

CƠ SỞ DỮ LIỆU

CHƯƠNG 3

NGÔN NGỮ ĐỊNH NGHĨA VÀ THAO TÁC DỮ LIỆU ĐỐI VỚI MÔ HÌNH QUAN HỆ

Nội dung

1. Phân loại các ngôn ngữ truy vấn
2. Ngôn ngữ đại số quan hệ
3. Ngôn ngữ truy vấn SQL

Nội dung

- Các cách tiếp cận đối với thiết kế ngôn ngữ của CSDL quan hệ
 - Giới thiệu một số ngôn ngữ và phân loại
 - ⇒ So sánh và đánh giá
 - Một tập hợp các ràng buộc về dữ liệu
- Một số ngôn ngữ dữ liệu mức cao
 - QBE (Query By Example)
 - SQL (Structured Query Language)
 - Kết luận

CSDL ví dụ 1

SINH_VIEN		
MaSV	TenSV	DiaChi
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

MON_HOC		
MaMon	TenMon	Khoa
21	Cấu trúc dữ liệu và giải thuật	Viễn Thông
23	Cơ sở dữ liệu	Viễn Thông
29	Kỹ thuật lập trình	Viễn Thông
18	Toán cao cấp	Viễn Thông

DANG_KY	
MaSV	MaMon
1108	21
1108	23
8507	23
8507	29

THAM_GIA	
MaSV	MaLop
3936	101
1108	113
8507	101
8452	113

LOP		
MaLop	TenLop	Khoa
113	VT01	Viễn Thông
101	VT02	Viễn Thông

CSDL ví dụ 2

NHA_PHAN_PHOI			
MaNPP	TenNPP	QuyMo	DiaChi
S1	Dustin	100	TP Hồ Chí Minh
S2	Rusty	60	Hà Nội
S3	Lubber	100	TP Hồ Chí Minh
S4	M&M	45	TP Hồ Chí Minh
S5	MBI	1000	Hà Nội
S6	Panda	170	Hải Phòng

SAN_PHAM		
MaSP	TenSP	MauSac
P1	Screw	red
P2	Screw	green
P3	Nut	red
P4	Bolt	blue
P5	Plier	green
P6	Scissors	red

PHAN_PHOI		
MaNPP	MaSP	SoLuong
S1	P1	500
S1	P2	500
S1	P3	400
S2	P4	300
S2	P6	200
S3	P5	50
S4	P3	100
S4	P5	1000
S5	P1	350
S5	P2	250

Đặt vấn đề - Các câu hỏi

- Tìm tên của các sinh viên nào sống ở Bundoora
 - Tìm các bộ của bảng SINH_VIEN có DiaChi = Bundoora
 - Đưa ra các giá trị của thuộc tính TenSV của các bộ này

SINH_VIEN		
MaSV	TenSV	DiaChi
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Đặt vấn đề - Các câu hỏi

- Tìm các sinh viên thuộc lớp học có mã số 113
 - Tìm các giá trị MaSV trong bảng THAM_GIA có MaLop tương ứng là 113
 - Đưa các bộ của bảng SINH_VIEN có MaSV trong các giá trị tìm thấy ở trên

THAM_GIA	
MaSV	MaLop
3936	101
1108	113
8507	101
8452	113

SINH_VIEN		
MaSV	TenSV	DiaChi
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Tổng quan

- Gồm các phép toán tương ứng với các thao tác trên các quan hệ
- Mỗi phép toán
 - Đầu vào: một hay nhiều quan hệ
 - Đầu ra: một quan hệ
- Biểu thức đại số quan hệ = chuỗi các phép toán
- Kết quả thực hiện một biểu thức đại số là một quan hệ
- Được cài đặt trong phần lớn các hệ CSDL hiện nay

Phân loại các phép toán đại số quan hệ

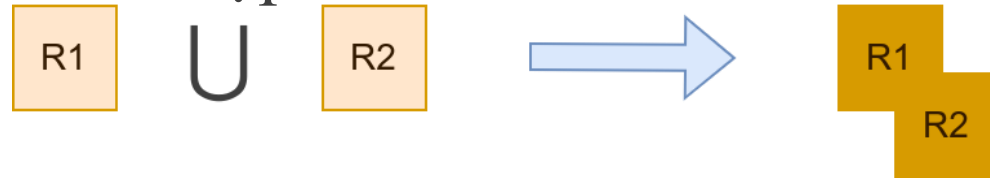
- Phép toán quan hệ
 - Phép chiếu (projection)
 - Phép chọn (selection)
 - Phép kết nối (join)
 - Phép chia (division)
- Phép toán tập hợp
 - Phép hợp (union)
 - Phép giao (intersection)
 - Phép trừ (difference)
 - Phép tích đề-các (cartesian product)

Phép toán tập hợp

- Định nghĩa: Quan hệ khả hợp
 - 2 quan hệ r và s được gọi là khả hợp nếu chúng được xác định trên cùng 1 miền giá trị
 - r xác định trên $D_1 \times D_2 \times \dots \times D_n$
 - s xác định trên $D'_1 \times D'_2 \times \dots \times D'_m \Rightarrow D_i = D'_i$ và $n=m$

Phép hợp

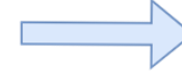
- Định nghĩa: gồm các bộ thuộc ít nhất 1 trong 2 quan hệ đầu vào
- 2 quan hệ đầu vào phải là khả hợp
- Cú pháp: $R = R1 \cup R2$



MON_HOC1	
MaMon	TenMon
S1	CSDL
S2	Toán Rời Rạc
S3	Kỹ thuật lập trình
S4	Mạng

\cup


MON_HOC2	
MaMon	TenMon
S5	Xác Suất Thống Kê
S6	Hệ Điều Hành



MON_HOC1	
MaMon	TenMon
S1	CSDL
S2	Toán Rời Rạc
S3	Kỹ thuật lập trình
S4	Mạng
S5	Xác Suất Thống Kê
S6	Hệ Điều Hành

Phép giao

- Định nghĩa: gồm các bộ thuộc cả hai quan hệ đầu vào

- Cú pháp: $R = R1 \cap R2$ 

MON_HOC1		\cap	MON_HOC2		\rightarrow	MON_HOC	
MaMon	TenMon		MaMon	TenMon		MaMon	TenMon
S1	CSDL		S5	Xác Suất Thống Kê		S1	CSDL
S2	Toán Rời Rạc		S6	Hệ Điều Hành			
S3	Kỹ thuật lập trình		S3	Kỹ thuật lập trình		S3	Kỹ thuật lập trình
S4	Mạng		S1	CSDL			

Phép trừ

- Định nghĩa: gồm các bộ thuộc quan hệ thứ nhất nhưng không thuộc quan hệ thứ hai
- Cú pháp: $R = R1 \setminus R2$ hoặc $R = R1 - R2$



MON_HOC1	
MaMon	TenMon
S1	CSDL
S2	Toán Rời Rạc
S3	Kỹ thuật lập trình
S4	Mạng

\setminus

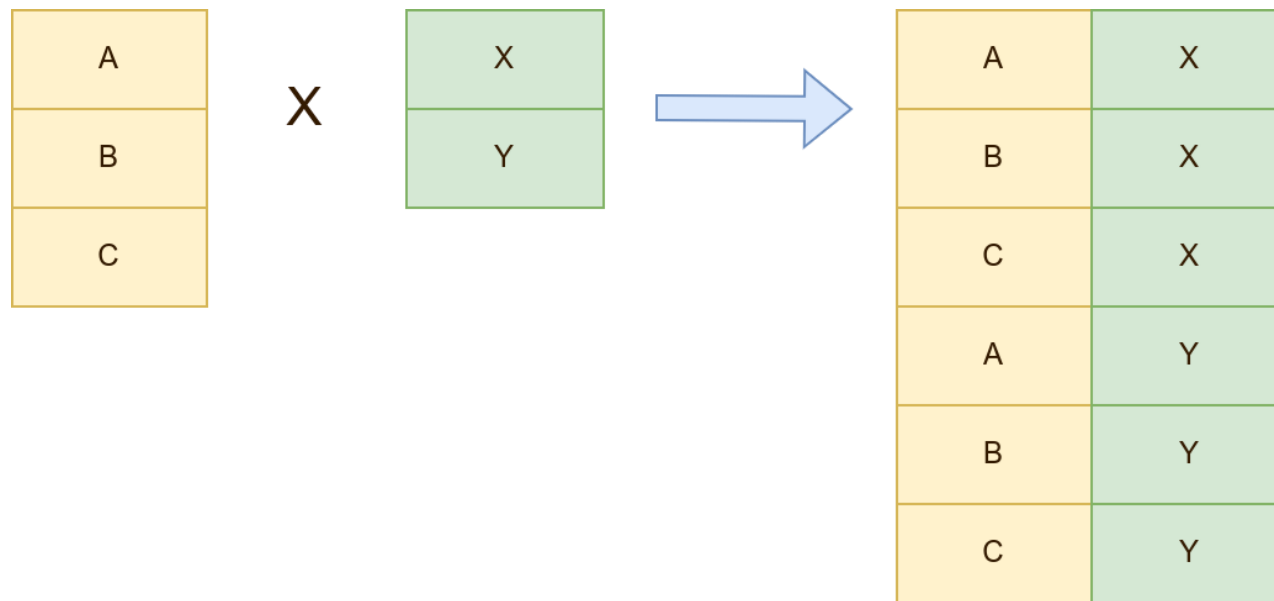
MON_HOC2	
MaMon	TenMon
S5	Xác Suất Thống Kê
S6	Hệ Điều Hành
S3	Kỹ thuật lập trình
S1	CSDL

\rightarrow

MON_HOC	
MaMon	TenMon
S2	Toán Rời Rạc
S4	Mạng

Phép tích Đề-các

- Định nghĩa: là kết nối giữa từng bộ của quan hệ thứ nhất với mỗi bộ của quan hệ thứ hai
- Cú pháp: $R = R1 \times R2$



Phép tích Đề-các

SINH_VIEN		
MaSV	TenSV	DiaChi
P1	Minh	Hà Nội
P2	Trang	Bắc Ninh
P3	Hưng	Hà Nam

X

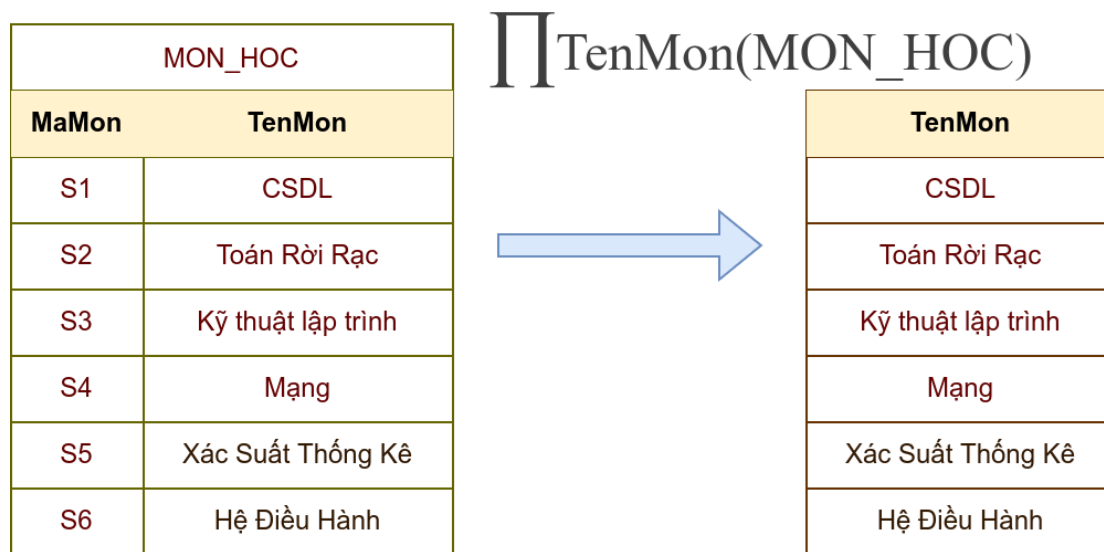
MON_HOC	
MaMon	TenMon
S2	Toán Rời Rạc
S4	Mạng



SINH_VIEN_MON_HOC				
MaSV	TenSV	DiaChi	MaMon	TenMon
P1	Minh	Hà Nội	S2	Toán Rời Rạc
P2	Trang	Bắc Ninh	S2	Toán Rời Rạc
P3	Hưng	Hà Nam	S2	Toán Rời Rạc
P1	Minh	Hà Nội	S4	Mạng
P2	Trang	Bắc Ninh	S4	Mạng
P3	Hưng	Hà Nam	S4	Mạng

Phép chiếu

- Định nghĩa: Lựa chọn một số thuộc tính từ một quan hệ
- Cú pháp: $\Pi_{A1,A2}(R)$
- Ví dụ: đưa ra danh sách tên của tất cả các sinh viên



Phép chọn

- Định nghĩa: Lựa chọn các bộ trong một quan hệ thỏa mãn điều kiện cho trước
- Cú pháp: $\sigma_{\langle \text{Điều kiện} \rangle}(\mathbf{R})$
- Ví dụ: đưa ra danh sách những sinh viên sống ở Bundoora

$\sigma_{\langle \text{DiaChi} = \text{'Bundoora'} \rangle}(\text{SINH_VIEN})$

SINH_VIEN		
MaSV	TenSV	DiaChi
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn



SINH_VIEN		
MaSV	TenSV	DiaChi
3936	Glen	Bundoora
8507	Norman	Bundoora

Phép chọn

- Điều kiện chọn còn gọi là biểu thức chọn.
- Biểu thức chọn F : một tổ hợp logic của các toán hạng. Mỗi toán hạng là một phép so sánh đơn giản giữa 2 biến là hai thuộc tính hoặc giữa 1 biến là 1 thuộc tính và 1 giá trị hằng.
 - Các phép so sánh trong F : $< > = \leq \geq \neq$
 - Các phép toán logic trong F : $\vee \wedge$

Ví dụ: chọn và chiếu

- Đưa ra tên của các sinh viên sống ở Bundoora
- Cú pháp: $\Pi_{\text{TenSV}}(\sigma_{\langle \text{DiaChi} = \text{'Bundoora'} \rangle}(\text{SINH_VIEN}))$

SINH_VIEN		
MaSV	TenSV	DiaChi
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn



TenSV
Glen
Norman

Phép kết nối 2 quan hệ

- Khái niệm ghép bộ: $u = (a_1, \dots, a_n); v = (b_1, \dots, b_m)$
 $(u, v) = (a_1, \dots, a_n, b_1, \dots, b_m)$
- Phép kết nối 2 quan hệ thực chất là phép ghép các cặp bộ của 2 quan hệ thỏa mãn 1 điều kiện nào đó trên chúng.
- Biểu thức kết nối là phép hội của các toán hạng, mỗi toán hạng là 1 phép so sánh đơn giản giữa 1 thuộc tính của quan hệ r và 1 thuộc tính của quan hệ s .
- Cú pháp: $R_1 \bowtie \langle\langle \text{điều kiện} \rangle\rangle R_2$

Phép kết nối - Ví dụ

- Đưa ra danh sách các sinh viên và mã khoá học mà sinh viên đó tham gia:
- $\text{SINH_VIEN} \bowtie_{\text{MaSV}=\text{MaSV}} \text{DANG_KY}$

SINH_VIEN		
MaSV	TenSV	DiaChi
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn



DANG_KY	
MaSV	MaMon
1108	21
1108	23
8507	23
8507	29



23	TenSV	DiaChi	MaMon
1108	Robert	Kew	21
1108	Robert	Kew	23
8507	Norman	Bundoora	23
8507	Norman	Bundoora	29

Phép kết nối bằng-kết nối tự nhiên

- Định nghĩa: Nếu phép so sánh trong điều kiện kết nối là phép so sánh bằng thì kết nối gọi là **kết nối bằng**
- Định nghĩa: Phép kết nối bằng trên các thuộc tính cùng tên của 2 quan hệ và sau khi kết nối 1 thuộc tính trong 1 cặp thuộc tính trùng tên đó sẽ bị loại khỏi quan hệ kết quả thì phép kết nối gọi là **kết nối tự nhiên**
- Cú pháp phép kết nối tự nhiên: $R1 * R2$

Phép kết nối bằng-kết nối tự nhiên

DANG_KY	
MaSV	MaMon
1108	21
1108	23
8507	23
8507	29

*

THAM_GIA	
MaSV	MaLop
3936	101
1108	113
8507	101
8452	113



MaSV	MaMon	MaLop
1108	21	113
1108	23	113
8507	23	101
8507	29	101

Ví dụ: chọn, chiếu, kết nối

- Đưa ra tên của các sinh viên sống ở Bundoora và mã lớp học mà sinh viên đó đăng ký:

$\Pi_{\text{TenSV}, \text{MaLop}}(\sigma_{\langle \text{DiaChi} = \text{'Bundoora'} \rangle}(\text{SINH_VIEN} \bowtie \text{MaSV} = \text{MaSV DANG_KY}))$

SINH_VIEN		
MaSV	TenSV	DiaChi
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

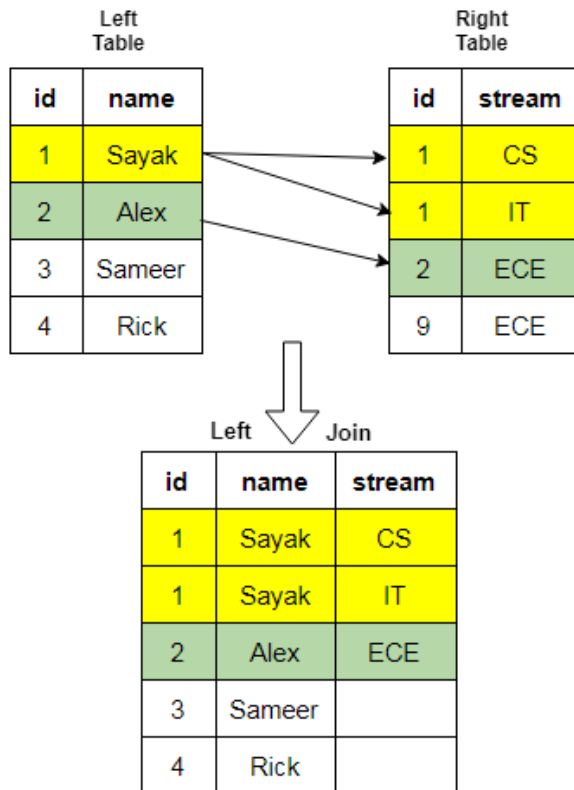
THAM_GIA	
MaSV	MaLop
3936	101
1108	113
8507	101
8452	113



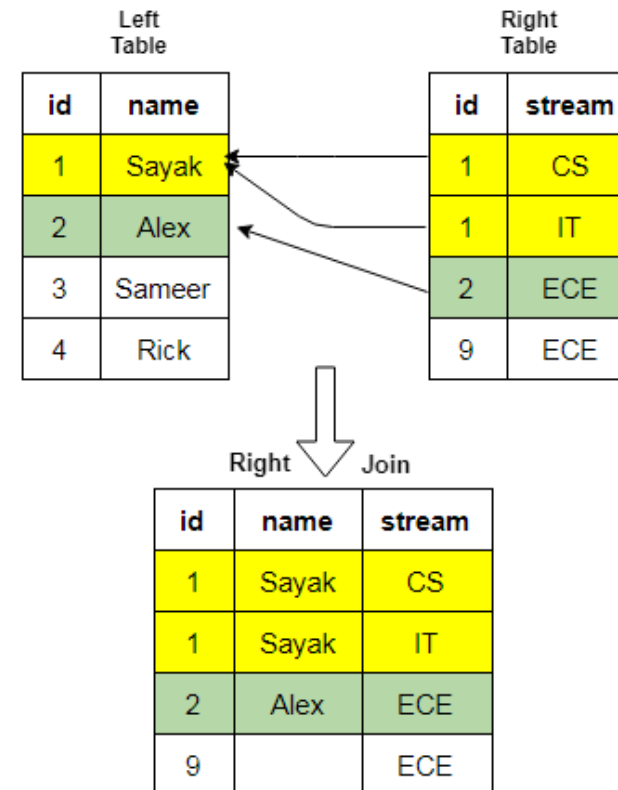
TenSV	MaLop
Glen	101
Norman	101

Phép kết nối ngoài

- Phép kết nối ngoài trái ⚡

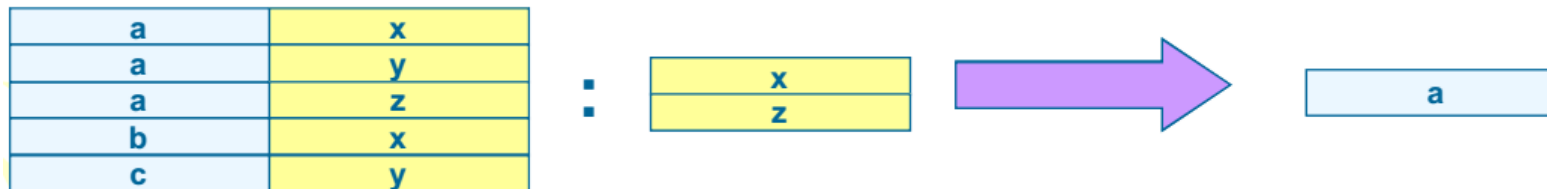


- Phép kết nối ngoài phải ⚡



Phép chia

- Định nghĩa: Phép chia giữa 1 quan hệ r bậc n và quan hệ s bậc m ($m < n$) với sơ đồ quan hệ của s là tập con của sơ đồ quan hệ của r là một tập các $(n-m)$ -bộ sao cho khi ghép mọi bộ thuộc s với t thì ta đều có một bộ thuộc r
- Cú pháp: $R = R1 : R2$



Phép chia

- Ví dụ: Đưa ra môn học được dạy ở tất cả các chương trình



Ngôn ngữ SQL

- 1975: SEQUEL
 - System-R
- 1976: SEQUEL2
- 1978/79: SQL
 - System-R
- 1986: chuẩn SQL-86
- 1989: chuẩn SQL-89
- 1992: chuẩn SQL-92
- 1996: chuẩn SQL-96



SQL

Các thành phần của SQL

- Ngôn ngữ định nghĩa dữ liệu (Data Definition Language)
 - Cấu trúc các bảng CSDL
 - Các mối liên hệ của dữ liệu
 - Quy tắc, ràng buộc áp đặt lên dữ liệu
- Ngôn ngữ thao tác dữ liệu (Data Manipulation Language)
 - Thêm, xóa, sửa dữ liệu trong CSDL
 - Truy vấn dữ liệu
- Ngôn ngữ điều khiển dữ liệu (Data Control Language)
 - Khai báo bảo mật thông tin
 - Quyền hạn của người dùng trong khai thác CSDL

Ngôn ngữ định nghĩa dữ liệu

- Các thông tin được định nghĩa bao gồm
 - Sơ đồ quan hệ
 - Kiểu dữ liệu hay miền giá trị của mỗi thuộc tính
 - Các ràng buộc toàn vẹn
 - Các chỉ số đối với mỗi bảng
 - Thông tin an toàn và ủy quyền đối với mỗi bảng
 - Cấu trúc lưu trữ vật lý của mỗi bảng trên đĩa
- Được biểu diễn bởi các lệnh định nghĩa dữ liệu

Quy ước đặt tên và kiểu dữ liệu

- Quy ước đặt tên
 - 32 ký tự: chữ cái, số, dấu _
- Kiểu dữ liệu (SQL-92)
 - char(n)
 - varchar(n)
 - int
 - smallint
 - numeric(p,d)
 - real, double
 - float(n)
 - date
 - time

Cú pháp

- Tạo bảng

```
CREATE TABLE tên-bảng(  
    cột-1 kiểu-dữ-liệu-1 [NOT NULL], ...,  
    cột-2 kiểu-dữ-liệu-2 [NOT NULL], ...,  
    ....  
    [CONSTRAINT tên-ràng-buộc kiểu-ràng-buộc]  
    ...  
);
```

- Xoá bảng

```
DROP TABLE tên-bảng
```


Tạo bảng - Ví dụ

```
CREATE TABLE NHA_PHAN_PHOI(  
    MaNPP char(4) NOT NULL,  
    TenNPP varchar(30) NOT NULL,  
    QuyMo smallint,  
    DiaChi varchar(20),  
    CONSTRAINT KhoachinhS primary key(sid)  
);
```

Tạo bảng - Ví dụ

```
CREATE TABLE SAN_PHAM(  
    MaSP char(4) NOT NULL,  
    TenSP varchar(30) NOT NULL,  
    MauSac char(8),  
    CONSTRAINT KhoachinhP primary key(pid)  
);
```

Tạo bảng - Ví dụ

```
CREATE TABLE PHAN_PHOI(  
    MaNPP char(4) NOT NULL,  
    MaSP char(4) NOT NULL,  
    SoLuong smallint,  
    primary key(MaNPP, MaSP),  
    foreign key(MaNPP) references NHA_PHAN_PHOI(MaNPP),  
    foreign key(MaSP) references SAN_PHOI(MaSP),  
    check(SoLuong > 0)  
);
```

Kiểu ràng buộc

- Ràng buộc toàn vẹn (RBTV) về giá trị miền
CONSTRAINT <tên ràng buộc>
CHECK <điều kiện>
- RBTV về khoá ngoại hay phụ thuộc tồn tại
CONSTRAINT <tên ràng buộc>
FOREIGN KEY (fki) **REFERENCES** tên-bảng(ki);

Thêm/xoá/sửa cột của các bảng

- Thêm

ALTER TABLE <tên bảng>

ADD COLUMN <tên cột> <kiểu dữ liệu> [**NOT NULL**];

- Xóa

ALTER TABLE <tên bảng>

DROP COLUMN <tên cột>;

- Sửa

ALTER TABLE <tên bảng>

CHANGE COLUMN <tên cột> **TO** <kiểu dữ liệu mới>;

Ví dụ

- **ALTER TABLE PHAN_PHOI ADD COLUMN GiaBan real NOT NULL;**
- **ALTER TABLE PHAN_PHOI DROP COLUMN GiaBan;**
- **ALTER TABLE NHA_PHAN_PHOI CHANGE COLUMN TenNPP TO varchar(20);**

Thêm/xóa các ràng buộc

- Thêm

ALTER TABLE <tên bảng>

ADD CONSTRAINT <tên ràng buộc> <kiểu ràng buộc>

- Xóa

ALTER TABLE <tên bảng>

DROP CONSTRAINT <tên ràng buộc>

Ngôn ngữ truy vấn dữ liệu

- Cú pháp câu lệnh SQL:

SELECT [**DISTINCT**] <DS cột>|*<Biểu thức>|<Hàm TV>

FROM <DS bảng>

[**WHERE** <Điều kiện tìm kiếm>]

[**GROUP BY** <DS cột> [**HAVING** <Điều kiện>]]

[**ORDER BY** <Danh sách cột> [**ASC**|**DESC**]]

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

Ngôn ngữ truy vấn dữ liệu

- Tìm thông tin từ các cột của bảng
SELECT <DS cột> **FROM** <Tên bảng>;
SELECT * **FROM** <Tên bảng>;

- Ví dụ
SELECT TenMon **FROM** MON_HOC;

MON_HOC	
MaMon	TenMon
S1	CSDL
S2	Toán Rời Rạc
S3	Kỹ thuật lập trình
S4	Mạng
S5	Xác Suất Thống Kê
S6	Hệ Điều Hành

$\Pi_{\text{TenMon}}(\text{MON_HOC})$



TenMon
CSDL
Toán Rời Rạc
Kỹ thuật lập trình
Mạng
Xác Suất Thống Kê
Hệ Điều Hành

Truy vấn không điều kiện trên một bảng

- Đưa ra tên của các mặt hàng
SELECT TenSP FROM SAN_PHAM;
- Đưa ra tên khác nhau của các mặt hàng
SELECT DISTINCT TenSP FROM SAN_PHAM;
- Đưa ra toàn bộ thông tin về các hãng cung ứng
SELECT * FROM NHA_PHAN_PHOI;
- Đưa ra mã số hãng cung ứng, mã mặt hàng được cung ứng và 10 lần số lượng mặt hàng đã được cung ứng
SELECT MaNPP, MaSP, SoLuong * 10 FROM PHAN_PHOI;

Truy vấn không điều kiện trên một bảng

- Chọn các bản ghi (dòng)

SELECT <DS cột> **FROM** <Tên bảng>

WHERE <Điều kiện tìm kiếm>

- Ví dụ

SELECT * FROM SINH_VIEN WHERE DiaChi='Bundoora' ;

$\sigma_{\langle \text{DiaChi} = \text{'Bundoora'} \rangle}(\text{SINH_VIEN})$

SINH_VIEN		
MaSV	TenSV	DiaChi
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn



SINH_VIEN		
MaSV	TenSV	DiaChi
3936	Glen	Bundoora
8507	Norman	Bundoora

Truy vấn không điều kiện trên một bảng

- Đưa ra tên của các hãng cung ứng có trụ sở tại London

```
SELECT TenNPP FROM NHA_PHAN_PHOI  
WHERE DiaChi = 'London';
```

- Đưa ra mã số và tên của các hãng cung ứng nằm ở London và có số nhân viên lớn hơn 75

```
SELECT MaNPP, TenNPP FROM NHA_PHAN_PHOI  
WHERE DiaChi = 'London' AND QuyMo > 75;
```

Biểu diễn điều kiện lựa chọn

- Các phép toán quan hệ: =, !=, <, >, <=, >=
- Các phép toán logic: NOT, AND, OR
- Phép toán phạm vi: BETWEEN, IN, LIKE
 - Kiểu dữ liệu số
 - attr BETWEEN val1 AND val2 ((attr>=val1) and (attr<=val2))
 - attr IN (val1, val2, ...) ((attr=val1) or (attr=val2) or ...)
 - Kiểu dữ liệu chuỗi
 - LIKE: sử dụng đối sánh mẫu chuỗi với các ký tự thay thế cho 1 ký tự bất kỳ (_, ?), thay thế cho 1 chuỗi ký tự bất kỳ (*, %)
 - (PostgreSQL sử dụng dấu % và dấu _)

Biểu diễn điều kiện lựa chọn

- Đưa ra thông tin của các hãng cung ứng có số nhân viên trong khoảng từ 100 đến 150

SELECT * FROM Supplier

WHERE size BETWEEN 100 AND 150;

- Đưa ra mã số của hãng cung ứng mặt hàng P1 hoặc P2

- Cách 1:

SELECT MaSP FROM PHAN_PHOI WHERE MaSP = 'P1' OR MaSP = 'P2';

- Cách 2:

SELECT MaSP FROM PHAN_PHOI WHERE MaSP IN ('P1', 'P2');

Biểu diễn điều kiện lựa chọn

- Đưa ra thông tin của hãng sản xuất có trụ sở đặt tại thành phố bắt đầu bằng chữ New

```
SELECT * FROM NHA_PHAN_PHOI  
WHERE DiaChi LIKE 'New%';
```

New York, New Jersey, New Mexico, New Hampshire

Loại trừ các bản ghi trùng nhau

- Từ khoá **DISTINCT**

SELECT DISTINCT <DS cột>

FROM <DS bảng>

- Ví dụ: đưa ra danh sách tên các khoa (Khoa) tương ứng với các môn học (MON_HOC). Mỗi giá trị chỉ hiện thị một lần

SELECT DISTINCT Khoa **FROM** MON_HOC

Truy vấn có sử dụng phép toán đổi tên

- SQL cho phép đổi tên các bảng và các cột trong một câu truy vấn (sau mệnh đề SELECT và FROM) sử dụng cấu trúc:
 - <tên cũ> **AS** <tên mới>
- Đưa ra tên và số nhân viên của các hãng cung ứng ở Paris
SELECT TenNPP **AS** HangOParis, QuyMo **AS** SoNhanVien
FROM NHA_PHAN_PHOI **WHERE** DiaChi = 'Paris';

Truy vấn phức tạp trên nhiều bảng

- Điều kiện kết nối

SELECT <DS cột> **FROM** <DS bảng>

WHERE <Điều kiện tìm kiếm>

- Ví dụ: đưa ra danh sách mã sinh viên (MaSV), tên sinh viên (TenSV), thành phố (DiaChi), mã môn học (MaMon) mà các sinh viên đã đăng ký

SELECT MaSV, TenSV, DiaChi, MaMon **FROM** SINH_VIEN, DANG_KY
WHERE SINH_VIEN.MaSV=DANG_KY.MaSV;

Kết nối tự nhiên

SELECT <DS cột> **FROM** A, B, C
WHERE A.CộtX = B.CộtX **AND** B.CộtY = C.CộtY

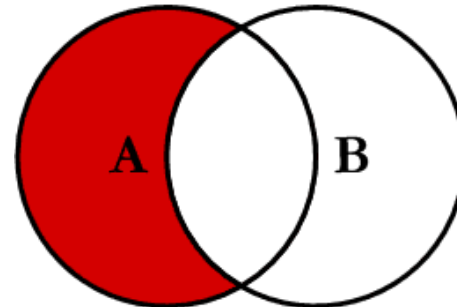
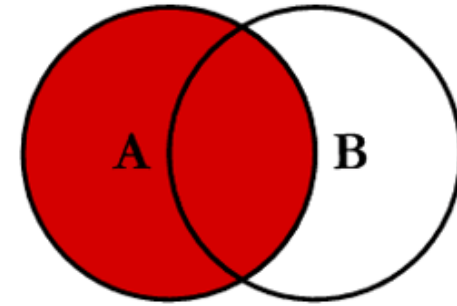
SELECT <DS cột>
FROM A **NATURAL JOIN** B **NATURAL JOIN** C

Kết nối hai bảng

- Kết nối ngoài trái

SELECT <DS cột> **FROM** A **LEFT JOIN** B
ON A.Key = B.Key

SELECT <DS cột> **FROM** A **LEFT JOIN** B
ON A.Key = B.Key **WHERE** B.Key IS NULL

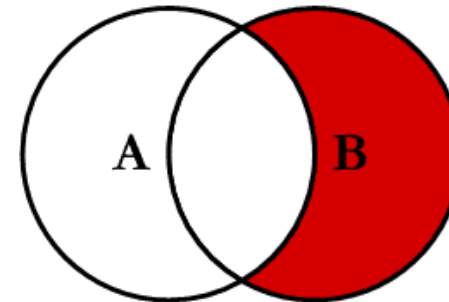
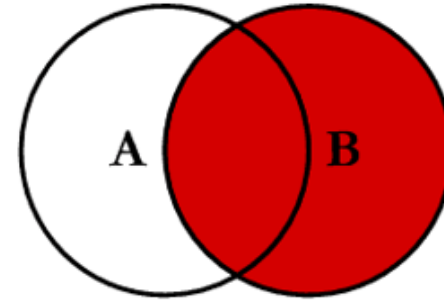


Kết nối hai bảng

- Kết nối ngoài phải

SELECT <DS cột> **FROM** A **RIGHT JOIN** B
ON A.Key = B.Key

SELECT <DS cột> **FROM** A **RIGHT JOIN** B
ON A.Key = B.Key **WHERE** A.Key IS NULL



Kết nối hai bảng

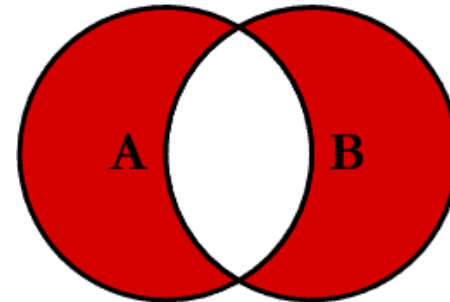
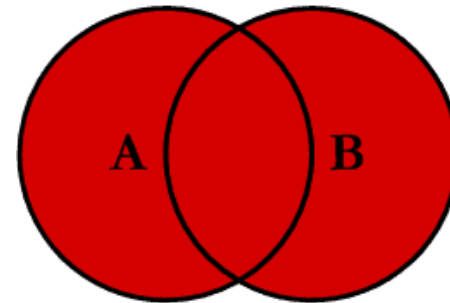
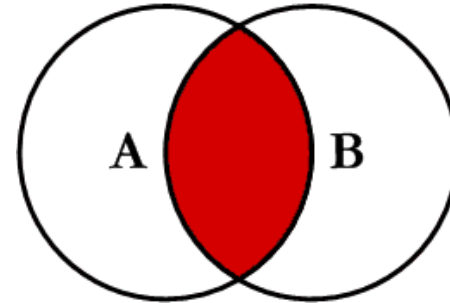
- Kết nối trong

SELECT <DS cột> **FROM** A **INNER JOIN** B
ON A.Key = B.Key

- Kết nối ngoài

SELECT <DS cột>
FROM A **FULL OUTER JOIN** B **ON** A.Key = B.Key

SELECT <DS cột>
FROM A **OUTER JOIN** B **ON** A.Key = B.Key
WHERE A.Key IS NULL **OR** B.Key IS NULL



Truy vấn phức tạp trên nhiều bảng

- Đưa ra tên của hãng có cung ứng mặt hàng P1

```
SELECT TenNPP FROM NHA_PHAN_PHOI S, PHAN_PHOI SP  
WHERE S.MaNPP = SP.MaNPP AND SP.MaSP = 'P1';
```

- Đưa ra tên và mã số của hãng cung ứng ít nhất một mặt hàng màu đỏ

```
SELECT TenNPP, S.MaNPP  
FROM NHA_PHAN_PHOI S, PHAN_PHOI SP, SAN_PHAM P  
WHERE S.MaNPP = SP.MaNPP AND P.MaSP = SP.MaSP AND P.MauSac = 'red';
```

Tìm kiếm có sắp xếp

- Sắp xếp các bản ghi kết quả theo một thứ tự cho trước
SELECT <DS cột> **FROM** <DS bảng>
[**WHERE** <Điều kiện tìm kiếm>]
ORDER BY <DS cột> [**ASC**|**DESC**]
- Ví dụ: đưa ra danh sách tên các sinh viên theo thứ tự tăng dần
SELECT TenSV **FROM** SINH_VIEN
ORDER BY TenSV **ASC**

Phân nhóm các bản ghi kết quả

- Phân nhóm các bản ghi kết quả theo giá trị của 1 hoặc nhiều thuộc tính
SELECT <DS cột> **FROM** <DS bảng>
[**WHERE** <Điều kiện tìm kiếm>]
[**GROUP BY** <DS cột>]
- Cột được chỉ ra trong mệnh đề **GROUP BY** được sử dụng làm cơ sở để chia nhóm. Cột này cũng bắt buộc phải được chỉ ra trong mệnh đề **SELECT**
- Ví dụ đưa ra tên các sinh viên nhóm theo thành phố của sinh viên đó
SELECT DiaChi, TenSV **FROM** SINH_VIEN
GROUP BY DiaChi

Điều kiện hiển thị các bản ghi kết quả

- Lựa chọn các bản ghi kết quả để hiển thị
SELECT <DS cột> **FROM** <DS bảng>
[**WHERE** <Điều kiện tìm kiếm>]
GROUP BY <Ds cột> **HAVING** <Điều kiện>
- Ví dụ: đưa ra tên các thành phố có nhiều hơn 3 sinh viên
SELECT DiaChi, **COUNT**(MaSV) **FROM** SINH_VIEN
GROUP BY DiaChi
HAVING COUNT(MaSV) > 3

Các phép toán tập hợp: UNION, MINUS, INTERSECT

- Ví dụ: đưa ra danh sách tên các môn học không có sinh viên nào tham dự

```
SELECT DISTINCT MON_HOC.TenMon FROM MON_HOC
```

```
MINUS
```

```
SELECT DISTINCT MON_HOC.TenMon
```

```
FROM SINH_VIEN, DANG_KY, MON_HOC
```

```
WHERE SINH_VIEN.MaSV = DANG_KY.MaSV AND DANG_KY.MaMon =  
MON_HOC.MaMon
```

- Tìm sid của hãng cung ứng đồng thời 2 mặt hàng P1 và P2

```
SELECT MaNPP FROM PHAN_PHOI WHERE MaSP = 'P1'
```

```
INTERSECT
```

```
SELECT MaNPP FROM PHAN_PHOI WHERE MaSP = 'P2'
```

Các câu truy vấn lồng nhau

- Là trường hợp các câu truy vấn (con) được viết lồng nhau
- Thường được sử dụng để
 - Kiểm tra thành viên tập hợp (IN, NOT IN)
 - So sánh tập hợp (>ALL, >=ALL, <ALL, <=ALL, =ALL, NOT IN, SOME,)
 - Ví dụ :
SELECT * FROM NHA_PHAN_PHOI
WHERE QuyMo >= ALL (SELECT QuyMo FROM NHA_PHAN_PHOI);
 - Kiểm tra các bảng rỗng (EXISTS hoặc NOT EXISTS)
- Các truy vấn con lồng nhau thông qua mệnh đề WHERE

Các câu truy vấn lồng nhau

- Kiểm tra thành viên tập hợp với IN và NOT IN:
- Đưa ra mã số của các hãng cung ứng đồng thời 2 mặt hàng P1 và P2:

```
SELECT DISTINCT MaNPP FROM PHAN_PHOI  
WHERE MaSP = 'P1' AND MaNPP IN (SELECT sid FROM PHAN_PHOI SP2 WHERE  
SP2.MaSP = 'P2');
```

- Đưa ra mã của các hãng không cung ứng mặt hàng P3:

```
SELECT MaNPP FROM PHAN_PHOI  
WHERE MaNPP NOT IN (SELECT MaNPP FROM PHAN_PHOI SP2 WHERE  
SP2.MaSP = 'P3');
```

Các câu truy vấn lồng nhau

- So sánh tập hợp: Sử dụng các phép toán <, >, >=, <=, =, != (<>) kèm với các mệnh đề ANY và ALL
- Đưa ra tên của các hãng có số nhân viên đông nhất:
SELECT TenNPP FROM NHA_PHAN_PHOI
WHERE QuyMo >= ALL(SELECT QuyMo FROM NHA_PHAN_PHOI)
- Đưa ra sid của hãng cung ứng một mặt hàng với số lượng bằng ít nhất 1 trong số lượng các mặt hàng được cung ứng bởi S2
SELECT MaNPP FROM PHAN_PHOI
WHERE MaNPP != 'S2' AND SoLuong = ANY(SELECT SoLuong FROM PHAN_PHOI
SP2 WHERE SP2.MaNPP = 'S2');

Các câu truy vấn lồng nhau

- Kiểm tra tập hợp rỗng với EXISTS và NOT EXISTS
 - EXISTS(câu truy vấn con): nhận giá trị đúng khi câu truy vấn con cho ra kết quả là một quan hệ khác rỗng
 - NOT EXISTS(câu truy vấn con): nhận giá trị đúng khi câu truy vấn con cho ra kết quả là một quan hệ rỗng

Các câu truy vấn lồng nhau

- Đưa ra thông tin của các nhà cung cấp đã cung ứng ít nhất một mặt hàng
SELECT * FROM NHA_PHAN_PHOI S
WHERE EXISTS (SELECT MaNPP FROM PHAN_PHOI SP WHERE S.MaNPP =
SP.MaNPP);
- Đưa ra thông tin của các nhà cung cấp không cung ứng mặt hàng nào
SELECT * FROM NHA_PHAN_PHOI S
WHERE NOT EXISTS (SELECT * FROM PHAN_PHOI SP WHERE S.MaNPP =
SP.MaNPP);

Các hàm thư viện

- Hàm tính toán trên nhóm các bản ghi
 - MAX/MIN
 - SUM
 - AVG
 - COUNT
- Hàm tính toán trên bản ghi
 - Hàm toán học: ABS, SQRT, LOG, EXP, SIGN, ROUND
 - Hàm xử lý chuỗi ký tự: LEN, LEFT, RIGHT, MID
 - Hàm xử lý thời gian: DATE, DAY, MONTH, YEAR, HOUR, MINUTE, SECOND
 - Hàm chuyển đổi kiểu giá trị: FORMAT

Các hàm thư viện

- SQL Server

```
SELECT * FROM GIANG_VIEN  
WHERE DATEPART(year, GETDATE()) - DATEPART(year, NgaySinh) > 40
```

- PostgreSQL

```
SELECT * FROM "GIANG_VIEN"  
WHERE date_part('year', current_date) - date_part('year',"NgaySinh")>40
```

Một số ví dụ với các hàm thư viện

- Có bao nhiêu mặt hàng khác nhau được cung ứng
SELECT COUNT(DISTINCT MaSP)
FROM PHAN_PHOI;
- Có tổng cộng bao nhiêu nhân viên làm cho các hãng ở Paris
SELECT SUM(size) FROM NHA_PHAN_PHOI
WHERE DiaChi = 'Paris';
- Đưa ra số lượng mặt hàng trung bình mà hãng S1 cung ứng
SELECT AVG(SoLuong) FROM PHAN_PHOI
WHERE MaNPP = 'S1';

Một số truy vấn phức tạp

- Đưa ra tên của hãng S1 và tổng số lượng các mặt hàng mà hãng đó cung ứng
SELECT TenNPP, **SUM**(SoLuong) **FROM** NHA_PHAN_PHOI S, PHAN_PHOI SP
WHERE S.MaNPP = SP.MaNPP AND S.MaNPP = 'S1'
GROUP BY TenNPP;
- Đưa ra mã số các hãng cung ứng và số lượng trung bình các mặt hàng được cung ứng bởi từng hãng
SELECT MaNPP, **AVG**(SoLuong) **FROM** PHAN_PHOI
GROUP BY MaNPP;
- Đưa ra mã số các hãng cung ứng mà số lượng mặt hàng trung bình được cung cấp bởi hãng đó là trong khoảng từ 75 đến 100
SELECT MaNPP, **AVG**(SoLuong) **FROM** PHAN_PHOI
GROUP BY MaNPP **HAVING** **AVG**(SoLuong) **BETWEEN** 75 **AND** 100

Thêm bản ghi vào bảng

```
INSERT INTO table[(col1,col2,...)]  
VALUES (exp1,exp2,...)
```

```
INSERT INTO table[(col1,col2,...)]  
SELECT col1,col2, ...  
FROM tab1, tab2, ...  
WHERE <dieu kien>
```

Thêm bản ghi vào bảng

- Ví dụ

INSERT INTO SINH_VIEN VALUES ('1179','Jane','California');

INSERT INTO SINH_VIEN(MaSV, TenSV, DiaChi) VALUE ('1180','David','New York');

INSERT INTO SINH_VIEN(TenSV, MaSV, DiaChi) VALUES ('Mary','1181','Texas');

**INSERT INTO SINH_VIEN(MaSV, TenSV, DiaChi)
VALUES ('1182','John','Ohio'), ('1183','Tom','Georgia'), ('1184','Declan','Arizona');**

Xóa bản ghi trong bảng

- Cú pháp

DELETE FROM <Tên bảng>

WHERE <Điều kiện xóa>;

- Ví dụ:

DELETE FROM PHAN_PHOI

WHERE MaNPP = 'S4';

DELETE FROM SINH_VIEN

WHERE DiaChi = 'Indiana';

Sửa dữ liệu trong bảng

- Cú pháp

UPDATE <tên bảng>**SET** (<Tên cột> = Giá trị mới , ...)
[**WHERE** <Điều kiện sửa đổi>];

- Ví dụ:

- Hãng S1 chuyển tới Milan

UPDATE NHA_PHAN_PHOI **SET** DiaChi = 'Milan'
WHERE MaNPP = 'S1';

- Tất cả các mặt hàng được cung cấp với số lượng nhỏ hơn 100 đều tăng số lượng lên 1.5 lần

UPDATE PHAN_PHOI **SET** SoLuong = SoLuong * 1.5
WHERE SoLuong < 100;