



# Chương 5

## SQL



# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

# Giới thiệu

- Ngôn ngữ ĐSQH
  - Cách thức truy vấn dữ liệu
  - Khó khăn cho người sử dụng
- SQL (Structured Query Language)
  - Ngôn ngữ cấp cao
  - Người sử dụng chỉ cần đưa ra nội dung cần truy vấn
  - Được phát triển bởi IBM (1970s)
  - Được gọi là SEQUEL
  - Được ANSI công nhận và phát triển thành chuẩn
    - SQL-86
    - SQL-92
    - SQL-99

# Giới thiệu (tt)

- SQL gồm
  - Định nghĩa dữ liệu (DDL)
  - Thao tác dữ liệu (DML)
  - Định nghĩa khung nhìn
  - Ràng buộc toàn vẹn
  - Phân quyền và bảo mật
  - Điều khiển giao tác
- SQL sử dụng thuật ngữ
  - Bảng ~ quan hệ
  - Cột ~ thuộc tính
  - Dòng ~ bộ

Lý thuyết : Chuẩn SQL-92

Ví dụ : SQL Server

# Nội dung chi tiết

- Giới thiệu
- **Định nghĩa dữ liệu**
  - Kiểu dữ liệu
  - Các lệnh định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

# Định nghĩa dữ liệu

- Là ngôn ngữ mô tả
  - Lược đồ cho mỗi quan hệ
  - Miền giá trị tương ứng của từng thuộc tính
  - Ràng buộc toàn vẹn
  - Chỉ mục trên mỗi quan hệ
- Gồm
  - CREATE TABLE (tạo bảng)
  - DROP TABLE (xóa bảng)
  - ALTER TABLE (sửa bảng)
  - CREATE DOMAIN (tạo miền giá trị)
  - CREATE DATABASE
  - ...

# Kiểu dữ liệu

- Số (numeric)
  - INTEGER
  - SMALLINT
  - NUMERIC, NUMERIC(p), NUMERIC(p,s)
  - DECIMAL, DECIMAL(p), DECIMAL(p,s)
  - REAL
  - DOUBLE PRECISION
  - FLOAT, FLOAT(p)

# Kiểu dữ liệu (tt)

- Chuỗi ký tự (character string)
  - CHARACTER, CHARACTER(n)
  - CHARACTER VARYING(x)
- Chuỗi bit (bit string)
  - BIT, BIT(x)
  - BIT VARYING(x)
- Ngày giờ (datetime)
  - DATE gồm ngày, tháng và năm
  - TIME gồm giờ, phút và giây
  - TIMESTAMP gồm ngày và giờ



# Lệnh tạo bảng

## ■ Để định nghĩa một bảng

- Tên bảng
- Các thuộc tính
  - Tên thuộc tính
  - Kiểu dữ liệu
  - Các RBTV trên thuộc tính

## ■ Cú pháp

```
CREATE TABLE <Tên_bảng> (  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],  
    ...  
    [<RBTV>]  
)
```

# Ví dụ - Tạo bảng

```
CREATE TABLE KHACHHANG (  
    MAKH CHAR(4),  
    HOTEN VARCHAR(40),  
    DCHI VARCHAR(50),  
    SODT VARCHAR(20),  
    NGSINH SMALLDATETIME,  
    DOANHISO MONEY,  
    NGDK SMALLDATETIME  
)
```

# Lệnh tạo bảng (tt)

## ■ <RBTV>

- NOT NULL
- NULL
- UNIQUE
- DEFAULT
- PRIMARY KEY
- FOREIGN KEY / REFERENCES
- CHECK

## ■ Đặt tên cho RBTV

<b>CONSTRAINT</b> <Ten_RBTV> <RBTV>
-------------------------------------

# Ví dụ - RBTV

```
CREATE TABLE KHACHHANG (  
    MAKH CHAR(4) PRIMARY KEY,  
    HOTEN VARCHAR(40) NOT NULL,  
    DCHI VARCHAR(50),  
    SODT VARCHAR(20) NOT NULL,  
    NGSINH SMALLDATETIME CHECK (NGSINH > '01-01-1900'),  
    DOANHISO MONEY DEFAULT (10000),  
    NGDK SMALLDATETIME  
)
```

# Ví dụ - RBTV

```
CREATE TABLE NHANVIEN (  
    MANV CHAR(4) PRIMARY KEY,  
    HOTEN VARCHAR(40) NOT NULL,  
    SODT VARCHAR(20) UNIQUE,  
    NGVL SMALLDATETIME DEFAULT (GETDATE())  
)
```

```
CREATE TABLE HOADON (  
    SOHD INT PRIMARY KEY,  
    NGHD SMALLDATETIME,  
    MAKH CHAR(4) REFERENCES KHACHHANG(MAKH),  
    MANV CHAR(4) FOREIGN KEY (MANV)  
        REFERENCES NHANVIEN(MANV),  
    TRIGIA MONEY  
)
```

# Ví dụ - Đặt tên cho RBTV

```
CREATE TABLE KHACHHANG (  
    MAKH CHAR(4) CONSTRAINT KH_MAKH_PK PRIMARY KEY,  
    HOTEN VARCHAR(40) CONSTRAINT KH_HOTEN_NN NOT NULL,  
    DCHI VARCHAR(50),  
    SODT VARCHAR(20) NOT NULL,  
    NGSINH SMALLDATETIME CONSTRAINT KH_NGSINH_CK  
        CHECK (NGSINH > '01-01-1900'),  
    DOANHISO MONEY CONSTRAINT KH_DOANHISO_DF DEFAULT (10000),  
    NGDK SMALLDATETIME  
)
```

# Ví dụ - Đặt tên cho RBTV

```
CREATE TABLE CTHD (  
    SOHD INT,  
    MASP CHAR(4),  
    SL INT,  
    CONSTRAINT CTHD_SOHD_MASP_PK PRIMARY KEY (SOHD, MASP),  
    CONSTRAINT CTHD_SOHD_FK FOREIGN KEY (SOHD)  
        REFERENCES HOADON(SOHD),  
    CONSTRAINT CTHD_MASP_FK FOREIGN KEY (MASP)  
        REFERENCES SANPHAM(MASP)  
)
```

# Lệnh sửa bảng

- Được dùng để
  - Thay đổi cấu trúc bảng
  - Thay đổi RBTV

- Thêm cột

```
ALTER TABLE <Tên_bảng> ADD COLUMN  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>]
```

- Xóa cột

```
ALTER TABLE <Tên_bảng> DROP COLUMN <Tên_cột>
```

- Mở rộng cột

```
ALTER TABLE <Tên_bảng> ALTER COLUMN  
    <Tên_cột> <Kiểu_dữ_liệu_mới>
```



# Lệnh sửa bảng (tt)

## ■ Thêm RBTV

```
ALTER TABLE <Tên_bảng> ADD  
    CONSTRAINT <Ten_RBTV> <RBTV>,  
    CONSTRAINT <Ten_RBTV> <RBTV>,  
    ...
```

## ■ Xóa RBTV

```
ALTER TABLE <Tên_bảng> DROP <Tên_RBTV>
```

# Ví dụ - Thay đổi cấu trúc bảng

```
ALTER TABLE SANPHAM ADD  
    GHICHU VARCHAR(20)
```

```
ALTER TABLE SANPHAM DROP COLUMN GHICHU
```

```
ALTER TABLE SANPHAM ALTER COLUMN  
    GHICHU VARCHAR(50)
```

# Ví dụ - Thay đổi RBTV

```
CREATE TABLE HOADON (
```

```
    SOHD INT,
```

```
    NGHD SMALLDATETIME,
```

```
    MAKH CHAR(4),
```

```
    MANV CHAR(4),
```

```
    TRIGIA MONEY
```

```
)
```

```
ALTER TABLE HOADON ADD
```

```
    CONSTRAINT HD_SOHD_PK PRIMARY KEY (SOHD),
```

```
    CONSTRAINT HD_MAKH_FK FOREIGN KEY (MAKH)
```

```
        REFERENCES KHACHHANG(MAKH),
```

```
    CONSTRAINT HD_MANV_FK FOREIGN KEY (MANV)
```

```
        REFERENCES NHANVIEN(MANV),
```

```
    CONSTRAINT HD_NGHD_DF DEFAULT (GETDATE()) FOR (NGHD)
```

# Lệnh xóa bảng

- Được dùng để xóa cấu trúc bảng
  - Tất cả dữ liệu của bảng cũng bị xóa

- Cú pháp

```
DROP TABLE <Tên_bảng>
```

- Ví dụ

```
DROP TABLE KHACHHANG
```

```
DROP TABLE HOADON
```

```
DROP TABLE SANPHAM
```

# Lệnh xóa bảng (tt)

## GIAOVIEN

HOTEN	HOCVI	HOCHAM	<u>MAGV</u>	GIOITINH	NGSINH	NGVL	HESO	MUCLUONG	MAKHOA
-------	-------	--------	-------------	----------	--------	------	------	----------	--------

## KHOA

TENKHOA	<u>MAKHOA</u>	TRGKHOA	NGTLAP
---------	---------------	---------	--------

# Lệnh tạo miền giá trị

- Tạo ra một kiểu dữ liệu mới kế thừa những kiểu dữ liệu có sẵn
- Cú pháp

```
CREATE DOMAIN <Tên_kdl_mới> AS <Kiểu_dữ_liệu>
```

- Ví dụ

```
CREATE DOMAIN Kieu_Ten AS VARCHAR(30)
```

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- **Truy vấn dữ liệu**
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - Hàm kết hợp và gom nhóm
  - Một số kiểu truy vấn khác
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

# Truy vấn dữ liệu

- Là ngôn ngữ rút trích dữ liệu thỏa một số điều kiện nào đó
- Dựa trên

Phép toán ĐSQH

+

Một số bổ sung

- Cho phép 1 bảng có nhiều dòng trùng nhau
- Bảng là *bag*  $\neq$  quan hệ là *set*



# Truy vấn cơ bản

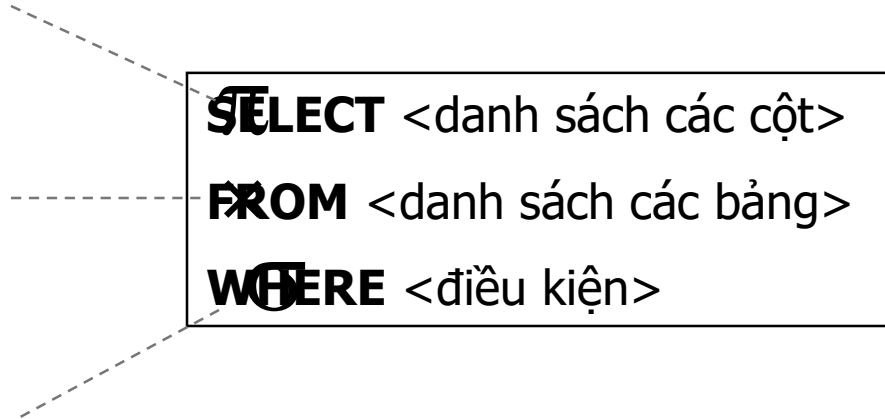
- Gồm 3 mệnh đề

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>
```

- <danh sách các cột>
  - Tên các cột cần được hiển thị trong kết quả truy vấn
- <danh sách các bảng>
  - Tên các bảng liên quan đến câu truy vấn
- <điều kiện>
  - Biểu thức boolean xác định dòng nào sẽ được rút trích
  - Nối các biểu thức: AND, OR, và NOT
  - Phép toán: < , > , ≤ , ≥ , ≠ , =, LIKE và BETWEEN

# Truy vấn cơ bản (tt)

- SQL và ĐSQH



**SELECT** <danh sách các cột>  
**FROM** <danh sách các bảng>  
**WHERE** <điều kiện>

**SELECT** L  
**FROM**  $\pi_C(\sigma_C(R)) \longrightarrow$   
**WHERE** C

# Ví dụ

Lấy tất cả các cột của  
quan hệ kết quả

```
SELECT *  
FROM KHACHHANG  
WHERE DOANHSO > 1000000
```

MAKH	HOTEN	DCHI	SODT	NGSINH	DOANHSO	NGDK
KH01	Nguyen Van A	731 Tran Hung Dao, Q5, TpHCM	08823451	22/10/1960	13,060,000	22/07/2006
KH03	Tran Ngoc Linh	45 Nguyen Canh Chan, Q1, TpHCM	0938776266	12/6/1980	3,860,000	05/08/2006

$\sigma_{DOANHSO > 1000000} (KHACHHANG)$

# Mệnh đề SELECT

```
SELECT MAKH, HOTEN, DCHI  
FROM KHACHHANG  
WHERE DOANHSO > 1000000 AND YEAR(NGDK) = 2006
```

MAKH	HOTEN	DCHI
KH01	Nguyen Van A	731 Tran Hung Dao, Q5, TpHCM
KH03	Tran Ngoc Linh	45 Nguyen Canh Chan, Q1, TpHCM

$$\pi_{\text{MAKH, HOTEN, DCHI}} (\sigma_{\text{DOANHSO} > 1000000 \wedge \text{YEAR(NGDK) = 2006}} (\text{KHACHHANG}))$$

# Mệnh đề SELECT (tt)

Tên bí danh

```
SELECT MANV, HOTEN AS TEN, NGVL AS 'NGAY VAO LAM'  
FROM NHANVIEN  
WHERE NGVL > 01/06/2006
```

MANV	TEN	NGAY VAO LAM
NV04	Ngo Thanh Tuan	24/6/2006
NV05	Nguyen Thi Truc Thanh	20/7/2006

$$\rho_{MANV, TEN, NGAY VAO LAM}(\pi_{MANV, HOTEN, NGVL}(\sigma_{NGVL > 01/06/2006}(NHANVIEN)))$$

# Mệnh đề SELECT (tt)

## Mở rộng

```
SELECT MASP, TENSX + ' ' + NUOCSX AS 'TEN SAN PHAM'  
FROM SANPHAM  
WHERE GIA >= 55000
```

MASP	TEN SAN PHAM
BB03	But bi Thai Lan
ST02	So tay loai 1 Viet Nam
ST04	So tay Thai Lan

$\rho_{MASP, TEN\ SAN\ PHAM}(\pi_{MASP, TENSX + NUOCSX}(\sigma_{GIA \geq 55000}(SANPHAM)))$

# Mệnh đề SELECT (tt)

## Mở rộng

```
SELECT MASP, GIA*1.1 AS 'GIA10%'  
FROM GIA  
WHERE GIA >= 55000
```

MASP	GIA10%
BB03	110.000
ST02	60.500
ST04	60.500

$$\rho_{MASP, GIA10\%}(\pi_{MASP, GIA*1.1}(\sigma_{GIA \geq 55000}(SANPHAM)))$$

# Mệnh đề SELECT (tt)

Loại bỏ các dòng trùng nhau

```
SELECT DISTINCT GIA  
FROM SANPHAM  
WHERE GIA > 50000
```

GIA
100000
55000
53000
53000

- Tồn chi phí
- Người dùng muốn thấy



# Ví dụ

- Cho biết MANV và HOTEN lập những hóa đơn trên 3.500.000

$R1 \leftarrow \text{NHANVIEN} \bowtie_{\text{MANV}=\text{MANV}} \text{HOADON}$

$KQ \leftarrow \pi_{\text{MANV}, \text{HOTEN}} (\sigma_{\text{TRIGIA} > 3500000}(R1))$

SELECT HOADON.MANV, HOTEN

FROM NHANVIEN, HOADON

WHERE TRIGIA > 3500000 AND NHANVIEN.MANV=HOADON.MANV

# Mệnh đề WHERE

SELECT HOADON.MANV, HOTEN

FROM NHANVIEN, HOADON

WHERE TRIGIA > 350000 AND NHANVIEN.MANV=HOADON.MANV

Biểu thức luận lý

↓  
TRUE

↓  
TRUE

# Mệnh đề WHERE (tt)

Độ ưu tiên

```
SELECT HOADON.MANV, TENNV  
FROM NHANVIEN, HOADON  
WHERE (GIA>3500000 OR GIA<1000000)  
AND NHANVIEN.MANV=HOADON.MANV
```

# Mệnh đề WHERE (tt)

**BETWEEN**

```
SELECT MASP, TENSP  
FROM SANPHAM  
WHERE GIA>20000 AND GIA<30000
```

```
SELECT MASP, TENSP  
FROM SANPHAM  
WHERE GIA BETWEEN 20000 AND 30000
```

# Mệnh đề WHERE (tt)

NOT BETWEEN

```
SELECT MASP, TENSPP  
FROM SANPHAM  
WHERE GIA NOT BETWEEN 20000 AND 30000
```

# Mệnh đề WHERE (tt)

LIKE

```
SELECT MAKH, HOTEN  
FROM KHACHHANG  
WHERE DCHI LIKE 'Nguyen _ _ _ _ '
```

Ký tự bất kỳ

```
SELECT MAKH, HOTEN  
FROM KHACHHANG  
WHERE DCHI LIKE 'Nguyen %'
```

Chuỗi bất kỳ

# Mệnh đề WHERE (tt)

NOT LIKE

```
SELECT MASP, TENSP  
FROM SANPHAM  
WHERE DVT LIKE 'CAY'
```

```
SELECT MASP, TENSP  
FROM SANPHAM  
WHERE DVT NOT LIKE 'CAY'
```

# Mệnh đề WHERE (tt)

ESCAPE

```
SELECT MAKH, HOTEN  
FROM KHACHHANG  
WHERE DCHI LIKE '% Nguyens_%' ESCAPE 's'
```



'Nguyen\_'



# Mệnh đề WHERE (tt)

Ngày giờ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE NGVL BETWEEN '2006-12-08' AND '2014-07-19'
```

'2006-12-08'	YYYY-MM-DD	'17:30:00'	HH:MI:SS
'12/08/2006'	MM/DD/YYYY	'05:30 PM'	
'December 8, 2006'			

'2006-12-08 17:30:00'

# Mệnh đề WHERE (tt)

## NULL

- Sử dụng trong trường hợp
  - Không biết (value unknown)
  - Không thể áp dụng (value inapplicable)
  - Không tồn tại (value withheld)
- Những biểu thức tính toán có liên quan đến giá trị NULL sẽ cho ra kết quả là NULL
  - $x$  có giá trị là NULL
  - $x + 3$  cho ra kết quả là NULL
  - $x + 3$  là một biểu thức không hợp lệ trong SQL
- Những biểu thức so sánh có liên quan đến giá trị NULL sẽ cho ra kết quả là UNKNOWN
  - $x = 3$  cho ra kết quả là UNKNOWN
  - $x = 3$  là một so sánh không hợp lệ trong SQL

# Mệnh đề WHERE (tt)

NULL

```
SELECT MANV, HOTEN  
FROM NHANVIEN  
WHERE SODT IS NULL
```

```
SELECT MANV, HOTEN  
FROM NHANVIEN  
WHERE SODT IS NOT NULL
```

# Mệnh đề FROM

Không sử dụng mệnh đề WHERE

```
SELECT MANV, SOHD  
FROM NHANVIEN, HOADON  
WHERE TRUE
```

MANV	SOHD
333445555	1
333445555	4
333445555	5
987987987	1
987987987	4
987987987	5
...	...

# Mệnh đề FROM (tt)

Tên bí danh

```
SELECT SOHD, HOTEN  
FROM HOADON, AS NV NHANVIEN AS NV  
WHERE MANV HOADON.MANV
```

```
SELECT MANV HOTEN, SOHD  
FROM NHANVIEN, HOADON HOADON HD  
WHERE MANV HOADON.MANV
```

# Ví dụ 1

- Với những chi tiết hóa đơn mua với số lượng trên 50, cho biết số hóa đơn, trị giá hóa đơn, mã và họ tên nhân viên lập hóa đơn.

## Ví dụ 2

- Tìm các sản phẩm (MASP,TENSP) do “Trung Quoc” hoặc “Thai Lan” sản xuất có giá từ 30.000 đến 40.000.

---

# Ví dụ 3

- In ra trị giá của từng hóa đơn và họ tên người lập hóa đơn đó.



---

# Ví dụ 4

- Tìm tất cả các hóa đơn của khách hàng 'Nguyen Van A'

# Mệnh đề ORDER BY

- Dùng để hiển thị kết quả câu truy vấn theo một thứ tự nào đó
- Cú pháp

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>  
ORDER BY <danh sách các cột>
```

- ASC: tăng (mặc định)
- DESC: giảm

# Mệnh đề ORDER BY (tt)

## ■ Ví dụ

```
SELECT NGDK, DOANHSO  
FROM KHACHHANG  
ORDER BY NGDK DESC, DOANHSO
```

NGDK	DOANHSO
10/09/2014	1.000.000
10/09/2014	3.350.000
23/12/2013	2.490.000
23/12/2013	3.210.000
11/07/2012	100.000
11/07/2012	2.040.000
11/07/2012	5.300.000

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
  - Truy vấn cơ bản
  - **Tập hợp, so sánh tập hợp và truy vấn lồng**
  - Hàm kết hợp và gom nhóm
  - Một số dạng truy vấn khác
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

# Phép toán tập hợp trong SQL

- SQL có cài đặt các phép toán
  - Hội (UNION)
  - Giao (INTERSECT)
  - Trừ (EXCEPT)
- Kết quả trả về là tập hợp
  - Loại bỏ các bộ trùng nhau
  - Để giữ lại các bộ trùng nhau
    - UNION ALL
    - INTERSECT ALL
    - EXCEPT ALL

# Phép toán tập hợp trong SQL (tt)

## ■ Cú pháp

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

**UNION [ALL]**

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

**INTERSECT [ALL]**

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

**EXCEPT [ALL]**

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

# Ví dụ 5

- Cho biết các số hóa đơn có
  - Nhân viên lập hóa đơn có họ là 'Nguyen' hoặc,
  - Khách hàng mua hàng có họ là 'Nguyen'

SELECT SOHD

FROM HOADON HD, NHANVIEN NV

WHERE HD.MANV=NV.MANV AND HOTEN LIKE 'Nguyen %'

UNION

SELECT SOHD

FROM HOADON HD, KHACHHANG KH

WHERE HD.MAKH=KH.MAKH AND HOTEN LIKE 'Nguyen %'

# Ví dụ 6

- Tìm hóa đơn có ngày hóa đơn bằng với ngày vào làm của nhân viên lập nó

```
SELECT MANV, NGHD FROM HOADON
```

```
INTERSECT
```

```
SELECT MANV, NGVL FROM NHANVIEN
```

```
SELECT HD.*
```

```
FROM HOADON HD, NHANVIEN NV
```

```
WHERE HD.MANV = NV.MANV AND NGHD = NGVL
```



# Ví dụ 7

- Tìm những nhân viên không lập hóa đơn nào

```
SELECT MANV FROM NHANVIEN  
EXCEPT  
SELECT MANV FROM HOADON
```

# Truy vấn lồng

SELECT NV.MANV, HOTEN

FROM NHANVIEN NV, HOADON HD

WHERE HOTEN LIKE 'Nguyễn %' AND NV.MANV = HD.MANV

Câu truy vấn cha  
(Outer query)

**SELECT** <danh sách các cột>

**FROM** <danh sách các bảng>

**WHERE** <so sánh tập hợp> (

**SELECT** <danh sách các cột>

**FROM** <danh sách các bảng>

**WHERE** <điều kiện>)

Câu truy vấn con  
(Subquery)

# Truy vấn lồng (tt)

- Các câu lệnh SELECT có thể lồng nhau ở nhiều mức
- Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic
- Câu truy vấn con thường trả về một tập các giá trị
- Mệnh đề WHERE của câu truy vấn cha
  - <biểu thức> <so sánh tập hợp> <truy vấn con>
  - So sánh tập hợp thường đi cùng với một số toán tử
    - IN, NOT IN
    - ALL
    - ANY hoặc SOME
  - Kiểm tra sự tồn tại
    - EXISTS
    - NOT EXISTS

# Truy vấn lồng (tt)

## ■ Có 2 loại truy vấn lồng

### - Lồng phân cấp

- Mệnh đề WHERE của truy vấn con không tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
- Khi thực hiện, câu truy vấn con sẽ được thực hiện trước

### - Lồng tương quan

- Mệnh đề WHERE của truy vấn con tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
- Khi thực hiện, câu truy vấn con sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của truy vấn cha

# Ví dụ - Lồng phân cấp

```
SELECT MANV, HOTEN  
FROM NHANVIEN NV, HOADON HD  
WHERE MAKH='KH01' AND HD.MANV=NV.MANV
```

```
SELECT MANV, HOTEN  
FROM NHANVIEN  
WHERE MANV IN ('NV01', 'NV02', 'NV03')
```

```
SELECT MANV  
FROM HOADON  
WHERE MAKH='KH01')
```

# Ví dụ 5

```
SELECT DISTINCT SOHD
FROM HOADON
FROM NHANVIEN NV, HOADON HD
WHERE NV.MANV=HD.MANV AND NV.HOTEN='Nguyen%'
UNION
SELECT SOHD
FROM NHANVIEN
WHERE HOTEN='Nguyen%' )
FROM KHACHHANG KH, HOADON HD
OR MAKH IN (
WHERE KH.MAKH=HD.MAKH AND KH.HOTEN='Nguyen%'
UNION
SELECT MAKH
FROM KHACHHANG
WHERE HOTEN='Nguyen%' )
```

# Ví dụ 7

- Tìm những nhân viên không lập hóa đơn nào

```
SELECT *  
FROM NHANVIEN  
WHERE MANV NOT IN (  
    SELECT MANV  
    FROM HOADON)
```

```
SELECT *  
FROM NHANVIEN  
WHERE MANV <> ALL (  
    SELECT MANV  
    FROM HOADON)
```

# Ví dụ 8

- Tìm những hóa đơn có trị giá lớn hơn trị giá của ít nhất một hóa đơn do nhân viên có mã 'NV01' lập

```
SELECT *  
FROM HOADON  
WHERE TRIGIA > ANY (  
    SELECT TRIGIA  
    FROM HOADON  
    WHERE MANV='NV01')
```

```
SELECT HD1.*  
FROM HOADON HD1, HOADON HD2  
WHERE HD1.TRIGIA > HD2.TRIGIA AND HD2.MANV='NV01'
```



# Ví dụ 9

- Tìm những hóa đơn có trị giá lớn hơn trị giá của tất cả hóa đơn do nhân viên có mã 'NV01' lập

```
SELECT *  
FROM HOADON  
WHERE TRIGIA > ALL (  
    SELECT TRIGIA  
    FROM HOADON  
    WHERE MANV='NV01')
```

# Ví dụ 10

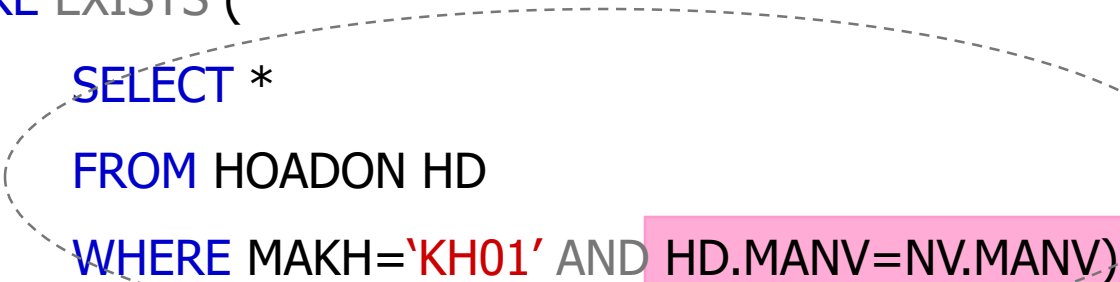
- Tìm những sản phẩm có trong tối thiểu một hóa đơn

```
SELECT *  
FROM SANPHAM  
WHERE MASP IN (SELECT MASP FROM CTHD)
```

# Ví dụ - Lồng tương quan

```
SELECT NV.MANV, HOTEN  
FROM NHANVIEN NV, HOADON HD  
WHERE MAKH='KH01' AND HD.MANV=N.V.MANV
```

```
SELECT MANV, HOTEN  
FROM NHANVIEN NV  
WHERE EXISTS (  
    SELECT *  
    FROM HOADON HD  
    WHERE MAKH='KH01' AND HD.MANV=N.V.MANV)
```



# Ví dụ 6

- Tìm hóa đơn của khách hàng có ngày đăng ký trùng với ngày hóa đơn và doanh số khách hàng bằng với trị giá hóa đơn đó

```
SELECT *  
FROM HOADON HD  
WHERE EXISTS (  
    SELECT *  
    FROM KHACHHANG KH  
    WHERE KH.MAKH=HD.MAKH  
    AND KH.NGDK=HD.NGHD  
    AND KH.DOANHISO=HD.TRIGIA)
```

# Ví dụ 7

- Tìm những nhân viên không lập hóa đơn nào

```
SELECT *  
FROM NHANVIEN NV  
WHERE NOT EXISTS (  
    SELECT *  
    FROM HOADON HD  
    WHERE HD.MANV=N.V.MANV)
```

# Ví dụ 8

- Tìm những hóa đơn có trị giá lớn hơn trị giá của ít nhất một hóa đơn do 'NV01' lập

```
SELECT *  
FROM HOADON HD1  
WHERE EXISTS (  
    SELECT *  
    FROM HOADON HD2  
    WHERE MANV='NV01'  
    AND HD1.TRIGIA>HD2.TRIGIA)
```

# Ví dụ 10

- Tìm những hóa đơn mua tối thiểu một sản phẩm do 'Viet Nam' sản xuất

```
SELECT *  
FROM HOADON HD  
WHERE EXISTS (  
    SELECT *  
    FROM CTHD CT  
    WHERE HD.SOHD=CT.SOHD  
        AND EXISTS (  
            SELECT *  
            FROM SANPHAM SP  
            WHERE CT.MASP=SP.MASP AND NUOCSX='Viet Nam'))
```

# Nhận xét IN và EXISTS

## ■ IN

- <tên cột> IN <câu truy vấn con>
- Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha

## ■ EXISTS

- Không cần có thuộc tính, hằng số hay biểu thức nào khác đứng trước
- Không nhất thiết liệt kê tên thuộc tính ở mệnh đề SELECT của truy vấn con
- Những câu truy vấn có = ANY hay IN đều có thể chuyển thành câu truy vấn có EXISTS



# Phép chia trong SQL

R	A	B	C	D	E
	$\alpha$	a	$\alpha$	a	1
	$\alpha$	a	$\gamma$	a	1
	$\alpha$	a	$\gamma$	b	1
	$\beta$	a	$\gamma$	a	1
	$\beta$	a	$\gamma$	b	3
	$\gamma$	a	$\gamma$	a	1
	$\gamma$	a	$\gamma$	b	1
	$\gamma$	a	$\beta$	b	1

S	D	E
$b_i$	a	1
	b	1

$R \div S$	A	B	C
$a_i$	$\alpha$	a	$\gamma$
	$\gamma$	a	$\gamma$

- $R \div S$  là tập các giá trị  $a_i$  trong R sao cho không có giá trị  $b_i$  nào trong S làm cho bộ  $(a_i, b_i)$  không tồn tại trong R

# Phép chia trong SQL (tt)

- Sử dụng NOT EXISTS để biểu diễn

```
SELECT R1.A, R1.B, R1.C
```

```
FROM R R1
```

```
WHERE NOT EXISTS (
```

```
    SELECT *
```

```
    FROM S
```

```
    WHERE NOT EXISTS (
```

```
        SELECT *
```

```
        FROM R R2
```

```
        WHERE R2.D=S.D AND R2.E=S.E
```

```
        AND R1.A=R2.A AND R1.B=R2.B AND R1.C=R2.C ))
```

# Ví dụ 11

- Tìm trị giá hóa đơn đã mua tất cả các sản phẩm
  - Tìm trị giá các hóa đơn mà không có sản phẩm nào là không mua
  - Tập bị chia: CTHD(MASP, SOHD)
  - Tập chia: SANPHAM(MASP)
  - Tập kết quả: KQ(SOHD)
  - Kết KQ với HOADON để lấy ra TRIGIA

# Ví dụ 11 (tt)

```
SELECT HD.TRIGIA
FROM HOADON HD
WHERE NOT EXISTS (
    SELECT *
    FROM SANPHAM SP
    WHERE NOT EXISTS (
        SELECT *
        FROM CTHD CT
        WHERE CTHD.MASP=SP.MASP
        AND CTHD.SOHD=HD.SOHD))
```

# Ví dụ 11 (tt)

```
SELECT HD.TRIGIA
FROM HOADON HD
WHERE NOT EXISTS (
    SELECT MASP
    FROM SANPHAM SP
    EXCEPT
    SELECT MASP
    FROM CTHD
    WHERE CTHD.SOHD = HD.SOHD
)
```

# Ví dụ 11 (tt)

```
SELECT HD.TRIGIA
FROM HOADON HD
WHERE NOT EXISTS (
    SELECT *
    FROM SANPHAM SP
    WHERE MASP NOT IN (
        SELECT MASP
        FROM CTHD
        WHERE CTHD.SOHD = HD.SOHD))
```

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - **Hàm kết hợp và gom nhóm**
  - Một số dạng truy vấn khác
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

# Hàm kết hợp

## ■ COUNT

- COUNT(\*) đếm số dòng
- COUNT(<tên thuộc tính>) đếm số giá trị khác NULL của thuộc tính
- COUNT(DISTINCT <tên thuộc tính>) đếm số giá trị khác nhau và khác NULL của thuộc tính

## ■ MIN

## ■ MAX

## ■ SUM

## ■ AVG

## ■ Các hàm kết hợp được đặt ở mệnh đề SELECT



# Ví dụ 12

- Tìm tổng trị giá, trị giá cao nhất, trị giá thấp nhất và trị giá trung bình của các hóa đơn

```
SELECT SUM(TRIGIA), MAX(TRIGIA), MIN(TRIGIA), AVG(TRIGIA)  
FROM HOADON
```

# Ví dụ 13

- Cho biết số lượng hóa đơn của nhân viên 'Nguyen Van B'

```
SELECT COUNT(*) AS SL_HD
FROM HOADON HD, NHANVIEN NV
WHERE HD.MANV=NV.MANV
      AND HOTEN='Nguyen Van B'
```

# Ví dụ 14

- Cho biết số lượng sản phẩm của từng nước.

NUOCSX	SL_SP
Singapore	2
Viet Nam	3
Trung Quoc	1

MASP	TENSP	DVT	NUOCSX	GIA
BC01	But chi	cay	Singapore	3,000
BC02	But chi	cay	Singapore	5,000
BC03	But chi	cay	Viet Nam	3,500
BC04	But chi	hop	Viet Nam	30,000
BB01	But bi	cay	Viet Nam	5,000
BB02	But bi	cay	Trung Quoc	7,000

# Gom nhóm

## ■ Cú pháp

**SELECT** <danh sách các cột>

**FROM** <danh sách các bảng>

**WHERE** <điều kiện>

**GROUP BY** <danh sách các cột gom nhóm>

## ■ Sau khi gom nhóm

- Mỗi nhóm các bộ sẽ có cùng giá trị tại các thuộc tính gom nhóm

# Ví dụ 14

- Cho biết số lượng hóa đơn đã lập của từng nhân viên

```
SELECT MANV, COUNT(*) AS SL_HD  
FROM HOADON  
GROUP BY MANV
```

```
SELECT HOTEN, COUNT(*) AS SL_HD  
FROM HOADON HD, NHANVIEN NV  
WHERE HD.MANV=N.V.MANV  
GROUP BY HOTEN
```

# Ví dụ 15

- Với mỗi nhân viên cho biết mã số, họ tên, số lượng hóa đơn và tổng trị giá hóa đơn mà họ đã lập

MANV	SOHD	TRIGIA
NV01	1004	180,000
NV01	1010	5,200,000
NV02	1002	840,000
NV02	1005	3,800,000
NV02	1014	3,150,000
NV03	1006	2,430,000
NV04	1016	200,000
NV04	1020	700,000
NV05	1021	230,000
NV05	1003	1,050,000
NV06	1031	420,000
NV06	1011	150,000

# Ví dụ 16

- Cho biết những nhân viên lập từ 2 hóa đơn trở lên

MANV	SOHD	TRIGIA
NV01	1004	180,000
NV01	1010	5,200,000
NV02	1002	840,000
NV02	1005	3,800,000
NV02	1014	3,150,000
NV03	1006	2,430,000
NV04	1009	200,000
NV04	1020	70,000
NV05	1021	230,000
NV05	1003	1,050,000
NV06	1031	420,000
NV06	1011	150,000

bị loại ra

# Điều kiện trên nhóm

## ■ Cú pháp

**SELECT** <danh sách các cột>

**FROM** <danh sách các bảng>

**WHERE** <điều kiện>

**GROUP BY** <danh sách các cột gom nhóm>

**HAVING** <điều kiện trên nhóm>



# Ví dụ 16

- Cho biết những nhân viên lập từ 2 hóa đơn trở lên

```
SELECT MANV  
FROM HOADON  
GROUP BY MANV  
HAVING COUNT(*) >= 2
```

# Ví dụ 17

- Cho biết những nhân viên (HOTEN) có trị giá hóa đơn đã lập trung bình lớn hơn 1.000.000

```
SELECT MANV, AVG(TRIGIA) AS TRIGIA_TB  
FROM HOADON  
GROUP BY MANV  
HAVING AVG(TRIGIA) > 1000000
```

```
SELECT HOTEN, AVG(TRIGIA) AS TRIGIA_TB  
FROM HOADON HD, NHANVIEN NV  
WHERE HD.MANV=NV.MANV  
GROUP BY HOTEN  
HAVING AVG(TRIGIA) > 1000000
```

# Nhận xét

## ■ Mệnh đề GROUP BY

- Các thuộc tính trong mệnh đề SELECT (trừ những thuộc tính trong các hàm kết hợp) phải xuất hiện trong mệnh đề GROUP BY

## ■ Mệnh đề HAVING

- Sử dụng các hàm kết hợp trong mệnh đề SELECT để kiểm tra một số điều kiện nào đó
- Chỉ kiểm tra điều kiện trên nhóm, không là điều kiện lọc trên từng bộ
- Sau khi gom nhóm điều kiện trên nhóm mới được thực hiện

# Nhận xét (tt)

- Thứ tự thực hiện câu truy vấn có mệnh đề GROUP BY và HAVING
  - (1) Chọn ra những dòng thỏa điều kiện trong mệnh đề WHERE
  - (2) Những dòng này sẽ được gom thành nhiều nhóm tương ứng với mệnh đề GROUP BY
  - (3) Áp dụng các hàm kết hợp cho mỗi nhóm
  - (4) Bỏ qua những nhóm không thỏa điều kiện trong mệnh đề HAVING
  - (5) Rút trích các giá trị của các cột và hàm kết hợp trong mệnh đề SELECT

# Ví dụ 18

- Tìm những nhân viên có trị giá hóa đơn đã lập trung bình cao nhất

```
SELECT MANV, AVG(TRIGIA) AS TRIGIA_TB
FROM HOADON
GROUP BY MANV
HAVING MAX(TRIGIA) > ALL (
    SELECT AVG(TRIGIA)
    FROM HOADON
    GROUP BY MANV)
```

# Ví dụ 19

- Tìm 3 hóa đơn có trị giá cao nhất

```
SELECT SOHD
FROM HOADON HD1
WHERE 2 >= (
    SELECT COUNT(*)
    FROM HOADON HD2
    WHERE HD2.TRIGIA > HD1.TRIGIA )
```

# Ví dụ 12

- Tìm tên nhân viên lập tất cả các hóa đơn

```
SELECT NV.MANV, HOTEN
FROM NHANVIEN NV, HOADON HD
WHERE NV.MANV=HD.MANV
GROUP BY NV.MANV, HOTEN
HAVING COUNT(*) = (
    SELECT COUNT(*)
    FROM HOADON )
```

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- **Truy vấn dữ liệu**
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - Hàm kết hợp và gom nhóm
  - **Một số dạng truy vấn khác**
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)



# Một số dạng truy vấn khác

- Truy vấn con ở mệnh đề FROM
- Điều kiện kết ở mệnh đề FROM
  - Phép kết tự nhiên
  - Phép kết ngoài
- Cấu trúc CASE

# Truy vấn con ở mệnh đề FROM

- Kết quả trả về của một câu truy vấn phụ là một bảng
  - Bảng trung gian trong quá trình truy vấn
  - Không có lưu trữ thật sự
- Cú pháp

**SELECT** <danh sách các cột>

**FROM** R1, R2, (<truy vấn con>) **AS** tên\_bảng

**WHERE** <điều kiện>

# Ví dụ 18

- Cho biết những khách hàng (HOTEN) có trị giá trung bình của các hóa đơn lớn hơn 500.000

```
SELECT MAKH, AVG(TRIGIA) AS TRIGIA_TB
FROM HOADON HD, KHACHHANG KH
GROUP BY MAKH, (SELECT MAKH, AVG(TRIGIA) AS TRIGIA_TB
FROM HOADON HD, KHACHHANG KH
HAVING AVG(TRIGIA) > 500000)
GROUP BY MAKH
SELECT MAKH, HOTEN, AVG(TRIGIA) AS TRIGIA_TB
HAVING AVG(TRIGIA) > 500000 ) AS TEMP
FROM HOADON HD, KHACHHANG KH
WHERE KH.MAKH=TEMP.MAKH
WHERE HD.MAKH=KH.MAKH
GROUP BY MAKH, HOTEN
HAVING AVG(TRIGIA) > 500000
```

# Điều kiện kết ở mệnh đề FROM

## ■ Kết bằng

```
SELECT <danh sách các cột>  
FROM R1 [INNER] JOIN R2 ON <biểu thức>  
WHERE <điều kiện>
```

## ■ Kết ngoài

```
SELECT <danh sách các cột>  
FROM R1 LEFT | RIGHT [OUTER] JOIN R2 ON <biểu thức>  
WHERE <điều kiện>
```

# Ví dụ 20

- Tìm mã và tên các nhân viên lập hóa đơn cho khách hàng 'KH01'

```
SELECT MANV, HOTEN  
FROM NHANVIEN NV, HOADON HD  
WHERE MAKH='KH01' AND NV.MANV=HD.MANV
```

```
SELECT MANV, HOTEN  
FROM NHANVIEN NV INNER JOIN HOADON HD ON NV.MANV=HD.MANV  
WHERE MAKH='KH01'
```

# Ví dụ 21

- Cho biết số hóa đơn và họ tên khách hàng nếu có

SOHD	HOTEN
1020	Le Ha Vinh
1021	Ha Duy Lap
1022	Null
1023	Null

```
SELECT SOHD, HOTEN  
FROM HOADON HD, KHACHHANG KH  
WHERE HD.MAKH=KH.MAKH
```

# Ví dụ 21 (tt)

SOHD	HOTEN
1020	Le Ha Vinh
1021	Ha Duy Lap
1022	Null
1023	Null

Mở rộng dữ liệu cho  
bảng HOADON



**SELECT** SOHD, HOTEN

**FROM** KHACHHANG **LEFT JOIN** KHACHHANG **ON** HOADON=HOADON

# Ví dụ 22

- Tìm số hóa đơn và tên các khách hàng mà nhân viên đã lập hóa đơn nếu có

HOADON join NHANVIEN

join

KHACHHANG

MAKH=MAKH

SELECT HD.SOHD, KH.HOTEN

FROM (HOADON HD JOIN NHANVIEN NV ON HD.MANV=NV.MANV)  
RIGHT JOIN KHACHHANG KH ON HD.MAKH=KH.MAKH



# Cấu trúc CASE

- Cho phép kiểm tra điều kiện và xuất thông tin theo từng trường hợp
- Cú pháp

```
CASE <tên cột>  
    WHEN <giá trị> THEN <biểu thức>  
    WHEN <giá trị> THEN <biểu thức>  
    ...  
    [ELSE <biểu thức>]  
END
```

## Ví dụ 23

- Cho biết mã và tên của sản phẩm có giá lớn hơn 50.000 nếu nước sản xuất là 'Viet Nam' và lớn hơn 200.000 nếu nước sản xuất là 'Singapore'

```
SELECT MASP, TENSP
FROM SANPHAM
WHERE GIA >= ( CASE NUOCSX
                WHEN 'Viet Nam' THEN 50000
                WHEN 'Singapore' THEN 200000
                END )
```

## Ví dụ 24

- Cho biết mã và giá sản phẩm sau khi tăng với tỷ lệ 5% nếu sản xuất tại 'Viet Nam' và 10% nếu sản xuất tại 'Singapore'

```
SELECT MASP,  
(CASE NUOCSX  
    WHEN 'Viet Nam' THEN GIA*1.05  
    WHEN 'Singapore' THEN GIA*1.1  
    END ) AS GIASAUKHITANG  
FROM SANPHAM
```

# Kết luận

**SELECT** <danh sách các cột>

**FROM** <danh sách các bảng>

[**WHERE** <điều kiện>]

[**GROUP BY** <các thuộc tính gom nhóm>]

[**HAVING** <điều kiện trên nhóm>]

[**ORDER BY** <các thuộc tính sắp thứ tự>]

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
- **Cập nhật dữ liệu**
  - Thêm (insert)
  - Xóa (delete)
  - Sửa (update)
- Khung nhìn (view)
- Chỉ mục (index)

# Lệnh INSERT

- Dùng để thêm 1 hay nhiều dòng vào bảng
- Để thêm dữ liệu
  - Tên quan hệ
  - Danh sách các thuộc tính cần thêm dữ liệu
  - Danh sách các giá trị tương ứng

# Lệnh INSERT (tt)

- Cú pháp (thêm 1 dòng)

```
INSERT INTO <tên bảng>(<danh sách các thuộc tính>)  
VALUES (<danh sách các giá trị>)
```

# Ví dụ

```
INSERT INTO NHANVIEN(MANV, HOTEN, DTHOAI)  
VALUES ('NV01', 'Nguyen Nhu Nhut', '0927345678')
```

```
INSERT INTO NHANVIEN(MANV, HOTEN, DTHOAI, NGVL)  
VALUES ('NV01', 'Nguyen Nhu Nhut', '0927345678', NULL)
```

```
INSERT INTO NHANVIEN  
VALUES ('NV01', 'Nguyen Nhu Nhut', '0927345678', '2006-04-13')
```



# Lệnh INSERT (tt)

## ■ Nhận xét

- Thứ tự các giá trị phải trùng với thứ tự các cột
- Có thể thêm giá trị NULL ở những thuộc tính không là khóa chính và NOT NULL
- Câu lệnh INSERT sẽ gặp lỗi nếu vi phạm RBTV
  - Khóa chính
  - Tham chiếu
  - NOT NULL - các thuộc tính có ràng buộc NOT NULL bắt buộc phải có giá trị

# Lệnh INSERT (tt)

- Cú pháp (thêm nhiều dòng)

```
INSERT INTO <tên bảng>(<danh sách các thuộc tính>)  
    <câu truy vấn con>
```

# Ví dụ

```
CREATE TABLE THONGKE_NV (  
    HOTEN VARCHAR(40),  
    SL_HD INT,  
    TRIGIAHD_TC MONEY  
)
```

```
INSERT INTO THONGKE_NV(HOTEN, SL_HD, TRIGIAHD_TC)  
SELECT HOTEN, COUNT(SOHD), SUM(TRIGIA)  
FROM HOADON HD, NHANVIEN NV  
WHERE HD.MANV=NV.MANV  
GROUP BY HOTEN
```

# Lệnh DELETE

- Dùng để xóa các dòng của bảng
- Cú pháp

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

# Ví dụ

```
DELETE FROM NHANVIEN  
WHERE HOTEN LIKE 'Tran%'
```

```
DELETE FROM NHANVIEN  
WHERE MANV='NV04'
```

```
DELETE FROM NHANVIEN
```

# Ví dụ 25

- Xóa đi những nhân viên không lập hóa đơn

```
DELETE FROM NHANVIEN  
WHERE MANV NOT IN (  
    SELECT MANV  
    FROM HOADON )
```

# Lệnh DELETE (tt)

## ■ Nhận xét

- Số lượng số dòng bị xóa phụ thuộc vào điều kiện ở mệnh đề WHERE
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị xóa
- Lệnh DELETE có thể gây ra vi phạm RB tham chiếu
  - Không cho xóa
  - Xóa luôn những dòng có giá trị đang tham chiếu đến
    - \* CASCADE
  - Đặt NULL cho những giá trị tham chiếu

# Lệnh DELETE (tt)

SOHD	NGHD	MAKH	MANV	TRIGIA
1001	23/07/2006	KH01	NV01	320,000
1002	12/08/2006	KH01	NV02	840,000
1003	23/08/2006	KH02	NV01	100,000
1004	01/09/2006	KH02	NV01	180,000
1005	20/10/2006	KH01	NV02	3,800,000
1006	16/10/2006	KH01	NV03	2,430,000

SOHD	MASP	SL
1001	TV02	10
1001	ST08	10
1002	BC04	20
1002	BB01	20
1003	BB03	10
1004	TV01	20
1005	TV05	50
1006	TV07	20



# Lệnh DELETE (tt)

MANV	HOTEN	DTHOAI	NGVL
NV01	Nguyen Nhu Nhut	0927345678	13/4/2006
NV02	Le Thi Phi Yen	0987567390	21/4/2006
NV03	Nguyen Van B	0997047382	27/4/2006
NV04	Ngo Thanh Tuan	0913758498	24/6/2006
NV05	Nguyen Thi Truc Thanh	0918590387	20/7/2006

SOHD	NGHD	MAKH	MANV	TRIGIA
1001	23/07/2006	KH01	NV01	320,000
1002	12/08/2006	KH01	NV02	840,000
1006	16/10/2006	KH01	<del>NV03</del>	2,430,000
1007	28/10/2006	KH03	<del>NV03</del>	510,000
1008	28/10/2006	KH01	<del>NV03</del>	440,000
1009	28/10/2006	KH03	NV04	200,000
1020	14/01/2007	KH09	NV04	70,000

# Lệnh UPDATE

- Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng
- Cú pháp

```
UPDATE <tên bảng>  
SET <tên thuộc tính>=<giá trị mới>,  
      <tên thuộc tính>=<giá trị mới>,  
      ...  
[WHERE <điều kiện>]
```

# Ví dụ

```
UPDATE NHANVIEN  
SET NGVL='08/12/2014'  
WHERE MANV='NV05'
```

```
UPDATE SANPHAM  
SET GIA=GIA*1.05
```

## Ví dụ 26

- Với sản phẩm có mã số 'ST10', hãy thay đổi nước sản xuất thành 'Viet Nam' và giá bán là 15.000

```
UPDATE SANPHAM
```

```
SET NUOCSX='Viet Nam', GIA=15000
```

```
WHERE MASP='ST10'
```

# Lệnh UPDATE

## ■ Nhận xét

- Những dòng thỏa điều kiện tại mệnh đề WHERE sẽ được cập nhật giá trị mới
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị cập nhật
- Lệnh UPDATE có thể gây ra vi phạm RB tham chiếu
  - Không cho sửa
  - Sửa luôn những dòng có giá trị đang tham chiếu đến
    - \* CASCADE

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- **Khung nhìn**
  - Định nghĩa
  - Truy vấn
  - Cập nhật
- Chỉ mục

# Khung nhìn

- Bảng là một quan hệ được tổ chức lưu trữ vật lý trong CSDL
- Khung nhìn cũng là một quan hệ
  - Không được lưu trữ vật lý (bảng ảo)
  - Không chứa dữ liệu
  - Được định nghĩa từ những bảng khác
  - Có thể truy vấn hay cập nhật thông qua khung nhìn

# Khung nhìn (tt)

- Tại sao phải sử dụng khung nhìn?
  - Che dấu tính phức tạp của dữ liệu
  - Đơn giản hóa các câu truy vấn
  - Hiển thị dữ liệu dưới dạng tiện dụng nhất
  - An toàn dữ liệu



# Định nghĩa khung nhìn

## ■ Cú pháp

```
CREATE VIEW <tên khung nhìn> AS  
    <câu truy vấn>
```

```
DROP VIEW <tên khung nhìn>
```

## ■ Bảng ảo này có

- Danh sách thuộc tính trùng với các thuộc tính trong mệnh đề SELECT
- Số dòng phụ thuộc vào điều kiện ở mệnh đề WHERE
- Dữ liệu được lấy từ các bảng ở mệnh đề FROM

# Ví dụ

```
CREATE VIEW NV_2006 AS
    SELECT MANV, HOTEN, NGVL
    FROM NHANVIEN
    WHERE YEAR(NGVL)=2006
```

```
CREATE VIEW TONGTG_SLHD_NV AS
    SELECT NV.MANV, HOTEN, COUNT(*) AS SLHD,
           SUM(TRIGIA) AS TONGTG
    FROM NHANVIEN NV, HOADON HD
    WHERE NV.MANV=HD.MANV
    GROUP BY NV.MANV, HOTEN
```

# Truy vấn trên khung nhìn

- Tuy không chứa dữ liệu nhưng có thể thực hiện các câu truy vấn trên khung nhìn

```
SELECT MANV  
FROM NV_2006  
WHERE HOTEN = 'Nguyen Nhu Nhut'
```

$$NV\_2006 \leftarrow \pi_{MANV, HOTEN, NGVL} (\sigma_{YEAR(NGVL)=2006} (NHANVIEN))$$
$$\pi_{MANV} (\sigma_{HOTEN='Nguyen\ Nhu\ Nhut'} (NV\_2006))$$

# Truy vấn trên khung nhìn (tt)

- Có thể viết câu truy vấn dữ liệu từ khung nhìn và bảng

```
SELECT HOTEN, SOHD, NGHD  
FROM NV_2006 NV, HOADON HD  
WHERE NV.MANV=HD.MANV
```

$$NV\_2006 \leftarrow \pi_{MANV, HOTEN} (\sigma_{YEAR(NGVL)=2006} (NHANVIEN))$$
$$TMP \leftarrow NV\_2006 \bowtie_{NV\_2006.MANV=HOADON.MANV} HOADON$$
$$\pi_{HOTEN, SOHD, NGHD} (TMP)$$

# Cập nhật trên khung nhìn

- Có thể dùng các câu lệnh INSERT, DELETE và UPDATE cho các khung nhìn đơn giản
  - Khung nhìn được xây dựng trên 1 bảng và có khóa chính của bảng
- Không thể cập nhật dữ liệu nếu
  - Khung nhìn có dùng từ khóa DISTINCT
  - Khung nhìn có sử dụng các hàm kết hợp
  - Khung nhìn có mệnh đề SELECT mở rộng
  - Khung nhìn được xây dựng từ bảng có RB trên cột
  - Khung nhìn được xây dựng từ nhiều bảng

# Cập nhật trên khung nhìn (tt)

- Sửa lại họ tên nhân viên mã 'NV03' vào làm năm 2006 là 'Phan Nhat Anh'

```
UPDATE NV_2006  
SET HOTEN='Phan Nhat Anh'  
WHERE MANV= 'NV03'
```

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- Khung nhìn (view)
- **Chỉ mục (index)**

# Chỉ mục

- Chỉ mục trên thuộc tính A là một cấu trúc dữ liệu làm cho việc tìm kiếm mẫu tin có chứa A hiệu quả hơn

SELECT \*

FROM SANPHAM

WHERE GIA>50000 AND DVT='Cay'

Đọc 10.000 bộ

Đọc 200 bộ

Bảng SANPHAM có 10.000 bộ

Có 200 sản phẩm có giá lớn hơn 50.000

Đọc 70 bộ



# Chỉ mục (tt)

## ■ Cú pháp

```
CREATE INDEX <tên chỉ mục> ON <tên bảng>(<tên cột>)
```

```
DROP INDEX <tên chỉ mục>
```

## ■ Ví dụ

```
CREATE INDEX GIA_IND ON SANPHAM(GIA)
```

```
CREATE INDEX GIA_DVT_IND ON SANPHAM(GIA, DVT)
```

# Chỉ mục (tt)

## ■ Nhận xét

- Tìm kiếm nhanh trong trường hợp so sánh với hằng số và phép kết
- Làm chậm đi các thao tác thêm, xóa và sửa
- Tốn chi phí
  - Lưu trữ chỉ mục
  - Truy xuất đĩa nhiều

## ■ Chọn lựa cài đặt chỉ mục hợp lý???

# Ví dụ

- Xét quan hệ
  - CTHD(SOHD, MASP, SL)
- Giả sử
  - CTHD được lưu trữ trong 10 block
    - Chi phí để đọc toàn bộ dữ liệu của CTHD là 10
  - Trung bình một hóa đơn có 3 sản phẩm và một sản phẩm nằm trong trong 3 chi tiết hóa đơn.
    - Dữ liệu được trải đều trong 10 block
    - Chi phí để tìm một sản phẩm hay một hóa đơn là 3
  - Khi sử dụng chỉ mục
    - Chi phí đọc hay cập nhật chỉ mục
  - Thao tác thêm cần 2 lần truy xuất đĩa

# Ví dụ (tt)

- Giả sử có 3 thao tác được thực hiện thường xuyên

- Q1

```
SELECT MASP, SL  
FROM CTHD  
WHERE SOHD='1011'
```

- Q2

```
SELECT SOHD  
FROM CTHD  
WHERE MASP='BB01' AND SL=20
```

- Q3

```
INSERT INTO CTHD  
VALUES ( '1024', 'TV03', 32)
```

# Ví dụ (tt)

## ■ Bảng so sánh chi phí

Thao tác	Không có chỉ mục	Chỉ mục trên SOHD	Chỉ mục trên MASP	Chỉ mục trên cả 2 thuộc tính
Q1	10	4	10	4
Q2	10	10	4	4
Q3	2	4	4	6
<b>Chi phí TB</b>	$2 + 8p_1 + 8p_2$	$4 + 6p_2$	$4 + 6p_1$	$6 - 2p_1 - 2p_2$

Khoảng thời gian thực hiện Q1 là  $p_1$

Khoảng thời gian thực hiện Q2 là  $p_2$

Khoảng thời gian thực hiện Q3 là  $1 - p_1 - p_2$

# Bài tập về nhà

## ■ Bài tập

- 6.1.1 đến 6.1.4 / 252 - 253
- 6.2.1 đến 6.2.3 / 262 - 263
- 6.3.1 đến 6.3.3 / 274 - 275
- 6.3.7 đến 6.3.10 / 275 - 276
- 6.4.1, 6.4.2, 6.4.6, 6.4.7 / 284 - 285
- 6.5.1, 6.5.2 / 290 - 291
- 6.7.1 đến 6.7.6 / 311 - 312

