



KHOA HỆ THỐNG THÔNG TIN
FACULTY OF INFORMATION SYSTEMS

TÀI LIỆU HƯỚNG DẪN THỰC HÀNH

MÔN CƠ SỞ DỮ LIỆU – IT004

BUỔI THỰC HÀNH 02

Hướng dẫn thực hành

Lê Võ Đình Kha – khalvd@uit.edu.vn



TÀI LIỆU HƯỚNG DẪN THỰC HÀNH

MÔN CƠ SỞ DỮ LIỆU – IT004

BUỔI THỰC HÀNH 02

Hướng dẫn thực hành

Lê Võ Đình Kha – khalvd@uit.edu.vn



TÀI LIỆU ĐƯỢC BIÊN SOẠN DỰA TRÊN CÁC NGUỒN TÀI LIỆU THAM KHẢO

1. Slides môn Cơ sở dữ liệu, Khoa Hệ thống Thông tin, Trường Đại học Công nghệ Thông tin, ĐHQG Tp. HCM.
2. Tài liệu thực hành môn Cơ sở dữ liệu Khoa HTTT, Giảng viên lý thuyết môn CSDL và quý Thầy Cô phụ trách giảng dạy môn CSDL.
3. Giáo trình Cơ sở dữ liệu, Đồng Thị Bích Thủy, Phạm Thị Bạch Huệ, Nguyễn Trần Minh Thư, NXB Khoa học và Kỹ thuật.

GIỚI THIỆU HỌC PHẦN THỰC HÀNH BUỔI SỐ 2

NỘI DUNG



Phần 1: Ôn tập buổi Thực hành số 1.

- Các kiểu dữ liệu trong SQL.
- Các lệnh về Database, về Table (Create, Drop, Alter).
- Các lệnh tạo ràng buộc toàn vẹn.
- Các lệnh thao tác dữ liệu (Insert, Update, Delete).

Phần 2: Các câu lệnh truy vấn khác

- Mệnh đề Select, From, Where, Order by.
- Các phép kết, Các phép toán tập hợp.
- Truy vấn lồng.

CÁC KIỂU DỮ LIỆU TRONG SQL

Trong Microsoft SQL mỗi cột (column), biến cục bộ (local variable) biểu thức (expression) và tham số (parameter) đều có một kiểu dữ liệu liên quan. Sau đây là 6 kiểu dữ liệu trong SQL.

1. Kiểu dữ liệu **Exact Numeric (Số chính xác, không sai số) trong SQL**
2. Kiểu dữ liệu **Approximate Numeric trong SQL**
3. Kiểu dữ liệu **Date và Time trong SQL**
4. Kiểu dữ liệu **Character String (Kiểu chuỗi) trong SQL**
5. Kiểu dữ liệu **Unicode Character String (Kiểu chuỗi có chứa Unicode) trong SQL**
6. Kiểu dữ liệu **Binary trong SQL**
7. Các kiểu dữ liệu khác

Tài liệu tham khảo về các kiểu dữ liệu:

<https://learn.microsoft.com/vi-vn/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15>

CÁC KIỂU DỮ LIỆU TRONG SQL

1. Kiểu dữ liệu Exact numeric (Số chính xác, không sai số) trong SQL

Kiểu dữ liệu	Mô tả
smallmoney	– 214,748.3648 tới +214,748.3647
money	-922,337,203,685,477.5808 tới +922,337,203,685,477.5807
numeric	$-10^{38} + 1$ tới $10^{38} - 1$
decimal	$-10^{38} + 1$ tới $10^{38} - 1$
bit	0 tới 1
tinyint	0 tới 225
smallint	-32,768 tới 32,767
int	-2,147,483,648 tới 2,147,483,647
bigint	-9,223,372,036,854,775,808 tới 9,223,372,036,854,775,807

CÁC KIỂU DỮ LIỆU TRONG SQL

2. Kiểu dữ liệu Approximate Numeric trong SQL

Kiểu dữ liệu	Mô tả
real	-3.40E + 38 tới 3.40E + 38
float	-1.79E + 308 tới 1.79E + 308

CÁC KIỂU DỮ LIỆU TRONG SQL

3. Kiểu dữ liệu Date và Time trong SQL

Kiểu dữ liệu	Mô tả
time	Lưu trữ 1 time dạng 12:30 P.M
date	Lưu trữ 1 date dạng June 30,1991
smalldatetime	Jan 1,1900 tới Jun 6, 2079
datetime	Jan 1, 1753 tới Dec 31, 9999

Lưu ý: Khi nhập dữ liệu, dữ liệu ngày tháng năm được đặt trong các cặp ngoặc đơn và theo đúng format 'yyyy-mm-dd'

Ví dụ: '2022-12-12'. Sinh viên có thể tìm hiểu thêm về các cách nhập. Gợi ý **SET DATEFORMAT dmy**

CÁC KIỂU DỮ LIỆU TRONG SQL

4. Kiểu dữ liệu Character String (Kiểu chuỗi) trong SQL

Kiểu dữ liệu	Mô tả
text	Không chứa Unicode, độ dài tối đa đến 2.147.483.647 ký tự, dữ liệu không phải Unicode có độ dài có thể thay đổi
varchar (max)	Không chứa Unicode, độ dài tối đa đến 231 ký tự, dữ liệu không phải Unicode có độ dài có thể thay đổi
varchar	Không chứa Unicode, độ dài tối đa đến 8.000 ký tự (dữ liệu không phải Unicode có độ dài có thể thay đổi)
char	Không chứa Unicode, độ dài tối đa đến 8.000 ký tự (dữ liệu không phải Unicode có độ dài không thể thay đổi)

CÁC KIỂU DỮ LIỆU TRONG SQL

5. Kiểu dữ liệu Unicode Character String (Kiểu chuỗi có chứa Unicode) trong SQL

Kiểu dữ liệu	Mô tả
ntext	Độ dài tối đa đến 1.073.741.823 ký tự
nvarchar (max)	Độ dài tối đa là 231 ký tự, Unicode có độ dài có thể thay đổi
nvarchar	Độ dài tối đa đến 4.000 ký tự (Unicode có độ dài có thể thay đổi)
nchar	Độ dài tối đa đến 4.000 ký tự (Unicode có độ dài cố định)

CÁC KIỂU DỮ LIỆU TRONG SQL

6. Kiểu dữ liệu Binary trong SQL

Kiểu dữ liệu	Mô tả
image	Độ dài tối đa là 2.147.483.647 byte (dữ liệu binary có độ dài có thể thay đổi)
varbinary (max)	Độ dài tối đa là 2 ³¹ byte, dữ liệu binary có độ dài có thể thay đổi (chỉ với SQL Server 2005)
varbinary	Độ dài tối đa là 8.000 byte (dữ liệu binary có độ dài có thể thay đổi)
binary	Độ dài tối đa là 8.000 byte (dữ liệu binary có độ dài không thể thay đổi)

CÁC KIỂU DỮ LIỆU TRONG SQL

7. Các kiểu dữ liệu khác

Kiểu dữ liệu	Mô tả
table	Lưu giữ một tập hợp kết quả để xử lý vào lần sau
cursor	Tham chiếu tới một đối tượng con trỏ (Cursor)
xml	Lưu giữ dữ liệu XML. Bạn có thể lưu giữ xml trong một column hoặc một biến (chỉ với SQL Server 2005)
uniqueidentifier	Lưu giữ một định danh chung (Globally Unique Identifier – GUID)
timestamp	Lưu giữ một số duy nhất mà được cập nhật mỗi khi một hàng được cập nhật
sql_variant	Lưu giữ các giá trị của các kiểu dữ liệu đa dạng được hỗ trợ bởi SQL Server, ngoại trừ text, ntext, và timestamp

CÁC LỆNH VỀ DATABASE

- **Tạo Database:** Để tạo một **Database** mới ta sử dụng cú pháp:

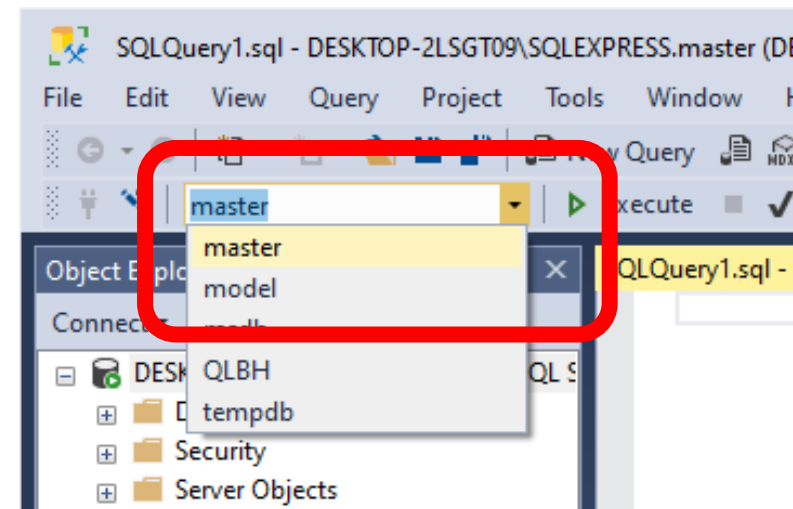
CREATE DATABASE <Tên Database>

Ví dụ: CRETAE DATABASE **QuanLyGiaoVu**

- **Sử dụng Database:** Sau khi tạo mới một Database ta cần dùng lệnh **USE** để chỉ định Database sẽ thao tác trên đó. Cú pháp như sau:

USE <Tên Database>

Ví dụ: USE **QuanLyGiaoVu**



- Chú ý "Available Database" có đúng Database cần thao tác chưa

CÁC LỆNH VỀ DATABASE

- **Xóa Database:** Để xóa một **Database** ta sử dụng cú pháp:

DROP DATABASE *<Tên Database>*

Ví dụ: DROP DATABASE **QuanLyGiaoVu**

- **Xóa nhiều Database:** Để xóa nhiều database cùng một lúc:

DROP DATABASE *<Tên Database 1>, ..., <Tên Database N>*

Ví dụ: DROP DATABASE **QuanLyGiaoVu, QuanLyCaSi, QuanLyBanHang**

- *Chú ý những database đang được sử dụng (Available Database) sẽ không xóa được.*

CÁC LỆNH VỀ TABLE

- SQL sử dụng các thuật ngữ tương đương với thuật ngữ về cơ sở dữ liệu như sau:

SQL	Cơ sở dữ liệu
Bảng	Quan hệ
Cột	Thuộc tính
Dòng dữ liệu	Bộ giá trị

- Để định nghĩa một bảng trong SQL ta cần: tên bảng, các cột, kiểu dữ liệu của cột và các ràng buộc toàn vẹn (RBTV – Integrity constraint) trên nó.

CÁC LỆNH VỀ TABLE

- Để tạo **Table**, ta sử dụng cú pháp sau:

CREATE TABLE *<Tên Table>*

(

<Tên column 1> <Kiểu dữ liệu> (integrity_constraint),

<Tên column 2> <Kiểu dữ liệu> (integrity_constraint),

)

CÁC LỆNH VỀ TABLE

- Ví dụ:

CREATE TABLE KHACHHANG

(

MaKH char(4) PRIMARY KEY,

HoTen varchar(40) NOT NULL,

DiaChi varchar(50) NOT NULL,

NgSinh smalldatetime,

DoanhSo money,

)

CÁC LỆNH VỀ TABLE

- Để xóa **Table**, ta sử dụng cú pháp sau:

DROP TABLE <Tên Table>

Ví dụ: DROP TABLE **KHACHHANG**

Lưu ý: Không thể **drop** được **table** khi nó đang có liên kết khóa ngoại.
Muốn xóa **table** đó trước hết phải xóa khóa ngoại đi.

Mở rộng các lệnh DROP khác:

- Để xóa một bảng View ta sử dụng cú pháp

DROP VIEW *<Tên View>*

- Để xóa một Procedure ta sử dụng cú pháp

DROP (PROCEDURE | PROC) *procedure_name*

- **Ngoài ra** còn các lệnh DROP khác như là 'DROP INDEX', 'DROP ROLE', 'DROP SCHEMA', 'DROP RULE'...

CÁC LỆNH VỀ TABLE

- **SQL** hỗ trợ **ALTER TABLE** cho phép chúng ta thêm, xóa hay sửa các thành phần của table.
- Để sửa **Table**, ta sử dụng cú pháp sau:

Thêm cột

ALTER TABLE <Tên Table> **ADD** <Tên cột> <Kiểu dữ liệu>

Xóa cột

ALTER TABLE <Tên Table> **DROP COLUMN** <Tên cột>

Sửa cột (Thay đổi kiểu dữ liệu)

ALTER TABLE <Tên Table> **ALTER COLUMN** <Tên cột> <Kiểu dữ liệu mới>

CÁC LỆNH VỀ TABLE

- Để thực hiện đổi tên (Rename) một bảng ta sử dụng Stored Procedure **sp_name**:

Sửa tên bảng

EXEC sp_rename <Tên Table>, <Tên Table mới>

Sửa tên cột

EXEC sp_rename <Tên Table.Tên Column>, <Tên Column mới>, <COLUMN>

- Các lệnh **sp_help**, **sp_tables**, **sp_columns**, **sp_pkeys**, **sp_fkeys**, **sp_helpconstraint**, **sp_helpsql**

- Lưu ý: Sinh viên có thể tham khảo thêm các lệnh sp_ khác tại: [TẠI ĐÂY](#)

RÀNG BUỘC TOÀN VỆN

- Việc ràng buộc toàn vẹn trong SQL Server được chia làm 2 loại chính
 - ❖ **Loại đơn giản:** Sử dụng **CONSTRAINT** để mô tả
 - ❖ **Loại phức tạp:** Sử dụng **TRIGGER** để thực hiện

RÀNG BUỘC TOÀN VỆN

- Một số ràng buộc thường sử dụng:
 - ✓ **NOT NULL:** Dữ liệu không được để trống.
 - ✓ **UNIQUE:** Dữ liệu không được trùng lặp.
 - ✓ **PRIMARY KEY:** Ràng buộc khóa chính (Not null + Unique)
 - ✓ **FOREIGN KEY:** Ràng buộc khóa ngoại.
 - ✓ **CHECK:** Kiểm tra dữ liệu nhập vào phải theo định dạng.
 - ✓ **DEFAULT:** Tạo giá trị mặc định.

- Các nguyên tắc chung

- ❖ Một **constraint** luôn gắn với một bảng.
- ❖ Nếu không đặt tên thì hệ thống sẽ tự động phát sinh tên cho **constraint**.
- ❖ Có thể tạo **constraint**:
 - o Cùng với thời điểm tạo bảng.
 - o Sau khi đã tạo bảng xong (dùng câu lệnh Alter).
- ❖ Có thể khai báo **constraint** ở mức cột hoặc mức bảng.
- ❖ Có thể xem các **constraint** hiện có trong database.

RÀNG BUỘC TOÀN VỆN

Cú pháp chi tiết từng loại constraint

(CONSTRAINT ten_constraint)

PRIMARY KEY (danh_sach_cot_khoa_chinh)

- Hoặc UNIQUE (danh_sach_cot)

- Hoặc

FOREIGN KEY (danh_sach_cot_khoa_ngoai)

REFERENCES bang_tham_chieu (ds_cot_tham_chieu)

- Hoặc CHECK (bieu_thuc_luan_ly)

- Hoặc DEFAULT gia_tri_mac_dinh FOR Ten_cot

RÀNG BUỘC TOÀN VỆN

Cách 1: Cách viết Constraint: Định nghĩa ràng buộc ngay sau kiểu dữ liệu

```
CREATE TABLE KHACHHANG
```

```
(
```

```
    MAKH char(4) PRIMARY KEY,
```

```
    HOTEN varchar(40) NOT NULL,
```

```
    SDT varchar(20) UNIQUE,
```

```
    DOANHISO money DEFAULT 0,
```

```
)
```

- **Lưu ý:** Nên dùng với Primary Key, Not Null, Unique, Default.

RÀNG BƯỚC TOÀN VỆN

Cách 2: Cách viết Constraint: Định nghĩa ràng buộc ngay tại dòng cuối cùng của lệnh tạo bảng.

```
CREATE TABLE CTHD
(  
    SOHD int,  
    MASP char(4),  
    SL int,  
    constraint pk_cthd primary key(SOHD,MASP),  
)
```

- Lưu ý: Nên dùng với Primary Key (Khóa chính có nhiều thuộc tính).
- Có thể viết tắt: Primary key(SOHD,MASP)

RÀNG BUỘC TOÀN VỆN

Cách 3: Cách viết Constraint: Thêm ràng buộc sau khi tạo bảng.

- Sử dụng lệnh sửa bảng (ALTER TABLE) và thêm (ADD) ràng buộc.

Ví dụ:

```
ALTER TABLE HOADON ADD FOREIGN KEY(MAKH) REFERENCES  
KHACHHANG(MAKH)
```

```
ALTER TABLE HOADON ADD CONSTRAINT fk02_HD FOREIGN KEY(MANV)  
REFERENCES NHANVIEN(MANV)
```

- Lưu ý:** Nên dùng với Foreign Key, Check
- Khóa ngoại ở bảng nào thì sửa bảng đó

RÀNG BUỘC TOÀN VỆN

Ràng buộc CHECK:

- Dùng kiểm tra dữ liệu nhập vào thỏa điều kiện nhất định.
- Áp dụng cho : Ràng buộc Miền giá trị, Ràng buộc Liên thuộc tính (Xem lý thuyết Chương 5).

Cách viết:

ALTER TABLE <Tên bảng> ADD CHECK (Điều kiện)

Ví dụ:

```
ALTER TABLE KHACHHANG ADD CHECK (DOANHSON >= 0)
ALTER TABLE SANPHAM ADD CONSTRAINT CHK_DVT
CHECK(( DVT = 'CAY' ) OR ( DVT = 'HOP' ) OR (DVT = 'CAI'))
```

RÀNG BUỘC TOÀN VỆN

Toán tử **LIKE**:

- Dùng kiểm tra dữ liệu kiểu chuỗi theo một định dạng nhất định.
- Dấu gạch dưới '_' : thay thế cho 1 ký tự.
- Dấu phần trăm '%': thay thế cho 0, 1, hoặc n ký tự.

• Ví dụ:

LIKE 'a__'	Chuỗi bắt đầu là a , sau a có 2 ký tự.
LIKE 'a%'	Chuỗi bắt đầu là a , sau a có thể có 0 hoặc n ký tự.
LIKE '_a___'	Ký tự thứ 2 là a, sau a có 3 ký tự

ALTER TABLE **SINHVIENT** ADD CHECK (MSSV LIKE '__52__')

CÁC LỆNH THAO TÁC DỮ LIỆU

- Các lệnh thao tác dữ liệu Data Manipulation Language (DML):

Thêm mới dữ liệu.

INSERT INTO <Tên bảng> **VALUES**('Giá trị 1', 'Giá trị 2', Giá trị 3)

Ví dụ:

INSERT INTO NHANVIEN VALUES

('NV01','Nguyen Nhu Nhut', '0927345678', '13/04/2006')

INSERT INTO NHANVIEN (MANV, HOTEN, SODT, NGVL)

VALUES ('NV01', 'Nguyen Nhut', '0927345678','13/04/2006')

CÁC LỆNH THAO TÁC DỮ LIỆU

- Các lệnh thao tác dữ liệu Data Manipulation Language (DML):

Xóa dữ liệu.

DELETE FROM <Tên bảng> **WHERE** <Điều kiện>

Ví dụ:

DELETE FROM NHANVIEN **WHERE** MANV='NV01'

DELETE FROM SINHVIEN **WHERE** MSSV LIKE '1852____'

CÁC LỆNH THAO TÁC DỮ LIỆU

- Các lệnh thao tác dữ liệu Data Manipulation Language (DML):

Cập nhật, chỉnh sửa dữ liệu.

UPDATE <Tên bảng> SET <Tên cột 1> = <Giá trị mới>,
<Tên cột 2> = <Giá trị mới> WHERE <Điều kiện>

Ví dụ:

UPDATE NHANVIEN SET HOTEN='Nguyễn Văn Tiên', SODT='083568711'
WHERE MANV='NV02'

CÁC LỆNH THAO TÁC DỮ LIỆU

- Các lệnh thao tác dữ liệu Data Manipulation Language (DML):

Sao chép bảng dữ liệu

SELECT * INTO <Tên bảng mới> FROM <Tên bảng cũ>

Ví dụ: Sao chép bảng KHACHHANG thành KHACHHANG_NEW

SELECT * INTO KHACHHANG_NEW FROM KHACHHANG

CÂU LỆNH TRUY VẤN

- Cú pháp tổng quát:

```
SELECT (DISTINCT) <Danh sách các thuộc tính / hàm >  
FROM <Danh sách các bảng>  
(WHERE <Điều kiện> )  
(GROUP BY <Danh sách thuộc tính gom nhóm> )  
(HAVING <Điều kiện trên nhóm> )  
(ORDER BY {<Thuộc tính sắp xếp> ASC / DESC })
```

CÂU LỆNH TRUY VẤN SELECT, FROM, WHERE

- Ví dụ: Cho biết **họ tên, ngày vào làm** của nhân viên có số điện thoại là **0342565857**
 - Quan hệ: **NHANVIEN**
 - Thuộc tính: **HOTEN, NGVL**
 - Điều kiện: **SODT = '0342565857'**

- Cách viết:

SELECT HOTEN, NGVL

FROM NHANVIEN

WHERE SODT = **'0342565857'**

- Các loại phép kết
 - JOIN
 - INNER JOIN
 - OUTER JOIN
 - FULL JOIN
 - LEFT JOIN
 - RIGHT JOIN
 - SELF JOIN

CÁC PHÉP KẾT

- Sử dụng Phép kết bằng: Để kết 02 hay nhiều bảng
 - Cách 1: Sử dụng **inner join on** <điều kiện kết> ở **FROM**

- Cú pháp 1:

SELECT COLUMN 1, COLUMN 2

FROM Table 1 INNER JOIN Table 2

ON <Điều kiện kết>

- Ví dụ: Liệt kê các nhân viên và tên phòng ban của nhân viên

Cách viết 1:

```
SELECT MaNV, HoTen, TenPH  
FROM NHANVIEN inner join PHONGBAN  
on NHANVIEN.Phong = PHONGBAN.MaPH
```

CÁC PHÉP KẾT

- Sử dụng Phép kết bằng: Để kết 02 hay nhiều bảng

- Cách 2: Đưa <điều kiện kết> xuống **WHERE**

- Cú pháp 2:

SELECT Column 1, Column 2 **AS** Name

FROM Table 1, Table 2 **AS** Name 1

WHERE <Điều kiện kết> <Điều kiện chọn>

ORDER BY Column1 **ASC**, Column2 **DESC**

- Ví dụ: Liệt kê các nhân viên và tên phòng ban của nhân viên

Cách viết 2: **SELECT** MaNV, HoTen, TenPH
FROM NHANVIEN, PHONGBAN
WHERE NHANVIEN.Phong = PHONGBAN.MaPH

Hoặc

SELECT MaNV, HoTen, TenPH
FROM NHANVIEN NV, PHONGBAN PB
WHERE NV.Phong = PB.MaPH

PHÉP TOÁN TẬP HỢP

Sử dụng phép toán tập hợp để kết hợp kết quả của 02 câu truy vấn

(Câu truy vấn 1)

<PHÉP TOÁN TẬP HỢP>

(Câu truy vấn 2)

- Các phép toán tập hợp:

- Hội: UNION
- Giao: INTERSECT
- Trừ: EXCEPT

PHÉP TOÁN TẬP HỢP

- Hội: **UNION**
- Ví dụ: Liệt kê các mã nhân viên thực hiện đề án DA01 **hoặc** DA02

```
(SELECT MaNV  
FROM PHANCONG  
WHERE MaDA = 'DA01')
```

UNION

```
(SELECT MaNV  
FROM PHANCONG  
WHERE MaDA = 'DA02')
```

PHÉP TOÁN TẬP HỢP

- Giao: **INTERSECT**
- Ví dụ: Liệt kê các mã nhân viên thực hiện cả 2 đề án DA01 và DA02

```
(SELECT MaNV  
FROM PHANCONG  
WHERE MaDA = 'DA01')
```

INTERSECT

```
(SELECT MaNV  
FROM PHANCONG  
WHERE MaDA = 'DA02')
```

PHÉP TOÁN TẬP HỢP

- Trừ: **EXCEPT**

- Ví dụ: Liệt kê các mã nhân viên thực hiện đề án DA01 **nhưng không thực hiện đề án DA02**

```
(SELECT MaNV  
FROM PHANCONG  
WHERE MaDA = 'DA01')
```

EXCEPT

```
(SELECT MaNV  
FROM PHANCONG  
WHERE MaDA = 'DA02')
```

CÂU TRUY VẤN LỒNG

— Lồng phân cấp

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

— Lồng tương quan

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

- Cú pháp:

SELECT <Danh sách các cột>

FROM <Danh sách các bảng>

WHERE<Phép so sánh tập hợp>(SELECT <Danh sách các cột>
FROM <Danh sách các bảng>
WHERE <Phép so sánh tập hợp>)

CÂU TRUY VẤN LỒNG

- Lồng phân cấp
 - So sánh tập hợp thường đi kèm:
 - + IN / NOT IN
 - + ALL
 - + SOME / ANY
 - Trước ALL, ANY có toán tử so sánh. Ví dụ: > ALL, = ANY
- Ví dụ: Xem thêm slide lý thuyết Chương 4 – NGÔN NGỮ SQL
- Lồng tương quan
 - Kiểm tra sự tồn tại:
 - + EXIST
 - + NOT EXIST

BÀI TẬP THỰC HÀNH

Yêu cầu: Sử dụng phần mềm **Microsoft SQL Server** và truy cập website môn học, tiến hành thực hiện các bài tập sau:

- Phần I bài tập **QuanLyBanHang** từ câu 2 đến câu 10.
- Phần II bài tập **QuanLyBanHang** từ câu 2 đến câu 5.
- Phần III bài tập **QuanLyBanHang** từ câu 1 đến câu 19.
- Phần I bài tập **QuanLyGiaoVu** từ câu 3 đến câu 8. và từ câu 11 đến câu 14.
- Phần II bài tập **QuanLyGiaoVu** từ câu 1 đến câu 4.
- Phần III bài tập **QuanLyGiaoVu** từ câu 1 đến câu 18.

HỎI - ĐÁP

