

Tính giá trị lớn nhất

Tính giá trị nhỏ nhất

# Thống kê dữ liệu với GROUP BY

- Thống kê trên toàn bộ dữ liệu

```
SELECT AVG(diemlan1)
FROM diemthi
```

```
SELECT MAX(YEAR(GETDATE())-YEAR(ngaysinh)),
       MIN(YEAR(GETDATE())-YEAR(ngaysinh)),
       AVG(YEAR(GETDATE())-YEAR(ngaysinh))
FROM sinhvien
WHERE noisinh='Hà nội '
```

# Thống kê dữ liệu với GROUP BY

- Thống kê trên các nhóm
  - Ví dụ
    - Viết câu lệnh SQL để hiển thị số sinh viên của mỗi lớp
    - Cho biết điểm trung bình thi lần 1 các môn học của các sinh viên.

# Thống kê dữ liệu với GROUP BY

- Thống kê trên các nhóm

```
SELECT lop.malop,tenlop,COUNT(masv) AS siso
FROM lop,sinhvien
WHERE lop.malop=sinhvien.malop
GROUP BY lop.malop,tenlop
```

```
SELECT sinhvien.masv,hodem,ten,
       sum(diemlan1*sodvht)/sum(sodvht)
FROM sinhvien,diemthi,monhoc
WHERE sinhvien.masv=diemthi.masv AND
       diemthi.mamonhoc=monhoc.mamonhoc
GROUP BY sinhvien.masv,hodem,ten
```

# Thống kê dữ liệu với GROUP BY

- Chỉ định điều kiện đối với hàm gộp (HAVING)
  - HAVING sử dụng sau GROUP BY
  - Khác với WHERE:
    - HAVING cho phép sử dụng hàm gộp còn WHERE thì không
- Ví dụ

```
SELECT sinhvien.masv, hodem, ten,  
       SUM(diemlan1*sodvht)/sum(sodvht)  
FROM sinhvien, diemthi, monhoc  
WHERE sinhvien.masv=diemthi.masv AND  
       diemthi.mamonhoc=monhoc.mamonhoc  
GROUP BY sinhvien.masv, hodem, ten  
HAVING sum(diemlan1*sodvht)/sum(sodvht) >=5
```



# SELECT

- Truy vấn thông thường
- Truy vấn có điều kiện
- Tạo mới bảng dữ liệu từ kết quả của SELECT
- Sắp xếp kết quả truy vấn
- Phép nối
- Thống kê dữ liệu với GROUP BY
- **Truy vấn con**

# Truy vấn con (Subquery)

- Là truy vấn được lồng trong một câu lệnh SELECT, INSERT, UPDATE, DELETE hoặc trong một truy vấn con khác
- Sử dụng để biểu diễn những truy vấn trong đó điều kiện truy vấn là kết quả của một truy vấn khác
- Cú pháp

```
(SELECT [ALL | DISTINCT] danh_sách_chọn  
FROM danh_sách_bảng  
[WHERE điều_kiện]  
[GROUP BY danh_sách_cột]  
[HAVING điều_kiện])
```

# Truy vấn con (Subquery)

- Một số chú ý khi sử dụng truy vấn con:
  - Một truy vấn con phải đặt trong ()
  - Kết quả của truy vấn con thường là một cột
  - Tên cột trong truy vấn con có thể là các cột trong truy vấn ngoài
  - Mệnh đề ORDER BY không có trong truy vấn con

# Truy vấn con (Subquery)

- Ví dụ

```
SELECT *  
FROM monhoc  
WHERE sodvht>=(SELECT sodvht  
                FROM monhoc  
                WHERE mamonhoc='TI-001')
```

```
SELECT hodem,ten  
FROM sinhvien JOIN lop ON sinhvien.malop=lop.malop  
WHERE tenlop='Tin K25' AND  
       ngaysinh<ALL(SELECT ngaysinh  
                    FROM sinhvien JOIN lop  
                    ON sinhvien.malop=lop.malop  
                    WHERE lop.tenlop='Toán K25')
```

# Ngôn ngữ thao tác dữ liệu

## DML – Data Manipulation Language

- SELECT
- INSERT
- UPDATE
- DELETE

# Bổ sung dữ liệu: INSERT

- Bổ sung từng dòng dữ liệu với mỗi câu lệnh INSERT
- Bổ sung nhiều dòng dữ liệu bằng cách truy xuất dữ liệu từ các bảng khác

# Bổ sung dữ liệu: INSERT

- Bổ sung từng dòng dữ liệu với mỗi câu lệnh INSERT

- Ví dụ

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

```
INSERT INTO khoa  
VALUES('DHT10','Khoa Luật','054821135')
```

```
INSERT INTO sinhvien(masv,hodem,ten,gioitinh,malop)  
VALUES('0241020008','Nguyễn Công','Chính',1,'C24102')
```

```
INSERT INTO sinhvien  
VALUES('0241020008','Nguyễn Công','Chính',  
      NULL,1,NULL,'C24102')
```

# Bổ sung dữ liệu: INSERT

- Bổ sung nhiều dòng dữ liệu từ các bảng khác

```
INSERT INTO tên_bảng[(danh_sách_cột)] câu_lệnh_SELECT
```

- Ví dụ

```
INSERT INTO luusinhvien  
SELECT hodem,ten,ngaysinh  
FROM sinhvien  
WHERE noisinh like '%Hà nội %'
```

- Chú ý

- Kết quả của câu lệnh SELECT phải có số cột bằng số cột được chỉ định trong bảng đích và phải tương thích về kiểu dữ liệu



# Ngôn ngữ thao tác dữ liệu

## DML – Data Manipulation Language

- SELECT
- INSERT
- **UPDATE**
- DELETE

# Bổ sung dữ liệu: UPDATE

- 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 monhoc  
SET sodvht = 3  
WHERE sodvht = 2
```

# Bổ sung dữ liệu: UPDATE

- Ví dụ

```
UPDATE nhakycbanhang
SET  thanh tien = so luong * gia
FROM mathang
WHERE nhakycbanhang.mahang = mathang.mahang
```

```
UPDATE nhakycbanhang
SET  thanh tien = so luong * gia
FROM mathang
WHERE mathang.mahang =(SELECT mathang.mahang
                        FROM mathang
                        WHERE mathang.mahang=nhakycbanhang.mahang)
```

# Ngôn ngữ thao tác dữ liệu

## DML – Data Manipulation Language

- SELECT
- INSERT
- UPDATE
- **DELETE**

# Xóa dữ liệu: DELETE

- Cú pháp

```
DELETE FROM tên_bảng  
[FROM danh_sách_bảng]  
[WHERE điều_kiện]
```

- Ví dụ

```
DELETE FROM sinhvien  
WHERE noisinh LIKE '%Hà nội %'
```

```
DELETE FROM sinhvien  
FROM lop  
WHERE lop.malop=sinhvien.malop AND tenlop='Tin K24'
```

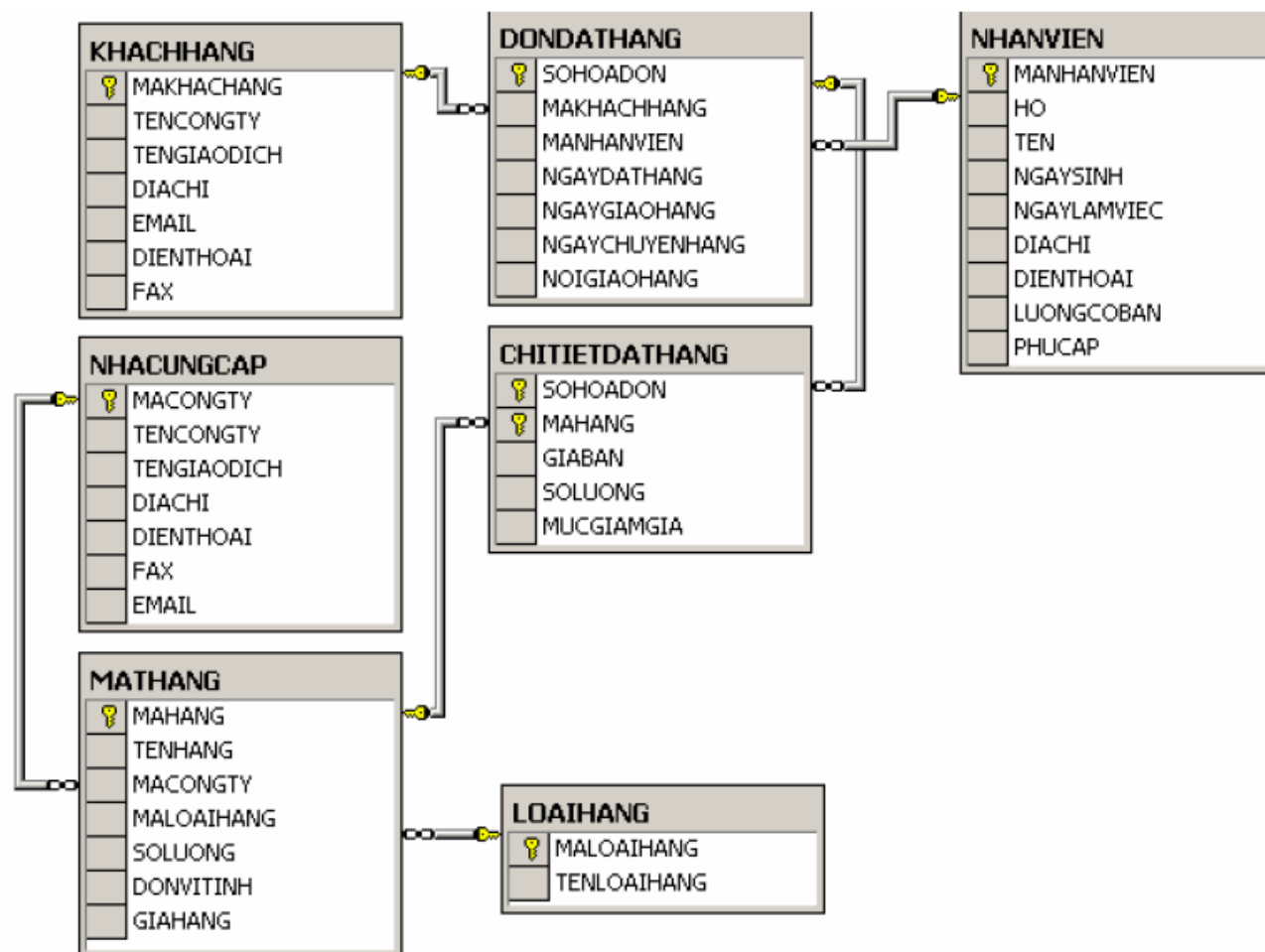
```
DELETE FROM lop  
WHERE malop NOT IN (SELECT DISTINCT malop  
                     FROM sinhvien)
```

# Xóa dữ liệu: DELETE

- Xóa toàn bộ dữ liệu trong bảng
  - Sử dụng câu lệnh DELETE... FROM không có WHERE
  - TRUNCATE

```
TRUNCATE TABLE tên_bảng
```

## Bài tập: Cho CSDL Quản lý bán hàng



Hãy viết các câu lệnh SQL thực hiện các truy vấn sau:

1. Cho biết danh sách các đối tác cung cấp hàng cho công ty.
2. Mã hàng, tên hàng và số lượng hiện có trong công ty
3. Địa chỉ, số điện thoại của nhà cung cấp có tên giao dịch VINAMILK là gì?
4. Cho biết mã và tên các mặt hàng có giá lớn hơn 100000 và số lượng hiện có ít hơn 50
5. Đơn đặt hàng số 1 do ai đặt, do nhân viên nào lập, thời gian và địa điểm giao hàng ở đâu?
6. Hiển thị những nhân viên có lương cơ bản cao nhất công ty
7. Nhân viên nào trong công ty bán được nhiều hàng nhất và số lượng bán được là bao nhiêu?
8. Hãy cho biết tổng số hàng của mỗi loại hàng
9. Tăng lương lên 50% cho những nhân viên bán được số lượng hàng  $\geq 100$  trong năm 2007.
10. Xóa những đơn hàng có ngày đặt hàng trước năm 2005 ra khỏi CSDL
11. Xóa khỏi bảng NHANVIEN những nhân viên đã làm việc cho công ty trên 40 năm



# Bài tập

10. Xóa những đơn hàng có ngày đặt hàng trước năm 2005 ra khỏi CSDL
11. Xóa khỏi bảng NHANVIEN những nhân viên đã làm việc cho công ty trên 40 năm