

Chương IV: Ngôn ngữ SQL

I Định nghĩa dữ liệu trong SQL

II Các lệnh truy vấn trong SQL

III Các lệnh cập nhật dữ liệu

IV Các lệnh an toàn dữ liệu

V Các lệnh nối dữ liệu (JOIN)

<

>

MAIN MENU

EXIT

TỔNG QUAN SQL

1. Giới thiệu:

- Là ngôn ngữ chuẩn đối với CSDL, ngôn ngữ rất mạnh, phổ dụng & dễ dùng, ngôn ngữ có ảnh hưởng nhất trong CSDL quan hệ.
- Chuẩn ANSI (American National Standard Institute) và ISO cho SQL được công bố vào năm 1989, 1992, 1999, 2003 (+ XML), 2006 và 2008, ... SQL Server 2019.
- Sử dụng tổ hợp cấu trúc đại số quan hệ và phép tính vị từ.

Về chương IV

< >

MAIN MENU

EXIT

TỔNG QUAN SQL

Khối lệnh SELECT ... FROM ... WHERE ...

SQL là ngôn ngữ phi thủ tục, chuẩn mực & điển hình. Phép toán cơ bản trong SQL là phép ánh xạ → mô tả bởi khối lệnh SELECT - FROM - WHERE, trong đó:

- mệnh đề SELECT ↔ phép chiếu (Π) của đại số quan hệ.
- mệnh đề FROM ↔ phép tích Decac (\times) của đại số quan hệ.
- mệnh đề WHERE ↔ phép chọn (σ) của đại số quan hệ.

$$\Pi_{A_1, A_2, \dots, A_n}(\sigma_P(R_1 \times R_2 \times \dots \times R_n))$$

↔

SELECT A₁, A₂, ..., A_n FROM R₁, R₂, ..., R_n WHERE P

TỔNG QUAN SQL

2. Đặc điểm:

- Cách đọc: FROM → WHERE → SELECT
- Đặc điểm của SQL: Ngôn ngữ tựa English

3. Thành phần:

- Ngôn ngữ định nghĩa dữ liệu (Data Definition Language - DDL).
- Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML).
- Ngôn ngữ điều khiển dữ liệu (Data Control Language - DCL).

Về chương IV

<

>

MAIN MENU

EXIT

TỔNG QUAN SQL

3. Thành phần:

- Ngôn ngữ định nghĩa dữ liệu DDL: Bao gồm những lệnh dùng để tạo, xóa hay thay đổi các đối tượng trong CSDL như Table, View, Schema, Domain, Trigger, Stored procedure. Đặc tả ủy trị (quyền truy cập), đặc tả các ràng buộc toàn vẹn (RBTV). Các từ khóa chính: CREATE, ALTER, DROP DATABASE|TABLE|INDEX|VIEW,...
- Ngôn ngữ thao tác dữ liệu DML: Bao gồm các lệnh dùng để trích chọn, bổ sung, biến đổi hoặc xóa dữ liệu. Các từ khóa chính: SELECT, INSERT, UPDATE, DELETE.
- Ngôn ngữ điều khiển dữ liệu DCL: Bao gồm các lệnh cho phép cấp hoặc loại bỏ quyền truy cập của người dùng hay của chương trình nhúng vào CSDL. Các từ khóa chính: GRANT, REVOKE và COMMIT, ROLLBACK.

Về chương IV

< >

MAIN MENU

EXIT

TỔNG QUAN SQL

4. Quy ước:

Qui ước	Sử dụng cho
TỪ VIẾT HOA	Từ khóa của SQL (“viết hoa theo quy ước”).
Ký tự %	(phần trăm) Đối sánh 1 xâu con hay thay thế cho 1 xâu con (*)
Ký tự _	(gạch dưới) Đối sánh 1 ký tự hay thay thế cho 1 ký tự (?)
[] ; () ; ; query	Tham số tùy chọn ; Tham số bắt buộc ; Hoặc ; câu truy vấn
Quan hệ; Bộ; Thuộc tính	Bảng; Dòng Hàng Bản ghi Record; Cột Trường Field
Định danh (Identifier)	Tên đối tượng định danh không có ký tự trắng và ký tự đặc biệt, không trùng với từ khóa.

Về chương IV

< >

MAIN MENU

EXIT

TỔNG QUAN SQL

5. Các kiểu dữ liệu căn bản (data-type)

Kiểu dữ liệu	Mô tả
char(n)	Kiểu chuỗi, độ dài cố định n ký tự
nchar(n)	Kiểu chuỗi, độ dài cố định n ký tự, hỗ trợ Unicode
varchar(n)	Kiểu chuỗi, độ dài tối đa n ký tự
nvarchar(n)	Kiểu chuỗi, độ dài tối đa n ký tự, hỗ trợ Unicode
text	Kiểu chuỗi độ dài lớn (tối đa 2,147,483,647 ký tự)
ntext	Như text, hỗ trợ Unicode (tối đa 1,073,741,823 ký tự)

Về chương IV

< >

MAIN MENU

EXIT

TỔNG QUAN SQL

5. Các kiểu dữ liệu căn bản (cont)

Kiểu dữ liệu	Mô tả
int	Số nguyên có giá trị từ -2 ³¹ đến 2 ³¹ - 1
tinyint	Số nguyên, từ 0 đến 255.
smallint	Số nguyên, từ -2 ¹⁵ đến 2 ¹⁵ - 1
bigint	Số nguyên, từ -2 ⁶³ đến 2 ⁶³ -1
numeric(m, n)	Kiểu số, độ chính xác cố định.
decimal(m, n)	Tương tự kiểu Numeric (MS. Access: real)
float	Số thực có giá trị từ -1.79E+308 đến 1.79E+308

Về chương IV



MAIN MENU

EXIT

TỔNG QUAN SQL

5. Các kiểu dữ liệu căn bản (cont)

Kiểu dữ liệu	Mô tả
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 trăm của giây)
smalldatetime	Kiểu ngày giờ (chính xác đến phút)
binary	Dữ liệu nhị phân, độ dài cố định (tối đa 8000 bytes)
varbinary	Dữ liệu nhị phân, độ dài tối đa 8000 bytes
image	Dữ liệu nhị phân (tối đa 2,147,483,647 bytes)

Về chương IV

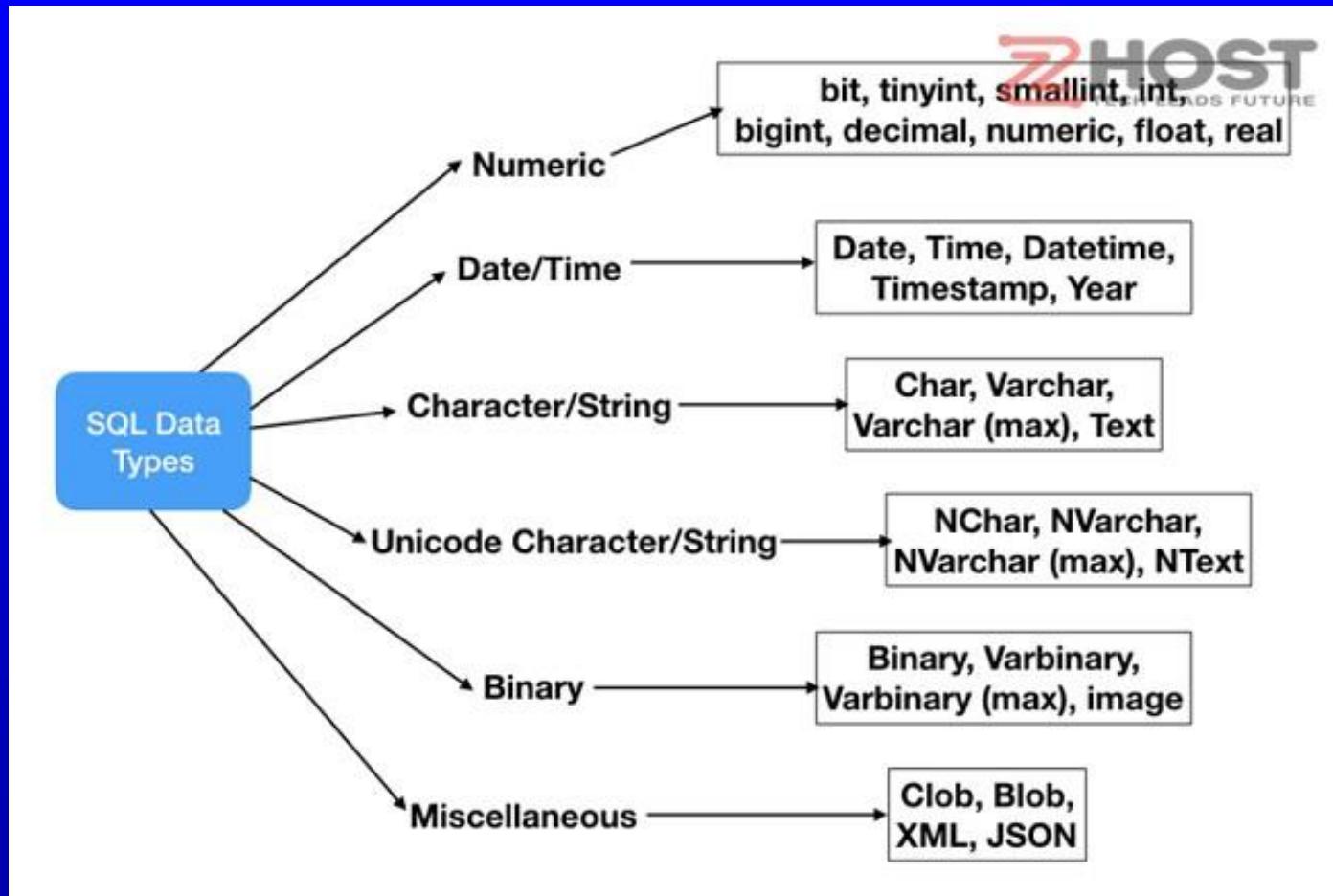


MAIN MENU

EXIT

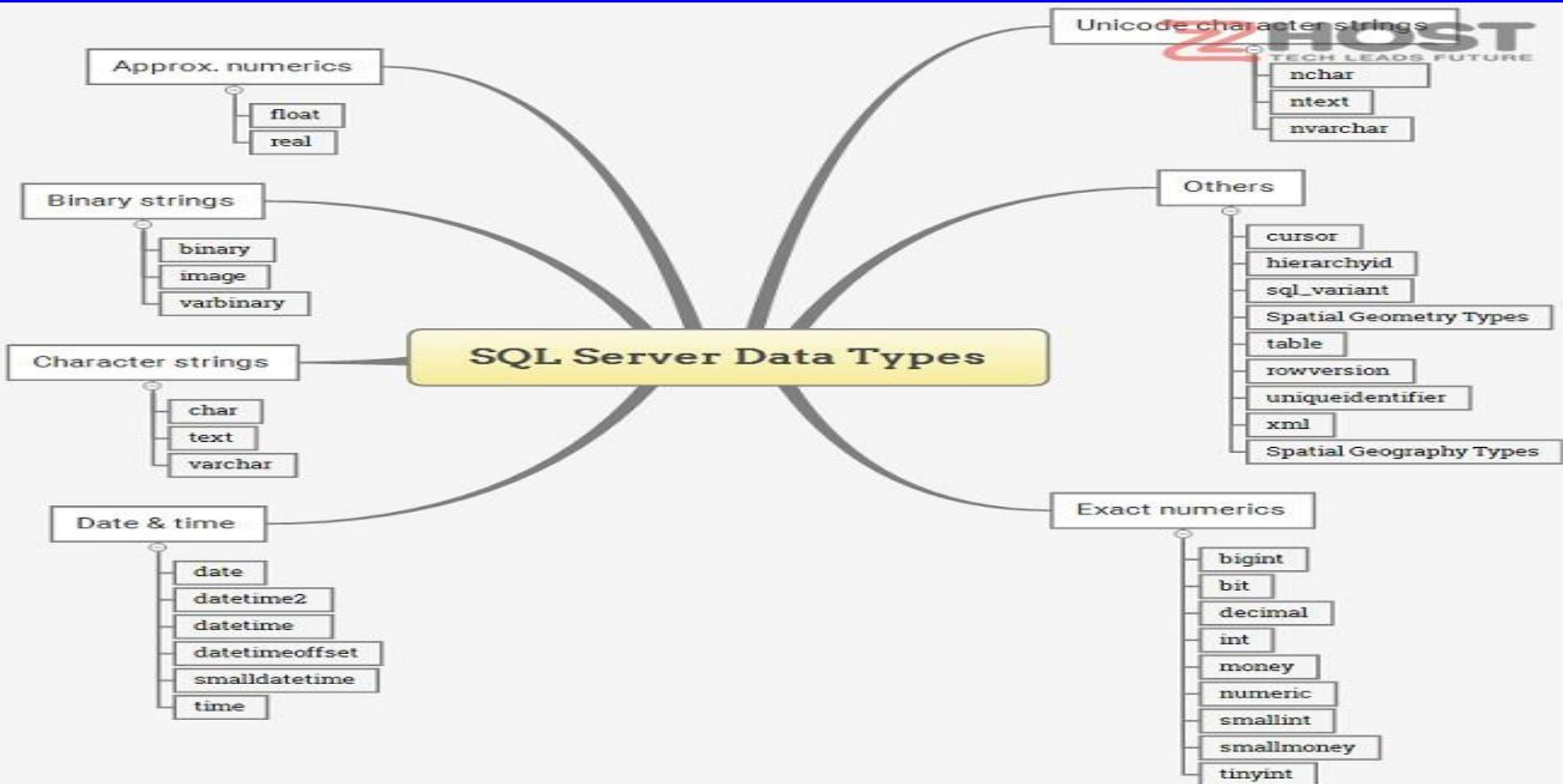
TỔNG QUAN SQL

5. Các kiểu dữ liệu căn bản (cont)



TỔNG QUAN SQL

5. Các kiểu dữ liệu căn bản (cont)



Về chương IV

< >

MAIN MENU

EXIT

TỔNG QUAN SQL

6. Độ ưu tiên toán tử

Mức ưu tiên	Phép toán
1	() , * (Multiply), / (Division), % (Modulo)
2	+ (Positive), - (Negative), + (Add), (+ Concatenate), - (Subtract),
3	=, >, <, >=, <=, <>, !=, !>, !< (Comparison operators)
4	NOT
5	AND
6	ALL, ANY, BETWEEN, IN, LIKE, OR, SOME
7	= (Assignment)

Về chương IV



MAIN MENU

EXIT

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

1. Tạo bảng (syntax):

CREATE TABLE <tên_bảng>

<tên_cột_1> <kiểu dữ liệu> [NOT NULL],

...

<tên_cột_N> <kiểu dữ liệu> [NOT NULL],

[CONSTRAINT <tên ràng buộc> <kiểu ràng buộc>]

);

Về chương IV

<

>

MAIN MENU

EXIT

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

trong đó:

<Tên_bảng>: Tên của bảng quan hệ cần tạo, tuân theo quy tắc đặt tên.

<Tên_cột>: Tên của một cột, phải là duy nhất trong bảng.

<RBTV_én>: ✓ NOT NULL: không chứa giá trị NULL.

✓ **Khóa chính**: xác định bởi PRIMARY KEY (NOT NULL).

✓ **Khóa ngoại**: FOREIGN KEY <cột> REFERENCES
<Bảng>(Cột) [ON DELETE SET NULL
ON UPDATE CASCADE]

✓ **Các tham số khác**: UNIQUE, CHECK, DEFAULT.

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

2. Thêm 1 cột đối với bảng

ALTER TABLE <tên_bảng>

ADD <tên_cột> <kiểu dữ liệu> [NOT NULL]

3. Sửa kiểu dữ liệu của 1 cột đối với bảng

ALTER TABLE <tên_bảng>

ALTER COLUMN <tên_cột> <kiểu dữ liệu mới>

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

4. Thêm 1 ràng buộc (trên cột) đối với bảng

ALTER TABLE <tên_bảng>

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

Vdụ:

Alter Table <bảng_1>

Add Constraint FK1 Foreign Key(Column Name)
References <bảng_2>(Column Name)

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

5. Xóa 1 cột đối với bảng

ALTER TABLE <tên_bảng>

DROP COLUMN <tên_cột>

Lưu ý: trường hợp cột cần xóa có ràng buộc → xóa tất cả các ràng buộc liên quan đến cột rồi mới thực hiện lệnh xóa cột.

6. Xóa 1 ràng buộc đối với bảng

ALTER TABLE <tên_bảng>

DROP CONSTRAINT <tên_ràng_buộc>

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

7. Xóa bảng

DROP TABLE <tên_bảng>

8. Tạo index

**CREATE INDEX <tên_index> ON <tên_bảng>
(<tên_cột>) ASC|DESC**

9. Xóa index

DROP INDEX <tên_index>

10. Tạo View (Khung nhìn, Bảng ảo)

**CREATE VIEW <tên_view> [Dsách cột] AS Select_Clause
[WITH CHECK OPTION][WITH READ ONLY]**

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

10. Tạo View (Khung nhìn, Bảng ảo) - cont

- Select_Clause là một câu lệnh SELECT.
- Bảng ảo View không là 1 cấu trúc dữ liệu như Table, dữ liệu trong View phụ thuộc dữ liệu bảng cơ sở tạo nên View.
- Không thể tạo index trên View.
- View được sử dụng để: Thao tác/Bảo mật dữ liệu: chỉ cho users xem những gì cần xem, import, export dữ liệu, .v.v.

11. Xóa view

DROP VIEW <tên_view>

Về chương IV

< >

MAIN MENU

EXIT

I. THỰC TẬP TẠI LỚP

Tạo Database thực tập:

CREATE DATABASE [S_P_SP]

Go

Set môi trường thực tập:

USE [S_P_SP]

Go

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 1: Tạo Table có tên NhanVien với các cột MaNV (Mã nhân viên), HoTen, Ngaysinh, Diachi, MaPhong, Phai (Phái), Luong:

CREATE TABLE NhanVien(

MaNV char(2) NOT NULL,

HoTen varchar(30) NOT NULL,

Ngaysinh datetime,

Diachi varchar(30),

Phai char(3),

MaPhong char(2),

Luong real

);

[Về chương IV](#)

<

>

[MAIN MENU](#)

[EXIT](#)

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 2: Ràng buộc cột MaNV là khóa chính của bảng NhanVien:

CREATE TABLE NhanVien(

MaNV char(2) NOT NULL,

HoTen varchar(30) NOT NULL,

Ngaysinh datetime,

Diachi varchar(30),

MaPhong char(2),

Phai char(3),

Luong real,

PRIMARY KEY(MaNV)

);

Về chương IV

<

>

MAIN MENU

EXIT

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 3: Set MaNV là khóa chính của bảng NhanVien (viết ngắn gọn):

CREATE TABLE NhanVien(

MaNV char(2) PRIMARY KEY,

HoTen varchar(30) NOT NULL,

Ngaysinh datetime,

Diachi varchar(30),

Maphong char(2),

Phai char(3),

Luong real

);

[Về chương IV](#)

<

>

[MAIN MENU](#)

[EXIT](#)

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 4: (SQL) chỉ định giá trị mặc định ‘Hà Nội’ cho cột DiaChi trong bảng NhanVien:

```
CREATE TABLE NhanVien(  
    MaNV char(2) PRIMARY KEY,  
    HoTen varchar(30) NOT NULL,  
    Ngaysinh datetime,  
    Diachi varchar(30) default 'Hà Nội',  
    Maphong char(2),  
    Phai char(3),  
    Luong real  
);
```

[Về chương IV](#)

<

>

[MAIN MENU](#)

[EXIT](#)

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 5: Tạo bảng TinhLuong gồm các cột MaNV, NgayTinhLuong, HSLuong, SoNgayCong, LuongThang, Thuong, TongLuong với khóa của bảng là MaNV, NgayTinhLuong.

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 5: Tạo bảng TinhLuong gồm các cột MaNV, NgayTinhLuong, HSLuong, SoNgayCong, LuongThang, Thuong, TongLuong với khóa của bảng là MaNV, NgayTinhLuong.

CREATE TABLE TinhLuong(

MaNV char(2) NOT NULL,

Ngaytinhluong datetime NOT NULL,

SoNgayCong int,

Luongthang real,

Thuong real,

TongLuong real,

PRIMARY KEY(MaNV, Ngaytinhluong)

);

Về chương IV

<

>

MAIN MENU

EXIT

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 6: Chỉ ra cột MaNV trong bảng TinhLuong là khóa ngoại tham chiếu đến cột MaNV trong bảng NhanVien.

ALTER TABLE TinhLuong

**ADD CONSTRAINT FK_TinhLuong FOREIGN KEY(MaNV)
REFERENCES NhanVien(MaNV)**

Ví dụ 7: Bổ sung cột MaDv (mã đơn vị) vào bảng NhanVien.

ALTER TABLE NhanVien

ADD MaDV CHAR(2) NOT NULL DEFAULT '00'

Ví dụ 8: Xóa cột MaDV trong bảng NhanVien:

ALTER TABLE NhanVien

DROP COLUMN MaDV

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 9: Thay đổi kiểu dữ liệu cho cột Thuong trong bảng TinhLuong từ kiểu real sang kiểu int:

ALTER TABLE TinhLuong

ALTER COLUMN Thuong int

Ví dụ 10: Xóa bảng TinhLuong:

DROP TABLE TinhLuong

Lưu ý: Xóa 1 bảng là xóa định nghĩa bảng và tất cả dữ liệu, cũng như các quyền (permission) định cho bảng. Trước khi xóa một bảng, gỡ bỏ/xóa tất cả các ràng buộc toàn vẹn RBTV (các phụ thuộc giữa bảng và các đối tượng khác).

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 11: Thêm cột: Trong bảng NhanVien, thêm cột MaNQL (Mã người quản lý):

ALTER TABLE NhanVien ← ADD COLUMN MaNQL char(2)

Ví dụ 12: Thêm RBTV:

ALTER TABLE NHANVIEN

**ADD CONSTRAINT FK_NHANVIEN FOREIGN KEY
(MaNQL) REFERENCES NhanVien(MaNV)**

Ví dụ 13: Xóa RBTV:

**ALTER TABLE NhanVien ← DROP CONSTRAINT
FK_NHANVIEN**

Ví dụ 14: Xóa cột MaNQL (Mã người quản lý):

ALTER TABLE NhanVien ← DROP COLUMN MaNQL

I. ĐỊNH NGHĨA DỮ LIỆU – THỰC TẬP

Ví dụ 15: Tạo bảng PhongBan, tạo RBTV trên PhongBan:

```
CREATE TABLE PHONGBAN(MaPhong char(2) NOT NULL,  
TenPhong varchar(30), TruongPhong char(2),  
CONSTRAINT PK_PHONGBAN PRIMARY KEY (MaPhong),  
CONSTRAINT U_PHONGBAN UNIQUE (TenPhong),  
CONSTRAINT FK_PHONGBAN FOREIGN KEY  
(TruongPhong) REFERENCES NHANVIEN (MaNV)  
)
```

Ví dụ 16: Xóa bảng PhongBan:

```
DROP TABLE PhongBan
```

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

Ví dụ 17:

Database S_P_SP:

$S(SID, SName, Status, City);$

$P(PID, PName, Color, Weight);$

trong đó,

$SP(SID, PID, Quantity, SDate);$

(Suppliers) S: Nhà cung cấp;

(Products) P: Sản phẩm/mặt hàng;

(Suppliers_Products) SP: Danh sách các mặt hàng đã bán;

[Về chương IV](#)

< >

[MAIN MENU](#)

[EXIT](#)

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

Ví dụ 17: $S(SID, SName, Status, City);$ $SP(SID, PID, Qty, SDate);$

$P(PID, PName, Color, Weight);$

CREATE TABLE S

(

SID char(2) PRIMARY KEY,

Sname varchar(30),

Status int,

City varchar(30)

);

[Về chương IV](#)

<

>

[MAIN MENU](#)

[EXIT](#)

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

2. Ví dụ 18: $S(SID, SName, Status, City);$ $SP(SID, PID, Qty);$

```
CREATE TABLE P(  
    PID char(2),  
    Pname varchar(30),  
    Color varchar(10),  
    Weight int,  
    PRIMARY KEY(PID)  
);
```

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

Ví dụ 19: $S(SID, SName, Status, City); SP(SID, PID, Qty, SDate);$

CREATE TABLE SP($P(PID, PName, Color, Weight);$

SID char(2) NOT NULL, PID char(2) NOT NULL,

Qty int, $Sdate$ date

-- CONSTRAINT PK PRIMARY KEY(SID, PID)

);

Thêm 1 cột Tính thuế với bảng SP(SID, PID, Quantity):

ALTER TABLE SP

ADD Tax real;

I. ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

Ví dụ 20: *Thêm ràng buộc với bảng SP(SID, PID, Quantity, Price):*

ALTER TABLE SP

ADD CONSTRAINT FK_SP1 FOREIGN KEY(S_ID) REFERENCES S(S_ID);

ALTER TABLE SP

ADD CONSTRAINT FK_SP2 FOREIGN KEY(P_ID) REFERENCES P(P_ID);

Xóa ràng buộc PK thuộc bảng SP(S_ID, P_ID):

ALTER TABLE SP

DROP CONSTRAINT PK

Về chương IV

<

>

MAIN MENU

EXIT

III. CÂU LỆNH TRUY VẤN TRONG SQL

Khối lệnh SELECT ... FROM ... WHERE ...

SQL là ngôn ngữ hỏi đáp (truy vấn), chuẩn mực & điển hình. Phép toán cơ bản trong SQL là phép ánh xạ → mô tả bởi khối lệnh SELECT - FROM - WHERE, trong đó:

- mệnh đề SELECT ↔ phép chiếu (Π) của đại số quan hệ.
- mệnh đề FROM ↔ phép tích Decac (\times) của đại số quan hệ.
- mệnh đề WHERE ↔ phép chọn (σ) của đại số quan hệ.

$$\pi_{A_1, A_2, \dots, A_n}(\sigma_P(R_1 \times R_2 \times \dots \times R_n))$$

↔

SELECT A₁, A₂, ..., A_n FROM R₁, R₂, ..., R_n WHERE P

III. CÂU LỆNH TRUY VẤN TRONG SQL

Cấu trúc cơ sở:

SELECT (π): Liệt kê các cột mong muốn trong kết quả của một truy vấn.

FROM (σ): Liệt kê các bảng cần tra cứu để đánh giá kết quả của biểu thức.

WHERE (\times): Xác định các bản ghi thỏa điều kiện \rightarrow kết xuất kết quả.

Mỗi truy vấn SQL \rightarrow có thể được dịch thành một biểu thức đại số quan hệ.

Về chương IV

< >

MAIN MENU

EXIT

III. CÂU LỆNH TRUY VẤN TRONG SQL

Dạng tổng quát khối SELECT - FROM - WHERE:

```
SELECT [DISTINCT] <Danh sách cột>|*|<bthức số học>
FROM <Danh sách bảng>|<Danh sách view>
[WHERE <biểu thức điều kiện>]
[GROUP BY <Dsách cột gom nhóm> [HAVING <biểu
thức điều kiện trên nhóm>][WITH ROLLUP|CUBE]]
[ORDER BY <Danh sách cột>|<biểu thức> [ASC|DESC]]
[UNION|INTERSECT|EXCEPT <câu truy vấn>]
[COMPUTE|FOR|BROWSE|INTO|Top N]
```

III. CÂU LỆNH TRUY VẤN TRONG SQL

Mệnh đề WHERE được biểu diễn với các dạng sau:

WHERE [NOT] <bthíc> <phép so sánh> <bthíc>

WHERE [NOT] <tên cột> [NOT] LIKE <xâu ký tự>

WHERE [NOT] <bthíc> [NOT] IN ({dsách|query})

WHERE [NOT] EXISTS (<câu truy vấn con|subquery>)

WHERE [NOT] <bthíc> <phép so sánh> {SOME | ANY | ALL(câu truy vấn con)}

WHERE [NOT] <bthíc logic>

WHERE [NOT] <bthíc logic> {AND|OR} [NOT] <bthíc logic>

III. CÂU LỆNH TRUY VẤN TRONG SQL

1. Truy vấn đơn giản trên 1 bảng (tìm kiếm không điều kiện)

S(SID, Sname, Status, City);

P(PID, Pname, Color, Weight);

SP(SID, PID, Quantity, Sdate);

Ví dụ 21: Đưa ra PID (mã các mặt hàng đã bán):

Results	
p#	
1	P1
2	P1
3	P2
4	P3
5	P1
6	P2
7	P4

III. CÂU LỆNH TRUY VẤN TRONG SQL

1. Truy vấn đơn giản trên 1 bảng (tìm kiếm không điều kiện)

S(SID, Sname, Status, City);

P(PID, Pname, Color, Weight);

SP(SID, PID, Qty, SDate);

Ví dụ 21: Đưa ra PID (mã các mặt hàng đã bán):

SELECT PID

FROM S_P_SP.dbo.SP

Nhận xét:

- Kết quả của truy vấn SQL là 1 bảng.
- Vậy, k.quả query trên → 1 bảng chứa cột P_ID

Results	
p#	
1	P1
2	P1
3	P2
4	P3
5	P1
6	P2
7	P4

III. CÂU LỆNH TRUY VẤN TRONG SQL

1. Truy vấn đơn giản trên 1 bảng (tìm kiếm không điều kiện)

- RA trên 1 quan hệ → 1 tập hợp → các bộ lặp lại không xuất hiện trong quan hệ.
- Trên thực tế, loại bỏ các bộ trùng lặp → tồn thời gian → SQL cho phép các hàng lặp lại trong bảng → Loại bỏ trùng lặp bằng từ khóa DISTINCT.

Ví dụ 22: Đưa ra mã P# (không trùng lặp) của các mặt hàng đã cung cấp: *SP(SID, PID, Qty, SDate)*;

	p#
1	P1
2	P2
3	P3
4	P4

III. CÂU LỆNH TRUY VẤN TRONG SQL

1. Truy vấn đơn giản trên 1 bảng (tìm kiếm không điều kiện)

- RA trên 1 quan hệ → 1 tập hợp → các bộ lặp lại không xuất hiện trong quan hệ.
- Trên thực tế, loại bỏ các bộ trùng lặp → tồn thời gian → SQL cho phép các hàng lặp lại trong bảng → Loại bỏ trùng lặp bằng từ khóa DISTINCT.

Ví dụ 22: Đưa ra mã PID (không trùng lặp) của các mặt hàng đã bán: *SP(SID, PID, Qty, SDate);*

```
SELECT DISTINCT PID  
FROM S_P_SP.dbo.SP
```

Results	
p#	
1	P1
2	P2
3	P3
4	P4

III. CÂU LỆNH TRUY VẤN TRONG SQL

1. Truy vấn đơn giản trên 1 bảng (tìm kiếm không điều kiện)

Ví dụ 23: Cho biết mã hiệu (SID) & số lượng các mặt hàng đã bán:

SP(SID, PID, Qty, SDate);

Ví dụ 24: Cho biết mã số (PID) các mặt hàng khác nhau đã bán:

Ví dụ 25: Sử dụng (*) hiển thị tất cả các dòng, cột (Select All) bảng SP:

III. CÂU LỆNH TRUY VẤN TRONG SQL

1. Truy vấn đơn giản trên 1 bảng (tìm kiếm không điều kiện)

Ví dụ 23: Cho biết mã hiệu (SID) & số lượng các mặt hàng đã bán:

```
SELECT SID, Qty          SP(SID, PID, Qty, SDate);  
FROM S_P_SP.dbo.SP
```

Ví dụ 24: Cho biết mã số (PID) các mặt hàng khác nhau đã bán:

```
SELECT DISTINCT PID  
FROM SP
```

Ví dụ 25: Sử dụng (*) hiển thị tất cả các dòng, cột (Select All) bảng SP:

```
SELECT *  
FROM SP
```

III. CÂU LỆNH TRUY VẤN TRONG SQL

2. Truy vấn đơn giản trên 1 bảng (tìm kiếm có điều kiện, WHERE)

WHERE <biểu thức điều kiện>

Biểu thức điều kiện thường sử dụng:

- Các toán tử so sánh =, <>, >, >=, <, <=
- Phép so sánh BETWEEN, NOT BETWEEN
- Phép kiểm tra thành viên tập hợp: IN, NOT IN
- Kiểm tra các giá trị rỗng hay chưa biết: IS NULL, IS NOT NULL
- Toán tử LIKE, NOT LIKE
- Các phép toán logic: AND, OR, NOT

III. CÂU LỆNH TRUY VẤN TRONG SQL

2. Truy vấn đơn giản trên 1 bảng (tìm kiếm có điều kiện, WHERE)

Ví dụ 26: Tìm mã số (SID) của các hàng đã bán mặt hàng P2:

SP(SID, PID, Qty, SDate);

Ví dụ 27: Tìm mã số SID của các hàng đã cung cấp các mặt hàng có số lượng giữa 10 và 20:

III. CÂU LỆNH TRUY VẤN TRONG SQL

2. Truy vấn đơn giản trên 1 bảng (tìm kiếm có điều kiện, WHERE)

Ví dụ 26: Tìm mã số (SID) của các hàng đã bán mặt hàng P2:

```
SELECT SID          SP(SID, PID, Qty, SDate);  
FROM SP  
WHERE PID = 'P2'
```

Ví dụ 27: Tìm mã số SID của các hàng đã cung cấp các mặt hàng có số lượng giữa 10 và 20:

```
SELECT SID  
FROM SP  
WHERE Qty >= 10 AND Qty <= 20
```

III. CÂU LỆNH TRUY VẤN TRONG SQL

3. Truy vấn đơn giản trên 1 bảng (tìm kiếm có xử lý xâu ký tự)

```
/*
SQL sử dụng ký tự '%' thay thế 1 xâu con/chuỗi con; ký
tự '_' thay thế 1 ký tự bất kỳ
*/
```

Ví dụ:

- A%B: Xâu ký tự bắt đầu là 'A' và kết thúc là 'B'.
- %A: Xâu bất kỳ, kết thúc là ký tự 'A'.
- A_B: Xâu gồm 3 ký tự, ký tự thứ 2 là ký tự bất kỳ.

III. CÂU LỆNH TRUY VẤN TRONG SQL

3. Truy vấn đơn giản trên 1 bảng (tìm kiếm có xử lý xâu ký tự)

Ví dụ 28: Đưa ra thông tin hàng cung ứng có tên Sant hay Santa gì đó:

$S(SID, Sname, Status, City);$

Ví dụ 29: Đưa ra thông tin hàng cung ứng có tên kết thúc là ‘ikes’:

III. CÂU LỆNH TRUY VẤN TRONG SQL

3. Truy vấn đơn giản trên 1 bảng (tìm kiếm có xử lý xâu ký tự)

Ví dụ 28: Đưa ra thông tin hàng cung ứng có tên Sant hay Santa gì đó:

```
SELECT *          S(SID, Sname, Status, City);  
FROM S  
WHERE Sname LIKE '%San%'
```

Ví dụ 29: Đưa ra thông tin hàng cung ứng có tên kết thúc là ‘ikes’:

```
SELECT *  
FROM S  
WHERE Sname LIKE '%ikes'
```

III. CÂU LỆNH TRUY VẤN TRONG SQL

4. Truy vấn trên nhiều bảng

$S(SID, SName, Status, City); SP(SID, PID, Qty, SDate);$

Ví dụ 30: Đưa ra Sname của các hãng đã bán mặt hàng ‘P2’:

Truy vấn đòi hỏi tra cứu đồng thời trên 2 bảng S và SP → phép kết tự nhiên (gồm 1 phép tích Decac, 1 phép chọn & 1 phép chiếu):

$$\pi_{Sname}(\sigma_{P='P2'}(S \times SP))$$

Trong SQL, truy vấn trên được biểu diễn?

III. CÂU LỆNH TRUY VẤN TRONG SQL

4. Truy vấn trên nhiều bảng

$S(SID, SName, Status, City); SP(SID, PID, Qty, SDate);$

Ví dụ 30: Đưa ra Sname của các hãng đã bán mặt hàng ‘P2’:

Truy vấn đòi hỏi tra cứu đồng thời trên 2 bảng S và SP → phép kết tự nhiên (gồm 1 phép tích Decac, 1 phép chọn & 1 phép chiếu):

$$\pi_{Sname}(\sigma_{P='P2'}(S \times SP))$$

Trong SQL, truy vấn trên được biểu diễn:

SELECT Sname

FROM S, SP

WHERE (S.SID=SP.SID) AND (PID= ‘P2’)

III. CÂU LỆNH TRUY VẤN TRONG SQL

5. Truy vấn trên nhiều bảng + phép đổi tên

S(SID, SName, Status, City); SP(SID, PID, Qty, SDate);

Ví dụ 31: Đưa ra Alias (Sname) của các hàng đã bán mặt hàng ‘P2’:

SQL cung cấp 1 cơ chế cho phép đổi tên, cú pháp:

Old_Name AS New_Name

```
SELECT Sname AS TenNhaCungCap  
FROM S, SP  
WHERE S.SID=SP.SID AND PID='P2'
```

Kết quả truy vấn → 1 bảng có 1 cột tên là TenNhaCungCap.

III. CÂU LỆNH TRUY VẤN TRONG SQL

5. Truy vấn trên nhiều bảng + phép đổi tên (cont)

Ví dụ 32: Đưa ra tên và địa chỉ của các hàng cung ứng có Status>100:

$S(SID, SName, Status, City);$

-- Mệnh đề AS *rất có ích* trong: $SP(SID, PID, Qty, SDate);$

Ví dụ 33: Đưa ra SID của các hàng đã bán cả 2 mặt hàng P1 và P2:

$\Pi_{SID}(\sigma_{PID='P1'}(SP)) \cap \Pi_{SID}(\sigma_{PID='P2'}(SP))$

III. CÂU LỆNH TRUY VẤN TRONG SQL

5. Truy vấn trên nhiều bảng + phép đổi tên (cont)

Ví dụ 32: Đưa ra tên và địa chỉ của các hàng cung ứng có Status>100:

SELECT Sname, City AS Address

FROM S WHERE Status>100

S(SID, SName, Status, City);

Mệnh đề AS *rất có ích* trong:

SP(SID, PID, Qty, SDate);

Ví dụ 33: Đưa ra SID của các hàng đã bán cả 2 mặt hàng P1 và P2:

SELECT T1.S_ID

FROM SP AS T1, SP AS T2

WHERE ((T1.SID=T2.SID) AND (T1.PID='P1') AND (T2.PID='P2'))

III. CÂU LỆNH TRUY VẤN TRONG SQL

6. Tìm kiếm có sắp xếp

Nhắc lại:

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

Ví dụ 34: Tìm tên (Pname) và số hiệu (PID) của các mặt hàng màu đỏ, sắp xếp theo thứ tự giảm dần của PID:

P (PID, Pname, Color, Weight);

Ghi chú: ASC là giá trị Default (Ascending).

III. CÂU LỆNH TRUY VẤN TRONG SQL

6. Tìm kiếm có sắp xếp

Nhắc lại:

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

Ví dụ 34: Tìm tên (Pname) và số hiệu (PID) của các mặt hàng màu đỏ, sắp xếp theo thứ tự giảm dần của PID:

```
SELECT Pname, PID          P (PID, Pname, Color, Weight);
      FROM P
     WHERE Color='Red' ORDER BY PID DESC;
```

Ghi chú: ASC là giá trị Default (Ascending).

III. CÂU LỆNH TRUY VẤN TRONG SQL

7. Tìm kiếm nhờ mệnh đề GROUP BY

Nhắc lại:

GROUP BY <Dsách cột gom nhóm> [HAVING <bíểu thức điều kiện trên nhóm>]

Ghi chú:

- HAVING lọc điều kiện, luôn đi sau GROUP BY.
- Sau HAVING là bíểu thức điều kiện, bíểu thức điều kiện này *không tác động lên toàn bộ bảng mà chỉ tác động lên từng nhóm các bản ghi đã chỉ ra tại GROUP BY*.
- HAVING kết hợp các phép toán thống kê: AVG, SUM, MAX, MIN, COUNT.

III. CÂU LỆNH TRUY VẤN TRONG SQL

7. Tìm kiếm nhờ mệnh đề GROUP BY - cont

Ví dụ 34 (refer 33, 57): Tìm SID cung cấp cả 2 mặt hàng P1 và P2:

SP(SID, PID, Qty, SDate);

Clause: CONTAINS

Ví dụ 35: Tìm SID bán ít nhất 2 mặt hàng:

III. CÂU LỆNH TRUY VẤN TRONG SQL

7. Tìm kiếm nhòm mệnh đề GROUP BY - cont

Ví dụ SQL (33, 57): Tìm SID cung cấp cả 2 mặt hàng P1 và P2:

SELECT SID FROM SP *SP(SID, PID, Qty, SDate);*

GROUP BY SID

HAVING SET(PID) CONTAINS('P1', 'P2')

Ví dụ 35: Tìm SID bán ít nhất 2 mặt hàng:

SELECT SID

FROM SP

GROUP BY SID

HAVING COUNT(PID)>=2

III. CÂU LỆNH TRUY VẤN TRONG SQL

8. Tìm kiếm dựa trên các phép toán tập hợp

Các phép toán tập hợp UNION, INTERSECT và EXCEPT tương đương với các phép \cup , \cap và $-$ của đại số quan hệ; Các bảng phải khả hợp (trùng tên cột, cột trùng tên có cùng miền giá trị)

Ví dụ SQL (38): Cho biết SID các hàng chưa cung ứng mặt hàng nào:

$S(SID, Sname, Status, City);$

$SP(SID, PID, Qty, SDate);$

III. CÂU LỆNH TRUY VẤN TRONG SQL

8. Tìm kiếm dựa trên các phép toán tập hợp

Các phép toán tập hợp UNION, INTERSECT và EXCEPT tương đương với các phép \cup , \cap và $-$ của đại số quan hệ; Các bảng phải khả hợp (cùng tập tên cột, các cột trùng tên có cùng miền giá trị)

Ví dụ SQL (38): Cho biết SID các hàng chưa cung ứng mặt hàng nào:

SELECT SID

FROM S

S(SID, Sname, Status, City);

EXCEPT

SELECT SID

SP(SID, PID, Qty, SDate);

FROM SP

[Về chương IV](#)

<

>

[MAIN MENU](#)

[EXIT](#)

III. CÂU LỆNH TRUY VẤN TRONG SQL

9. Truy vấn con

SQL cung cấp cơ chế cho phép lồng truy vấn con (sub-query). Mỗi truy vấn con là 1 biểu thức SELECT-FROM-[WHERE].

SQL cho phép dùng các phép so sánh: <ALL, <=ALL, =ALL, .v.v.

SQL cho phép sử dụng lượng từ ANY (hay SOME) trong các phép: < SOME; > SOME; <= SOME; >= SOME; = SOME và <> SOME.
Đễ thấy:

1. = SOME \equiv IN
2. <> SOME \neq NOT IN
3. <> ALL \equiv NOT IN

III. CÂU LỆNH TRUY VẤN TRONG SQL

10. So sánh tập hợp: Sử dụng toán tử IN

Ví dụ 36: Sử dụng toán tử IN, tìm mã (SID), tên (Sname) các hàng đã bán mặt hàng P2 (vdu 26):

$S(SID, Sname, Status, City);$

$SP(SID, PID, Qty, SDate);$

III. CÂU LỆNH TRUY VẤN TRONG SQL

10. So sánh tập hợp: Sử dụng toán tử IN

Ví dụ 36: Sử dụng toán tử IN, tìm mã (SID), tên (Sname) các hàng đã bán mặt hàng P2 (vdu 26):

```
SELECT SID, Sname  
FROM S  
WHERE SID IN  
(SELECT SID  
FROM SP  
WHERE PID='P2')
```

*S(SID, Sname, Status, City);
SP(SID, PID, Qty, SDate);*

III. CÂU LỆNH TRUY VẤN TRONG SQL

10. So sánh tập hợp: Sử dụng toán tử ALL

Ví dụ 37: Đưa ra Sname của các hãng có hiện trạng (Status) là lớn nhất:

Ví dụ 38: Đưa ra danh sách hãng chưa bán mặt hàng nào (subquery):

S(SID, Sname, Status, City);

SP(SID, PID, Qty, SDate);

/ <>ALL ≡ NOT IN */ /* Cung ứng tất cả các mặt hàng ????? */*

Về chương IV

<

>

MAIN MENU

EXIT

III. CÂU LỆNH TRUY VẤN TRONG SQL

10. So sánh tập hợp: Sử dụng toán tử ALL

Ví dụ 37: Đưa ra Sname của các hãng có hiện trạng (Status) là lớn nhất:

SELECT Sname, Status FROM S

WHERE Status >= ALL(SELECT Status FROM S)

Ví dụ 38: đưa ra danh sách hãng chưa bán mặt hàng nào (sub-query):

SELECT * FROM S WHERE SID <> ALL(

**SELECT SID
FROM SP
)**

S(SID, Sname, Status, City);

SP(SID, PID, Qty, SDate);

/ <>ALL ≡ NOT IN */ /* Cung ứng tất cả các mặt hàng ????? */*

Về chương IV

<

>

MAIN MENU

EXIT

III. CÂU LỆNH TRUY VẤN TRONG SQL

11. Kiểm tra các bảng rỗng: Cấu trúc EXISTS

SQL bao hàm đặc trưng kiểm tra liệu 1 truy vấn có kết quả khác rỗng hay không: **Cấu trúc EXISTS** cho giá trị true nếu kết quả truy vấn khác rỗng. Tương tự kiểm tra không tồn tại: **NOT EXISTS**

Ví dụ 39: Dưa ra Sname của các hãng đã bán ít nhất 1 mặt hàng nào đó:

S(SID, Sname, Status, City);

SP(SID, PID, Qty, SDate);

III. CÂU LỆNH TRUY VẤN TRONG SQL

11. Kiểm tra các bảng rỗng: Cấu trúc EXISTS

SQL bao hàm đặc trưng kiểm tra liệu 1 truy vấn có kết quả khác rỗng hay không: **Cấu trúc EXISTS** cho giá trị true nếu kết quả truy vấn khác rỗng. Tương tự kiểm tra không tồn tại: **NOT EXISTS**

Ví dụ 39: Dưa ra Sname của các hãng đã bán ít nhất 1 mặt hàng nào đó:

SELECT Sname

S(SID, Sname, Status, City);

FROM S

SP(SID, PID, Qty, SDate);

WHERE EXISTS(

SELECT * FROM SP

WHERE SP.SID=S.SID)

III. CÂU LỆNH TRUY VẤN TRONG SQL

12. Các hàm thư viện

- SQL cung cấp hàm thư viện: AVG, SUM, COUNT, MAX, MIN, ...
- Đầu vào với AVG và SUM phải là số, riêng hàm COUNT có thể có đối số là *, như Count(*).
- *Các hàm thư viện không áp dụng trên biểu thức logic của WHERE:*

SELECT * FROM SP WHERE Qty=MAX(Qty) ???

III. CÂU LỆNH TRUY VẤN TRONG SQL

12. Các hàm thư viện

SQL cung cấp hàm thư viện: AVG, SUM, COUNT, MAX, MIN, ... Đầu vào với AVG và SUM phải là số, riêng hàm COUNT có thể có đối số là *, như Count(*) .

Các hàm thư viện không áp dụng trên biểu thức logic của WHERE:

SELECT * FROM SP WHERE Qty=MAX(Qty) ← báo lỗi ← sử dụng trong SELECT ...GROUP BY ... HAVING, hoặc subquery.

Ví dụ 40: Tính trọng lượng trung bình của các mặt hàng màu đỏ:

P(PID, Pname, Color, Weight);

III. CÂU LỆNH TRUY VẤN TRONG SQL

12. Các hàm thư viện

SQL cung cấp hàm thư viện: AVG, SUM, COUNT, MAX, MIN, ... Đầu vào với AVG và SUM phải là số, riêng hàm COUNT có thể có đối số là *, như Count(*) .

Các hàm thư viện không áp dụng trên biểu thức logic của WHERE:

SELECT * FROM SP WHERE Qty=MAX(Qty) ← báo lỗi ← sử dụng trong SELECT ... GROUP BY ... HAVING, hoặc subquery.

Ví dụ 40: Tính trọng lượng trung bình của các mặt hàng màu đỏ:

SELECT AVG(Weight) *P(PID, Pname, Color, Weight);*
FROM P
WHERE Color='Red';

III. CÂU LỆNH TRUY VẤN TRONG SQL

12. Các hàm thư viện (cont)

Các hàm thư viện không chỉ áp dụng trên từng record thỏa điều kiện → áp dụng hàm tập hợp trên Group.

Ví dụ 41: Với mỗi hàng cung ứng, đưa ra SID và đếm SID bán bao nhiêu mặt hàng:
SP(SID, PID, Price, Qty, SDate);

Hàm Count(*) đếm số bản ghi thỏa yêu cầu truy vấn mà ko quan tâm tới các cột của bảng, ko quan tâm giá trị NULL (phân biệt với Count(A)).

Về chương IV

< >

MAIN MENU

EXIT

III. CÂU LỆNH TRUY VẤN TRONG SQL

12. Các hàm thư viện (cont)

Các hàm thư viện không chỉ áp dụng trên từng record thỏa điều kiện → áp dụng hàm tập hợp trên Group.

Ví dụ 41: Với mỗi hàng cung ứng, đưa ra SID và đếm SID bán bao nhiêu mặt hàng:

SP(SID, PID, Price, Qty, SDate);

```
SELECT SID, COUNT(Distinct PID) as NumProducts  
FROM SP  
GROUP BY SID
```

Hàm Count(*) đếm số bản ghi thỏa yêu cầu truy vấn mà ko quan tâm tới các cột của bảng, ko quan tâm giá trị NULL (phân biệt với Count(A)).

[Về chương IV](#)

< >

[MAIN MENU](#)

[EXIT](#)

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có các quan hệ: S, P và SP.

Ví dụ 42: Đưa ra thông tin về 3 lần bán hàng đầu tiên:

SP(SID, PID, Price, Qty, SDate);

Ví dụ 43: Thêm cột Price, Tax vào SP (sau đó nhập value tương ứng):

Ví dụ 44: Đưa ra mã nhà cung cấp, mã mặt hàng, số lượng bán, giá bán, thành tiền (thành tiền bằng giá bán nhân số lượng):

Ví dụ 45: Vẫn ví dụ trên, sử dụng phép đổi tên (AS) gợi nhớ all column:

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có các quan hệ: S, P và SP.

Ví dụ 42: Đưa ra thông tin về 3 lần bán hàng đầu tiên:

SELECT TOP 3 * FROM SP *SP(SID, PID, Price, Qty, SDate);*

Ví dụ 43: Thêm cột Price, Tax vào SP (sau đó nhập value tương ứng):

ALTER TABLE SP

ADD Price real; -- ADD Tax real;

Ví dụ 44: Đưa ra mã nhà cung cấp, mã mặt hàng, số lượng bán, giá bán, thành tiền (thành tiền bằng giá bán nhân số lượng):

SELECT SID, PID, Qty, Price, Qty*Price As Amount FROM SP

Ví dụ 45: Vẫn ví dụ trên, sử dụng phép đổi tên (AS) gợi nhớ all column:

```
SELECT SID AS MãĐại lý, PID AS MãSảnphẩm, Qty AS Sốlượng,  
Price AS Giábán, Qty*Price AS Thànhtiền FROM SP
```

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 46: Tính tổng doanh thu:

SP(SID, PID, Sdate, Price, Qty, Tax);

Ví dụ 47: Đếm số lần bán hàng:

Ví dụ 48: Xem thông tin bán hàng trong ngày 03/10/2022:

Ví dụ 49: Xem thông tin về các sản phẩm có giá từ 16 đến 18:

Ví dụ 50: Toán tử OR → show các lần bán sản phẩm P1 hoặc P2 hoặc P3:

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 46: Tính tổng doanh thu:

```
SELECT Sum(Qty*Price) AS Total_Revenue  
FROM SP  
          SP(SID, PID, Sdate, Price, Qty, Tax);
```

Ví dụ 47: Đếm số lần bán hàng:

```
SELECT Count(*) FROM SP
```

Ví dụ 48: Xem thông tin bán hàng trong ngày 03-10-2022:

```
Select * From SP Where Convert(nvarchar, Sdate, 103)=‘03/10/2022’
```

Ví dụ 49: Xem thông tin về các sản phẩm có giá từ 16 đến 18:

```
SELECT * FROM SP WHERE Price BETWEEN 16 AND 18
```

Ví dụ 50: Toán tử OR → show các lần bán sản phẩm P1 hoặc P2 hoặc P3:

```
SELECT * FROM SP  
WHERE PID='P1' OR PID='P2' OR PID='P3'
```

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 51: Toán tử IN → show các lần bán sản phẩm P1 hoặc P2 hoặc P3:

Ví dụ 52: Đưa ra các hàng cung ứng có địa chỉ bắt đầu bằng ‘3’:

Ví dụ 53: Xem thông tin về lần bán chạy nhất của sản phẩm (số lượng):

SP(SID, PID, Price, Qty, Sdate);

Ví dụ 54: Đưa ra mã & tổng số lượng đã bán của từng sản phẩm:

Ví dụ 55: Sắp xếp các sản phẩm theo t/tự tăng dần của Amount:

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 51: Toán tử IN → show các lần bán sản phẩm P1 hoặc P2 hoặc P3:

SELECT * FROM SP WHERE PID IN ('P1','P2','P3')

Ví dụ 52: Đưa ra các hàng cung ứng có địa chỉ bắt đầu bằng ‘3’:

SELECT * FROM S WHERE City LIKE '3%'

Ví dụ 53: Xem thông tin về lần bán chạy nhất của sản phẩm (số lượng):

SELECT * FROM SP SP(SID, PID, Price, Qty, Tax, Sdate);

WHERE Qty=(SELECT Max(Qty) FROM SP)

Ví dụ 54: Đưa ra mã & tổng số lượng đã bán của từng sản phẩm:

SELECT PID, Sum(Qty) FROM SP GROUP BY PID

Ví dụ 55: Sắp xếp các sản phẩm theo t/tự tăng dần của Amount:

SELECT Qty*Price As Amount FROM SP ORDER BY Amount

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 56: Hiển thị t.tin bán hàng SID, sDate, Qty theo từng hàng c.cấp:

Ví dụ 57: Cho biết mã số những nhà cung cấp mặt hàng P1 và P2:

SP(SID, PID, Price, Qty, Tax, Sdate);

Ví dụ 58: Đưa ra mã và tên các nhà cung ứng đã bán mặt hàng P1.

Ví dụ 59: Liệt kê những lần bán hàng không tính thuế?

III. TRUY VĂN TRONG SQL – THỦC TẬP

Ví dụ 56: Hiển thị t.tin bán hàng SID, sDate, Qty theo từng hàng c.cấp:

SELECT SID, Sdate, Qty FROM SP GROUP BY S_ID, Sdate, Qty;

Ví dụ 57: Cho biết mã số những nhà cung cấp mặt hàng P1 và P2:

SELECT SID FROM SP WHERE PID='P1'

INTERSECT *SP(SID, PID, Price, Qty, Tax, Sdate);*

SELECT SID FROM SP WHERE PID='P2'

Ví dụ 58: Đưa ra mã và tên các nhà cung ứng đã bán mặt hàng P1.

SELECT S.SID, Sname

FROM S, SP WHERE (S.SID=SP.SID) AND (PID='P1')

Ví dụ 59: Liệt kê những lần bán hàng không tính thuế?

SELECT * FROM SP WHERE Tax IS NULL

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 60: Cho biết số lần bán mặt hàng P2:

SP(SID, PID, Price, Qty, Tax, Sdate);

Ví dụ 61: Tính hiệu số giữa lần bán nhiều nhất và lần bán ít nhất của mặt hàng P2 ?

Ví dụ 62: Tìm những mã mặt hàng bán trước ngày 06/08/2022 một ngày?

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 60: Cho biết số lần bán mặt hàng P2 :

```
SELECT COUNT(*) AS SoLuongP2 FROM SP  
WHERE PID = 'P2'          SP(SID, PID, Price, Qty, Tax, Sdate);
```

Ví dụ 61: Tính hiệu số giữa lần bán nhiều nhất và lần bán ít nhất của mặt hàng P2 ?

```
SELECT Max(Qty)-Min(Qty) AS HiệuSốMaxMin  
FROM SP WHERE PID='P2'
```

Ví dụ 62: Tìm những mã mặt hàng bán trước ngày 06/08/2022 một ngày?

```
SELECT PID FROM SP  
WHERE DateDiff(d, Sdate, '2022-08-06') = 1;
```

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 63: Với các mặt hàng đã bán, cho biết mã số mặt hàng và địa chỉ của hàng cung ứng:

$SP(SID, PID, Price, Qty, Tax, Sdate);$
 $S(SID, Sname, Status, City);$
 $P(PID, Pname, Color, Weight);$

Ví dụ 64: Với các mặt hàng đã bán, cho biết mã số, tên, màu của mặt hàng đó:

Ví dụ 65: Sử dụng toán tử ANY tìm tên những mặt hàng đã bán bởi S1:

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 63: Với các mặt hàng đã bán, cho biết mã số mặt hàng và địa chỉ của hãng cung ứng:

SP(SID, PID, Price, Qty, Tax, Sdate);

**SELECT PID, City FROM S, SP
WHERE SP.SID=S.SID**

S(SID, Sname, Status, City);

P(PID, Pname, Color, Weight);

Ví dụ 64: Với các mặt hàng đã bán, cho biết mã số, tên, màu của mặt hàng đó:

**SELECT P.PID, Pname, Color FROM P, SP
WHERE SP.PID=P.PID**

Ví dụ 65: Sử dụng toán tử ANY tìm tên những mặt hàng đã bán bởi S1:

**SELECT Pname FROM P WHERE PID=ANY(
SELECT PID FROM SP WHERE SID='S1')**

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 66: Sử dụng hàm COUNT(*) → xác định tên các hãng đã bán ít nhất một mặt hàng nào đó (vdụ 39):

Ví dụ 67: Sử dụng toán tử IN, tìm mã (SID), tên (Sname) các hãng cung ứng mặt hàng P2 (vdụ 36):

S(SID, Sname, Status, City); SP(SID, PID, Price, Qty, Tax, Sdate);

Về chương IV

< >

MAIN MENU

EXIT

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 66: Sử dụng hàm COUNT(*) → xác định tên các hàng đã bán ít nhất một mặt hàng nào đó (vdụ 39):

```
SELECT Sname FROM S  
WHERE 0 < (SELECT COUNT (*) FROM SP  
WHERE SP.SID=S.SID)
```

Ví dụ 67: Sử dụng toán tử IN, tìm mã (SID), tên (Sname) các hàng cung ứng mặt hàng P2 (vdụ 36):

```
SELECT S.SID, Sname FROM S  
WHERE 'P2' IN (  
SELECT PID FROM SP WHERE SP.SID=S.SID)
```

S(SID, Sname, Status, City); SP(SID, PID, Price, Qty, Tax, Sdate);

Về chương IV

< >

MAIN MENU

EXIT

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 68: Tìm tên các hàng đã cung cấp ít nhất một mặt hàng màu đỏ:

SP(SID, PID, Price, Qty, Tax, Sdate);

S(SID, Sname, Status, City);

P(PID, Pname, Color, Weight);

Ví dụ 69: Tìm tên các hàng đã cung cấp ít nhất một mặt hàng màu đỏ với số lượng > 100:

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 68: Tìm tên các hàng đã cung cấp ít nhất một mặt hàng màu đỏ:

```
SELECT Sname FROM S    SP(S_ID, P_ID, Price, Qty, Tax, Sdate);
WHERE S.SID=ANY(          S(S_ID, Sname, Status, City);
SELECT S.SID FROM SP, P      P(P_ID, Pname, Color, Weight);
WHERE S.SID=SP.SID AND P.PID=SP.PID AND Color='Red')
```

Ví dụ 69: Tìm tên các hàng đã cung cấp ít nhất một mặt hàng màu đỏ với số lượng > 100:

```
SELECT Sname FROM S WHERE S.SID=ANY(
SELECT S.SID FROM SP, P
WHERE S.SID=SP.SID AND P.PID=SP.PID AND Color='Red'
AND Qty>100)
```

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 70: Tạo View có tên ViewS1 chứa dữ liệu bán hàng của hãng S1:

SP(SID, PID, Price, Qty, Tax, Sdate);

Ví dụ 71: Dưa ra chỉ mã sản phẩm có tổng số lượng bán được nhiều nhất:

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 70: Tạo View có tên ViewS1 chứa dữ liệu bán hàng của hãng S1:

SP(SID, PID, Price, Qty, Tax, Sdate);

CREATE VIEW ViewS1 AS

SELECT * FROM SP WHERE SID='S1'

Ví dụ 71: Đưa ra chỉ mã sản phẩm có tổng số lượng bán được nhiều nhất:

(Để thực hiện yêu cầu này, ta tạo ra một View chứa PID và tổng số lượng đã bán của từng sản phẩm. Sau đó truy vấn trên View)

CREATE VIEW ViewP(PID, TongSL) AS

SELECT PID, SUM(Qty) FROM SP GROUP BY PID;

SELECT PID FROM ViewP

WHERE TongSL=(SELECT MAX(TongSL) FROM ViewP);

Về chương IV

< >

MAIN MENU

EXIT

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 72: Tìm tên các hàng đã cung cấp tất cả các mặt hàng:

SP(SID, PID, Price, Qty, Tax, Sdate);

S(SID, Sname, Status, City);

P(PID, Pname, Color, Weight);

Nhắc lại rằng, hàm Count(*) đếm số bản ghi thỏa yêu cầu truy vấn mà không quan tâm tới các cột của bảng; và P_ID là PK của bảng P

Về chương IV

< >

MAIN MENU

EXIT

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 72 (v1: False): Tìm tên các hãng đã cung cấp tất cả các mặt hàng:

```
SELECT Sname FROM S      SP(SID, PID, Price, Qty, Tax, Sdate);  
WHERE SID IN(           S(SID, Sname, Status, City);  
SELECT SID              P(PID, Pname, Color, Weight);  
FROM SP  
GROUP BY SID  
HAVING COUNT(PID)  
>=  
(SELECT COUNT(*) FROM P))
```

Nhắc lại rằng, hàm Count(*) đếm số bản ghi thỏa yêu cầu truy vấn mà không quan tâm tới các cột của bảng; và P_ID là PK của bảng P

[Về chương IV](#)

< >

[MAIN MENU](#)

[EXIT](#)

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 72 (v2: True): Tìm tên các hàng đã cung cấp tất cả các mặt hàng:

```
SELECT Sname FROM S      SP(SID, PID, Price, Qty, Tax, Sdate);  
WHERE S_ID IN(           S(SID, Sname, Status, City);  
SELECT S_ID                P(PID, Pname, Color, Weight);  
FROM ViewSP – ViewSP As (Select distinct S_ID, P_ID From SP)  
GROUP BY S_ID  
HAVING COUNT(P_ID)  
>=  
(SELECT COUNT(*) FROM P))
```

Nhắc lại rằng, hàm Count(*) đếm số bản ghi thỏa yêu cầu truy vấn mà không quan tâm tới các cột của bảng; và P_ID là PK của bảng P

Về chương IV

< >

MAIN MENU

EXIT

III. TRUY VẤN TRONG SQL – THỦC TẬP

Ví dụ 72 (v3: True): Tìm tên các hàng đã cung cấp tất cả các mặt hàng:

SP(SID, PID, Price, Qty, Tax, Sdate);

S(SID, Sname, Status, City);

P(PID, Pname, Color, Weight);

```
SELECT Sname FROM S
WHERE SID IN(
SELECT SID  FROM TmpSP
GROUP BY SID
HAVING COUNT(PID)
>=
(SELECT COUNT(*) FROM P))
```

TmpSP: Select distinct SID,
PID into TmpSP From SP

Nhắc lại rằng, hàm Count(*) đếm số bản ghi thỏa yêu cầu truy vấn
mà không quan tâm tới các cột của bảng; và P_ID là PK của bảng P

Về chương IV

< >

MAIN MENU

EXIT

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

PhongBan(MaPhong, TenPhong, TruongPhong);

NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);

Duan(MaDA, TenDA, MaPhong);

PhanViec(MaNV, MaDA, Thgian);

Ví dụ 73: Tìm mã nhân viên, họ tên tất cả các nhân viên nữ:

Ví dụ 74: Tìm mã, họ tên các nhân viên tham gia vào dự án có mã ‘D1’:

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

PhongBan(MaPhong, TenPhong, TruongPhong);

NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);

Duan(MaDA, TenDA, MaPhong);

PhanViec(MaNV, MaDA, Thgian);

Ví dụ 73: Tìm mã nhân viên, họ tên tất cả các nhân viên nữ:

SELECT MaNV, HoTen FROM NHANVIEN WHERE Phai='Nu'

Ví dụ 74: Tìm mã, họ tên các nhân viên tham gia vào dự án có mã ‘D1’:

**SELECT NV.MaNV, HoTen
FROM NHANVIEN NV, PHANVIEC
WHERE NV.MaNV= PHANVIEC.MaNV AND MaDA='D1'**

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

PhongBan(MaPhong, TenPhong, TruongPhong);

NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);

Ví dụ 75: Tìm mã, họ tên tất cả các nhân viên phòng Tài chính

Ví dụ 76: Tìm mã nhân viên, họ tên và lương nhân viên, trong đó lương được hiển thị tăng thêm 20% cho mọi nhân viên

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

PhongBan(MaPhong, TenPhong, TruongPhong);

NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);

Ví dụ 75: Tìm mã, họ tên tất cả các nhân viên phòng Tài chính

```
SELECT MaNV, HoTen FROM NHANVIEN NV, PHONGBAN P  
WHERE NV.MaPhong=P.MaPhong AND TenPhong = 'Tài chính'
```

Ví dụ 76: Tìm mã nhân viên, họ tên và lương nhân viên, trong đó lương được hiển thị tăng thêm 20% cho mọi nhân viên

```
SELECT MaNV, HoTen, Luong*1.2  
FROM NHANVIEN
```

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);

Ví dụ 77: Tìm tất cả những nhân viên có họ bắt đầu bằng T như
Trương, Trần ...

Ví dụ 78: Cho biết danh sách nhân viên thuộc phòng P1

Ví dụ 79: Tìm tất cả những nhân viên phòng P1 và lương tương ứng,
sắp xếp giảm dần theo lương

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);

Ví dụ 77: Tìm tất cả những nhân viên có họ bắt đầu bằng T như
Trương, Trần ...

SELECT * FROM NhanVien WHERE HoTen LIKE 'T%'

Ví dụ 78: Cho biết danh sách nhân viên thuộc phòng P1

SELECT * FROM NhanVien WHERE MaPhong='P1'

Ví dụ 79: Tìm tất cả những nhân viên phòng P1 và lương tương ứng,
sắp xếp giảm dần theo lương

SELECT * FROM NhanVien WHERE MaPhong = 'P1'

ORDER BY Luong DESC

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

*NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);
PhongBan(MaPhong, TenPhong, TruongPhong);*

Ví dụ 80: Tìm tất cả nhân viên nam làm việc trong phòng Kỹ thuật

Ví dụ 81: Cho biết danh sách nviên phòng ‘P3’ có lương >500 ?

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

*NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);
PhongBan(MaPhong, TenPhong, TruongPhong);*

Ví dụ 80: Tìm tất cả nhân viên nam làm việc trong phòng Kỹ thuật

```
SELECT * FROM NHANVIEN WHERE MaPhong IN (
    SELECT MaPhong FROM PHONGBAN
    WHERE TenPhong='Ky thuat') AND Phai = 'Nam'
```

Ví dụ 81: Cho biết danh sách nviên phòng ‘P3’ có lương >500 ?

```
SELECT * FROM NhanVien
WHERE MaPhong= 'P3' AND Luong > 500
```

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

*NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);
PhongBan(MaPhong, TenPhong, TruongPhong);*

Ví dụ 82: Đưa ra danh sách nvien phòng ‘P1’, ‘P2’ và ‘P3’:

Ví dụ 83: Đưa ra d.sách tên phòng và tổng lương các nvien trong phòng?

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

*NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);
PhongBan(MaPhong, TenPhong, TruongPhong);*

Ví dụ 82: Đưa ra danh sách nviên phòng ‘P1’, ‘P2’ và ‘P3’:

SELECT * FROM NhanVien WHERE MaPhong IN ('P1','P2','P3')

Ví dụ 83: Đưa ra d.sách tên phòng và tổng lương các nviên trong phòng?

**SELECT TenPhong, SUM(Luong) AS TongLuong
FROM NhanVien, PhongBan
WHERE NhanVien.MaPhong=PhongBan.MaPhong
GROUP BY NhanVien.MaPhong, TenPhong**

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

*NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);
PhongBan(MaPhong, TenPhong, TruongPhong);*

Ví dụ 84: Đưa ra danh sách phòng và mức lương cao nhất trong phòng?

Ví dụ 85: Cho biết mã phòng mà mức lương cao nhất ≥ 2000 ?

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

*NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);
PhongBan(MaPhong, TenPhong, TruongPhong);*

Ví dụ 84: Đưa ra danh sách phòng và mức lương cao nhất trong phòng?

```
SELECT TenPhong, MAX(Luong) AS MaxLuong  
FROM NhanVien, PhongBan  
WHERE NhanVien.MaPhong=PhongBan.MaPhong  
GROUP BY NhanVien.MaPhong, TenPhong
```

Ví dụ 85: Cho biết mã phòng mà mức lương cao nhất ≥ 2000 ?

```
SELECT MaPhong FROM NhanVien  
GROUP BY MaPhong HAVING Max(Luong)>=2000
```

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);

Ví dụ 86: Tìm tất cả nhân viên có lương lớn hơn mọi nhân viên phòng P2

```
SELECT * FROM NhanVien WHERE Luong > ALL (  
SELECT Luong FROM NhanVien WHERE MaPhong='P2')
```

Ví dụ 87: Tìm tất cả nhân viên có lương lớn hơn ít nhất một nhân viên phòng P2

```
SELECT * FROM NhanVien WHERE Luong > SOME (  
SELECT Luong FROM NhanVien WHERE MaPhong='P2')
```

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);

PhongBan(MaPhong, TenPhong, TruongPhong);

Ví dụ 88: Cho biết mức lương trung bình và cao nhất của phòng P2

Ví dụ 89: Cho biết tên phòng mà lương trung bình của các nhân viên
trong phòng lớn hơn 700.

III. TRUY VẤN TRONG SQL – THỰC TẬP

Thực tập trên CSDL có 4 quan hệ: NV, PhongBan, DuAn, PhanViec.

NhanVien(MaNV, HoTen, Ngaysinh, Diachi, MaPhong, Phai, Luong);

PhongBan(MaPhong, TenPhong, TruongPhong);

Ví dụ 88: Cho biết mức lương trung bình và cao nhất của phòng P2

```
SELECT AVG(Luong) AS LuongTrbinh, Max(Luong) AS  
MaxLuong FROM NhanVien WHERE MaPhong='P2'
```

Ví dụ 89: Cho biết tên phòng mà lương trung bình của các nhân viên
trong phòng lớn hơn 700.

```
SELECT TenPhong, AVG(Luong) FROM NhanVien, PhongBan  
WHERE NhanVien.MaPhong = PhongBan.MaPhong  
GROUP BY TenPhong HAVING AVG(Luong)>700
```

II. TRUY VẤN TRONG SQL – BÀI TẬP TẠI LỚP

Cửa hàng buôn bán tạp hóa xây dựng csdliệu quản lý các mặt hàng:

1. MATHANG(MaMH: int, TenMH: string, DonviTinh: string, DonGia: double, NgayNhap: date, NhomHang: int, SoLuongTonKho: int)
2. NHOMMATHANG (MaNhomMH: int, TenNhomMH: string);
3. KHACHHANG (MaKH: int, TenKH: string, DiaChi: string, SoĐT: string, DiemThuong: int);
4. MUAHANG (MaMH: int, MaKH: int, NgayMua: date, SoLuong: int);

II. TRUY VẤN TRONG SQL – BÀI TẬP TẠI LỚP

Q1: Đưa ra tên, địa chỉ của những khách hàng mua các mặt hàng mặt hàng thuộc nhóm "Hàng điện tử" từ ngày 01/05/2022 đến 30/05/2022.

Q2: Đưa ra tên của những mặt hàng bán chạy nhất (có doanh thu lớn nhất). Doanh thu được tính theo công thức: Số lượng bán được * Đơn giá.

Q3: Đếm số lượng mặt hàng thuộc nhóm mặt hàng có tên: "Hàng gia dụng".

Q4: Đưa ra tên, địa chỉ, số điện thoại của những khách hàng có tổng giá trị hàng mua là lớn nhất trong khoảng thời gian từ ngày 01/1/2022 đến 25/10/2022. Trong đó, Tổng giá trị hàng = Σ (Số lượng * Đơn giá);

Về chương IV

< >

MAIN MENU

EXIT

II. TRUY VẤN TRONG SQL – BÀI TẬP TẠI LỚP

A1: Đưa ra tên, địa chỉ của những khách hàng mua các mặt hàng mặt hàng thuộc nhóm "Hàng điện tử" từ ngày 01/05/2022 đến 30/05/2022.

```
SELECT KH.TenKH, KH.DiaChi, MATHANG.TenMH, [NgayMua],  
[TenNhómMH] FROM [PE_DB_TEST1].[dbo].[MUAHANG],  
[PE_DB_TEST1].[dbo].[MATHANG],  
[PE_DB_TEST1].[dbo].[KHACHHANG] KH,  
[PE_DB_TEST1].[dbo].NHOMMATHANG  
WHERE (MUAHANG.MaMH=MATHANG.MaMH) AND  
(MUAHANG.MaKH=KH.MaKH) AND (NgayMua BETWEEN  
'2022/05/1' AND '2022/05/30') AND  
(MATHANG.NhomHang=NHOMMATHANG.MaNhómMH) AND  
(NHOMMATHANG.TenNhómMH LIKE N'Hàng điện tử');
```

Về chương IV

< >

MAIN MENU

EXIT

II. TRUY VẤN TRONG SQL – BÀI TẬP TẠI LỚP

A2: Đưa ra tên của những mặt hàng bán chạy nhất (có doanh thu max).

Doanh thu được tính theo công thức: Số lượng bán được * Đơn giá.

CREATE View TmpView As

```
SELECT MUAHANG.MaMH, Sum(SoLuong*DonGia) As DoanhThu
FROM [PE_DB_TEST1].[dbo].[MUAHANG],
[PE_DB_TEST1].[dbo].[MATHANG] WHERE
(MUAHANG.MaMH=MATHANG.MaMH) GROUP BY
MUAHANG.MaMH;
```

Go;

```
SELECT TenMH, DoanhThu As DoanhThuMax
FROM [PE_DB_TEST1].[dbo].[MATHANG], TmpView
WHERE (MATHANG.MaMH=TmpView.MaMH) AND
(DoanhThu=(SELECT MAX(DoanhThu) FROM TmpView));
```

II. TRUY VẤN TRONG SQL – BÀI TẬP TẠI LỚP

A3: Đếm số lượng mặt hàng thuộc nhóm mặt hàng: "Hàng gia dụng".

CREATE View TmpView2 As

SELECT MaMH, COUNT(SoLuong) As SoLuong

FROM [PE_DB_TEST1].[dbo].[MUAHANG]

GROUP BY MaMH;

SELECT MATHANG.TenMH, SoLuong,

NHOMMATHANG.TenNhomMH

FROM TmpView2, [PE_DB_TEST1].[dbo].[MATHANG],

[PE_DB_TEST1].[dbo].[NHOMMATHANG]

WHERE (TmpView2.MaMH=MATHANG.MaMH) AND

(MATHANG.NhomHang=NHOMMATHANG.MaNhomMH) AND

(NHOMMATHANG.TenNhomMH LIKE N'Hàng gia dụng');

II. TRUY VẤN TRONG SQL – BÀI TẬP TẠI LỚP

A4: Đưa ra tên, địa chỉ, số điện thoại của những khách hàng có tổng giá trị hàng mua là lớn nhất trong khoảng thời gian từ ngày 01/1/2022 đến 25/10/2022. Trong đó, Tổng giá trị hàng = Σ (Số lượng * Đơn giá);

```
CREATE ViewTmpView3(MaKH, DoanhThuKH) As
SELECT MUAHANG.MaKH, Sum(SoLuong*DonGia) As
DoanhThuKH
FROM [PE_DB_TEST1].[dbo].[MUAHANG],
[PE_DB_TEST1].[dbo].[MATHANG] WHERE
(MUAHANG.MaMH=MATHANG.MaMH) GROUP BY
MUAHANG.MaKH;
```

II. TRUY VẤN TRONG SQL – BÀI TẬP TẠI LỚP

A4 (cont): Đưa ra tên, địa chỉ, số điện thoại của những khách hàng có tổng giá trị hàng mua là lớn nhất trong khoảng thời gian từ ngày 01/1/2022 đến 25/10/2022. Trong đó, Tổng giá trị hàng = Σ (Số lượng * Đơn giá);

```
SELECT [TenKH],[DiaChi],[SoDT]
FROM [PE_DB_TEST1].[dbo].[KHACHHANG] KH, TmpView3,
MUAHANG WHERE
(TmpView3.MaKH=KH.MaKH) AND
(MUAHANG.NgayMua BETWEEN '2022/01/01' AND '2022/10/01')
AND
(MUAHANG.MaKH=KH.MaKH) AND (DoanhThuKH=(SELECT
MAX(DoanhThuKH) FROM TmpView3))
```

III. CÁC LỆNH CẬP NHẬT DỮ LIỆU

Mục I và II hạn chế trong thực nghiệm về truy vấn CSDL (truy vấn → trích rút thông tin). Mục III trình bày các thao tác thay đổi thông tin trong CSDL. Cụ thể về các lệnh INSERT, DELETE & UPDATE data.

1. Lệnh INSERT: thêm một hoặc nhiều bản ghi vào 1 bảng.

Dạng tổng quát lệnh INSERT:

INSERT [INTO] <tên bảng> [<Danh sách cột>]

[VALUES (các giá trị)] [<câu truy vấn>]

III. CÁC LỆNH CẬP NHẬT DỮ LIỆU

2. Lệnh DELETE: xóa một hoặc nhiều bản ghi trong 1 bảng.

Dạng tổng quát lệnh DELETE:

DELETE FROM {<đường dẫn>}|<tên bảng>|<tên view>
[WHERE <biểu thức điều kiện>]

3. Lệnh UPDATE: thay đổi dữ liệu trong 1 bảng.

Dạng tổng quát lệnh UPDATE:

UPDATE <tên bảng>
SET <cột 1>=<biểu thức 1>, .., <cột N>=<biểu thức N>
[WHERE <biểu thức điều kiện>]

III. CÁC LỆNH CẬP NHẬT DỮ LIỆU

3. Lệnh UPDATE: thay đổi dữ liệu trong >=1 bảng.

Dạng lệnh UPDATE:

UPDATE <tên bảng>

**SET <cột 1>=<biểu thức 1>, .., <cột N>=<biểu thức N>
[FROM {<bảng(s)>}] [WHERE <biểu thức điều kiện>]**

Dạng lệnh UPDATE với cấu trúc CASE:

UPDATE <bảng> SET <cột j> = CASE

WHEN <biểu thức điều kiện 1>] THEN <Result 1> ...

ELSE <Result k>

END

Về chương IV

< >

MAIN MENU

EXIT

III. CÁC LỆNH CẬP NHẬT DỮ LIỆU

4. Thực tập lệnh Cập nhật $P (P_ID, Pname, Color, Weight);$

$S(S_ID, Sname, Status, City); SP(S_ID, P_ID, Qty, Price, Tax, Sdate);$

Ví dụ 90: Thêm một hàng cung ứng mới:

Ví dụ 91: Xóa hàng cung ứng S9:

Ví dụ 92: Thay đổi màu sắc của P1 thành ‘green’:

III. CÁC LỆNH CẬP NHẬT DỮ LIỆU

4. Thực tập lệnh Cập nhật $P(P_ID, Pname, Color, Weight);$

$S(S_ID, Sname, Status, City); SP(S_ID, P_ID, Qty, Price, Tax, Sdate);$

Ví dụ 90: Thêm một hàng cung ứng mới:

INSERT INTO S

VALUES('S9', 'Hoa Quynh', Null, 'Hoa Lac')

Ví dụ 91: Xóa hàng cung ứng S9:

DELETE FROM SP WHERE S_ID='S9';

DELETE FROM S WHERE S_ID='S9'

Ví dụ 92: Thay đổi màu sắc của P1 thành ‘green’:

UPDATE P

SET Color='green' WHERE P_ID='P1'

III. CÁC LỆNH CẬP NHẬT DỮ LIỆU

4. Thực tập lệnh Cập nhật $P(P_ID, Pname, Color, Weight);$

$S(S_ID, Sname, Status, City); SP(S_ID, P_ID, Qty, Price, Tax, Sdate);$

Ví dụ 93: Xóa các hàng cung ứng chưa cung cấp sản phẩm nào:

Ví dụ 94: Xóa các mặt hàng đã được cung cấp sau ngày 01-06-2022:

Ví dụ 95: Bổ sung thêm mặt hàng mới với các thông số: mã hàng P7, tên mặt hàng Apple iPad 4 128GB, màu đen, trọng lượng 600 gram:

III. CÁC LỆNH CẬP NHẬT DỮ LIỆU

4. Thực tập lệnh Cập nhật $P(P_ID, Pname, Color, Weight);$

$S(S_ID, Sname, Status, City); SP(S_ID, P_ID, Qty, Price, Tax, Sdate);$

Ví dụ 93: Xóa các hàng cung ứng chưa cung cấp sản phẩm nào:

DELETE FROM S WHERE S_ID NOT IN (
SELECT S_ID FROM SP)

Ví dụ 94: Xóa các mặt hàng đã được cung cấp sau ngày 01-06-2022:

DELETE FROM SP WHERE Sdate >'01/06/2022'

Ví dụ 95: Bổ sung thêm mặt hàng mới với các thông số: mã hàng P7, tên mặt hàng Apple iPad 4 128GB, màu đen, trọng lượng 600 gram:

INSERT INTO P(P_ID, Pname, Color, Weight)
VALUES ('P7','Apple iPad 4 128GB', 'Black', 0.6)

III. CÁC LỆNH CẬP NHẬT DỮ LIỆU

4. Thực tập lệnh Cập nhật

$P(P_ID, Pname, Color, Weight);$

$S(S_ID, Sname, Status, City); SP(S_ID, P_ID, Price, Qty, ...);$

Ví dụ 96: Tăng giá 10% cho những mặt hàng xuất bán có số lượng >100:

Ví dụ 97: Xóa các mặt hàng có giá hơn 400:

Heuristic: SELECT INTO <tempTable> ...

III. CÁC LỆNH CẬP NHẬT DỮ LIỆU

4. Thực tập lệnh Cập nhật $P(P_ID, Pname, Color, Weight);$

$S(S_ID, Sname, Status, City); SP(S_ID, P_ID, Qty);$

Ví dụ 96: Tăng giá 10% cho những mặt hàng xuất bán có số lượng >100:

UPDATE SP SET Price = Price*1.1 WHERE Qty>=100

Ví dụ 97: Xóa các mặt hàng có giá hơn 400:

DELETE FROM SP WHERE Price>400

Heuristic: **SELECT INTO <tempTable> ...**

IV. CÁC LỆNH AN TOÀN DỮ LIỆU

1. Cấp quyền truy cập

GRANT <tên_quyền> ON <đối tượng>

TO <tên_người_dùng> [WITH GRANT OPTION]

Tên_quyền: *SELECT (chọn)* *READ (đọc)*
WRITE (ghi) *INSERT (bổ sung)*
UPDATE (cập nhật) *DELETE (xóa)*
RUN (thực hiện)

Đối_tượng: tên bảng, tên VIEW hoặc tên chương trình.

Tên_người_dùng: tên một người (user) hoặc nhóm người

WITH GRANT OPTION: Người dùng có thể chuyển quyền cho người khác.

IV. CÁC LỆNH AN TOÀN DỮ LIỆU

2. Hủy quyền truy cập

```
REVOKE <tên_quyền> ON <đối tượng>  
FROM <tên_người_dùng>
```

Tên_quyền: *SELECT (chọn)* *READ (đọc)*
 WRITE (ghi) *INSERT (bổ sung)*
 UPDATE (cập nhật) *DELETE (xoá)*
 RUN (thực hiện)

Đối_tượng: Tên bảng, tên VIEW hoặc tên chương trình

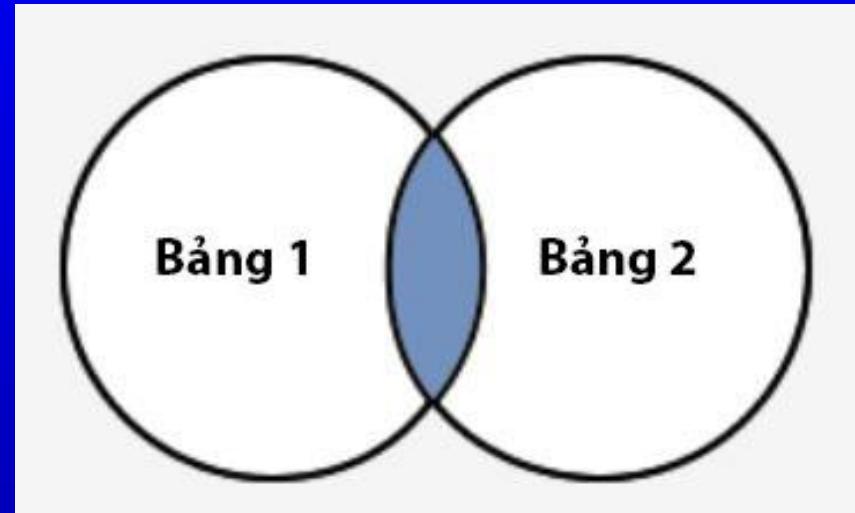
Tên_người_dùng: tên một người (user) hoặc nhóm người

V. CÁC LỆNH NỐI DỮ LIỆU (JOIN)

5.1. Inner Join

Trả về các dòng của các bảng thỏa điều kiện JOIN

```
SELECT C.CID, C.name, S.name  
FROM Class C  
INNER JOIN Student S  
ON C.CID=S.CID
```



Class (CID, Cname);

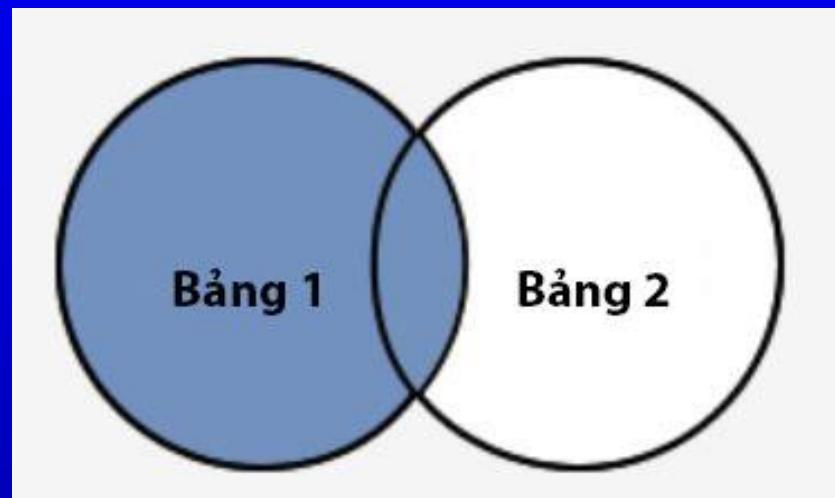
Student(ID, Name, CID);

V. CÁC LỆNH NỐI DỮ LIỆU (JOIN)

5.2. Outer Join

5.2.1. Left Join: Trả về các dòng thỏa điều kiện JOIN + (các dòng còn lại của bảng trái + NULL).

```
SELECT C.CID, C.name, S.name  
FROM Class C  
LEFT JOIN Student S  
ON C.CID=S.CID
```



Class (CID, Cname);

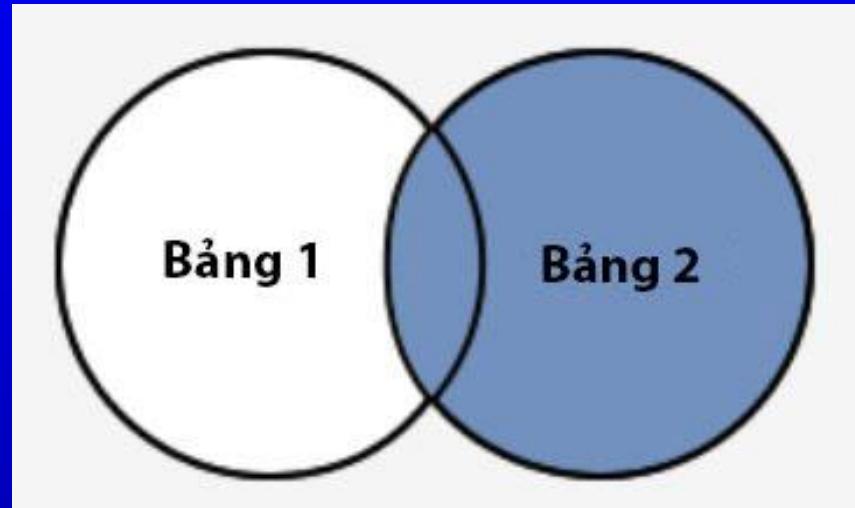
Student(ID, Name, CID);

V. CÁC LỆNH NỐI DỮ LIỆU (JOIN)

5.2. Outer Join

5.2.2. Right Join: Trả về các dòng thỏa điều kiện JOIN + (các dòng còn lại của bảng phải + NULL).

```
SELECT C.CID, C.name, S.name  
FROM Class C  
RIGHT JOIN Student S  
ON C.CID=S.CID
```



Class (CID, Cname);

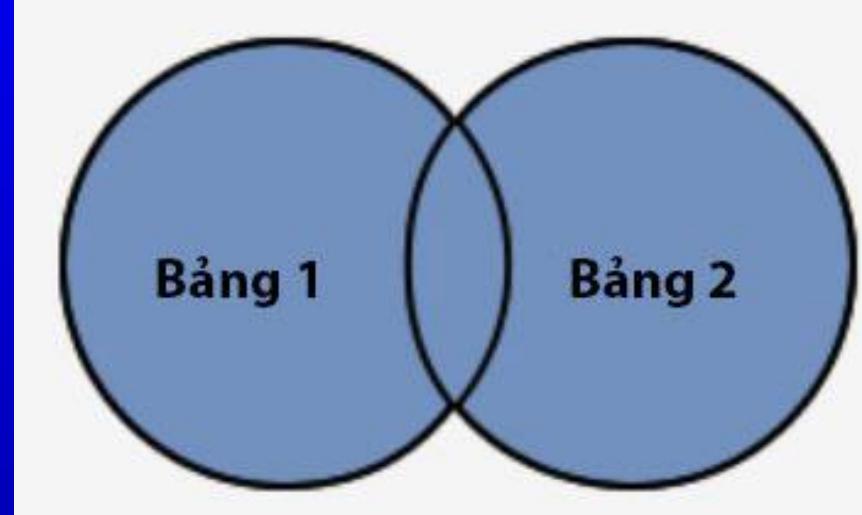
Student(ID, Name, CID);

V. CÁC LỆNH NỐI DỮ LIỆU (JOIN)

5.2. Outer Join

5.2.3. Full Join: Trả về các dòng thỏa điều kiện JOIN + (các dòng còn lại của bảng trái + NULL) + (các dòng còn lại của bảng phải + NULL).

```
SELECT C.CID, C.name, S.name  
FROM Class C  
FULL JOIN Student S  
ON C.CID=S.CID
```



Class (CID, Cname);

Student(ID, Name, CID);

BÀI TẬP

- Tạo sơ đồ quan hệ SV

Create Table SV (MASV Text (10) CONSTRAINT *khoachinh* PRIMARY KEY, HOTEN Text (30), NU YesNo, NGAYSINH Date, MALOP Text (10), DIACHI Text (50), HOCBONG Double)

- Tạo sơ đồ quan hệ KQ

Create Table KQ (MASV Text (10), MAMH Text (10), DIEMTHI Double, CONSTRAINT *khoachinh* PRIMARY KEY (MASV, MAMH))

BÀI TẬP

1. Lập danh sách svien gồm: MASV, HOTEN, HOCBONG
2. Lập danh sách sinh viên nữ, có học bổng, thuộc lớp có mã: 'THK5'
3. Lập danh sách sinh viên gồm: MASV, HOTEN, HOCBONG trong đó HOCBONG được sắp xếp giảm dần
4. Lập danh sách sinh viên gồm: MASV, HOTEN, TENLOP
(*TENLOP lấy từ bảng LOP*)
5. Lập danh sách sinh viên gồm: MASV, HOTEN, NU, NGAYSINH, DIEMTHI với điểm thi môn 'CSDL' >=8
6. Lập danh sách sinh viên có họ là NGUYEN

Về chương IV

< >

MAIN MENU

EXIT

BÀI TẬP

7. Lập danh sách sinh viên đăng ký ít nhất 3 môn học.
8. Lập danh sách sinh viên có học bỗng cao nhất.
9. Lập danh sách sinh viên gồm: MASV, HOTEN, DIEMTHI có điểm thi môn 'CSDL' cao nhất.

Câu 8: SELECT *
FROM SV

WHERE HOCBONG>= ALL (SELECT HOCBONG FROM SV);

Câu 9: SELECT SV.MASV, HOTEN, DIEMTHI
FROM SV, KQ
WHERE SV.MASV=KQ.MASV AND MAMH='CSDL' AND
DIEMTHI >= ALL (SELECT DIEMTHI FROM KQ
WHERE MAMH='CSDL');