

Quản trị Cơ sở dữ liệu và Phần mềm ứng dụng

Bộ môn CNTT – TMĐT
Khoa Thương mại điện tử



Chương III

Ngôn ngữ SQL



Chương III: Ngôn ngữ SQL

1. Đại số quan hệ và ngôn ngữ SQL
2. Lệnh định nghĩa dữ liệu
3. Lệnh cập nhật dữ liệu
4. Lệnh truy vấn dữ liệu



Chương III

1. Đại số quan hệ và ngôn ngữ SQL

1.1. Đại số quan hệ

1.2. Ngôn ngữ SQL

2. Lệnh định nghĩa dữ liệu

3. Lệnh cập nhật dữ liệu

4. Lệnh truy vấn dữ liệu



1.1.Đại số quan hệ

- Đại số quan hệ cung cấp tám phép toán tác động trên các quan hệ và cho kết quả cũng là một quan hệ gồm:
 - Các phép toán tập hợp: Hợp, trừ, giao, tích Đề các.
 - Các phép toán quan hệ: Chọn, chiếu, kết nối, chia.



Đại số quan hệ(t)

○ Quan hệ khả hợp

- Định nghĩa: Hai quan hệ là khả hợp nếu chúng được xác định trên cùng tập thuộc tính và các thuộc tính cùng tên có cùng miền giá trị.

a. Phép hợp (union)

- Định nghĩa:

- *Phép hợp của hai quan hệ khả hợp r và s , ký hiệu là $r \cup s$, là tập tất cả các bộ thuộc r hoặc thuộc s hoặc thuộc cả hai quan hệ.*

- Ví dụ:

A	B	C
a1	b1	c1
a1	b1	c2
a1	b2	c2
a2	b2	c2
a3	b2	c2

 \cup

A	B	C
a1	b1	c1
a1	b2	c1
a1	b2	c2

 =

A	B	C
a1	b1	c1
a1	b1	c2
a1	b2	c1
a1	b2	c2
a2	b2	c2
a3	b2	c2

b. Phép giao

- Định nghĩa:

- *Phép giao của hai quan hệ khả hợp r và s , ký hiệu là $r \cap s$, là tập tất cả các bộ thuộc cả hai quan hệ r và s .*

- Ví dụ:

A	B	C
a1	b1	c1
a1	b1	c2
a1	b2	c2
a2	b2	c2
a3	b2	c2

 \cap

A	B	C
a1	b1	c1
a1	b2	c1
a1	b2	c2

 =

A	B	C
a1	b1	c1
a1	b2	c2

c. Phép trừ

- Định nghĩa:

- *Phép trừ của hai quan hệ khả hợp r và s , ký hiệu là $r-s$, là tập tất cả các bộ thuộc r nhưng không thuộc s .*

- Ví dụ:

A	B	C
a1	b1	c1
a1	b1	c2
a1	b2	c2
a2	b2	c2
a3	b2	c2

 -

A	B	C
a1	b1	c1
a1	b2	c1
a1	b2	c2

 =

A	B	C
a1	b1	c2
a2	b2	c2
a3	b2	c2



d. Phép tích Đề các

○ Định nghĩa:

- Cho quan hệ r xác định trên tập thuộc tính $\{A_1, \dots, A_n\}$ và quan hệ s xác định trên tập thuộc tính $\{B_1, \dots, B_m\}$. Tích Đề các của hai quan hệ r và s ký hiệu là $r \times s$ là tập tất cả các $(m+n)$ -bộ có n thành phần đầu tiên là một bộ thuộc r và m thành phần sau là một bộ thuộc s .

Phép tích Đề các(t)

- Ví dụ:

A	B	C
a1	b1	1
a2	b2	2
a3	b3	3

 ×

D	E
1	e1
2	e2
3	e3

 =

A	B	C	D	E
a1	b1	1	1	e1
a1	b1	1	2	e2
a1	b1	1	3	e3
a2	b2	2	1	e1
a2	b2	2	2	e2
a2	b2	2	3	e3
a3	b3	3	1	e1
a3	b3	3	2	e2
a3	b3	3	3	e3

e. Phép chiếu

- Định nghĩa;

- Cho quan hệ r xác định trên tập thuộc tính $U = \{A_1, \dots, A_n\}$. $X \in U$. Phép chiếu của r trên tập thuộc tính X , ký hiệu là $\Pi_X(r)$, là tập các bộ của r xác định trên X .

- Ví dụ:

- $\Pi_{A,B}(r) =$

A	B
a1	b1
a1	b2
a2	b2
a3	b2

 $\Pi_C(s) =$

C
c1
c2



-
- Phép chiếu trên một quan hệ thực chất là phép toán loại bỏ đi một số thuộc tính và chỉ giữ lại những thuộc tính còn lại của quan hệ đó.

f. Phép chọn

- Định nghĩa:

- Cho r là một quan hệ, F là một biểu thức điều kiện. Phép chọn trên r với biểu thức chọn F , ký hiệu $\sigma_F(r)$, là tập tất cả các bộ của r thỏa mãn điều kiện F .

- Ví dụ:

$$\sigma_{A = a1}(r) =$$

A	B	C
a1	b1	c1
a1	b2	c2
a1	b1	c2

$$\sigma_{A = a1 \wedge C = c2}(r) =$$

A	B	C
a1	b1	c2
a1	b2	c2



-
- Phép chọn là phép toán lọc ra một tập con các bộ của quan hệ đã cho thỏa mãn một điều kiện xác định. Điều kiện đó được gọi là điều kiện chọn hay biểu thức chọn.

g. Phép chia

○ Định nghĩa:

- Cho hai quan hệ r xác định trên n thuộc tính, s xác định trên m thuộc tính ($n > m$, $s \neq \emptyset$), phép chia quan hệ r cho quan hệ s , ký hiệu $r \div s$, là các bộ t sao cho với mọi bộ v thuộc s thì t ghép với v thuộc r .

○ Ví dụ:

A	B
a1	b1
a1	b2
a1	b3
a2	b4

 \div

B
b1
b2
b3

 $=$

A
a1

Bài giảng CSDL và Phần mềm ứng dụng

h. Phép kết nối

- Khái niệm:

- *Phép kết nối hai quan hệ r, s , ký hiệu $r \bowtie s$ là phép ghép các cặp bộ của hai quan hệ thỏa mãn một điều kiện kết nối hay một biểu thức kết nối F .*

- Ví dụ: $F = (C \leq D)$

A	B	C	D	E
a1	b1	1	1	e1
a1	b1	1	2	e2
a1	b1	1	3	e3
a2	b2	2	2	e2
a2	b2	2	3	e3
a3	b3	3	3	e3

Bài giảng CSDL và Phần mềm ứng dụng



Biểu diễn câu hỏi bằng đại số quan hệ



III.1.2. Ngôn ngữ SQL

- SQL (Structured Query Language) : Ngôn ngữ truy vấn có cấu trúc
 - Tiền thân SEQUEL, IBM phát triển cho hệ CSDL thử nghiệm System/R, 1974
 - 1986, ANSI/ISO công nhận một chuẩn ngôn ngữ sử dụng trên csdl quan hệ.
 - SQL_86 (ANSI)
 - SQL_89 (sửa đổi nhỏ)
 - **SQL_92 (chuẩn hiện tại, ANSI/ISO)**
 - *SQL_99 (mở rộng – đối tượng)*
 - *SQL_2003 (hỗ trợ XML và nhiều tính năng khác)*
 - *SQL_2006 ...*



-
- Các hệ quản trị cơ sở dữ liệu quan hệ thương mại hiện có như Oracle, SQL Server, Informix, DB2,... đều chọn SQL làm ngôn ngữ cho sản phẩm của mình.
 - SQL cài đặt trong các hệ quản trị CSDL thương mại có một số khác biệt so với SQL do ANSI/ISO đề xuất.
 - Các câu lệnh SQL cung cấp có thể được nhúng vào trong các ngôn ngữ lập trình nhằm xây dựng các ứng dụng tương tác với cơ sở dữ liệu.



a. Các thành phần

- Ngôn ngữ định nghĩa dữ liệu (DDL)
 - Định nghĩa/thay đổi/ xóa lược đồ quan hệ
 - Định nghĩa khung nhìn
 - Đặc tả quyền truy nhập
 - Đặc tả ràng buộc toàn vẹn
- Ngôn ngữ thao tác dữ liệu (DML)
 - Thêm/xóa/sửa/truy vấn các bộ giá trị trong quan hệ
- Ngôn ngữ điều khiển giao dịch
 - Đặc tả sự bắt đầu và kết thúc giao dịch
 - Điều khiển tương tranh



b. Các dạng quan hệ

- Bảng cơ sở
 - Quan hệ được lưu trữ vật lý trong csdl
- Kết quả truy vấn
 - Kết quả truy vấn trên các quan hệ là một quan hệ.
- Khung nhìn
 - Quan hệ ảo được định nghĩa bởi một biểu thức truy vấn



Chương III

1. Đại số quan hệ và ngôn ngữ SQL
2. Lệnh định nghĩa dữ liệu
 - 2.1. Tạo csdl, xóa csdl
 - 2.2. Tạo bảng
 - 2.3. Sửa đổi cấu trúc bảng
 - 2.4. Xoá bảng
 - 2.5. Tạo khung nhìn
3. Lệnh cập nhật dữ liệu
4. Lệnh truy vấn dữ liệu



2.1. Tạo csdl, xóa csdl

- Tạo csdl
 - CREATE DATABASE <tên cơ sở dữ liệu>
- Xóa csdl
 - DROP DATABASE <tên cơ sở dữ liệu>

2.2. Tạo bảng

- **Cú pháp**

```
CREATE TABLE <Tên bảng>  
(<Tên cột> <Kiểu dữ liệu> [NOT NULL],...  
[CONSTRAINT <Tên ràng buộc> <Kiểu ràng buộc>])
```

- **Ví dụ:** *Tạo bảng về khách hàng có các trường là mã khách hàng, tên khách hàng, địa chỉ, số điện thoại và khóa chính là mã khách hàng.*

```
CREATE TABLE KHACH_HANG  
(MaKH char(5) NOT NULL,  
TenCongTy varchar(40),  
DiaChi varchar (60),  
DienThoaiCD char(7),  
CONSTRAINT khoa_chinh  
PRIMARY KEY (MaKH))
```

2.3. Sửa đổi cấu trúc bảng

	Cú pháp	Ví dụ
Thêm cột	ALTER TABLE <Tên bảng> ADD COLUMN <Tên cột> <kiểu dữ liệu>	<i>Thêm cột giá vào bảng CUNG_UNG ứng</i> ALTER TABLE CUNG_UNG ADD COLUMN Gia int
Xóa cột	ALTER TABLE <Tên bảng> DROP COLUMN <Tên cột>	<i>Xóa cột giá trong bảng CUNG_UNG</i> ALTER TABLE CUNG_UNG DROP COLUMN Gia
Sửa kiểu dữ liệu của cột	ALTER TABLE <Tên bảng> CHANGE COLUMN <Tên cột> <kiểu dữ liệu mới>	<i>Sửa đổi kiểu dữ liệu của cột giá trong bảng CUNG_UNG là số thực</i> ALTER TABLE CUNG_UNG CHANGE COLUMN Gia real

Sửa đổi cấu trúc bảng(t)

	Cú pháp	Ví dụ
Thêm ràng buộc	ALTER TABLE <Tên bảng> ADD CONSTRAINT <Tên ràng buộc> <kiểu ràng buộc>	<i>Thêm một ràng buộc khóa ngoài vào bảng CUNG_UNG, cột MaNCC tham chiếu tới mã MaNCC của bảng NHA_CUNG_CAP</i> ALTER TABLE CUNG_UNG ADD CONSTRAINT khoa_ngoai FOREIGN KEY (MaNCC) REFERENCES NHA_CUNG_CAP(MaNCC)
Xóa ràng buộc	ALTER TABLE <Tên bảng> DROP CONSTRAINT <Tên ràng buộc>	<i>Xóa ràng buộc khóa ngoài trong bảng Sự cung ứng</i> ALTER TABLE CUNG_UNG DROP CONSTRAINT khoa_ngoai



2.4. Xóa bảng

- Cú pháp
 - **DROP TABLE** <Tên bảng>
- Ví dụ: *Xóa bảng KHACH_HANG*
 - **DROP TABLE** KHACH_HANG



2.5. Tạo khung nhìn

- Cú pháp:
 - **CREAT VIEW** <tên khung nhìn> [(danh sách tên cột)] **AS** <Câu truy vấn>
- Ví dụ
 - Bảng cơ sở:
 - NHAN_VIEN(MaNv, TenNV, DiaChi, MaPhong, Luong, NamLenLuong, DanhGia)
 - PHONG_BAN(MaPhong, TenPhong, NguoiQuanLy)



Tạo khung nhìn(t)

- Khung nhìn: phục vụ cho đối tượng là trưởng phòng của phòng kỹ thuật
 - **CREAT VIEW KT**(TênNV, Đchỉ, Lương, NămLênlương, Đánh giá) **AS**
SELECT TenNV, DiaChi, MaPhong, Luong, NamLenLuong, DanhGia
FROM NHAN_VIEN
WHERE MaPhong IN
SELECT MaPhong FROM PHONG_BAN
WHERE TenPhong = 'Kỹ thuật'



Chương III

1. Đại số quan hệ và ngôn ngữ SQL
2. Lệnh định nghĩa dữ liệu
3. Lệnh cập nhật dữ liệu
 - 3.1. Thêm một bộ giá trị
 - 3.2. Xóa bộ giá trị
 - 3.3. Thay đổi thuộc tính các bộ
4. Lệnh truy vấn dữ liệu



3.1. Thêm bộ giá trị

- Cú pháp:
 - **INSERT INTO** <Tên bảng> [(danh sách tên cột)] { **VALUES** (các giá trị) | <Câu truy vấn> }
- Ví dụ: *Thêm một bộ giá trị vào bảng NHA_CUNG_CAP.*
 - **INSERT INTO** NHA_CUNG_CAP (MaNCC, TenCongTy, DiaChi) **VALUES** ('S1', 'Hải Hà', 'Hà Nội')



3.2. Xóa bộ giá trị

- Cú pháp:
 - **DELETE FROM** { <Tên bảng| Tên khung nhìn> } [**WHERE** <Biểu thức điều kiện>]

- Ví dụ:

Xóa các bộ có mã nhà cung cấp = S1 tại hai bảng CUNG_UNG và NHA_CUNG_CAP

- **DELETE * FROM CUNG_UNG**
WHERE MaNCC = 'S1'
- **DELETE * FROM NHA_CUNG_CAP**
WHERE MaNCC = 'S1'



3.3. Thay đổi giá trị thuộc tính các bộ

- Cú pháp:

- **UPDATE** <Tên bảng> **SET** <Tên cột >
= <biểu thức> [**WHERE** <Biểu thức
điều kiện>]

- Ví dụ:

Thay đổi thuộc tính tên của bộ giá trị có MaHang là = P1 trong bảng HANG_HOA.

- **UPDATE** Items **SET** TenHang = 'Bánh ngọt' **WHERE** MaHang= 'P1'



Lược đồ csdl quan hệ của siêu thị M

- EMPS(**ENAME**, SALARY, DNAME)
- DEPTS(**DNAME**, *DEPT#*, MGR)
- ITEMS(**INAME**, *ITEM#*, DNAME)
- CUSTOMERS(**CNAME**, CADDR, BALANCE)
- SUPPLIERS(**SNAME**, SADDR)
- ORDERS(**O#**, DATE, CNAME)
- SUPPLIES(**SNAME**, **INAME**, PRICE)
- INCLUDES(**O#**, **INAME**, QUANTITY)



Lược đồ csdl quan hệ của siêu thị M

- NHAN_VIEN (TenNV, Luong, TenPhong)
- PHONG_BAN(TenPhong, MaPhong, NguoiQuanLy)
- MAT_HANG(TenHang, MaHang, TenPhong)
- KHACH_HANG(TenKH, DiaChi, SoDuTK)
- NHA_CUNG_CAP(TenNCC, DiaChi)
- DON_DAT_HANG(MaDDH, NgayLap, TenKH)
- CUNG_UNG(TenNCC, TenHang, Gia)
- CHI_TIET_DON_HANG(MaDDH, TenHang, SoLuong)



Thay đổi


- Nhân viên được quản lý theo mã nhân viên.
- Phòng ban được quản lý theo mã phòng ban.
- Mặt hàng được quản lý theo mã hàng.
- Khách hàng được quản lý theo mã khách hàng.
- Nhà cung cấp được quản lý theo mã nhà cung cấp.
- Các mặt hàng phân thành nhiều nhóm hàng.
- Mỗi gian hàng quản lý/bán một nhóm hàng.



Lược đồ csdl quan hệ mới của siêu thị M

- NHAN_VIEN (MaNV, Ho, Ten, Luong, MaPhong,...)
- PHONG_BAN(MaPhong, TenPhong, NguoiQL,...)
- MAT_HANG(MaHang, TenHang, MaNhom,...)
- NHOM_HANG(MaNhom, TenNhom, MaPhong,...)
- KHACH_HANG(MaKH, TenKH, DiaChi, SoDuTK,...)
- NHA_CUNG_CAP(MaNCC, TenNCC, DiaChi,...)
- DON_DAT_HANG(MaDDH, NgayLap, MaKH,...)
- CUNG_UNG(MaNCC, MaHang, Gia,...)
- CHI_TIET_DON_HANG(MaDDH, MaHang, SoLuong,...)

Chú thích: "...” lược đồ có thể bổ sung thêm một số thuộc tính khác (không quan trọng)



Tạo cơ sở dữ liệu cho siêu thị M theo lược đồ đã được thay đổi

- Bảng PHONG_BAN

```
CREATE TABLE PHONG_BAN  
(MaPhong char(5) not null,  
TenPhong nvarchar(100),  
NguoiQL char(5),  
CONSTRAINT khoa_chinhP  
PRIMARYKEY (MaPhong))
```



Bảng NHAN_VIEN

```
CREATE TABLE NHAN_VIEN
(MaNv char(7) NOT NULL,
Ho nvarchar(50),
Ten nvarchar(50),
Luong real,
MaPhong char(5),
CONSTRAINT khoa_chinhNV PRIMARYKEY(MaNv),
CONSTRAINT khoa_ngoaiNP FOREIGNKEY(MaPhong)
REFERENCES PHONG_BAN(MaPhong)
ON DELETE CASCADE
ON UPDATE CASCADE)
```




Bảng KHACH_HANG

```
CREATE TABLE KHACH_HANG  
(MaKH char(5) NOT NULL,  
TenCongTy nvarchar(100),  
DiaChi nvarchar(100),  
CONSTRAINT khoa_chinhK  
PRIMARY KEY (MaKH))
```



Bảng NHA_CUNG_CAP

```
CREATE TABLE NHA_CUNG_CAP  
(MaNCC char(5) NOT NULL,  
TenCongTy nvarchar(100),  
DiaChi nvarchar(100),  
CONSTRAINT Khoa_chinhC  
PRIMARY KEY (MaNCC))
```



Bảng NHOM_HANG

```
CREATE TABLE NHOM_HANG
(MaNhom char(5) NOT NULL,
TenNhom nvarchar(100),
MaPhong char(5),
CONSTRAINT Khoa_chinhNH
PRIMARY KEY (MaNhom),
CONSTRAINT duy_nhatNH
UNIQUE (MaPhong),
CONSTRAINT Khóa_ngoaiNH
FOREIGN KEY (MaPhong) REFERENCES
PHONG_BAN(MaPhong))
```



Bảng MAT_HANG

```
CREATE TABLE MAT_HANG
(MaHang char(5) NOT NULL,
TenHang nvarchar(100),
NhomHang char(5),
CONSTRAINT Khoa_chinhH
PRIMARY KEY (MaHang),
CONSTRAINT Khóa_ngoaiH
FOREIGN KEY (NhomHang) REFERENCES
    NHOM_HANG(MaNhom)
ON UPDATE CASCADE
ON DELETE CASCADE)
```



Bảng DON_DAT_HANG

```
CREATE TABLE DON_DAT_HANG
(SoHieuDH char(7) NOT NULL,
MaKH char(5),
NgàyDat datetime,
CONSTRAINT Khoa_chinhD
PRIMARY KEY (SoHieuDH),
CONSTRAINT Khoa_ngoaiD
FOREIGN KEY (MaKH) REFERENCES
    KHACH_HANG(MaKH)
ON UPDATE CASCADE
ON DELETE CASCADE)
```



Bảng CHI_TIET_DON_HANG

```
CREATE TABLE CHI_TIET_DH
(SoHieuDH char(7) NOT NULL,
MaHang char(5) NOT NULL,
SoLuong int,
Gia real,
CONSTRAINT Khoa_chinhCT
PRIMARY KEY (SoHieuDH, MaHang),
CONSTRAINT Khóa_ngoaiCD
FOREIGN KEY (SoHieuDH) REFERENCES DON_DAT_HANG(SoHieuDH)
ON UPDATE CASCADE
ON DELETE CASCADE,
CONSTRAINT Khóa_ngoaiCM
FOREIGN KEY (MaHang) REFERENCES MAT_HANG(MaHang)
ON UPDATE CASCADE
ON DELETE CASCADE)
```



Bảng CUNG_UNG

```
CREATE TABLE CUNG_UNG
(MaNCC char(5) NOT NULL,
MaHang char(5) NOT NULL,
Gia real,
CONSTRAINT khoa_chinhCC
PRIMARY KEY (MaNCC, MaHang),
CONSTRAINT khoa_ngoaiCC
FOREIGN KEY (MaNCC) REFERENCES NHA_CUNG_CAP(MaNCC)
ON UPDATE CASCADE
ON DELETE CASCADE,
CONSTRAINT khoa_ngoaiCH
FOREIGN KEY (MaHang) REFERENCES MAT_HANG(MaHang)
ON UPDATE CASCADE
ON DELETE CASCADE)
```



Thay đổi ràng buộc trong bảng PHONG_BAN

Tạo một ràng buộc sao cho cột NguoiQL nhận những giá trị không trùng lặp.

Tạo một ràng buộc khóa ngoài giữa cột NguoiQL và cột MaNV trong bảng NHAN_VIEN

→ Tạo mối liên kết 1- 1 giữa hai bảng PHONG_BAN và NHAN_VIEN với các trường(cột) liên kết là NguoiQL và MaNV

```
ALTER TABLE PHONG_BAN ADD  
CONSTRAINT duy_nhat  
    UNIQUE(NguoiQL),  
CONSTRAINT Khoa_ngoai  
FOREIGN KEY (NguoiQL)  
REFERENCES NHAN_VIEN(MaNV)
```




Chương III

1. Đại số quan hệ và ngôn ngữ SQL
2. Lệnh định nghĩa dữ liệu
3. Lệnh thao tác dữ liệu
4. Lệnh truy vấn dữ liệu
 - 4.1. Mệnh đề truy vấn tổng quát
 - 4.2. Câu truy vấn lồng nhau



4.1. Mệnh đề truy vấn tổng quát

SELECT[**DISTINCT**] < danh sách cột > | * | < biểu thức số học >

FROM < danh sách tên bảng > | < danh sách các Khung nhìn >

[**WHERE** < biểu thức điều kiện >]

[**GROUP BY** < danh sách tên cột >]

[**HAVING** < biểu thức điều kiện >]

[**ORDER BY** < danh sách tên cột > | < biểu thức >

[**ASC**|**DESC**]

[**UNION**|**INTERSECT**|**MINUS** < Câu truy vấn >]



a. Mệnh đề **SELECT**


- Mệnh đề **SELECT** tương ứng với phép chiếu trong đại số quan hệ, được sử dụng để liệt kê các thuộc tính mong muốn
- Mệnh đề **SELECT** cho phép:
 - Lựa chọn một/nhiều/tất cả (*) các thuộc tính
 - *Hiển thị tất cả thông tin của nhân viên trong bảng NHAN_VIEN*
`SELECT * FROM NHAN_VIEN`
 - Lấy các bộ giá trị không trùng nhau (**DISTINCT**)
 - *Cho biết họ của các nhân viên*
`SELECT DISTINCT Ho
FROM NHAN_VIEN`



Mệnh đề SELECT(t)

- Lấy kết quả từ các biểu thức số học (phép tính +, -, *, /) của hằng hoặc các thuộc tính
 - *Cho biết danh mục mặt hàng và giá của mặt hàng đó (tính theo việt nam đồng):*

```
SELECT MAT_HANG.TenHang,  
CUNG_UNG.Gia * 16.02  
FROM MAT_HANG, CUNG_UNG  
WHERE MAT_HANG.MaHang =  
CUNG_UNG.MaHang
```
 - Kết quả của biểu thức số học là NULL nếu một giá trị đầu vào là NULL.

- 
-
- SQL sử dụng cách viết <tên quan hệ>.<tên thuộc tính> để che dấu tính lập lờ trong trường hợp tên các thuộc tính trong các quan hệ trùng nhau.
 - SQL sử dụng các phép nối logic NOT, AND OR. Toán hạng của các phép nối logic có thể là các biểu thức chứa các toán tử so sánh >, >=, <>, <, <=.
 - Toán tử BETWEEN được dùng để chỉ các giá trị nằm giữa các khoảng giá trị.



b. Mệnh đề FROM

- Mệnh đề **FROM** tương ứng với phép tích Đề các của các quan hệ được xét.
- Ví dụ: Tìm giá mua vào của các mặt hàng thuộc loại máy tính xách tay:

```
SELECT MAT_HANG.TenHang,  
CUNG_UNG.Gia*16.02  
FROM CUNG_UNG, NHOM_HANG, MAT_HANG  
WHERE  
((MAT_HANG.MaHang=CUNG_UNG.MaHang)  
AND (NHOM_HANG.MaNhom =  
MAT_HANG.MaNhom) AND (TenNhom='laptop'))
```



c. Mệnh đề WHERE

- Mệnh đề **WHERE** tương ứng chọn dựa trên các thuộc tính của các quan hệ xuất hiện sau FROM.
- Ví dụ: 1. Tìm các mặt hàng có số lượng đặt hàng lớn hơn 1 trăm

```
SELECT MAT_HANG.TenHang,  
       CHI_TIET_DH.SoLuong  
FROM   CHI_TIET_DH, MAT_HANG  
WHERE  (MAT_HANG.MaHang =  
        CHI_TIET_DH.MaHang) AND (SoLuong > 100)
```



Mệnh đề WHERE (t)

- Kết quả của biểu thức điều kiện là TRUE hoặc FALSE
- Nếu 1 trong những đầu vào của biểu thức điều kiện có giá trị NULL thì kết quả trả về là FALSE
 - WHERE[NOT] <biểu thức> <phép so sánh> <biểu thức>
 - WHERE[NOT] <biểu thức logic>
 - WHERE[NOT] <biểu thức logic> {AND|OR} [NOT] <biểu thức logic>
 - WHERE[NOT] <tên cột> [NOT] LIKE <xâu ký tự mẫu>



Mệnh đề WHERE(t)

- WHERE[NOT] <biểu thức> [NOT] IN ({danh sách | câu truy vấn})
- WHERE[NOT] EXISTS (<câu truy vấn con>)
- WHERE[NOT] <biểu thức> <phép so sánh> {SOME|ANY|ALL (Câu truy vấn con)}
- WHERE <cột> IS NULL



Các phép đổi tên

- Mệnh đề AS cho phép đổi tên các bảng hoặc các cột
 - < tên cũ > AS < tên mới >
- Đổi tên cột:
 - Các bảng trong mệnh đề FROM cùng tên
 - Sử dụng biểu thức số học trong câu lệnh SELECT
 - Thay đổi tên cột trong bảng kết quả
 - Ví dụ: *Đổi tên cột dữ liệu hiển thị*
SELECT MAT_HANG.TenHang,
CUNG_UNG.Gia*16.02 AS GiaSanPham
FROM CUNG_UNG, NHOM_HANG, MAT_HANG
WHERE ((MAT_HANG.MaHang=CUNG_UNG.MaHang)
AND (NHOM_HANG.MaNhom = MAT_HANG.MaNhom)
AND (TenNhom='laptop'))



Các phép đổi tên(t)

- Đổi tên bảng

- So sánh các bộ (dòng) trong cùng một bảng

- Ví dụ: *Lấy tên của các hãng cung ứng có cung ứng các sản phẩm màn hình LCD + TV (mã MH013) rẻ hơn hãng hula (mã là CHULA).*

- SELECT T.MaNCC

- FROM CUNG_UNG AS T, CUNG_UNG AS S

- WHERE T.MaHang='MH013' AND T.Gia < S.Gia AND S.MaNCC = 'CHULA' AND

- S.MaHang = 'MH013'



Các phép toán trên chuỗi

- Toán tử **LIKE** và **NOT LIKE** trong mệnh đề WHERE cho phép đối sánh giá trị của một cột với một mẫu khi tìm kiếm. Mẫu có thể được biểu diễn bởi các ký tự thay thế:
 - Ký tự (%): Tương ứng với một chuỗi con bất kỳ.
 - Ký tự (_): Tương ứng với một ký tự bất kỳ.
 - Nếu trong xâu mẫu có chứa các ký tự thay thế (% , _ , \) thì đặt ký tự \ trước các ký tự đó.



Các phép toán trên chuỗi(t)

- Ví dụ: Mẫu
 - '% Nam An %': Tương ứng với các xâu ký tự có chứa xâu con là 'Nam An'.
 - '___%': Tương ứng với các xâu ký tự có ít nhất hai ký tự.
 - '20\\%%': Tương ứng với các xâu ký tự bắt đầu bằng 20%
- Ví dụ: *Tìm tên chính xác của nhân viên tiếp xúc phía khách hàng có chứa xâu 'Hoa'.*
 - ```
SELECT NguoiLL
FROM KHACH_HANG
WHERE NguoiLL LIKE '%Hoa%'
```



# Các hàm thư viện

---

- Hàm thư viện lấy một tập các giá trị làm đầu vào và trả kết quả là một giá trị đơn. Các hàm tính gộp:
  - Tính trung bình:
    - `AVG([DISTINCT] <cột có kiểu dữ liệu số>)`
  - Tính tổng
    - `SUM([DISTINCT] <cột có kiểu dữ liệu số>)`
  - Tính Min, Max
    - `Min(<[DISTINCT] > <cột >`
  - Đếm
    - `COUNT([DISTINCT] <cột> | *)`



## Các hàm thư viện(t)

---

- Tất cả các hàm tính gộp (trừ Count(\*) bỏ qua các giá trị đầu vào NULL

- COUNT() = 0

- Ví dụ: *Kiểm kê lượng hàng hoá theo loại mặt hàng còn trong kho*

```
SELECT MaNhom, Sum(TonKho)
FROM MAT_HANG
GROUP BY MaNhom
```



## d. Mệnh đề **GROUP BY** và **HAVING**

---

- Mệnh đề **GROUP BY** nhóm các bộ có cùng giá trị trên các thuộc tính nào đó

Ví dụ: In danh sách loại hàng hóa cùng lượng hàng hoá theo loại mặt hàng còn trong kho với điều kiện lượng hàng lớn hơn 100.

```
SELECT MaNhom, Sum(TonKho)
```

```
FROM MAT_HANG
```

```
GROUP BY MaNhom
```

```
HAVING Sum(TonKho) > 100
```

- Mệnh đề **HAVING** xuất hiện sau khi tạo nhóm, đưa ra điều kiện cho nhóm.





## Mệnh đề GROUP BY và HAVING(t)

---

- Nếu WHERE và HAVING cùng ở trong một câu truy vấn thì:
  - Biểu thức điều kiện trong WHERE sẽ được thực hiện trước.
  - Các bộ thỏa mãn điều kiện trong WHERE sẽ được nhóm vào bởi GROUPBY.
  - Mệnh đề HAVING (nếu có) sẽ được áp dụng trên mỗi nhóm. Các nhóm không thỏa mãn mệnh đề HAVING sẽ bị xóa bỏ.



## e. Mệnh đề ORDER BY

---

- Mệnh đề **ORDER BY** cho phép trình bày kết quả câu truy vấn theo thứ tự.
  - Mặc định liệt kê theo thứ tự tăng
- Ví dụ: Đưa ra danh sách họ tên nhân viên với mức lương từ cao tới thấp.

```
SELECT Ho, Ten, Luong
FROM NHAN_VIEN
ORDER BY Luong DESC
```



## f. Tìm kiếm nhờ các phép toán tập hợp

---

- Phép UNION tương ứng với phép hợp trong đại số quan hệ.
  - Ví dụ: Tìm các hãng là đối tác của siêu thị (khách hàng hoặc nhà cung cấp)
    - (SELECT TenCongTy  
FROM KHACH\_HANG)  
UNION  
(SELECT TenCongTy  
FROM NHA\_CUNG\_CAP)



## Tìm kiếm nhờ các phép toán tập hợp(t)

---

- Phép INTERSECT tương ứng với phép giao trong đại số quan hệ
  - Ví dụ: *Tìm các hãng vừa là khách hàng vừa là nhà cung cấp cho siêu thị.*
    - ```
(SELECT TenCongTy
FROM KHACH_HANG)
INTERSECT
(SELECT TenCongTy
FROM NHA_CUNG_CAP)
```



Tìm kiếm nhờ phép toán tập hợp(t)

- Phép MINUS tương ứng với phép trừ trong đại số quan hệ

- Ví dụ: Tìm các hãng chỉ là khách hàng (không là nhà cung cấp) của siêu thị.

```
(SELECT MaKH
FROM KHACH_HANG)
MINUS
(SELECT MaKH
FROM CUNG_UNG)
```



4.2. Các câu truy vấn lồng nhau

- Câu truy vấn lồng là một câu truy vấn có chứa câu truy vấn con.
- Câu truy vấn con là một biểu thức truy vấn (**SELECT-FROM-WHERE**) lồng trong một truy vấn khác như:
 - **SELECT**
 - **INSERT..INTO**
 - **UPDATE**
 - **DELETE**
 - Câu truy vấn con khác



Các câu truy vấn lồng nhau(t)

- Cú pháp tạo truy vấn con
 - i) <biểu thức> [**NOT**] **IN** (<câu truy vấn con>)
 - ii) [**NOT**] **EXISTS** (<câu truy vấn con>)
 - iii) <biểu thức> <phép so sánh> {**SOME**|**ANY**|**ALL**} (<câu truy vấn con>)



Mệnh đề WHERE

- WHERE[NOT] <biểu thức> [NOT] IN ({ danh sách | câu truy vấn con })
- WHERE[NOT] EXISTS (<câu truy vấn con>)
- WHERE[NOT] <biểu thức> <phép so sánh> { SOME | ANY | ALL } (<câu truy vấn con>)



**a. <biểu thức> [NOT] IN
(<Câu truy vấn con>)**

Xác định kết quả trong truy vấn cha (không) phải thuộc tập hợp các bộ là kết quả của truy vấn con.



Ví dụ

- **IN**

- Ví dụ: Tìm những MaHang từ bảng CUNG_UNG mà mặt hàng đó có lượng lưu kho >0.

- SELECT MaHang
FROM CUNG_UNG
WHERE MaHang **IN**
(Select MaHang from HANG_HOA where
TonKho >0)

- **NOT IN**

- Ví dụ: Tìm những MaHang từ bảng CUNG_UNG mà mặt hàng đó có lượng lưu kho <0.

- SELECT MaHang
FROM CUNG_UNG
WHERE MaHang **NOT IN**
(Select MaHang from HANG_HOA where
TonKho >0)




b. [NOT] EXISTS (<Câu truy vấn con>)

Kiểm tra quan hệ là rỗng hay không. Nếu kết quả trả về của truy vấn con chứa ít nhất một dòng thì điều kiện tồn tại thoả mãn.



Ví dụ trong câu lệnh SELECT

- *Hiển thị thông tin của những nhà cung cấp cung cấp ít nhất một sản phẩm*
- SELECT *
FROM NHA_CUNG_CAP
WHERE **EXISTS**
(select *
from CUNG_UNG
where NHA_CUNG_CAP.MaNCC =
CUNG_UNG.MaNCC

- 
-
- *Hiển thị thông tin của những khách hàng đã có yêu cầu về hàng hoá*

- SELECT *

FROM KHACH_HANG

WHERE EXISTS

(Select MaKH

from DON_DAT_HANG

Where KHACH_HANG.MaKH=

DONG_DAT_HANG.MaKH)




Ví dụ trong câu lệnh DELETE

- *Xóa trong bảng Sự cung ứng (nếu có) tất cả những nhà cung cấp vừa là nhà cung cấp vừa là là khách hàng.*
- DELETE *
FROM NHA_CUNG_CAP
WHERE **EXISTS**
(select *
from KHACH_HANG
where KHACH_HANG.MaKH =
NHA_CUNG_CAP.MaNCC)



Ví dụ trong câu lệnh INSERT

- *Thêm một bộ giá trị vào bảng Customer và kiểm tra xem mã của khách hàng = 'CQTHG' đã tồn tại hay chưa.*
- INSERT INTO KHACH_HANG
(MaKH, TenCongty, ThanhPho)
SELECT DISTINCT 'CQTHG', 'Công ty Quốc tế
Hoàng Gia', 'Hà Nội'
FROM KHACH_HANG
WHERE **NOT EXISTS**
(select *
from KHACH_HANG
where MaKH = 'CQTHG')



**c. <biểu thức><phép so sánh>
{SOME|ANY|ALL} (<Câu truy vấn
con>)**

So sánh kết quả với truy vấn con

Ví dụ

○ **SOME/ANY**

- { < | <= | >= | > | <> | = } **SOME**
- <>**SOME** ⇔ **NOT IN**
- =**SOME** ⇔ **IN**

- Ví dụ: *Hiện thông tin về việc cung ứng các mặt hàng có mã MH013 có giá nhỏ hơn giá của nhà cung cấp có mã CHULA cung ứng.*

○ SELECT *

FROM CUNG_UNG

WHERE MaHang='MH013' and Gia < **SOME**

(select CUNG_UNG.Gia

from CUNG_UNG

where MaNCC = 'CHULA');

Ví dụ

- **ALL**

- {<| <=| >=| >| <>| =} **ALL**
- <>**ALL** ⇔ **NOT IN**
- *Ví dụ: Tìm nhà cung ứng cung ứng mặt hàng laptop với giá trung bình thấp nhất*

```
SELECT MaNCC
FROM CUNG_UNG
where MaNhom =12
GROUP BY MaNCC
HAVING avg (Gia) < ALL
(select avg(Gia)
from CUNG_UNG inner join MAT_HANG on
CUNG_UNG.MaHang = MAT_HANG.MaHang
where MaNhom =12
group by MaNCC);
```



Phụ lục

- Kiểu dữ liệu:
 - **char(n)**: Xâu ký tự có độ dài cố định
 - **varchar(n)**: Xâu ký tự có độ dài thay đổi với độ dài tối đa là n
 - **int**: Số nguyên
 - **smallint**: Số nguyên nhỏ
 - **numeric (p,d)**: Số thập phân: phần thập phân p, phần nguyên d
 - **real, double precision**: Số thập phân dấu phẩy động và số thập phân dấu phẩy động với độ chính xác kép
 - **float(n)**: Số thập phân dấu phẩy động với độ chính xác ít nhất là n chữ số
 - **date**: Ngày gồm năm, tháng, ngày
 - **time**: Thời gian gồm giờ, phút, giây
 - NULL