



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

MÔN CƠ SỞ DỮ LIỆU - IT004

CHƯƠNG 4: NGÔN NGỮ SQL (Structured Query Language)

ThS. TẠ VIỆT PHƯƠNG

phuongtv@uit.edu.vn

Nội dung

I. Giới thiệu

II. Ngôn ngữ định nghĩa dữ liệu

III. Ngôn ngữ thao tác dữ liệu

IV. Ngôn ngữ truy vấn dữ liệu



GIỚI THIỆU

Ngôn ngữ truy vấn có cấu trúc SQL

- **Structured Query Language**
- Là ngôn ngữ chuẩn để truy vấn và thao tác trên CSDL quan hệ
- Là ngôn ngữ phi thủ tục
- Khởi nguồn của SQL là SEQUEL - *Structured English Query Language*, năm 1974)
- **Các chuẩn SQL**
 - SQL89
 - SQL92 (SQL2)
 - SQL99 (SQL3) ...
 - Hiện tại: SQL:2019

Ngôn ngữ truy vấn có cấu trúc SQL

Bao gồm:

- **Ngôn ngữ định nghĩa dữ liệu** (*Data Definition Language - DDL*): cho phép khai báo cấu trúc bảng, các mối quan hệ và các ràng buộc.
- **Ngôn ngữ thao tác dữ liệu** (*Data Manipulation Language - DML*): cho phép thêm, xóa, sửa dữ liệu.
- **Ngôn ngữ truy vấn dữ liệu** (*Structured Query Language – SQL*): cho phép truy vấn dữ liệu.
- **Ngôn ngữ điều khiển dữ liệu** (*Data Control Language – DCL*): khai báo bảo mật thông tin, cấp quyền và thu hồi quyền khai thác trên cơ sở dữ liệu.

Ngôn ngữ truy vấn có cấu trúc SQL

- SQL sử dụng các thuật ngữ tương đương với thuật ngữ về mô hình dữ liệu quan hệ như sau:

SQL	Mô hình dữ liệu quan hệ
Bảng	Quan hệ
Cột	Thuộc tính
Dòng	Bộ

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU

2. Ngôn ngữ định nghĩa dữ liệu

1. Lệnh tạo bảng (CREATE)

Cú pháp

Một số kiểu dữ liệu

2. Lệnh sửa cấu trúc bảng (ALTER)

Thêm thuộc tính

Sửa kiểu dữ liệu của thuộc tính

Xoá thuộc tính

Thêm ràng buộc toàn vẹn

Xoá ràng buộc toàn vẹn

3. Lệnh xóa bảng (DROP)

2.1. Lệnh tạo bảng

Cú pháp

CREATE TABLE <tên_bảng>

(

<tên_cột1> <kiểu_dữ_liệu> [not null],

<tên_cột2> <kiểu_dữ_liệu> [not null],

...

<tên_cột1> <kiểu_dữ_liệu> [not null],

khai báo khóa chính, khóa ngoại, ràng buộc

)

2.1. Lệnh tạo bảng

Một số kiểu dữ liệu

Kiểu dữ liệu	SQL Server
Chuỗi ký tự	varchar(n), char(n),nvarchar(n), nchar(n)
Số	tinyint,smallint, int, numeric(m,n), decimal(m,n),float, real, smallmoney, money
Ngày tháng	smalldatetime, datetime
Luận lý	bit

2.1. Lệnh tạo bảng

Ví dụ:

Lược đồ CSDL quản lý bán hàng gồm có các quan hệ sau:

KHACHHANG (MAKH, HOTEN, DCHI, SODT, NGSINH,
DOANHISO, NGDK, CMND)

NHANVIEN (MANV,HOTEN, NGVL, SODT)

SANPHAM (MASP,TENSP, DVT, NUOCSX, GIA)

HOADON (SOHD, NGHD, MAKH, MANV, TRIGIA)

CTHD (SOHD,MASP,SL)

2.1. Lệnh tạo bảng

Create table KHACHHANG

(

MAKH	char(4) primary key,
HOTEN	varchar(40),
DCHI	varchar(50),
SODT	varchar(20),
NGSINH	smalldatetime,
DOANHISO	money,
NGDK	smalldatetime,
CMND	varchar(10)

)

2.1. Lệnh tạo bảng

Create table CTHD

```
(  
    SOHD int foreign key  
        references HOADON(SOHD),  
    MASP char(4) foreign key  
        references SANPHAM(MASP),  
    SL int,  
    constraint PK_CTHD primary key (SOHD, MASP)  
)
```

2.2. Lệnh sửa cấu trúc bảng

❖ Thêm thuộc tính

ALTER TABLE tênbảng ADD tên cột kiểu dữ liệu

— Ví dụ: thêm cột Ghi_chu vào bảng khách hàng

ALTER TABLE KHACHHANG ADD GHI_CHU varchar(20)

❖ Sửa kiểu dữ liệu thuộc tính

***ALTER TABLE tênbảng ALTER COLUMN tên cột
kiểu dữ liệu_mới***

❖ Lưu ý:

Không phải sửa bất kỳ kiểu dữ liệu nào cũng được

2.2. Lệnh sửa cấu trúc bảng

- **Ví dụ:** Sửa Cột Ghi_chu thành kiểu dữ liệu varchar(50)
ALTER TABLE KHACHHANG ALTER COLUMN GHI_CHU varchar(50)
- Nếu sửa kiểu dữ liệu của cột Ghi_chu thành varchar(5), mà trước đó đã nhập giá trị cho cột Ghi_chu có độ dài hơn 5 ký tự thì không được phép.
- Hoặc sửa từ kiểu chuỗi ký tự sang kiểu số, ...

❖ **Xóa thuộc tính**

ALTER TABLE tên_bảng DROP COLUMN tên_cột

- Ví dụ: xóa cột Ghi_chu trong bảng KHACHHANG
ALTER TABLE NHANVIEN DROP COLUMN Ghi_chu

2.2. Lệnh sửa cấu trúc bảng

❖ Thêm ràng buộc toàn vẹn:

**ALTER TABLE <tên_bảng>
ADD CONSTRAINT
<tên_ràng_buộc>**

UNIQUE tên_cột

PRIMARY KEY (tên_cột)

FOREIGN KEY (tên_cột)

REFERENCES tên_bảng
(cột_là_khóa_chính)

CHECK (tên_cột điều_kiện)

2.2. Lệnh sửa cấu trúc bảng

- ❖ Ràng buộc toàn vẹn trong SQL được chia làm 2 loại chính:
 - Loại đơn giản: Sử dụng **Constraint** để mô tả
 - Loại phức tạp: Sử dụng **Trigger** để thực hiện
- ❖ Có thể khai báo RBTV ở mức cột hoặc mức bảng

2.2. Lệnh sửa cấu trúc bảng

- ❖ **Các loại RBTV đơn giản:**
 - Kiểm tra duy nhất: Primary Key, Unique
 - Kiểm tra rỗng/khác rỗng: NULL/NOT NULL
 - Kiểm tra tồn tại: Foreign Key ... References...
 - Kiểm tra miền giá trị: Check, Default
- ❖ **Có thể tạo RBTV cùng với thời điểm tạo bảng hoặc sau khi tạo bảng**

2.2. Lệnh sửa cấu trúc bảng

• Ví dụ

- ALTER TABLE **NHANVIEN** ADD CONSTRAINT **PK_NV**
PRIMARY KEY (MANV)
- ALTER TABLE **CTHD** ADD CONSTRAINT **FK_CT_SP**
FOREIGN KEY (MASP) REFERENCES **SANPHAM**(MASP)
- ALTER TABLE **SANPHAM** ADD CONSTRAINT **CK_GIA**
CHECK (GIA >=500)
- ALTER TABLE **KHACHHANG** ADD CONSTRAINT **UQ_KH**
UNIQUE (CMND)

2.2. Lệnh sửa cấu trúc bảng

Xóa ràng buộc toàn vẹn

***ALTER TABLE tên_bảng DROP CONSTRAINT
tên_ràng_buộc***

—Ví dụ:

- Alter table CTHD drop constraint FK_CT_SP
- Alter table SANPHAM drop constraint ck_gia
- **Lưu ý:** đối với ràng buộc khóa chính, muốn xóa ràng buộc này phải xóa hết các ràng buộc khóa ngoại tham chiếu tới nó

2.3. Lệnh xóa bảng (DROP)

- **Cú pháp**

DROP TABLE tên_bảng

- **Ví dụ:** xóa bảng KHACHHANG.

DROP TABLE KHACHHANG

- **Lưu ý:** khi muốn xóa một bảng phải xóa tất cả những khóa ngoại tham chiếu tới bảng đó trước.

NGÔN NGỮ THAO TÁC DỮ LIỆU

3. Ngôn ngữ thao tác dữ liệu

- **Gồm các lệnh:**

1. Lệnh thêm dữ liệu (**INSERT**)
2. Lệnh sửa dữ liệu (**UPDATE**)
3. Lệnh xóa dữ liệu (**DELETE**)

3.1. Lệnh Insert

•Cú pháp

- INSERT INTO tên_bảng (cột1,...,cột n) VALUES (giá_trị_1,..., giá_trị_n)
- INSERT INTO tên_bảng VALUES (giá_trị_1, giá_trị_2,..., giá_trị_n)
- SELECT * INTO tên-bảng-mới from tên-bảng-có-sẵn
- INSERT INTO tên-bảng-tạo-trước select * from tên-bảng-có-sẵn

•Ví dụ:

- insert into SANPHAM values ('BC01','But chi', 'cay', 'Singapore', 3000)
- insert into SANPHAM (masp,tensp,dvt,nuocsx,gia) values ('BC01','But chi','cay','Singapore',3000)

3.1. Lệnh Insert

❖ Lưu ý khi sử dụng Lệnh Insert:

- Thứ tự các giá trị phải trùng với thứ tự các cột
- Câu lệnh **Insert** sẽ gặp lỗi nếu vi phạm RBTV:
 - ✓ Khóa chính
 - ✓ Khóa ngoại
 - ✓ Not null (các thuộc tính có RBTV Not null bắt buộc phải có giá trị)

3.1. Lệnh Insert

• Ví dụ của select...into

- Select * into SANPHAM_NEW from SANPHAM
- Select * into SANPHAM_NEW from SANPHAM WHERE điều-kiện

• Ví dụ của insert into....select....

- insert into SANPHAM_COPY select * from SANPHAM
- insert into SANPHAM_COPY select * from SANPHAM where điều-kiện

3.2. Lệnh Update

- **Cú pháp**

UPDATE tên_bảng

SET cột_1 = giá_trị_1, cột_2 = giá_trị_2

[WHERE điều_kiện]

- **Ví dụ:** Tăng giá 10% đối với những sản phẩm do “VietNam” sản xuất

UPDATE SANPHAM

SET Gia = Gia*1.1

WHERE Nuocsx='VietNam'

3.2. Lệnh Update

❖ Lưu ý khi dung câu lệnh Update:

- Những dòng thỏa điều kiện tại mệnh đề WHERE sẽ được cập nhật giá trị mới
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các giá trị của thuộc tính đó đều sẽ bị cập nhật
- Lệnh **UPDATE** có thể gây vi phạm RBTV khóa ngoại: không cho sửa

3.3. Lệnh Delete

- **Cú pháp**

DELETE FROM tên_bảng [WHERE điều_kiện]

- **Ví dụ:**

- **Xóa toàn bộ nhân viên**

DELETE FROM NHANVIEN

- **Xóa những sản phẩm do Trung Quốc sản xuất có giá thấp hơn 10000**

**DELETE FROM SANPHAM
WHERE (Gia < 10000) and (Nuocsx='Trung Quoc')**

3.3. Lệnh Delete

❖ Lưu ý khi sử dụng lệnh DELETE:

- Những dòng thỏa điều kiện tại mệnh đề WHERE sẽ bị xóa đi
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng sẽ bị xóa
- Lệnh DELETE có thể gây vi phạm RBTV khóa ngoại: không cho xóa

Bài tập tại lớp

- ❖ Cho lược đồ CSDL sau:
- ❖ **SINHVIEN**(MaSV,Hoten,GT,Ngaysinh,CMND,MaKhoa)
- ❖ **KHOA** (MaKhoa,TenKhoa,SoluongGV,SoluongSV)
- ❖ **HOADONHOCPhi**(MAHDHP,MASV,SOTIEN)
- Yêu cầu:
 - 1. Sử dụng ngôn ngữ SQL để tạo bảng cho các quan hệ trên
 - 2. Tạo ràng buộc khóa chính, khóa ngoại cho các bảng trên
 - 3. Trong bảng **SINHVIEN** tạo ràng buộc kiểm tra CMND là duy nhất
 - 4. Giảm 10% học phí cho sinh viên do tình hình dịch COVID-19

NGÔN NGỮ TRUY VẤN DỮ LIỆU

4. Ngôn ngữ truy vấn dữ liệu

- Là ngôn ngữ chuẩn, có cấu trúc dùng để truy vấn và thao tác trên CSDL quan hệ.

- Câu truy vấn tổng quát:

SELECT [DISTINCT] danh_sách_cột | hàm

FROM danh sách các quan hệ (hay bảng, table)

[WHERE điều_kiện]

[GROUP BY danh_sách_cột_gom_nhóm]

[HAVING điều_kiện_trên_nhóm]

[ORDER BY cột1 ASC | DESC, cột2 ASC | DESC,...]

4. Ngôn ngữ truy vấn dữ liệu

❖ Toán tử so sánh:

- =, >, <, >=, <=, <>
- BETWEEN
- IS NULL, IS NOT NULL
- LIKE (% , _)
- IN, NOT IN
- EXISTS, NOT EXISTS
- SOME, ALL, ANY

❖ Toán tử logic: AND, OR.

❖ Các phép toán: +, - , * , /

❖ Các hàm xử lý ngày (DAY()), tháng (MONTH()), năm (YEAR())

4. Ngôn ngữ truy vấn dữ liệu

All: trả về giá trị là TRUE nếu tất cả các giá trị trong câu truy vấn con đáp ứng điều kiện, được sử dụng với Select, Where, having.

- Any: trả về TRUE nếu BẤT KỲ giá trị nào trong số các giá trị truy vấn con đáp ứng điều kiện

ANY có nghĩa là điều kiện sẽ đúng nếu hoạt động đúng với bất kỳ giá trị nào trong phạm vi.

4. Ngôn ngữ truy vấn dữ liệu

- 5 hàm: **COUNT(), SUM(), MAX(), MIN(), AVG()**
- Phân loại câu SELECT:
 - SELECT đơn giản.
 - SELECT có mệnh đề ORDER BY.
 - SELECT lồng (câu SELECT lồng câu SELECT khác).
 - SELECT gom nhóm (GROUP BY).
 - SELECT gom nhóm (GROUP BY) có điều kiện HAVING.

4. Ngôn ngữ truy vấn dữ liệu

Bài tập: Cho lược đồ CSDL “quản lý đề án công ty” như sau

NHANVIEN (MaNV, HoTen, Phai, Luong, NTNS, Ma_NQL, MaPH)

PHONGBAN (MaPH, TenPH, TRPH)

DEAN (MaDA, TenDA, Phong, NamThucHien)

PHANCONG (MaNV, MaDA, ThoiGian)

MANV	HOTEN	NTNS	PHAI	MA_NQL	MaPH	LUONG
001	Vuong Ngoc Quyen	22/10/1957	Nu		QL	3.000.000
002	Nguyen Thanh Tung	09/01/1955	Nam	001	NC	2.500.000
003	Le Thi Nhan	18/12/1960	Nu	001	DH	2.500.000
004	Dinh Ba Tien	09/01/1968	Nam	002	NC	2.200.000
005	Bui Thuy Vu	19/07/1972	Nam	003	DH	2.200.000
006	Nguyen Manh Hung	15/09/1973	Nam	002	NC	2.000.000
007	Tran Thanh Tam	31/07/1975	Nu	002	NC	2.200.000
008	Tran Hong Minh	04/07/1976	Nu	004	NC	1.800.000

NHANVIEN

PHANCONG

DEAN

MADA	TENDA	PHONG	NamThucHien
TH001	Tin hoc hoa 1	NC	2002
TH002	Tin hoc hoa 2	NC	2003
DT001	Dao tao 1	DH	2004
DT002	Dao tao 2	DH	2004

PHONGBAN

MAPH	TENPH	TRPH
QL	Quan Ly	001
DH	Dieu Hanh	003
NC	Nghien Cuu	002

MANV	MADA	THOIGIAN
001	TH001	30,0
001	TH002	12,5
002	TH001	10,0
002	TH002	10,0
002	DT001	10,0
002	DT002	10,0
003	TH001	37,5
004	DT001	22,5
004	DT002	10,0
006	DT001	30,5
007	TH001	20,0
007	TH002	10,0
008	DT002	12,5

4.1. BETWEEN, ORDER BY, IS NULL

Câu hỏi 13: Sử dụng =, >, >=, ... Danh sách các nhân viên sinh trong khoảng từ năm 1978 đến 1983?

➡ Select MaNV, HoTen From NhanVien
where Year(NTNS) >= 1978 AND Year(NTNS) <= 1983

Câu hỏi 14: Sử dụng BETWEEN, ORDER BY. Danh sách các nhân viên sinh trong khoảng từ năm 1978 đến 1983? Sắp xếp theo mức lương giảm dần.

➡ Select * From NhanVien where Year(NTNS) BETWEEN 1978 and 1983 ORDER BY Luong DESC

Câu hỏi 15: Sử dụng IS NULL. Cho biết những nhân viên không có người quản lý trực tiếp? (không chịu sự quản lý trực tiếp của người nào)

➡ Select MaNV, HoTen, NTNS, Ma_NQL from NhanVien where Ma_NQL is Null

4.2. SQL - SO SÁNH IN & NOT IN

Câu hỏi 16: **Sử dụng Is Not Null**. Cho biết những nhân viên có người quản lý trực tiếp? Thông tin hiển thị gồm: mã nhân viên, họ tên, mã người quản lý.

➡ Select MaNV, HoTen, Ma_NQL from NhanVien
where Ma_NQL is not Null

Câu hỏi 17: **Sử dụng IN (so sánh với một tập hợp giá trị cụ thể)**. Cho biết họ tên nhân viên thuộc phòng 'NC' hoặc phòng 'DH'?

➡ Select DISTINCT Hoten From NhanVien where MaPH in ('NC','DH')

Câu hỏi 18: **Sử dụng IN (so sánh với một tập hợp giá trị chọn từ câu SELECT khác)**. Cho biết họ tên nhân viên thuộc phòng 'NC' hoặc phòng 'DH'?

➡ Select Hoten from NhanVien where MaPH in (Select MaPH from PHONGBAN where MaPH='NC' OR MaPH='DH')

4.2. SQL - SO SÁNH IN & NOT IN

Câu hỏi 19 (tt): Cho biết mã số, họ tên, ngày tháng năm sinh của những nhân viên đã tham gia đề án?

➡ Select MaNV, HoTen, NTNS from NhanVien
where MaNV in (Select MaNv From PhanCong)

Câu hỏi 20: **Sử dụng NOT IN**. Cho biết mã số, họ tên, ngày tháng năm sinh của những nhân viên không tham gia đề án nào?

Gợi ý cho mệnh đề NOT IN: thực hiện câu truy vấn “tìm nhân viên có tham gia đề án (dựa vào bảng PhanCong)”, sau đó lấy phần bù.

➡ Select MaNV, HoTen, NTNS from NhanVien
where MaNV not in (Select MaNv From PhanCong)

Câu hỏi 21 (tt): Cho biết tên phòng ban không chủ trì các đề án triển khai năm 2005? Gợi ý: thực hiện câu truy vấn “tìm phòng ban chủ trì các đề án triển khai năm 2005”, sau đó lấy phần bù.

➡ Select TenPH from PhongBan where MaPH not in (Select DISTINCT Phong from DEAN where NamThucHien=2005)

4.3. SQL – SO SÁNH LIKE

Câu hỏi 22: so sánh chuỗi = chuỗi. Liệt kê mã nhân viên, ngày tháng năm sinh, mức lương của nhân viên có tên “Nguyễn Tường Linh”?

```
Select MaNV, NTNS, Luong from NhanVien  
where HoTen = 'Nguyễn Tường Linh'
```

Câu hỏi 23: Sử dụng LIKE (%: thay thế 1 chuỗi ký tự). Tìm những nhân viên có họ Nguyễn.

```
Select MaNV, HoTen from NhanVien where HoTen like 'Nguyễn %'
```

Câu hỏi 24 (tt): Tìm những nhân viên có tên Lan.

```
Select MaNV, HoTen from NhanVien where HoTen like '% Lan'
```

Câu hỏi 25 (tt): Tìm những nhân viên có tên lót là “Văn”.

```
Select MaNV, HoTen from NhanVien where HoTen like '% Văn %'
```

Câu hỏi 26: Sử dụng LIKE (_: thay thế 1 ký tự bất kỳ). Tìm những nhân viên tên có tên ‘Nguyễn La_’ (ví dụ Lam, Lan)

```
Select MaNV, HoTen from NhanVien where HoTen like 'Nguyễn La_'
```

4.4. SQL – PHÉP KẾT (TÍCH DESCARTES GIỮA 2 TABLE VÀ ĐIỀU KIỆN CHỌN ĐẶT Ở MỆNH ĐỀ WHERE)

Câu 27. Phép kết: Cho biết mã số, họ tên, ngày tháng năm sinh của những nhân viên đã tham gia đề án?

```
Select MaNV, NTNS, Luong from NhanVien, PhanCong  
where Nhanvien.MaNV=PhanCong.MaNV
```

Câu 28. Phép kết: Cho biết họ tên nhân viên tham gia đề án 'TH01'

```
Select HoTen from NhanVien, PhanCong  
where MaDA='TH01' and Nhanvien.MaNV=PhanCong.MaNV
```

Câu 29. Phép kết chính nó (Self join): Cho biết họ tên nhân viên và họ tên người quản lý của nhân viên đó

```
Select A.HoTen, B.HoTen From NhanVien as A, NhanVien as B  
Where A.MaNV=B.Ma_NQL
```

4.4. SQL – PHÉP KẾT (TÍCH DESCARTES GIỮA 2 TABLE VÀ ĐIỀU KIỆN CHỌN ĐẶT Ở MỆNH ĐỀ WHERE)

Cú pháp:

SELECT column_name

FROM table1 **INNER JOIN** table2

ON table1.column_name = table2.column_name

Cho biết mã số, họ tên, ngày tháng năm sinh của những nhân viên đã tham gia đề án

```
Select MaNV, HoTen, NTNS from NhanVien inner join PhanCong  
on Nhanvien.MaNV=PhanCong.MaNV
```

Cho biết họ tên nhân viên tham gia đề án 'TH01'

```
Select HoTen From NhanVien inner join PhanCong  
ON NhanVien.MaNV=PhanCong.MaNV  
Where MaDA='TH01'
```

4.5. SQL – PHÉP HỢP, PHÉP GIAO

Câu 30. Phép hợp: Sử dụng mệnh đề UNION hợp hai tập hợp. Cho biết họ tên nhân viên thuộc phòng 'NC' hoặc phòng 'DH'

(Select HoTen From NhanVien where MaPH='NC')

UNION

(Select HoTen From NhanVien where MaPH='DH')

Câu 31. Phép giao: Sử dụng mệnh đề INTERSECT giao hai tập hợp. Cho biết họ tên nhân viên thuộc cả hai phòng 'NC' và 'DH'

(Select HoTen From NhanVien where MaPH='NC')

INTERSECT

(Select HoTen From NhanVien where MaPH='DH')

4.5. SQL – PHÉP HỢP, PHÉP GIAO

Câu 32. Phép giao: Cho biết mã nhân viên tham gia cả hai đề án 'TH01' và 'TH02'

(Select MaNV From PhanCong where MaDA='TH01')

INTERSECT

(Select MaNV From PhanCong where MaDA='TH02')

Câu 33. Phép giao: Sử dụng từ khóa IN Cho biết họ tên nhân viên tham gia cả hai đề án 'TH01' và 'TH02'

(Select Hoten From NhanVien

Where MaNV IN ((Select MaNV From PhanCong where
MaDA='TH01')

INTERSECT

(Select MaNV From PhanCong where MaDA='TH02'))

4.6. SQL – PHÉP TRỪ

Câu 34. Phép trừ: Sử dụng mệnh đề **EXCEPT**.

Cho biết mã số của những nhân viên không tham gia đề án nào?

Gợi ý: Tìm tập hợp B “Nhân viên có tham gia đề án (bảng PhanCong), sau đó lấy phần bù bằng cách: Tất cả nhân viên tập hợp A trừ B.

(Select MaNV From NhanVien)

EXCEPT

(Select MaNV From PhanCong)

Câu 35. Phép trừ: Cho biết phòng ban không chủ trì các đề án triển khai năm 2005?

Gợi ý: thực hiện tìm tập hợp B “phòng ban chủ trì các đề án triển khai năm 2005”, sau đó lấy phần bù bằng cách: tất cả phòng ban (tập hợp A) trừ B

(Select MaPH from PhongBan)

EXCEPT

(Select Phong as MaPH from DEAN where NamThucHien=2005)

4.6. SQL – PHÉP TRỪ

Câu 36. Phép trừ: Sử dụng NOT IN. Cho biết mã số, họ tên, ngày tháng năm sinh của những nhân viên không tham gia đề án nào?

```
Select MaNV, HoTen, NTNS from NhanVien  
Where MaNV NOT IN (select MaNV From PhanCong)
```

Câu 37. Phép trừ: Cho biết tên phòng ban không chủ trì các đề án triển khai năm 2005?

```
Select TenPH From PhongBan where MaPH not in (select  
DISTINCT Phong From DeAn Where NamThucHien=2005)
```


4.7. SQL – TRUY VẤN LÒNG

Cú pháp:

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <so sánh tập hợp>

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

4.7. SQL – TRUY VẤN LÒNG

- Các câu truy vấn có thể lồng nhau ở nhiều mức
- Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic
- Câu truy vấn con thường trả về một tập các giá trị
- So sánh tập hợp thường đi cùng một số toán tử như: IN, NOT IN, ALL, ANY hoặc SOME.
- Kiểm tra sự tồn tại: EXISTS, NOT EXISTS

4.7. SQL – TRUY VẤN LÒNG

❖ Có 2 loại truy vấn lồng:

▪ Lồng phân cấp

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

▪ Lồng tương quan

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

4.7. SQL – TRUY VẤN LÒNG

- ❖ Cho lược đồ cơ sở dữ liệu quản lý hàng hóa như sau:
- ❖ KHACHHANG (MaKH, HoTen, DiaChi,DT,NGSinh,NgayDK,DoanhSo)
- ❖ NHANVIEN (MaNV, Hoten, SoDT, NgayLamviec)
- ❖ SANPHAM (MaSP, TenSP, DVT,NuocSX,Gia)
- ❖ HoaDon (SoHD,NGHD, MaKH,MaNV,TriGia)
- ❖ CTHD (SoHD,MaSP,SL)

4.7. SQL – TRUY VẤN LÒNG

❖ Ví dụ lồng phân cấp:

Câu 1: Tìm những nhân viên không lập hóa đơn nào

```
Select MaNV, HoTen, NTNS From NhanVien  
Where MaNV NOT IN (select MaNV From HOADON)
```

Câu 2: Tìm những nhân viên lập hóa đơn cho khách hàng có mã số là 'KH01'

```
Select MaNV, HoTen, NTNS From NhanVien  
Where MaNV IN (select MaNV From HOADON where  
MaKH='KH01')
```

4.7. SQL – TRUY VẤN LÒNG

❖ Ví dụ lồng phân cấp:

Câu 3: Tìm những hóa đơn có trị giá lớn hơn trị giá của ít nhất một hóa đơn do nhân viên có mã 'NV01' lập

Select * From HoaDon

Where TriGia > Any (select Trigia from Hoadon where MaNV='NV01')

Câu 4: Tìm những hóa đơn có trị giá lớn hơn trị giá của tất cả hóa đơn do nhân viên có mã 'NV01' lập.

Select * From HoaDon

Where TriGia > All (select Trigia from Hoadon where MaNV='NV01')

4.7. SQL – TRUY VẤN LÒNG

❖ Ví dụ lồng tương quan:

Câu 1: Tìm những nhân viên lập hóa đơn cho khách hàng có mã số 'KH01'

```
Select MaNV, HoTen From NhanVien  
Where Exists (Select * From Hoadon where MaKH='KH01' and  
Hoadon.MaNV = NhanVien.MaNV)
```

Câu 2: Tìm những nhân viên không lập hóa đơn nào

```
Select * From NhanVien  
Where NOT Exists (Select * From Hoadon  
where Hoadon.MaNV = NhanVien.MaNV)
```

Câu 3: Tìm những hóa đơn có trị giá lớn hơn trị giá của ít nhất một hóa đơn do nhân viên có mã số 'NV01' lập

```
Select * From Hoadon as HD1  
Where Exists (select * From Hoadon as HD2  
where MaNV='NV01' and HD1.Trigial > HD2.Trigial)
```

4.7. SQL – TRUY VẤN LÒNG

Đối với Exists không nhất thiết liệt kê tên thuộc tính ở mệnh đề SELECT của truy vấn con

Những câu truy vấn có any hoặc in đều có thể chuyển thành câu truy vấn có Exists

4.8. Phép chia tập hợp (/ hay ÷)

Biểu diễn phép chia trong SQL:

Sử dụng NOT EXISTS để biểu diễn

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

```
FROM R R1
```

```
WHERE NOT EXISTS (
```

```
    SELECT *
```

```
    FROM S
```

```
    WHERE NOT EXISTS (
```

```
        SELECT *
```

```
        FROM R R2
```

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

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

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

4.9. SQL – MỆNH ĐỀ GROUP BY

Quan hệ NV

nhóm

Q	S
a	10
a	2
b	9
b	5
c	10
c	8
c	6
c	4
c	10
d	16
d	18
d	50

Chia các dòng thành các nhóm dựa trên tập thuộc tính chia nhóm

Q	Count(S)
a	2
b	2
c	5
d	3

Tương tự cho các hàm SUM, MIN, MAX, AVG

Các thuộc tính GROUP BY: Q

Câu SQL:
Select Q, **count(S)**
From NV
Group by Q

4.10. HÀM COUNT,SUM,MAX,MIN,AVG

a) Sử dụng các hàm COUNT, SUM, MIN, MAX, AVG trên 1 nhóm lớn (trên toàn bộ quan hệ):

– Câu hỏi 1: Số lượng nhân viên của công ty.

Select COUNT(MaNV) as SoNV from NhanVien

– Câu hỏi 2: Số lượng nhân viên quản lý trực tiếp nhân viên khác.

Select COUNT (DISTINCT Ma_NQL) from NhanVien

– Câu hỏi 3: Tìm mức lương lớn nhất, mức lương trung bình, tổng lương của công ty.

Select MAX(Luong), AVG(Luong), SUM(Luong) from NhanVien

– Câu hỏi 4: Cho biết nhân viên có mức lương lớn nhất.

Select HoTen from NhanVien

Where Luong = (Select MAX(Luong) from NhanVien)

4.11. SQL – MỆNH ĐỀ GROUP BY

Câu hỏi 5: Cho biết nhân viên có mức lương trên mức lương trung bình của công ty.

Select HoTen from NhanVien where **Luong** > (Select **AVG(Luong)** from NhanVien)

b) Sử dụng các hàm COUNT, SUM, MIN, MAX, AVG trên từng nhóm nhỏ: mệnh đề GROUP BY

- Chia các dòng thành các nhóm nhỏ dựa trên tập thuộc tính chia nhóm.
- Thực **hiện các phép toán trên nhóm** như: Count (thực hiện phép đếm), Sum (tính tổng), Min(lấy giá trị nhỏ nhất), Max(lấy giá trị lớn nhất), AVG (lấy giá trị trung bình).

4.11. SQL – MỆNH ĐỀ GROUP BY

Câu hỏi 6: Cho biết số lượng nhân viên theo từng phái (giới tính)?

Do cột phái có 2 giá trị “nam” và “nữ”, trường hợp này ta chia bảng NhanVien thành 2 nhóm nhỏ. Thuộc tính chia nhóm là thuộc tính “Phai”.

➡ **Select Phai, count(MaNV) as SoNV from NhanVien
Group by Phai**

Câu hỏi 7: Cho biết số lượng nhân viên theo từng phòng?

Do cột MaPH có 3 giá trị “NC” và “DH” và “QL”, trường hợp này ta chia bảng nhân viên thành 3 nhóm nhỏ. Thuộc tính chia nhóm là thuộc tính “MaPH”.

➡ **Select MaPH, count(Manv) from NhanVien Group by MaPH**

Tương tự: cho biết tổng lương của mỗi phòng, cho biết mức lương thấp nhất của từng phòng, mức lương cao nhất, mức lương trung bình của từng phòng

4.11. SQL – MỆNH ĐỀ GROUP BY

Câu hỏi 8: Cho biết tên phòng và số lượng nhân viên theo từng phòng?

Thuộc tính chia nhóm là (TenPH) thay cho MaPH.

➡ **Select** TenPH, count(Manv) **as** SoLuongNV
From NhanVien n, PhongBan p **Where** n.MaPh=p.MaPH
Group by TenPH

Câu hỏi 9: Với mỗi phòng, cho biết số lượng nhân viên theo từng phái?

Do cột MaPH có 3 giá trị “NC” và “DH” và “QL”, mỗi phòng chia nhỏ theo từng phái: 2 nhóm “Nam” và “Nữ”, trường hợp này ta chia bảng nhân viên thành 6 nhóm nhỏ. Như vậy, tập thuộc tính chia nhóm cho câu truy vấn là (Phong, Phai).

➡ **Select** MaPH, Phai, count(MaNV) **from** NhanVien
Group by MaPH, Phai

4.11. SQL – MỆNH ĐỀ GROUP BY

Câu hỏi 10: Đếm số đề án của từng nhân viên tham gia?

- Do cột MaNV có 7 giá trị “NV001”,...“NV008” (không có nhân viên “005”), trường hợp này ta chia bảng PhanCong thành 7 nhóm nhỏ. Với mỗi nhóm nhỏ (MaNV), ta đếm số đề án (count(MADA)) tham gia. Thuộc tính chia nhóm là thuộc tính “MaNV”.

➔ **Select** MaNV, count(MaDA) **as** SoDATG **From** PhanCong
Group by MaNV

- Tương tự: tính tổng số giờ làm việc của mỗi nhân viên (SUM), thời gian làm việc thấp nhất của mỗi nhân viên (MIN), thời gian làm việc lớn nhất của mỗi nhân viên (MAX), thời gian làm việc trung bình,...

4.11. SQL – MỆNH ĐỀ GROUP BY

Câu hỏi 11: Cho biết mã, tên nhân viên và số đề án mà n/v đã tham gia?



```
Select n.MaNV, HoTen, count(MaDA) as SoDATG  
From PhanCong pc, NhanVien n where pc.manv=n.manv  
Group by MaNV, HoTen
```

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

4.12. SQL – MỆNH ĐỀ HAVING

- Lọc kết quả theo điều kiện, sau khi đã gom nhóm
- Điều kiện của HAVING là điều kiện về các hàm tính toán trên nhóm (Count, Sum, Min, Max, AVG) và các thuộc tính trong danh sách GROUP BY.

Câu hỏi 12: Cho biết những nhân viên tham gia từ 2 đề án trở lên?

```
Select MaNV, count(MaDA) as SoDATG From PhanCong  
Group by MaNV  
Having count(MaDA) >=2
```

Câu hỏi 13: Cho biết mã phòng ban có trên 4 nhân viên?

```
Select MaPH, count(Manv) from NhanVien Group by MaPH  
Having count(Manv)>4
```

4.12. SQL – MỆNH ĐỀ HAVING

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

4.12. SQL – MỆNH ĐỀ HAVING

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

4.13. MỘT SỐ VÍ DỤ KHÁC

Câu 1: Cho biết giá trị hóa đơn cao nhất, thấp nhất

```
Select Max(Trigia) as Trigiacaonhat, Min(Trigia) as Trigiathapnhap  
FROM HOADON
```

Câu 2: Cho biết số lượng hóa đơn do nhân viên “Nguyen Van A” lập.

```
Select count(*) as soluonghoadon  
FROM HOADON HD, NHANVIEN NV  
WHERE HD.MANV=NHANVIEN.MANV AND HOTEN=“Nguyen Van  
A ”
```

4.13. MỘT SỐ VÍ DỤ KHÁC

Câu 3: Tính doanh thu bán hàng trong năm 2022

```
Select SUM(Trigia) as Doanhthu_Nam2022  
FROM HOADON  
WHERE YEAR(NGHD)='2022'
```

Câu 4: Cho biết trị giá trung bình của tất cả các hóa đơn trong năm 2022

```
Select AVG(Trigia) as TrigiaTrungbinh_Nam2022  
FROM HOADON  
WHERE YEAR(NGHD)='2022'
```

4.13. MỘT SỐ VÍ DỤ KHÁC

Câu 5: Cho biết số lượng sản phẩm của từng nước?

```
Select NuocSX, count(*) as SoluongSP  
FROM SANPHAM  
GROUP BY NuocSX
```

Câu 6: Với mỗi nhân viên cho biết mã số, số lượng hóa đơn và tổng trị giá mà họ đã lập

```
Select MaNV, count (*) as SL_HD, sum (Trigia) as Tong_Trigia_HD  
FROM HOADON  
GROUP BY MANV
```

4.13. MỘT SỐ VÍ DỤ KHÁC

Câu 7: Cho biết những nhân viên lập từ 2 hóa đơn trở lên?

```
Select MaNV FROM HOADON  
GROUP BY MANV  
HAVING Count(*) >=2
```

Câu 8: Cho biết những nhân viên (HOTEN) có trị giá hóa đơn đã lập trung bình lớn hơn 1 triệu đồng

```
Select Hoten, avg(Trigia) as Trigia_TB  
FROM HOADON AS HD, NHANVIEN AS NV  
WHERE NHANVIEN.MANV=HOADON.MANV  
GROUP BY HOTEN  
HAVING AVG(Trigia)>1000000
```

4.13. MỘT SỐ VÍ DỤ KHÁC

Câu 9: Tìm những nhân viên có trị giá hóa đơn đã lập trung bình cao nhất

```
Select MaNV, AVG(TriGia) as Trigia_TB
```

```
FROM HOADON
```

```
GROUP BY MANV
```

```
HAVING AVG(TriGia) >=All
```

```
(Select AVG(TriGia)
```

```
FROM HOADON
```

```
GROUP BY MANV)
```


Tài liệu tham khảo

1. Slides bài giảng Quản lý dự án CNTT, Khoa HTTT, Trường Đại học CNTT.
2. ThS. Nguyễn Thị Kim Phụng, Slides bài giảng Cơ sở dữ liệu, Khoa HTTT, Trường Đại học CNTT
3. ThS. Nguyễn Hải Châu, Slides bài giảng Cơ sở dữ liệu, Đại học Công nghệ, ĐH Quốc gia Hà Nội
4. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, Seven Edition, 2016



THANK YOU!

Q & A

ThS. TẠ VIỆT PHƯƠNG
phuongtv@uit.edu.vn