

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN


BÀI TẬP THỰC HÀNH HỆ CƠ SỞ DỮ LIỆU

Bài tập được thiết kế theo từng TUẦN, có thời lượng 3 tiết/buổi thực hành với sự hướng dẫn của GV. Cuối mỗi buổi thực hành, sinh viên nộp lại phần bài tập mình đã thực hiện cho GV hướng dẫn. Những câu hỏi mở rộng/khó giúp sinh viên trau dồi thêm kiến thức của môn học. Sinh viên phải có trách nhiệm nghiên cứu, tìm câu trả lời nếu chưa thực hiện xong trong giờ thực hành.

TP. Hồ Chí Minh
Năm 2022

BÀI TẬP TUẦN 1

Số tiết: 3

Mục tiêu:

- Khởi động và sử dụng SQL Server Management Studio
- Tạo Database - cơ sở dữ liệu (CSDL) và thực hiện các thao tác cơ bản trên CSDL bằng lệnh và bằng công cụ design
- Tạo các Table (Bảng dữ liệu) và nhập dữ liệu bằng công cụ design
- Tạo lược đồ quan hệ (Relationship Diagram)
- Biết các kiểu dữ liệu (DataType) trong SQL Server
- Biết tạo, sửa, xóa và áp dụng các kiểu dữ liệu trong SQL Server
- Biết sử dụng một số thủ tục trợ giúp về database và Datatype

PHẦN 1: SQL SERVER DATABASE ENGINE & SQL SERVER MANAGEMENT STUDIO

1. Giới thiệu

- **SQL Server Database Engine** là dịch vụ (service) cốt lõi của MS SQL Server, cung cấp chức năng lưu trữ, xử lý, kiểm soát truy suất dữ liệu, quản lý giao tác. Dịch vụ chạy background, đáp ứng yêu cầu từ nhiều user. Dịch vụ có thể tự động khởi động (automatic) khi khởi động hệ điều hành hoặc có thể khởi động bằng tay (manual).

Khởi động một instance của SQL Server Database Engine là khởi động một SQL Server service (server). Sau khi khởi động SQL Server service thành công, người dùng có thể thiết lập một kết nối đến server và sử dụng các chức năng được cung cấp từ server.

- **SQL Server Management Studio (SSMS)** là một môi trường tích hợp cho phép truy cập, cấu hình, quản lý, quản trị và phát triển tất cả các dịch vụ của MS SQL Server. Nó kết hợp một nhóm công cụ đồ họa và trình soạn thảo script hỗ trợ cho người lập trình và người quản trị database tương tác với các dịch vụ của MS SQL Server.

SQL Server Management Studio kết hợp các đặc trưng của Enterprise Manager, Query Analyzer, and Analysis Manager vào một môi trường thống nhất. Bên cạnh đó, SQL Server Management Studio làm việc với tất cả công cụ của SQL Server như Reporting Services and Integration Services. Người lập trình cũng như người quản trị dễ dàng thao tác trên một môi trường đồng nhất và thân thiện.

2. Khởi động SQL Server Management Studio (SSMS)

- Đảm bảo SQL Server service đã khởi động (đang chạy) trên localhost

- Vào Start menu → chọn program → chọn Microsoft SQL Server 2012 → chọn SQL Server Management Studio → khai báo hộp thoại để connect tới Sever

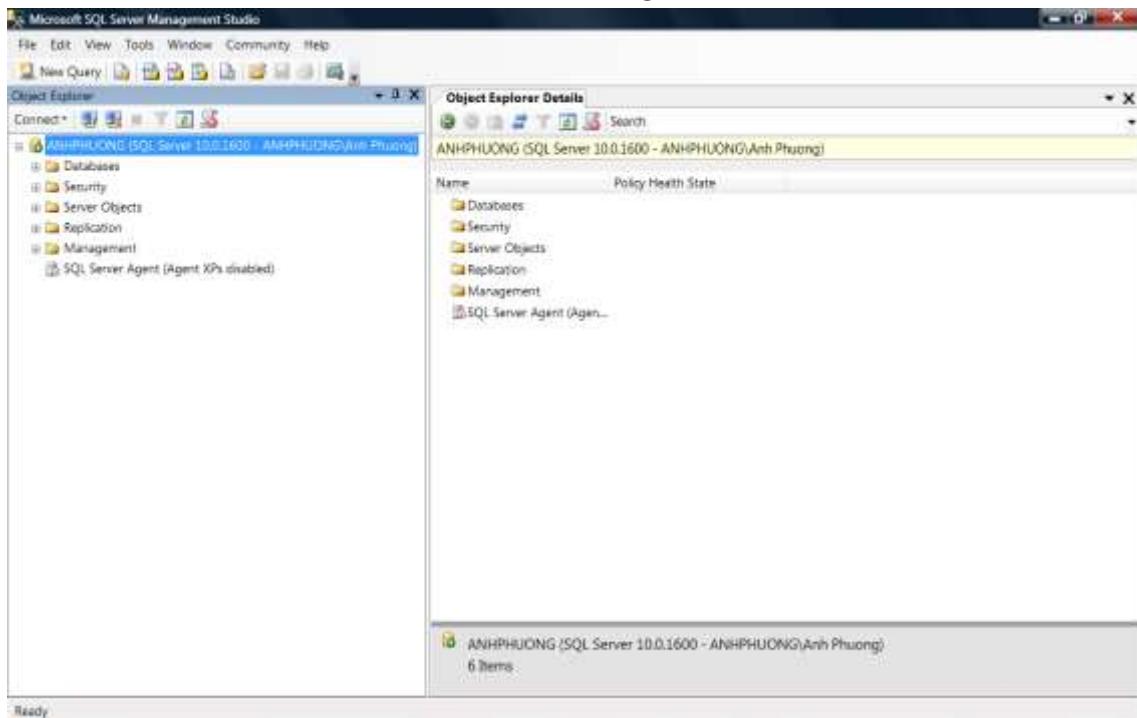


Hình 1.1 Kết nối vào SQL Server 2012

Khai báo các mục trên hộp thoại Connect :

- **Server Type:** chọn loại service của SQL Server mà người dùng muốn sử dụng
 - Chọn Database engine
- **Server Name:** chọn tên của Server mà người dùng muốn kết nối
 - Chọn “.”: đăng nhập vào một thể hiện mặc định của SQL Server trên cùng máy tính
 - Hoặc mở hộp Server name → chọn <Browse for more...> → chọn Database Engine → chọn tên server ...
- **Authentication:** chọn một trong hai kiểu xác thực bạn muốn để connect vào server
 - **Windows Authentication:** thông tin đăng nhập vào Windows được chuyển thành tài khoản đăng nhập SQL Server.
 - **SQL Server Authentication:** sử dụng tài khoản được quản lý bởi SQL Server → nhập User name và Password
(SQL Server có 1 tài khoản quản trị tối cao là ‘sa’)

Sau khi nhấn nút Connect xuất hiện màn hình giao diện SSMS :



3. Làm quen SQL Server Management Studio (SSMS)

3.1 Tìm hiểu sơ lược cửa sổ, thực đơn, thanh công cụ của SSMS

3.2 Sử dụng Object Explorer quan sát các đối tượng trong server

- Nếu Object Explorer chưa xuất hiện → vào menu View để chọn mở
- Các đối tượng trong server được phân nhóm, và hiển thị trên giao diện dạng cây → Lần lượt mở các nhánh của cây : Databases, ...

Thực hiện quan sát và cho biết:

Server hiện hành đang connect có tên server ? hiện đang kết nối bằng tài khoản nào ?

Danh sách các database đang tồn tại trong server nằm ở mục nào ?

System Databases bao gồm những database tên gì ?

Mở xem master database nằm trong mục System database : master database gồm những loại đối tượng nào ?

- Nhấn mouse phải vào tên server -> chọn properties -> đọc thông tin xuất hiện trên hộp thoại
- Nhấn mouse phải vào tên master database -> chọn properties -> đọc thông tin xuất hiện trên hộp thoại

3.3 Mở database master, khảo sát các đối tượng

- Vào đối tượng Table, tìm hiểu cấu trúc và dữ liệu của bảng (*lưu ý, chỉ được chọn xem không nên xoá hay sửa dữ liệu*): Sysdatabases, SysObjects, systypes, syslogins, sysusers, sysmessages, syspermissions...

- Vào đối tượng Stored Procedures, tìm hiểu nội dung của các thủ tục sau (*lưu ý, chỉ được chọn xem không nên xoá hay sửa*): sp_help, sp_helpdb, sp_helpconstraint, sp_rename, sp_renamedb, sp_table, sp_addlogin, sp_addmessage, sp_addrole ...
- Lần lượt vào đối tượng còn lại User, Role, ...

3.4 Tìm hiểu các mục trong menu Help : Lần lượt tìm hiểu các lệnh Create DataBase, Create Table, Alter Table, Select Statement, Select into, Update Statement, Insert Statement, DataType, Triggers...
(*Hướng dẫn: Gõ tên lệnh/từ khóa cần tìm và nhấn Enter*)

3.5 Khởi động màn hình Query Editor

Nhập dòng lệnh sau trên cửa sổ Query Editor:

`USE master`

`SELECT * from dbo.MSreplication_options`

Nhấn F5 để thực thi và quan sát kết quả hiển thị.

3.6 Tìm hiểu các mục trong menu Tools→Option

PHẦN 2: TẠO VÀ QUẢN LÝ CƠ SỞ DỮ LIỆU (DATABASE)

PHẦN LÝ THUYẾT

I. Giới thiệu Database

1. **Database:** Cơ sở dữ liệu trong SQL server là cơ sở dữ liệu quan hệ, bao gồm một tập các quan hệ, mỗi quan hệ là một bảng dữ liệu bao gồm các dòng (bộ dữ liệu hay record) và cột (field) .
2. Một CSDL được lưu trữ trong tối thiểu 2 tập tin: một file **.mdf** và một file **.ldf** . Một CSDL cũng có thể được lưu trong nhiều hơn 2 tập tin: một file **.mdf**, một hay nhiều file **.ndf** , một hay nhiều file **.ldf**
 - **File dữ liệu cơ bản (Primary data file) :** file **.mdf**, mỗi CSDL có duy nhất 1 file dữ liệu cơ bản (bắt buộc). Primary data file chứa thông tin về CSDL, thông tin về các tập tin khác của CSDL và chứa dữ liệu.
 - **Các file dữ liệu thứ cấp (Secondary data files) :** file **.ndf**, một CSDL có thể không có hoặc có một/nhiều file dữ liệu thứ cấp (tùy chọn), chứa dữ liệu.
 - **File nhật ký giao tác (Transaction log file) :** file **.ldf**, mỗi CSDL có tối thiểu 1 hoặc có thể có nhiều file nhật ký, dùng để lưu trữ thông tin cần thiết cho việc phục hồi các giao tác (transaction) mà người dùng đã thực hiện.

Như vậy , một cơ sở dữ liệu (Database) có thể nhìn ở 2 góc độ :

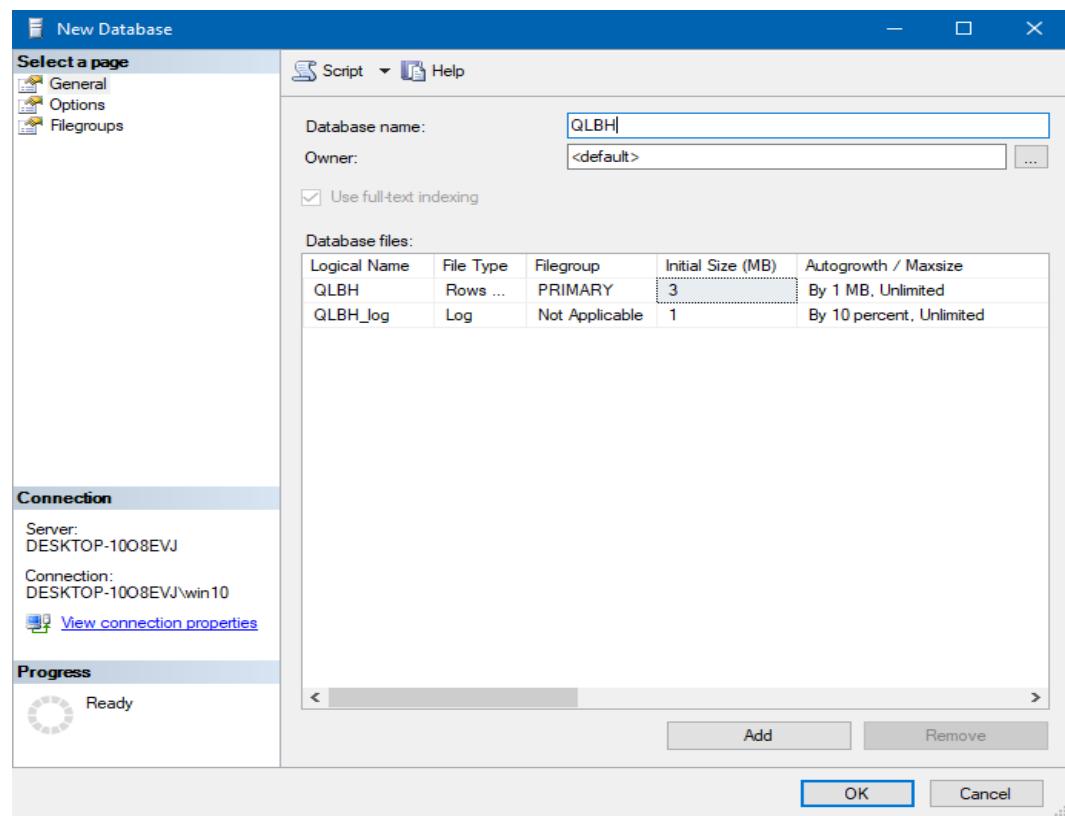
- **Về mặt vật lý:** một CSDL bao gồm hai hay nhiều tập tin nằm trên một hay nhiều đĩa. Các file lưu trữ CSDL chỉ thấy được bởi nhà quản trị và trong suốt đối với người dùng cuối.

- Khi tạo 1 CSDL, các file dữ liệu và file log được tạo và lưu trữ tại vị trí do người quản trị xác định.
 - Các file dữ liệu thường nằm trên những đĩa vật lý khác nhau để tăng tốc độ đọc/ghi dữ liệu.
 - o **Về mặt Logic:** một CSDL bao gồm các thành phần (đối tượng), hiển thị với người dùng như các Tables, Views, Procedures, ...
3. **Filegroup :** một khái niệm dùng gom nhóm các file dữ liệu (data file) . Một filegroup gồm 1 hay nhiều file dữ liệu. Một file dữ liệu luôn thuộc về một filegroup. Có 3 loại: Primary filegroup, user-define filegroup và default filegroup
- o **Primary Filegroup:** là filegroup mặc định, luôn chứa file dữ liệu cơ bản (.mdf) và có thể chứa các file dữ liệu thứ cấp khác (.ndf). Tất cả các bảng hệ thống phải nằm trong primary filegroup.
 - o **User-defined filegroup:** người quản trị có thể tạo thêm filegroup ngoài Primary filegroup bằng lệnh CREATE/ALTER DATABASE
 - o **Default filegroup:** là bất kỳ filegroup nào trong CSDL. Thường thì primary filegroup chính là default filegroup nhưng owner có quyền thay đổi. Tất cả bảng và index mặc định đều được tạo ra trong default filegroup.

II. Các lệnh định nghĩa dữ liệu (DDL - Data Definition Language)

1. Tạo cơ sở dữ liệu bằng cách sử dụng công cụ SQL Server Management Studio (SSMS)

- Click chuột phải trên mục Database → Chọn New Database
- Trong ô Database name: Nhập tên Database
- Trong Database files:
 - o Logical Name: nhập tên file luận lý
 - o Initial Size: nhập kích thước khởi tạo của tập tin
 - o Autogrowth/Maxsize: nhập kích thước tối đa và kích cỡ gia tăng của file
 - o Path: chọn vị trí lưu file
 - o File Name: nhập tên file vật lý
- Nhấn nút Add nếu muốn tạo thêm file thứ cấp, nhấn nút remove để xóa file
- Nhấn nút OK để tạo Database



2. Tạo cơ sở dữ liệu bằng lệnh T-SQL

Cú pháp:

```
CREATE DATABASE database_name
[ ON
[ <filespec> [ ,...n ] ]
[ ,<filegroup> [ ,...n ] ]
]
```

```
[ LOG ON { <filespec> [ ,...n ] } ]
```

Cú pháp <Filespec> :

```
(NAME = logical_name,
FILENAME = 'path\filename',
SIZE = size_in_MB,
MAXSIZE = size_in_MB | UNLIMITED,
FILEGROWTH = %_or_MB)
```

Ví dụ 1 : Tạo CSDL QLSV

```
CREATE DATABASE QLSV
ON PRIMARY
( NAME=QLSV_Data,
  FILENAME='T:\BTSQL\QLSV_Data.mdf',
  SIZE=10MB,
  MAXSIZE=15MB,
  FILEGROWTH=20%)
LOG ON
( NAME=QLSV_Log,
  FILENAME= 'T:\BTSQL\QLSV_Log.ldf',
  SIZE=3MB,
  MAXSIZE=5MB,
  FILEGROWTH=1MB)
```

Ví dụ 2: Tạo CSDL QLBH

```
CREATE DATABASE QLBH
ON PRIMARY
( NAME = QLBH_data1,
  FILENAME = 'T:\BTSQL\QLBH_data1.mdf',
  SIZE = 10, MAXSIZE = 50, FILEGROWTH = 15% ),
( NAME = QLBH_data2,
  FILENAME = 'T:\BTSQL\QLBH_data2.ndf',
  SIZE = 10, MAXSIZE = 50, FILEGROWTH = 15% ),
FILEGROUP QLBHGroup1
( NAME = QLBH_data3,
  FILENAME = 'T:\BTSQL\QLBH_data3.ndf',
  SIZE = 10, MAXSIZE = 50, FILEGROWTH = 5 ),
( NAME = QLBH_data4,
  FILENAME = 'T:\BTSQL\QLBH_data4.ndf',
  SIZE = 10, MAXSIZE = 50, FILEGROWTH = 5 )
LOG ON
( NAME = 'QLBH_log',
  FILENAME = 'T:\BTSQL\QLBH_log.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
```

Chú ý : không thể thiếu dấu phẩy (đánh dấu highlight)

3. Mở CSDL (chọn sử dụng CSDL)

Cú pháp: **use database_name**

Ví dụ: **use QLSV**

4. Kiểm tra không gian sử dụng của CSDL

Cú pháp: **sp_spaceused**

Ví dụ:

```
use QLSV
sp_spaceused
```

5. Xem thông tin của tất cả các CSDL trong server hiện hành

Cú pháp: `sp_helpdb`

Ví dụ: `sp_helpdb`

6. Xem thông tin một CSDL

Cú pháp: `sp_helpdb database_name`

Ví dụ: `sp_helpdb QLSV`

7. Thay đổi cấu trúc CSDL

Cú pháp:

`ALTER DATABASE database_name`

```
ADD FILE filespec [TO FILEGROUP filegroup_name]
ADD LOG FILE filespec
| REMOVE FILE logical_filename
| ADD FILEGROUP filegroup_name
| REMOVE FILEGROUP filegroup_name
| MODIFY FILE filespec
| MODIFY FILEGROUP filegroup_name
filegroup_property
|SET optionspec [WITH termination]
```

Ví dụ:

- a) Thêm filegroup **GroupOrder** vào CSDL QLBH

```
ALTER DATABASE QLBH
ADD FILEGROUP GroupOrder
```

- b) Chính sửa Size của tập tin

```
ALTER DATABASE QLBH
MODIFY FILE (NAME = 'QLBH_log', size =10MB)
```

- c) Bổ sung thêm một tập tin dữ liệu thứ cấp QLBH_data5

```
ALTER DATABASE QLBH
ADD File (Name =QLBH_data5,
Filename ='T:\BTSQ\QLBH_data5.ndf',
SIZE =10 MB,
Maxsize =20MB) TO FILEGROUP GroupOrder
```

- d) Xóa file thứ cấp QLBH_data5

```
ALTER DATABASE QLBH
REMOVE FILE QLBH_data5
```

- e) Xóa filegroup GroupOrder: chú ý filegroup muốn xóa phải trống

```
ALTER DATABASE QLBH
REMOVE FILEGROUP GroupOrder
```

- f) Thay đổi thuộc tính CSDL

Cú pháp:

```
ALTER DATABASE database_name
SET <option> [, status]
```

<Option> :

- AUTO_SHRINK
- CURSOR_CLOSE_ON_COMMIT
- RECOVERY FULL | BULK_LOGGED | SIMPLE
- SINGLE_USER | RESTRICTED_USER | MULTI_USER
- READ_ONLY | READ_WRITE

Ví dụ:

```
ALTER DATABASE QLBH
SET Read_Only
```

8. Đổi tên cơ sở dữ liệu

Cú pháp: `sp_renamedb [@dbname =] 'old_name',
[@newname =] 'new_name'`

Ví dụ: `Sp_Rename 'QLBH', 'Banhang'`

9. Xóa cơ sở dữ liệu

Khi 1 CSDL bị xóa thì tất cả các file vật lý của CSDL sẽ bị xóa

Cú pháp:

```
DROP DATABASE database_name
```

Ví dụ:

```
Drop database Banhang
```

10. Kiểu dữ liệu (Data Type)

Có 2 nhóm:

- **System-Supplied datatype:** Các kiểu dữ liệu cơ bản được hỗ trợ bởi SQL Server.
- **User-defined datatype:** Các kiểu dữ liệu mà người dùng tự định nghĩa dựa trên các kiểu dữ liệu cơ bản.

a) Tạo một User-Defined Data Type

Dùng thủ tục hệ thống `sp_addtype` để tạo một user-defined data type.

Cú pháp:

```
sp_addtype type, system_data_type [,NULL' | 'NOT NULL']
```

Ví dụ: Tạo kiểu dữ liệu tên là **isbn** với kiểu dữ liệu cơ bản là **smallint** và **không chấp nhận giá trị Null**

```
EXEC sp_addtype isbn, 'smallint', 'NOT NULL'
```

b) Xem các user-defined data types trong CSDL hiện hành

Dùng thủ tục `sp_help` hoặc truy vấn trong `information_schema.domains`

Ví dụ: `Use QLBH
Sp_help`

hoặc

```
SELECT domain_name, data_type, character_maximum_length
FROM information_schema.domains
ORDER BY domain_name
```

c) **Xoá một User-Defined Data Type:** dùng thủ tục hệ thống `sp_droptype` để xóa một user-defined data type từ bảng systypes. Một user-defined data type không thể xóa được nếu nó được tham chiếu bởi các bảng và những đối tượng khác.

Cú pháp: **`sp_droptype type`**

Ví dụ:

`EXEC sp_droptype isbn`

PHẦN THỰC HÀNH

1. Tạo CSDL QLSach bằng SSMS với giá trị các tham số như sau:

Tham số	Giá trị
Database name	QLSach
Tên logic của file data chính	QLSach_data
Tên và đường dẫn của file data chính	T:\QLTV\QLSach_Data.mdf
Kích cỡ khởi tạo của file data chính	20 MB
Kích cỡ tối đa của file data chính	40 MB
Gia số gia tăng của file data chính	1 MB
Tên logic của file transaction log	QLSach_Log
Tên và đường dẫn của file transaction log	T:\QLTV\QLSach_Log.ldf
Kích cỡ khởi tạo của file transaction log	6 MB
Kích cỡ tối đa của file transaction log	8 MB
Gia số gia tăng của file transaction log	1 MB

- a. Sau khi tạo thành công CSDL QLSach, xem thuộc tính (properties) của CSDL QLSach : Nhấp phải chuột tại tên CSDL, chọn properties. Đọc hiểu thông tin hiển thị về CSDL ?
- b. Tại cửa sổ properties của CSDL, khai báo thêm
 - Một Filegroup mới có tên là DuLieuSach
 - Một tập tin dữ liệu (data file) thứ hai nằm trong filegroup vừa tạo ở trên và có thông số như sau : Tên logic của data file là QlSach_Data2; Tên tập tin và đường dẫn vật lý của data file là T:\QLTV\QlSach_Data2.ndf ; các thông số về kích thước tùy ý.
 - Thiết lập thuộc tính ReadOnly, sau đó đóng cửa sổ properties. Quan sát màu sắc của CSDL. Bỏ thuộc tính ReadOnly.
 - Thay đổi Owner

2. Tạo cơ sở dữ liệu bằng lệnh T-SQL tại cửa sổ Query Analyzer (lưu ý: kiểm tra kết quả sau mỗi thay đổi)

- a. Dùng lệnh **CREATE DATABASE**, tạo **CSDL QLBH** với các tham số được liệt kê trong bảng dưới.

Tham số	Giá trị
Database name	QLBH
Tên logic của file data chính	QLBH_data1
Tên tập tin và đường dẫn của file data chính	T:\QLBH_data1.mdf
Kích cỡ khởi tạo của file data chính	10 MB
Kích cỡ tối đa của file data chính	40 MB
Gia số gia tăng file data chính	1 MB
Tên logic của file transaction log	QLBH_Log
Tên tập tin và đường dẫn của file transaction log	T:\QLBH.ldf
Kích cỡ khởi tạo của file transaction log	6 MB
Kích cỡ tối đa của file transaction log	8 MB
Gia số gia tăng file transaction log	1 MB

- b. Xem lại thuộc tính của CSDL QLBH bằng cách Click phải vào tên CSDL chọn Properties và bằng thủ tục hệ thống sp_helpdb, sp_spaceused, sp_helpfile.
- c. Thêm một filegroup có tên là DuLieuQLBH (*Hướng dẫn: dùng lệnh ALTER DATABASE ... ADD FILEGROUP ...*)
- d. Thêm một secondary data file có tên logic là QLBH_data2 trong filegroup vừa tạo : tên và đường dẫn file vật lý T:\QLBH_data2.ndf , các thông số khác tùy chọn (*HD: Dùng lệnh ALTER DATABASE ADD FILE TO FILEGROUP ...*)
- e. Sử dụng sp_helpfilegroup xem các filegroup đã có .
- f. Dùng lệnh Alter Database ... Set ... để cấu hình cho CSDL QLBH có thuộc tính là Read_Only. Dùng sp_helpDB để xem lại thuộc tính của CSDL. Hủy bỏ thuộc tính Read_Only.
- g. Dùng lệnh Alter DataBase ... MODIFY FILE ... để tăng SIZE của QLBH_data1 thành 50 MB. Tương tự tăng SIZE của tập tin QLBH_log thành 10 MB.

Để thay đổi SIZE của các tập tin bằng công cụ Design bạn làm như thế nào? Bạn hãy thực hiện thay đổi kích thước của tập tin QLBH_log với kích thước là 15MB. Nếu thay đổi kích cỡ nhỏ hơn ban đầu có được không? Nếu thay đổi kích cỡ MAXSIZE nhỏ hơn kích cỡ SIZE thì có được không? Giải thích.

3. Tạo CSDL QLSV, các thông số tùy chọn.

Dùng công cụ design tạo các bảng trong CSDL QLSV :

LOP (**MaLop** char(5) , **TenLop** NVarchar(20), **SiSoDuKien** Int,
NgayKhaiGiang DateTime)

SINHVIEN (**MaSV char(5)**, **TenHo NVarchar(40)**, NgaySinh
DateTime, **MaLop char(5)**)

MONHOC(**MaMH char(5)**, Tenmh Nvarchar(30), SoTC int)

KETQUA(**MaSV char(5)**, **MaMH char(5)**, Diem real)

Lưu ý: cột in đậm và gạch chân là khóa chính và Not Null; cột in đậm là Not Null

- Tạo Diagram giữa các bảng vừa tạo.
- Nhập dữ liệu tùy ý vào các bảng bằng SSMS, mỗi bảng khoảng 3 mẫu tin.

Lưu ý : thứ tự nhập dữ liệu giữa các bảng.

Giả sử bạn nhập dữ liệu cho bảng KETQUA trước, sau đó mới nhập dữ liệu cho các bảng còn lại thì bạn có nhập được không? Vì sao? Theo bạn nên nhập dữ liệu theo thứ tự nào?

- Dùng tác vụ General Script phát sinh mã lệnh cho CSDL QLSV và các đối tượng của CSDL, file lưu với tên qlsv.sql
- Vào Query Analyzer, mở và đọc các lệnh trong tập tin Script vừa tạo. Nếu các lệnh trong file này được thực thi thì kết quả sẽ là gì ?
Hãy xóa CSDL QLSV đang tồn tại, rồi thực hiện chạy các lệnh trong file script này để tạo lại CSDL QLSV và các table.

4. Kiểu dữ liệu (datatype)

- Tìm hiểu và trả lời các câu hỏi sau:

- Có mấy loại datatype trong SQL Server, hãy liệt kê ?
- Các system datatype được SQL Server lưu trữ trong Table nào và ở trong CSDL nào ?
- Các User-defined datatype được SQL Server lưu trữ trong Table nào, ở trong CSDL nào?

- Vào Query Analyzer, chọn QLBH là CSDL hiện hành, định nghĩa các datatype:

Kiểu dữ liệu	Mô tả dữ liệu
Mavung	10 ký tự
STT	Số nguyên, không vượt quá 30,000
SoDienThoai	13 ký tự , chấp nhận NULL
Shortstring	Số ký tự thay đổi đến 15 ký tự

HD: Dùng thủ tục **sp_addtype** để định nghĩa

Ví dụ: **exec sp_addtype SoDienThoai, 'char(13)', NULL**

- Các User-defined datatype vừa định nghĩa được lưu trữ ở đâu và phạm vi sử dụng của nó ở đâu (trong toàn bộ một instance hay chỉ ở trong CSDL hiện hành).
- Thực hiện liệt kê danh sách các User-Defined datatype vừa định nghĩa

```
SELECT domain_name, data_type, character_maximum_length  
FROM information_schema.domains  
ORDER BY domain_name
```

Hoặc SELECT * From Systype

- e. Tạo 1 bảng có tên là **ThongTinKH** trong CSDL QLBH và sử dụng User-defined data type vừa định nghĩa ở trên. Bảng gồm các cột : **MaKH** (khóa chính) có kiểu dữ liệu STT , **Vung** có kiểu dữ liệu Mavung , **Diachi** có kiểu dữ liệu Shortstring, **DienThoai** có kiểu dữ liệu SoDienThoai.
Bạn có tạo được không? Nếu được bạn nhập thử 2 record bằng giao diện design.
- f. Muốn User-Defined datatype được dùng trong tất cả các CSDL thì bạn định nghĩa nó ở đâu?
- g. Xóa kiểu dữ liệu SoDienThoai.

BÀI TẬP TUẦN 2 & 3

Số tiết: 6

Mục tiêu:

- ☞ Tạo bảng bằng lệnh T-SQL
- ☞ Tạo các ràng buộc (constraint) bằng lệnh T-SQL
- ☞ Phát sinh tập tin script

PHẦN LÝ THUYẾT

1) Bảng dữ liệu – Table

Bảng là một loại đối tượng của CSDL dùng để lưu trữ dữ liệu. Dữ liệu trong bảng được tổ chức thành các hàng (rows) và cột (columns). Mỗi hàng trong bảng biểu diễn một bản ghi (record). Mỗi bản ghi trong một bảng sẽ là duy nhất (phân biệt) với các bản ghi khác trong bảng. Mỗi cột biểu diễn một thuộc tính (attribute). Giá trị của thuộc tính thuộc một domain (kiểu dữ liệu) xác định. Tên cột trong 1 bảng không được trùng nhau nhưng tên cột có thể trùng nhau trong những bảng khác nhau của cùng 1 CSDL.

SQL Server cho phép:

Tối đa 2 triệu bảng trong 1 CSDL.

Tối đa 1024 cột trong 1 bảng

Tối đa 8060 bytes trong 1 hàng

2) Tạo Table

Cú pháp:

CREATE TABLE

```
[ database_name.[ owner ] .[owner.] table_name
({ <column_definition>
    |column_name AS computed_column_expression
    |<table_constraint> ::= [ CONSTRAINT
        constraint_name ] }
    [ { PRIMARY KEY | UNIQUE } [ ,...n ]
)
[ ON { filegroup | DEFAULT } ]
[ TEXTIMAGE_ON { filegroup | DEFAULT } ]
```

Ví dụ:

- a) Tạo bảng không có ràng buộc

```
CREATE TABLE Sanpham
( Masp CHAR(5),
  Tensp VARCHAR(15),
  Dvt VARCHAR(10),
  Dongia SMALLMONEY,
  SlTon INT )
```

- b) Tạo bảng chứa cột mà giá trị sẽ phát sinh tự động (Identity)

```
CREATE TABLE NhaCungCap
(MaNCC int Identity,
 TenNCC VarChar(25) )
```

- c) Tạo bảng có cột tính toán

```
CREATE TABLE cthoadon
( Sohd int,
  MaHang char(5),
  SoLuong int,
  DonGia money,
  ThanhTien AS SoLuong*DonGia )
```

- d) Khai báo Filegroup FGROUP1 chứa Table

```
CREATE TABLE KH
( MaKh int Identity(1000,1),
  TenKH Varchar(40) ) ON FGROUP1
```

3) Sửa cấu trúc của bảng**Cú pháp:**

```
ALTER TABLE <table_name>
{ALTER COLUMN <column_name> <new_data_type>}
| {ADD [<column_name> <data_type>]}
| {DROP COLUMN <column_name>}
```

Ví dụ:

- a) Thêm cột

```
ALTER TABLE SanPham ADD NgayNhap SmallDateTime
```

- b) Sửa kiểu dữ liệu cho cột

```
ALTER TABLE SanPham ALTER COLUMN NgayNhap DateTime
NOT NULL
```

- c) Xóa cột

```
ALTER TABLE Sanpham DROP COLUMN NgayNhap
```

4) Xóa bảng

Cú pháp: DROP TABLE <Table_Name>

Ví dụ: `DROP TABLE SanPham`

5) Xem thông tin Table

Cú pháp: sp_help <table_name>

Ví dụ: `Sp_help cthoadon`

Toàn vẹn dữ liệu :

- TVDL là đề cập đến trạng thái của tất cả các giá trị dữ liệu lưu trữ trong CSDL là đúng. Nếu dữ liệu được lưu trữ trong CSDL không đúng thì gọi là vi phạm TVDL.
- Để đảm bảo TVDL, SQL Server hỗ trợ thiết lập các loại ràng buộc toàn vẹn dữ liệu : Primary key, Foreign key, Not Null, Default, Identity, Check, ...
- Có 2 cách để cài đặt các ràng buộc
 - Tạo ràng buộc cùng lúc tạo table, bằng lệnh CREATE TABLE
 - Tạo ràng buộc bằng lệnh ALTER TABLE (sau khi tạo table)

6) Ràng buộc toàn vẹn dữ liệu NULL/NOT NULL, Default, Check

Cú pháp:

```
<column_constraint> ::= [ CONSTRAINT constraint_name ]
{ [ NULL | NOT NULL ]
| [ { PRIMARY KEY | UNIQUE } ]
| [ FOREIGN KEY ]
REFERENCES ref_table [ ( ref_column ) ]
[ ON DELETE { CASCADE | NO ACTION } ]
[ ON UPDATE { CASCADE | NO ACTION } ]
]
| CHECK ( logical_expression )
}
```

a) Ràng buộc NULL/NOT NULL

- NULL dùng để chỉ các giá trị chưa biết, hay sẽ được bổ sung sau. Nó khác với giá trị rỗng (empty) hay zero. Hai giá trị null không được xem là bằng nhau. Khi so sánh hai giá trị null, hay 1 giá trị null với 1 giá trị khác thì kết quả trả về sẽ là unknown.
- NOT NULL một ràng buộc trên cột : giá trị của cột phải là một giá trị xác định, không thể là “chưa biết” hay “bổ sung sau”.
- **Ví dụ: lệnh tạo table SanPham với các cột có ràng buộc NOT NULL và NULL**

```
USE QLBH
CREATE TABLE SanPham
(   Masp      smallint NOT NULL,
    Tensp     char(20) NOT NULL,
    Mota      char(30) NULL,
    Gia       smallmoney NOT NULL
)
```

- **Ràng buộc DEFAULT:** cho phép thiết lập giá trị “mặc định” tự động được gán cho 1 cột khi thêm 1 bản ghi mới vào bảng. DEFAULT có thể

áp dụng cho bất kỳ cột nào trong bảng ngoại trừ cột có kiểu timestamp hay có thuộc tính IDENTITY.

b.1. Tạo ràng buộc Default

Cách 1: Tạo ràng buộc khi tạo Table

Cú pháp: CREATE TABLE <TableName>

(<Column_Name> <DataType> DEFAULT (<expression>))

Ví dụ:

```
CREATE TABLE HoaDon
(MaHD int,
LoaiHD Char(1) DEFAULT 'X',
NgayLap DateTime NOT NULL)
```

Cách 2: tạo ràng buộc sau khi đã tạo Table

Cú pháp: ALTER TABLE tablename

ADD [CONSTRAINT constraintname]

DEFAULT expression FOR columnname

Ví dụ: table HoaDon đã được tạo, lệnh sau bổ sung ràng buộc Default cho cột NgayLap

```
ALTER TABLE HoaDon
ADD CONSTRAINT Ngay_DF DEFAULT Getdate() FOR NgayLap
```

Cách 3: sử dụng lệnh tạo default object

Cú pháp: CREATE DEFAULT default_name AS 'value'

Sau khi tạo một default object, nó cần được gắn kết vào 1 cột hay kiểu dữ liệu người dùng.

sp_bindefault default_name, object_name [, FUTUREONLY]

Xóa gắn kết default làm cho nó không còn áp dụng vào cột của bảng hay kiểu dữ liệu người dùng.

sp_unbindefault object_name [, FUTUREONLY]

Ví dụ:

```
CREATE DEFAULT sp_mota AS 'unknown'
GO
sp_bindefault sp_mota, 'sanpham.mota'
GO
sp_unbindefault 'sanpham.mota'
```

b.2. Xoá ràng buộc Default

Nếu tạo Default bằng cách 1 và 2, sử dụng lệnh

ALTER TABLE <TenTable>

DROP CONSTRAINT <constraint_name>

Ví dụ : `ALTER TABLE HoaDon`

`DROP CONSTRAINT Ngay_DF`

Nếu tạo Default bằng cách 3, sử dụng lệnh

`DROP DEFAULT { default } [,...n]`

Ví dụ:

```
DROP DEFAULT sp_mota
```

- c) **Ràng buộc Check:** cho phép kiểm soát dữ liệu nhập vào phải thỏa mãn điều kiện của *bíểu thức logic*

Cách tạo:

Cách 1: Tạo ràng buộc khi tạo Table

Cú pháp: CREATE TABLE <Table_Name>

```
(<Column_Name> <Data_Type>[,...]
CONSTRAINT ConstraintName] CHECK (NOT FOR
REPLICATION] <Logical expresion>),....)
```

Ví dụ:

```
CREATE TABLE Nhanvien (
MaNV char(4) CHECK (Manv LIKE '[0-9][0-9][0-9][0-9]'),
Hoten Varchar(40),
LCB int CHECK (LCB BETWEEN 0 AND 50000),
HSPC real,
Thanhpho varchar(10)
CONSTRAINT chkCity CHECK(Thanhpho IN ('Berkeley',
'Boston', 'Chicago', 'Dallas'))
)
```

Cách 2: Tạo ràng buộc sau khi đã tạo Table

Cú pháp: ALTER TABLE <Table_Name>

```
[WITH CHECK | WITH NOCHECK] ADD
[CONSTRAINT ConstraintName]
CHECK (NOT FOR REPLICATION] <Logical expresion>),....)
```

Ví dụ: thêm ràng buộc check trên cột HSPC

```
ALTER TABLE Nhanvien
ADD CONSTRAINT NV_HSPC CHECK (HSPC>=0.1 AND
HSPC<0.5)
```

Cách 3 : Có thể tạo ràng buộc check bằng đối tượng rule

Rule: Định nghĩa miền giá trị mà có thể kết buộc vào các cột của bảng hay các kiểu dữ liệu do người dùng định nghĩa. Một rule cần được tạo trước khi kết buộc vào đối tượng khác.

Tạo một Rule:

```
CREATE RULE rulename AS condition_expression
```

Kết buộc rule vào một cột:

```
sp_bindrule rulename, tablename.columnname
```

Kết buộc Rule vào user-defined datatype:

```
sp_binrule rulename, datatypename[, futureonly]
```

Ví dụ:

```

CREATE RULE ActiveDate AS
    @Date Between '01/01/70' AND Getdate()
Exec sp_bindrule ActiveDate, 'Orders.OrderDate'

```

Xoá ràng buộc CHECK (đã tạo bằng cách 1 và 2)

```

ALTER TABLE <TenTable>
DROP CONSTRAINT <CheckConstraintName>

```

Ví dụ:

```

ALTER TABLE NhanVien
DROP CONSTRAINT NV_HSPC

```

Lưu ý : sử dụng lệnh **Sp_helpconstraint** để xem thông tin về các ràng buộc trong table. Ví dụ :

```
EXEC Sp_helpconstraint Nhanvien
```

7) Các ràng buộc Primary Key, Foreign key, Unique

Cú pháp: < column_constraint > ::= [CONSTRAINT constraint_name]

```

{ [ NULL | NOT NULL ]
| [ { PRIMARY KEY | UNIQUE } ]
| [ FOREIGN KEY ]
REFERENCES ref_table [ ( ref_column ) ]
[ ON DELETE { CASCADE | NO ACTION } ]
[ ON UPDATE { CASCADE | NO ACTION } ]
]
| CHECK ( logical_expression )
}

```

- a) **Ràng buộc Primary Key (Primary Key Constraint):** Primary key của một bảng gồm một hay nhiều cột dùng để nhận diện record (bản ghi). Mỗi bảng chỉ có một Primary key. Giá trị của Primary key không Null và không trùng nhau giữa các record. Đặc tính này của Primary key sẽ được SQL Server kiểm tra thông qua **ràng buộc Primary key**. SQL Server cũng tự động tạo một chỉ mục khi tạo bảng có primary key.

Cú pháp:

Cách 1: Tạo ràng buộc khi tạo Table

```
CREATE TABLE TableName
```

```

(columnname datatype [,...],[CONSTRAINT constraint_name]
PRIMARY KEY [CLUSTERED|NONCLUSTERED]
{(column [ASC |DESC][,...,n])}
[WITH FILLFACTOR = fillfactor]
[ON {filegroup|DEFAULT}]

```

```

CREATE TABLE DEAN
(
    MADA smallint PRIMARY KEY NOT NULL,
    TENDA varchar(50) NOT NULL
)
--Hoặc
CREATE TABLE DEAN
(
    MADA smallint NOT NULL,
    TENDA varchar(50) NOT NULL,
    PRIMARY KEY (MADA)
)
--nếu khóa gồm nhiều cột
CREATE TABLE DEAN
(
    MADA smallint NOT NULL,
    TENDA varchar(50) NOT NULL,
    NGAY datetime NOT NULL
    PRIMARY KEY (MADA, NGAY)
)

```

Cách 2: thiết lập ràng buộc sau khi đã tạo Table

Cú pháp: ALTER TABLE TableName

```

        ADD [CONSTRAINT constraint_name]
        PRIMARY KEY {(column [ASC |DESC][,...,n])}
        [ON {filegroup|DEFAULT}]

```

Ví dụ 1:

```

CREATE TABLE DEAN2
(
    MADA smallint NOT NULL,
    TENDA varchar(50) NOT NULL
)
Go
ALTER TABLE DEAN2
    ADD CONSTRAINT MaDA_PK PRIMARY KEY (MADA)
Go
EXEC Sp_helpconstraint DEAN2

```

- b) **Ràng buộc Unique (Unique Constraint):** đảm bảo các giá trị trong một cột (hay một nhóm các cột) không trùng nhau. Chấp nhận một giá trị Null. Một bảng có thể có nhiều Unique constraint. Ràng buộc Unique thường áp dụng trên một / nhiều cột không phải là khóa chính nhưng có đặc tính giá trị là duy nhất (không trùng lặp).

Cách tạo:

Cách 1: Tạo ràng buộc khi tạo Table

Cú pháp: CREATE TABLE TableName

```

        (columnname datatype [,...],[CONSTRAINT constraint_name]
        UNIQUE [CLUSTERED|NONCLUSTERED]
        {(column [ASC |DESC][,...,n])})

```

[WITH FILLFACTOR = fillfactor]
[ON {filegroup|DEFAULT}]

Ví dụ:

```
CREATE TABLE jobs
(job_id smallint Primary key,
 job_name varchar(20) UNIQUE ,
 job_desc varchar(50))
```

Cách 2: Tạo ràng buộc sau khi tạo Table

Cú pháp: ALTER TABLE TableName

ADD [CONSTRAINT constraint_name]
UNIQUE {(column [ASC |DESC][,...,n])}
[ON {filegroup|DEFAULT}]

Ví dụ:

```
CREATE TABLE jobs2
(job_id smallint Primary key,
 job_name varchar(20) ,
 job_desc varchar(50))
```

Go

```
ALTER TABLE jobs2
ADD CONSTRAINT name_uni UNIQUE (job_name)
Go
```

```
EXEC Sp_helpconstraint jobs2
```

- c) **Ràng buộc Foreign key (Foreign Key Constraint):** Khoá ngoại chỉ có thể tham chiếu đến cột trong bảng chính có đặc tính

- Là khóa chính hay 1 phần của khóa chính (nếu khóa chính gồm nhiều cột)
- Là cột có ràng buộc unique
- Là cột có chỉ mục unique

Một bảng có thể có tối đa 253 khoá ngoại và có thể tham chiếu đến 253 bảng khác nhau.

Có thể tạo ràng buộc Foreign key khi tạo Table , hoặc sau khi tạo Table.

Cú pháp 1 :

```
CREATE TABLE TableName
(columnName datatype [, ...],
[CONSTRAINT constraintName]
FOREIGN KEY[(column[, ..n])] ]
REFERENCES ref_table [ ( ref_column [, ..n]) ] )
[ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT} ]
[ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT} ]
)
```

Cú pháp 2 :**ALTER TABLE TableName**

```
[WITH CHECK | WITH NOCHECK] ADD
[CONSTRAINT constraintName]
FOREIGN KEY [(column[,..n])] 
REFERENCES ref_table [ ( ref_column [,..n]) ] )
[ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT} ]
[ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT} ]
```

- **ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT}**

Chỉ định hành động xảy ra trên cột là khóa ngoại mỗi khi giá trị mà nó tham chiếu đến trong bảng chính bị **xóa**.

NO ACTION : SQL Server sẽ đưa ra thông báo lỗi và việc xoá hàng trên bảng chính sẽ bị từ chối. NO ACTION là thiết lập mặc định.

CASCADE : SQL Server sẽ xóa hàng tương ứng trên bảng chứa khóa ngoại

SET NULL : SQL Server sẽ gán NULL những giá trị khóa ngoại tham chiếu đến giá trị bị xóa trong bảng chính.

SET DEFAULT : SQL Server sẽ gán giá trị Default cho những giá trị khóa ngoại tham chiếu đến giá trị bị xóa trong bảng chính.

Tương tự với ON UPDATE...

Ví dụ 1

```
CREATE TABLE Congviec1
(Macv int primary key,
Tencv varchar(30))
-- tạo bảng nhanvien1 có ràng buộc khóa ngoại
CREATE TABLE NhanVien1
(MaNV int primary key,
Hoten varchar(30),
Macongviec int REFERENCES Congviec1(Macv)
```

Ví dụ 2

```
CREATE TABLE Congviec2
(Macv int primary key,
Tencv varchar(30))
--tạo bảng nhanvien2 có ràng buộc khóa ngoại
CREATE TABLE NhanVien2
(MaNV int primary key,
Hoten varchar(30),
Macongviec int REFERENCES Congviec2(Macv)
ON DELETE Cascade
ON UPDATE Cascade )
```

Ví dụ 3:

```

CREATE TABLE Congviec3
(Macv int primary key,
Tencv varchar(30))

-- 

CREATE TABLE NhanVien3
(MaNV int primary key,
Hoten varchar(30),
Macongviec int )
--thêm ràng buộc khóa ngoại trong bảng Nhanvien3
ALTER TABLE Nhanvien3 WITH CHECK
ADD CONSTRAINT Macv_FK Foreign key (Macongviec)
REFERENCES Congviec3(Macv)
    ON DELETE Set NULL
    ON UPDATE Cascade

```

- **WITH CHECK:** SQL Server sẽ kiểm tra dữ liệu tồn tại có vi phạm ràng buộc hay không. Nếu có vi phạm, lệnh tạo constraint sẽ không thành công.

d) Liệt kê tất cả các ràng buộc trong Table

Cú pháp: Sp_helpConstraint Table_Name

Ví dụ: Sp_helpConstraint NhanVien3

e) Xóa ràng buộc

Cú pháp:

ALTER TABLE <TenTable> DROP CONSTRAINT <ConstraintName>

Ví dụ: ALTER TABLE NhanVien3 DROP CONSTRAINT Macv_FK

PHẦN THỰC HÀNH**BÀI TẬP 1:**

Hệ thống quản lý bán hàng của một siêu thị được mô tả như sau:

- Siêu thị bán nhiều sản phẩm khác nhau. Các sản phẩm được phân loại theo từng nhóm sản phẩm, mỗi nhóm sản phẩm có một mã nhóm (MANHOM) duy nhất, mỗi mã nhóm hàng xác định tên nhóm hàng (TENNHOME), tất nhiên một nhóm hàng có thể có nhiều sản phẩm. Mỗi sản phẩm được có một mã số (MASP) duy nhất, mỗi mã sản phẩm xác định các thông tin về sản phẩm đó như : tên sản phẩm (TENSP), mô tả sản phẩm (MOTA), đơn vị tính (DonViTinh), đơn giá mua (GIAMUA), số lượng tồn (SLTON).
- Siêu thị nhập hàng từ nhiều nhà cung cấp khác nhau. Mỗi sản phẩm được cung cấp từ một nhà cung cấp. Hệ thống phải lưu trữ các thông tin về các nhà cung cấp. Mỗi nhà cung cấp có một mã số (MaNCC) duy nhất, mỗi

mã nhà cung cấp sẽ xác định tên nhà cung cấp (TenNCC), địa chỉ (Diachi), số điện thoại (Phone), số fax (Sofax) và địa chỉ mail (DCMail).

- Siêu thị bán hàng cho nhiều khách hàng. Mỗi khách hàng có một mã khách hàng (MAKH) duy nhất, mỗi Makh xác định các thông tin về khách hàng như : họ tên khách hàng (HOTEN), địa chỉ (DIACHI), số điện thoại (DIENTHOAI), địa chỉ mail (DCMail), điểm tích lũy (DiemTL). Siêu thị phân loại khách hàng thành 3 loại khách hàng: VIP, TV, VL. Khách hàng VIP là những khách hàng đã là thành viên trên 5 năm và có tổng số hóa đơn mua hàng trên 100, khách hàng TV (thành viên là các khách hàng đã làm thẻ thành viên nhưng không đủ điều kiện của khách hàng VIP). Khách hàng vãng lai (VL) là khách hàng chưa có thẻ thành viên. Đối với khách hàng vãng lai thì MaKH sẽ được hệ thống tự cấp phát MaKH cho mỗi lần mua hàng do đó hệ thống không cần lưu các thông tin còn lại của khách hàng vãng lai.
 - Mỗi lần mua hàng, khách hàng có một hóa đơn. Mỗi hóa đơn có một số hóa đơn (SOHD) duy nhất, mỗi số hóa đơn xác định một khách hàng (MAKH), ngày lập hóa đơn (NGAYLAPHD), ngày giao hàng (NGAYGIAO) và nơi chuyển hàng (NoiChuyen). Ưng với mỗi hóa đơn mua hàng, siêu thị qui định: Nếu khách hàng VIP sẽ được tặng 20% tổng tiền vào điểm tích lũy của khách hàng, nếu là thành viên là 10% tổng tiền, vãng lai thì không được tặng điểm tích lũy. Dựa vào điểm tích lũy siêu thị sẽ tặng phiếu quà tặng vào cuối năm cho các khách hàng. Mỗi hóa đơn có thể mua nhiều sản phẩm, với số lượng cụ thể (SOLUONG) và đơn giá bán (DONGIA) của từng sản phẩm .
- Dựa vào mô tả, xây dựng mô hình thực thể kết hợp ERD, sau đó xây dựng lược đồ cơ sở dữ liệu và xác định các ràng buộc khóa chính và khóa ngoại.
 - Xác định các qui tắc nghiệp vụ của hệ thống trên.
 - Thực hiện tạo các table trong CSDL QLBH theo thiết kế sau

Nhóm Sản Phẩm			Ràng buộc cần cài đặt
MaNhóm	Int	Not null	PK constraint
TenNhóm	Nvarchar(15)		
Sản Phẩm			
MaSP	int	Not null	PK constraint
TenSP	nvarchar(40)	Not null	Not NULL
MaNCC	Int		FK constraint
MoTa	nvarchar(50)		
MaNhóm	Int		FK constraint
Đơn Vị Tính	nvarchar(20)		

GiaGoc	Money	>0	Check constraint
SLTON	Int	>=0	Check constraint
HoaDon			
MaHD	Int	Not null	PK constraint
NgayLapHD	DateTime	>=Ngày hiện hành. Giá trị mặc định là ngày hiện hành	Check constraint và Default constraint
NgayGiao	DateTime		
Noichuyen	NVarchar(60)	Not Null	Not NULL
MaKH	Char(5)		FK constraint
CT_HoaDon			
MaHD	Int	Not null	PK = {MaHD, MaSP}
MaSP	int	Not null	FK1 = {MaHD} FK2 = {MaSP}
Soluong	SmallInt	>0	Check constraint
Dongia	Money		
ChietKhau	Money	>=0	Check constraint
NhaCungCap			
MaNCC	Int	Not null	PK constraint
TenNCC	Nvarchar(40)	Not Null	Not NULL
Diachi	Nvarchar(60)		
Phone	NVarchar(24)		
SoFax	NVarchar(24)		
DCMail	NVarchar(50)		
KhachHang			
MaKH	Char(5)	Not null	PK constraint
TenKH	Nvarchar(40)	Not null	Not NULL
LoaiKH	Nvarchar(3)	Là ‘VIP’ hoặc ‘TV’ hoặc ‘VL’	Check constraint
DiaChi	Nvarchar(60)		
Phone	NVarchar(24)		
DCMail	NVarchar(50)		
DiemTL	Int	>=0	Check constraint

Hướng dẫn : Thực hiện tạo các table bằng sử dụng kết hợp 2 phương pháp

- Sử dụng lệnh CREATE TABLE ... tạo table và các ràng buộc
- Sử dụng lệnh CREATE TABLE ... để tạo cấu trúc table và lệnh ALTER TABLE ... để thiết lập các ràng buộc.

(xem Hướng dẫn trang 15->24)

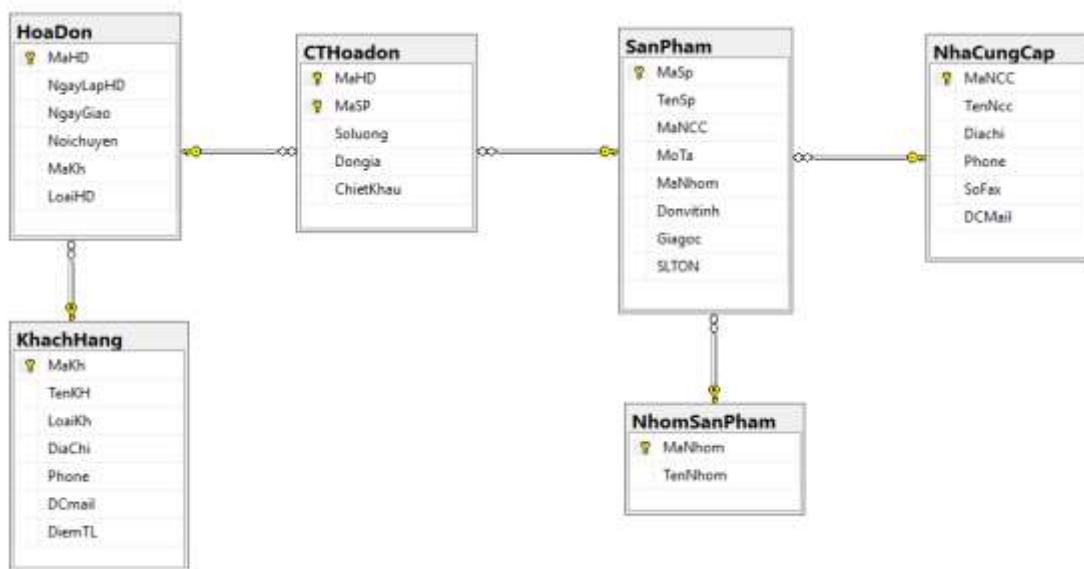
4. Viết lệnh thực hiện :

- a. Thêm cột LoaiHD vào bảng HoaDon ?

LoaiHD : kiểu dữ liệu char(1), có giá trị : ‘N’ hoặc ‘X’ hoặc ‘C’ hoặc ‘T’ ; giá trị mặc định là ‘N’ (với ngữ nghĩa ‘N’ : Nhập, ‘X’: Xuất , ‘C’ : Chuyển từ cửa hàng này sang cửa hàng khác, ‘T’ : Trả)

- b. Tạo thêm ràng buộc trên bảng HoaDon : NgayGiao>=NgayLapHD ?

5. Tạo diagram cho CSDL QLBH ? Khóa chính và khóa ngoại giữa các table được biểu diễn trên Diagram , hãy quan sát và nhận diện ?



6. Thực hiện phát sinh tập tin script cho toàn bộ CSDL QLBH. Đọc hiểu các lệnh trong file script. Lưu lại file script.

BÀI TẬP 2 (làm thêm tại nhà và nộp lại cho GV)

1. Dùng T-SQL tạo CSDL Movies với các tham số sau:

- Tập tin Data file có: Name: Movies_data; Filename: C:\Movies\Movies_data.mdf; Size: 25 MB; Maxsize: 40 MB; FileGrowth: 1 MB.
- Tập tin Log file có: Name: Movies_log; Filename: C:\Movies\Movies_log.ldf; Size: 6 MB; Maxsize: 8 MB; FileGrowth: 1 MB.

2. Thực hiện và kiểm tra kết quả các yêu cầu sau :

- Thêm một Data file thứ 2 có: Name: Movies_data2; Filename: C:\Movies\Movies_data2.ndf; Size: 10 MB; thông số khác không cần chỉ định.
- Lần lượt cấu hình CSDL Movies với chế độ single_user, restricted user, multi user. Sau đó cài đặt lại là multi_user. Nhớ dùng lệnh để kiểm tra lại.
- Tăng kích cỡ của data file thứ 2 từ 10 MB lên 15 MB. Kiểm tra lại.

- Cấu hình CSDL về chế độ tự động SHRINK
- Phát sinh tập tin Script từ CSDL Movies, tạo script cho tất cả các đối tượng. Lưu vào đĩa với tên file Movies.sql
- Xoá CSDL Movies

3. Mở tập tin Movies.sql, thực hiện:

- Bổ sung thêm câu lệnh tạo một filegroup tên là DataGroup
- Hiệu chỉnh maxsize của tập tin transaction log thành 10 MB
- Size của tập tin data file thứ 2 thành 10 MB.
- Thêm data file thứ 3 thuộc filegroup DataGroup (thông số tùy ý)
- Lưu tập tin script
- Thực thi toàn bộ tập tin script.
- Dùng sp_helpDB để kiểm tra sự tồn tại của Movies và các thông số của nó.

4. Các bảng trong CSDL Movies bao gồm

Tên bảng	Mô tả bảng
Movie	Danh sách các phim có trong cửa hàng
Customer	Thông tin khách hàng
Category	Danh sách các loại phim
Rental	Thông tin thuê phim
Rental_detail	Chi tiết thuê phim

Bạn hãy suy nghĩ xem mỗi bảng trên cần lưu những thông tin cụ thể nào (tức là các cột nào), kiểu dữ liệu ra sao? Khóa chính của từng bảng, mối quan hệ giữa các bảng, có những ràng buộc toàn vẹn nào?

5. Thực hiện định nghĩa các user-defined datatype sau trong CSDL Movies.
Kiểm tra sau khi tạo.

Kiểu dữ liệu	Mô tả
Movie_num	Int, không chấp nhận Null
Category_num	Int, không chấp nhận Null
Cust_num	Int, không chấp nhận Null
Invoice_num	Int, không chấp nhận Null

6. Thực hiện tạo các bảng vào CSDL Movies, kiểm tra kết quả bằng sp_help

Customer

Tên cột	Kiểu dữ liệu	cho phép Null
Cust_num	cust_num IDENTITY(300,1)	No
Lname	varchar(20)	No
Fname	varchar(20)	No
Address1	varchar(30)	Yes
Address2	varchar(20)	Yes

City	varchar(20)	Yes
State	Char(2)	Yes
Zip	Char(10)	Yes
Phone	Varchar(10)	No
Join_date	Smalldatetime	No

Category

Tên cột	Kiểu dữ liệu	cho phép Null
Category_num	category_num IDENTITY(1,1)	No
Description	Varchar(20)	No

Movie

Tên cột	Kiểu dữ liệu	cho phép Null
Movie_num	Movie_num	No
Title	Cust_num	No
Category_Num	category_num	No
Date_purch	Smalldatetime	Yes
Rental_price	Int	Yes
Rating	Char(5)	Yes

Rental

Tên cột	Kiểu dữ liệu	cho phép Null
Invoice_num	Invoice_num	No
Cust_num	Cust_num	No
Rental_date	Smalldatetime	No
Due_date	Smalldatetime	No

Rental_Detail

Tên cột	Kiểu dữ liệu	cho phép Null
Invoice_num	Invoice_num	No
Line_num	Int	No
Movie_num	Movie_num	No
Rental_price	Smallmoney	No

7. Thực hiện phát sinh tập tin script cho CSDL Movies với các lựa chọn sau, lưu tên Movies.sql:
 - All Tables, All user-defined data types
 - Generate the CREATE <object> command for each object
 - Generate the DROP <object> command for each object
8. Thực hiện tạo Diagram cho CSDL Movies. Lưu diagram với tên là Movies.

9. Thực hiện định nghĩa các khoá chính (Primary Key Constraint) cho các bảng, kiểm tra kết quả bằng lệnh sp_helpconstraint

Tên bảng	Khóa chính	Tên của Primary Constraint
Movie	Movie_num	PK_movie
Customer	Cust_num	PK_customer
Category	Category_num	PK_category
Rental	Invoice_num	PK_rental

10. Thực hiện định nghĩa các khoá ngoại (Foreign Key Constraint) cho các bảng, kiểm tra kết quả bằng lệnh sp_helpconstraint

Tên bảng	Cột khóa ngoại	Tên bảng được tham chiếu	Cột được tham chiếu	Tên của Foreign Constraint	Cascade
Movie	Category_num	Category	Category_num	FK_movie	
Rental	Cust_num	Customer	Cust_num	FK_rental	
Rental_detail	Invoice_num	Rental	Invoice_num	FK_detail_invoice	delete
Rental_detail	Movie_num	Movie	Movie_num	PK_detail_movie	

11. Mở lại Diagram có tên Movie, xem khóa chính, mối quan hệ giữa các bảng.

12. Thực hiện định nghĩa các giá trị mặc định (Default Constraint) cho các cột ở các bảng như mô tả sau, kiểm tra kết quả bằng lệnh sp_helpconstraint

Tên bảng	Cột có giá trị Default	Giá trị Default	Tên của Default Constraint
Movie	Date_purch	Ngày hiện hành	DK_movie_date_purch
Customer	join_date	Ngày hiện hành	DK_customer_join_date
Rental	Rental_date	Ngày hiện hành	DK_rental_rental_date
Rental	Due_date	Ngày hiện hành + 2	DK_rental_due_date

13. Thực hiện định nghĩa các ràng buộc miền giá trị (Check Constraint) cho các cột như mô tả sau, kiểm tra kết quả bằng lệnh sp_helpconstraint

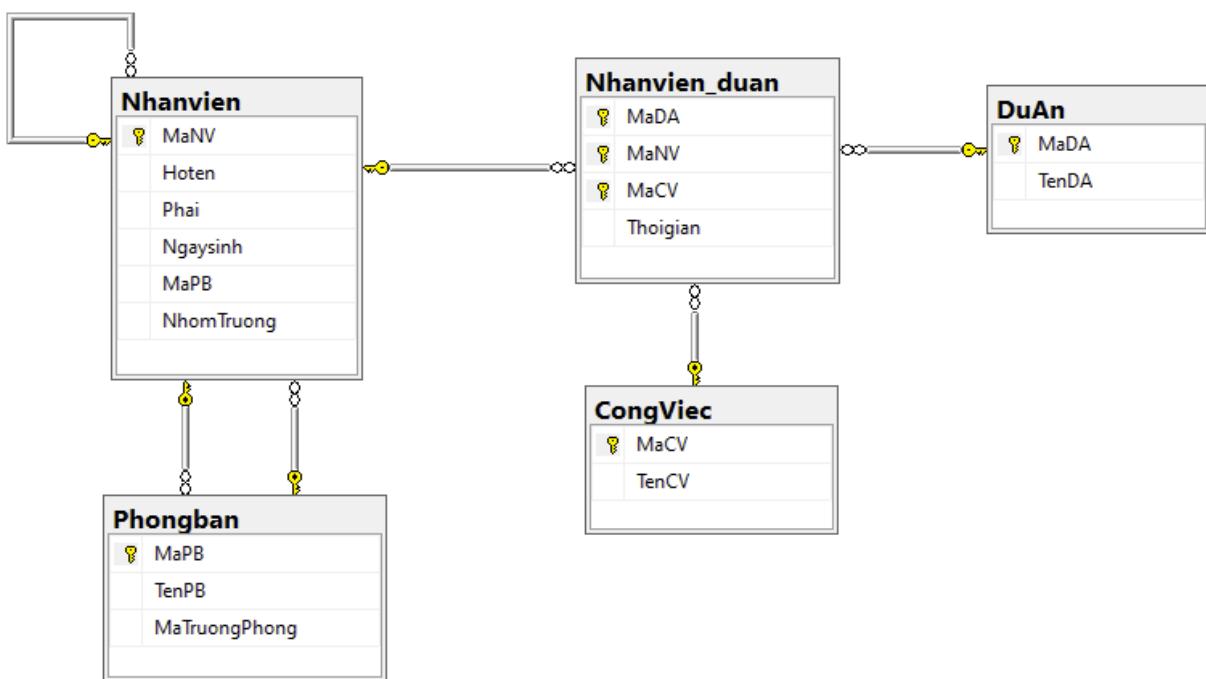
Tên bảng	Cột có ràng buộc check	Miền giá trị	Tên của Check Constraint
Movie	Rating	'G', 'PG', 'R', 'NC17', 'NR'	CK_movie
Rental	Due_date	>= Rental_date	CK_Due_date

14. Thực hiện phát sinh tập tin script cho các đối tượng trong CSDL Movie.

Tên của tập tin là ConstraintsMovies.sql. Với lựa chọn Script Primary Keys, Foreign Keys, Default, and Check Constraints.

BÀI TẬP 3 (tùy chọn)

CSDL QlDuAn gồm các quan hệ và relationships như sau :



Các relationship bao gồm :

- Một phòng ban có nhiều nhân viên, một nhân viên chỉ thuộc một phòng ban,
- Mỗi phòng ban có một trưởng phòng.
- Một nhân viên có thể là nhóm trưởng của một nhóm nhân viên
- Nhân viên có thể tham gia vào nhiều dự án. Ở mỗi dự án, một nhân viên tham gia với một công việc xác định

Yêu cầu :

- Cài đặt database QlDuAn theo mô tả trên (Kiểu dữ liệu tự chọn)
- Sau khi cài đặt database, nhập dữ liệu theo mô tả sau :
 - Phòng ‘Triển khai & bảo trì’ gồm 10 nhân viên. Trưởng phòng là nhân viên tên ‘Nguyễn Năm’ . Các nhân viên trong phòng được chia 3 nhóm. Mỗi nhóm có 1 nhóm trưởng (gạch chân) :
 - An, Minh , Tuấn
 - Lan, Hùng, Vân
 - Mai, Hà, Việt
 - Nhân viên An đã tham gia vào 2 dự án :
 - An tham gia dự án ‘Alpha B1 SAP’ với vị trí công việc ‘triển khai vòng 1’
 - An tham gia dự án ‘Delta B1 SAP’ với vị trí công việc ‘triển khai tổng thể’
 - ...

BÀI TẬP TUẦN 4, 5, 6, 7, 8**Số tiết: 15****Mục tiêu:**

- ☞ Nắm cú pháp căn bản và thực hiện được thao tác thêm, xóa, sửa dữ liệu bằng lệnh INSERT, UPDATE, DELETE.
- ☞ Tuân thủ các ràng buộc đã thiết lập khi thêm, xóa, sửa dữ liệu
- ☞ Nắm cú pháp và thực hiện truy vấn dữ liệu bằng lệnh SELECT

TUẦN 4 (3 tiết)**PHẦN I: Lệnh INSERT – UPDATE – DELETE căn bản****PHẦN LÝ THUYẾT****1) Cập nhật dữ liệu****a) Thêm dữ liệu vào table**

Cú pháp: `INSERT [INTO] <table_name>`
`VALUES (<values>) [, ...]`

Ví dụ:

```
Insert into NhomSanPham values(1, N'Điện tử')
Insert into SanPham(MaSP,TenSP,Manhom) values(1,
N'Loa bluetooth' , 1 )
```

b) Sửa dữ liệu trong table

Cú pháp 1: `UPDATE <table_name>`
`SET <column_name> = value [, ...]`
`[WHERE <condition>]`

Ví dụ:

```
Update SanPham
set TenSP = N'Loa Sony AS001' where MaSP = 1
```

c) Xóa dữ liệu trong table

Cú pháp 1: `DELETE FROM <table_name>`
`[WHERE <condition>]`

Ví dụ:

```
Delete from SanPham where MaSP = 1
```

Cách khác : Dùng lệnh TRUNCATE TABLE để xóa tất cả các dòng của table

`TRUNCATE TABLE <table_name>`

Nhanh hơn lệnh DELETE. Lệnh Truncate không kích hoạt Trigger.

Ví dụ:

TRUNCATE TABLE NewProducts

Ghi chú : cú pháp 2 của lệnh Update và Delete xem Bài tập tuần 9.

2) Truy vấn dữ liệu

Cú pháp:

```
SELECT [ALL | DISTINCT] [TOP n [WITH TIES]] select_list
[ INTO new_table ]
FROM table_source
[ WHERE search_condition ]
[ GROUP BY group_by_expression ]
[ HAVING search_condition ]
[ ORDER BY order_expression [ ASC | DESC ] ]
```

- *SELECT: chỉ định các cột muốn truy vấn*
- *FROM: chỉ định table nguồn*
- *WHERE: Biểu thức điều kiện lọc bộ dữ liệu*
- *GROUP BY: tạo nhóm*
- *HAVING: Biểu thức điều kiện lọc bộ dữ liệu sau khi nhóm dữ liệu*
- *ORDER BY: Sắp xếp bộ dữ liệu*

a) Truy vấn đơn giản

Hiển thị tất cả các cột có trong một Table :

Cú pháp: SELECT * FROM <tablename>

Ví dụ: SELECT * FROM Customers

Hiển thị một vài cột trong một Table :

```
SELECT OrderID, OrderDate, CustomerID FROM Orders
```

Truy vấn loại bỏ các dòng bị trùng:

Cú pháp: SELECT Distinct ... FROM ...

Ví dụ:

```
SELECT DISTINCT Order_Date as 'Date of Order'
FROM Orders
```

Truy vấn chọn một số bộ dữ liệu thỏa biểu thức điều kiện trong

WHERE :

Ví dụ 1:

```
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE UnitPrice < 275
```

Ví dụ 2:

```
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE ProductName like 'A%' AND UnitPrice < 100
```

b) Truy vấn trên nhiều bảng

Ví dụ 1:

```
SELECT c.CustomerID, CompanyName, OrderID, OrderDate
```

```
FROM Customers C INNER JOIN Orders O
    ON C.CustomerID = O.CustomerID
```

Ví dụ 2: Truy vấn soft join

```
SELECT e.Firstname+' '+e.Lastname AS 'Employee',
       m.Firstname+' '+m.Lastname AS 'Manager'
  FROM Employees e JOIN Employees m
    ON m.EmployeeID = e.ReportsTo
```

c) **Lệnh Union:** Union nối kết quả từ nhiều câu lệnh select

Ví dụ:

```
SELECT Firstname+' '+Lastname AS name, Homephone
  FROM Employees
UNION
  SELECT Companyname, Phone
  FROM Customers
```

d) **Truy vấn kết nhóm: hiển thị thông tin theo nhóm**

- Sử dụng các hàm tổng hợp - Aggregate Functions: tổng hợp thông tin từ nhiều bộ. Các hàm tổng hợp: COUNT(), SUM(), MAX(), MIN(), AVG()
- Sử dụng mệnh đề GROUP BY : chỉ định các thuộc tính tạo nhóm. Những thuộc tính này xuất hiện trong mệnh đề select. Hàm thống kê nếu có sẽ được áp dụng cho các bộ trong cùng một nhóm.
- Sử dụng WHERE để lọc bộ dữ liệu trước khi tạo nhóm, HAVING để lọc bộ dữ liệu sau khi tạo nhóm

Ví dụ 1:

```
SELECT count(*) as TotalEmps
  FROM Employees
```

Ví dụ 2:

```
SELECT Title, count(*) as TotalbyTitle
  FROM Employees
 GROUP BY Title
 Go
SELECT City, Country, count(*) as TotalbyCity
  FROM Employees
 GROUP BY City, Country
 Go
SELECT orderid, SUM(quantity*unitprice) AS 'Total'
  FROM [Order Details]
 GROUP BY orderid
```

Ví dụ 3 : sử dụng điều kiện lọc sau khi nhóm dữ liệu

```
SELECT orderid, SUM(quantity*unitprice) AS 'Total'
  FROM [Order Details]
 GROUP BY orderid
 HAVING SUM(quantity*unitprice)>10000
```

Ví dụ 4 : sử dụng điều kiện lọc trước và sau khi nhóm dữ liệu

```
SELECT orderid, SUM(quantity*unitprice) AS 'Total'
```

```

FROM [Order Details]
WHERE orderid >= 11000
GROUP BY orderid
HAVING SUM(quantity*unitprice)>10000

```

e) Lệnh SELECT... INTO...

Tạo một bảng mới dựa vào kết quả của câu lệnh select.

Ví dụ:

```

SELECT C.CustomerID AS NameId, OrderID, OrderDate
INTO Customer_Order
FROM Customers C INNER JOIN Orders O
ON C.CustomerID = O.CustomerID
WHERE month(OrderDate) = 7

```

f) Truy vấn lồng nhau - Nested Queries

Nested query là một query chứa một query khác_ goi là *subquery* (*truy vấn con*). Subquery thường xuất hiện trong mệnh đề WHERE hay HAVING của query ngoài. Subquery cũng có thể xuất hiện trong mệnh đề FROM của query ngoài.

Subquery có thể trả về một giá trị đơn hoặc một tập giá trị. Nếu subquery trả về **một giá trị đơn**, các toán tử so sánh dùng với subquery (trong WHERE hay HAVING) thường là : = , > , >= , < , <= , <> , ...

Nếu subquery trả về **một tập giá trị**, các toán tử so sánh thường dùng là: **IN**, **NOT IN**, **operator ALL**, **operator ANY** (với operator là = , <> , > , >= , < , <=).

Các toán tử **EXISTS**, **NOT EXISTS** sử dụng trong WHERE hay HAVING thường đi sau là một subquery

- Toán tử so sánh = , > , >= , < , <= , <> .

Ví dụ 1 :

```

SELECT *
FROM Products
WHERE unitprice > (select avg(unitprice)
from products )

```

- **IN**: so sánh một giá trị v với một tập giá trị V, kết quả là TRUE nếu v tồn tại trong V.

Ví dụ 2:

```

SELECT Orderid, orderdate
FROM Orders
WHERE CustomerID IN (Select CustomerID from Customers
where city = 'London')

```

- **NOT IN**: so sánh một giá trị v với một tập giá trị V, kết quả là TRUE nếu v không tồn tại trong V

Ví dụ 3 :

```
SELECT Orderid, orderdate
FROM Orders
WHERE CustomerID NOT IN (Select CustomerID from
Customers where city = 'London')
```

- **operator ANY** (với operator là =, <>, >, >=, <, <=) : kết quả là TRUE nếu các giá trị trong tập V thỏa mãn phép toán **operator với ít nhất là một** giá trị trong tập V.

Ví dụ 4 :

```
SELECT Orderid, orderdate
FROM Orders
WHERE CustomerID =ANY (Select CustomerID from
Customers where city = 'London')
```

- **operator ALL** (với operator là =, <>, >, >=, <, <=) : kết quả là TRUE nếu các giá trị trong tập V thỏa mãn phép toán **operator với tất cả** giá trị trong tập V

Ví dụ 5 :

```
SELECT Productid, Productname, Unitprice
FROM Products
WHERE Unitprice >ALL (Select Unitprice from Products
where ProductName like 'A%')
```

- **EXISTS**: trả về giá trị TRUE nếu kết quả của subquery chứa ít nhất một bộ giá trị.

Ví dụ 6 :

```
SELECT *
FROM Products p
WHERE EXISTS ( Select * from [Order Details] od
where p.Productid=od.Productid)
```

- **NOT EXISTS**: trả về giá trị TRUE nếu kết quả của subquery không chứa bộ giá trị nào.

Ví dụ 7 :

```
SELECT *
FROM Products p
WHERE NOT EXISTS ( Select * from [Order Details] od
where p.Productid=od.Productid)
```

Pivot SQL Server

Trong SQL Server (Transact-SQL), mệnh đề PIVOT cho phép phân tích bảng chéo (cross tabulation) chuyển dữ liệu từ bảng này sang bảng khác, tức là lấy kết quả tổng hợp rồi chuyển từ dòng thành cột.

The diagram illustrates the transformation of a source table into a result table using the PIVOT operator. On the left, a source table named 'nhanvien' is shown with columns: employee_number, last_name, first_name, salary, and dept_id. The rows contain data for employees 12009, 34974, 34987, 45001, and 75623. An orange arrow points from the source table to a result table on the right. The result table has three columns: TotalSalaryByDept, 30, and 45. Below the result table, the URL 'TechOnTheNet.com' is mentioned.

employee_number	last_name	first_name	salary	dept_id
12009	Sutherland	Barbara	54000	45
34974	Yates	Fred	80000	45
34987	Erickson	Neil	42000	45
45001	Parker	Sally	57500	30
75623	Gates	Steve	65000	30

TotalSalaryByDept	30	45
TotalSalary	122500	176000

Ví dụ tính tổng rồi chuyển hàng thành cột trong bảng dữ liệu

Cú pháp mệnh đề PIVOT

```

SELECT cot_dautien AS <bidanh_cot_dautien>,
[giatri_chuyen1], [giatri_chuyen2], ... [giatri_chuyen_n]
FROM
(<bang_nguon>) AS <bidanh_bang_nguon>
PIVOT
(
ham_tong (<cot_tong>)
FOR <cot_chuyen>
IN ([giatri_chuyen1], [giatri_chuyen2], ... [giatri_chuyen_n])
) AS <bidanh_bang_chuyen>;
    
```

Tên biến hoặc giá trị biến

cot_dautien: Cột hoặc biểu thức sẽ thành cột đầu tiên trong bảng mới sau khi chuyển.

bidanh_cot_dautien: Tên của cột đầu tiên trong bảng mới sau khi chuyển.

giatri_chuyen1, giatri_chuyen2, ... giatri_chuyen_n: Danh sách các giá trị cần chuyển.

bang_nguon: Lệnh [SELECT](#) đưa dữ liệu nguồn (dữ liệu ban đầu) vào bảng mới.

bidanh_bang_nguon: Bí danh của bang_nguon

ham_tong: Hàm tính tổng như [SUM trong SQL Server](#), COUNT, MIN, MAX hay AVG.

cot_tong: Cột hoặc biểu thức được dùng với ham_tong.

cot_chuyen: Cột chứa giá trị cần chuyển.

bidanh_bang_chuyen: Bí danh của bảng sau khi chuyển.

Mệnh đề PIVOT có thể dùng trong các phiên bản sau của SQL Server: SQL Server 2014, SQL Server 2012, SQL Server 2008 R2, SQL Server 2008, SQL Server 2005.

Ví dụ với mệnh đề PIVOT

Ta có bảng nhanvien với các dữ liệu như dưới đây.

so_nhanvien	ho	ten	luong	id_phong
12009	Nguyen	Huong	54000	45
34974	Pham	Hoa	80000	45
34987	Phan	Lan	42000	45
45001	Tran	Hua	57500	30
75623	Vu	Hong	65000	30

Chạy lệnh SQL dưới đây để tạo truy vấn chéo bằng mệnh đề PIVOT.

```
SELECT 'TongLuong' AS TongLuongTheoPhong,
```

```
[30], [45]
```

```
FROM
```

```
(SELECT id_phong, luong
```

```
FROM nhanvien) AS BangNguon
```

```
PIVOT
```

```
(
```

```
SUM(luong)
```

```
FOR id_phong IN ([30], [45])
```

```
) AS BangChuyen;
```

Kết quả trả về sẽ như dưới đây.

TongLuongTheoPhong	30	45
TongLuong	122500	176000

Ví dụ trên tạo một bảng sau khi đã chuyển dữ liệu, cho biết tổng lương của phòng có ID là 30 và phòng có ID là 45. Kết quả nằm trên 1 hàng với 2 cột, mỗi cột là 1 phòng.

```
INSERT INTO phong
(id_phong, ten_phong)
VALUES
(30, 'Ketoan');
```

```
INSERT INTO phong
(id_phong, ten_phong)
VALUES
(45, 'Banhang');
```

```
INSERT INTO nhanvien
(so_nhanvien, ho, ten, luong, id_phong)
VALUES
(12009, 'Nguyen', 'Huong', 54000, 45);
```

```
INSERT INTO nhanvien
(so_nhanvien, ho, ten, luong, id_phong)
VALUES
(34974, 'Pham', 'Hoa', 80000, 45);
```

```
INSERT INTO nhanvien
(so_nhanvien, ho, ten, luong, id_phong)
VALUES
(34987, 'Phan', 'Lan', 42000, 45);
```

```
INSERT INTO nhanvien
(so_nhanvien, ho, ten, luong, id_phong)
VALUES
45001, 'Tran', 'Hue', 57500, 30);
```

```
INSERT INTO nhanvien
(so_nhanvien, ho, ten, luong, id_phong)
VALUES
(75623, 'Vu', 'Hong', 65000, 30);
```

Những lỗi thường gặp khi dùng PIVOT trong SQL Server

PIVOT không hợp nhất nhiều hàng

Hãy xét ví dụ bên dưới. Chúng ta sẽ triển khai cả PIVOT và UNPIVOT cho bảng này và so sánh kết quả của cả hai bảng.

	Name	Year	Sales
1	Pankaj	2010	72500
2	Rahul	2010	60500
3	Sandeep	2010	52000
4	Pankaj	2011	45000
5	Sandeep	2011	82500
6	Rahul	2011	35600
7	Pankaj	2012	32500
8	Pankaj	2010	20500
9	Rahul	2011	200...
10	Sandeep	2010	32000

```
SELECT Name,[Year] , Sales FROM
(
    SELECT [Year], Pankaj,Rahul,Sandeep FROM
    (SELECT Name, [Year] , Sales FROM Employee )Tab1
```

```
PIVOT
```

```
(
```

```
    SUM(Sales) FOR Name IN (Pankaj,Rahul,Sandeep)) AS Tab2
```

```
)Tab
```

```
UNPIVOT
```

```
(
```

```
    Sales FOR Name IN (Pankaj,Rahul,Sandeep)
```

```
) AS TAb2
```

Kết quả:

	Name	Year	Sales
1	Pankaj	2010	93000
2	Rahul	2010	60500
3	Sandeep	2010	84000
4	Pankaj	2011	45000
5	Rahul	2011	236100
6	Sandeep	2011	82500
7	Pankaj	2012	32500

Bạn có thể thấy hai bảng cho kết quả giống nhau.

Toán tử UNPIVOT

Nó là phương thức ngược lại của toán tử PIVOT trong SQL Server. Toán tử này thực hiện thao tác ngược lại với PIVOT bằng cách chuyển đổi dữ liệu từ các cột sang hàng. Toán tử UNPIVOT cũng xoay bảng PIVOT thành bảng bình thường. Nó được giới thiệu lần đầu tiên trong phiên bản SQL Server 2005.

Cú pháp:

Cú pháp sau minh họa UNPIVOT trong SQL Server:

```
SELECT (column_names)
FROM (table_name)
UNPIVOT
(
    Aggregate_function (column to be aggregated)
    FOR PivotColumn IN (pivot column values)
) AS (alias_name)
```

Ví dụ:

Hãy xem xét ví dụ sau để hiểu cách hoạt động của UNPIVOT. Đầu tiên, chúng ta sẽ tạo một bảng gốc và PIVOT, sau đó áp dụng toán tử UNPIVOT trên bảng này.

Đoạn code sau khai báo lần đầu một biến bảng tạm @Tab:

```
DECLARE @Tab TABLE
(
    Year int,
    North varchar(45),
    South varchar(45)
)
```

Tiếp theo, bạn sẽ chèn các giá trị vào bảng như sau:

```
INSERT INTO @Tab
SELECT Year, North, South FROM
(SELECT Region, Year, Sales FROM pivot_demo ) AS Tab1
PIVOT
(SUM(Sales) FOR Region IN (North, South)) AS PivotTable
ORDER BY PivotTable.Year
```

Giờ triển khai UNPIVOT vào lệnh dưới:

```
SELECT Region, Year, Sales FROM @Tab t
UNPIVOT
(
Sales FOR Region IN (North, South)
) AS UnpivotTable
```

Kết quả:

```
--Declare Temp Variable
DECLARE @Tab TABLE
(
    Year int,
    North varchar(45),
    South varchar(45)
)
--Insert Value in Temp Variable
INSERT INTO @Tab
SELECT Year, North, South FROM
(SELECT Region, Year, Sales FROM pivot_demo ) AS Tab1
PIVOT
(SUM(Sales) FOR Region IN (North, South)) AS PivotTable
ORDER BY PivotTable.Year
--Perform UNPIVOT Operation
SELECT Region, Year, Sales FROM @Tab t
UNPIVOT
(
    Sales FOR Region IN (North, South)
) AS UnpivotTable |
```

% < Results Messages

Region	Year	Sales
North	2010	93000
South	2010	112500
North	2011	80600
South	2011	82500
South	2012	32500

PHẦN THỰC HÀNH

Mở CSDL QLBH, thực hiện các lệnh sau:

1. Sử dụng **lệnh Insert** để nhập dữ liệu vào các bảng trong CSDL QLBH : dữ liệu nhập như trong hình sau. Chú ý đến cấu trúc table và các ràng buộc đã cài đặt khi nhập dữ liệu (xem *Bài tập 1 trang 24->27*)

Table NhomSanPham

I-35E18473... dbo.NhomHang		SQLQuery2.sql
Manhom	TenNhóm	
1	Điện Tử	.
2	Gia Dụng	.
3	Dụng Cụ Gia Đình	.
4	Các Mặt Hàng Khác	.
NULL	NULL	.

Table NhaCungCap

IBM-35E18473...bo.NhaCungCap						SQLQuery1.sql	Administrator (54)
	Mancc	TenNcc	Diachi	Phone	SoFax	DCMail	
	1	Công ty TNHH Nam Phương ...	1 Lê Lợi Phường 4 Quận Gò Vấp ...	083843456...	32343434 ...	NamPhuong@yahoo.com	
*	2	Công Ty Lan Ngọc	... 12 Cao bá Quát Quận 1 Tp. Hồ Chí Minh ...	086234567...	83434355 ...	LanNgoc@gmail.com	
	NULL	NULL	NULL	NULL	NULL	NULL	

Table SanPham

	Masp	Tensp	DonVitinh	GiaGoc	SITon	Manhom	MaNCC	MoTa
	1	Máy Tính	Cái	7000.0000	100	1	1	Máy Sony Ram 2GB
	2	Bàn phím	Cái	1000.0000	50	1	1	Bàn phím 101 phím
	3	Chuột	Cái	800.0000	150	1	1	Chuột không dây
	4	CPU	Cái	3000.0000	200	1	1	CPU
	5	USB	Cái	500.0000	100	1	1	8GB
	6	Lò Vi Sóng	Cái	1000000.0000	20	3	2	NULL
►*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table KhachHang

	Makh	Tenkh	DiaChi	DienThoai	LoaiKH	SoFax	DCMail	DiemTL
	KH1 ...	Nguyễn Thu Hằng	12 Nguyễn Du		VL	NULL	NULL	NULL
	KH2 ...	Lê Minh	34 Điện Biên Phủ	0123943455	TV	NULL	LeMinh@yahoo.com	... 100
►	KH3 ...	Nguyễn Minh Trung	3 Lê Lợi Quận Gò Vấp	098343434	VIP	NULL	Trung@gmail.com	... 800
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table HoaDon

	Mahd	NgayLap	Makh	NgayGiao	Noichuyen
	1	2015-09-30 00:00:00.000	KH1	2015-10-05 00:00:00.000	Cửa Hàng ABC 3 Lý Chính Thắng Quận 3
	2	2015-07-29 00:00:00.000	KH2	2015-08-10 00:00:00.000	23 Lê Lợi Quận Gò Vấp
	3	2015-10-01 00:00:00.000	KH3	2015-10-01 00:00:00.000	2 Nguyễn Du Quận Gò Vấp
►*	NULL	NULL	NULL	NULL	NULL

Table CT_HoaDon

	Mahd	Masp	Dongia	Soluong
	1	1	8000.0000	5
	1	2	1200.0000	4
	1	3	1000.0000	15
	2	2	1200.0000	9
	2	4	800.0000	5
	3	2	3500.0000	20
	3	3	1000.0000	15
►*	NULL	NULL	NULL	NULL

2. Dùng lệnh **Update** chỉnh sửa dữ liệu theo yêu cầu :

- Tăng đơn giá bán lên 5% cho các sản phẩm có mã là 2
- Tăng số lượng tồn lên 100 cho các sản phẩm có nhóm mặt hàng là 3 của nhà cung cấp có mã là 2

- c) Cập nhật cột mô tả với nội dung tùy ý cho sản phẩm có tên là Lò vi sóng.
- d) Trên bảng KhachHang, cập nhật mã khách hàng ‘KH3’ thành ‘VI003’
HD : Đọc và hiểu thông báo lỗi khi thực hiện câu lệnh update trên MaKH trên bảng Khachhang. Để cập nhật thành công, hãy tạm thời gỡ bỏ tham chiếu bằng cách sửa MaKH là NULL tại dòng có giá trị ‘KH3’ trên bảng HoaDon.
- e) Tương tự, sửa mã khách hàng ‘KH1’ thành ‘VL001’ , ‘KH2’ thành ‘T0002’
HD : Thiết lập lại ràng buộc khóa ngoại trên bảng HoaDon để cho phép khi sửa MaKH trên KhachHang thì tự động sửa MaKH trên bảng HoaDon.

3. Dùng lệnh Delete thực hiện các yêu cầu sau:

- a) Xóa dòng trong NhomHang có mã 4
- b) Xóa dòng trong CT_Hoadon có MaHD là 1 và MaSP là 3
- c) Xóa dòng trong bảng HoaDon có mã là 1
HD : Đọc và hiểu thông báo lỗi khi thực hiện xóa dòng có mã là 1 trong bảng HoaDon. Để xóa thành công, hãy xóa các dòng trong CT_Hoadon có tham chiếu tới (có MaHD = 1) , sau đó thực hiện xóa dòng trong bảng HoaDon.
- d) Tương tự , xóa dòng trong bảng HoaDon có mã là 2
HD : Thiết lập lại ràng buộc khóa ngoại trên bảng CT_HoaDon để cho phép khi xóa dòng trên bảng HoaDon thì tự động xóa các dòng trong CT_HoaDon mà có tham chiếu đến dòng đang xóa trong HoaDon.

TUẦN 5 (3 TIẾT)

PHẦN 2: Lệnh SELECT

BÀI TẬP 1: LỆNH SELECT – TRUY VẤN ĐƠN GIẢN

Từ bài tập này chúng ta sử dụng CSDL NorthWind. Dưới đây là mô tả tóm tắt các table trong CSDL NorthWind. Hãy tìm hiểu cấu trúc của các bảng và tạo diagram để thấy mối quan hệ giữa các table trong CSDL trước khi thực hiện các bài tập truy vấn.

Tên bảng	Mô tả bảng
[Categories]	Dữ liệu về nhóm hàng (nhóm sản phẩm)
[Products]	Dữ liệu về các mặt hàng (sản phẩm)
[Suppliers]	Dữ liệu về các nhà cung cấp hàng
[Orders]	Dữ liệu các hóa đơn bán hàng
[Order Details]	Dữ liệu chi tiết hóa đơn
[Customers]	Dữ liệu về các khách hàng
[Employees]	Dữ liệu về các nhân viên của công ty

[Shippers]	Dữ liệu về công ty giao hàng
[Territories]	Danh sách mã, tên tỉnh bang/thành phố
[Region]	Danh sách mã, tên vùng
[EmployeeTerritories]	Dữ liệu mô tả mối quan hệ giữa nhân viên và tỉnh bang/thành phố

Viết lệnh thực hiện các truy vấn sau :

1. Liệt kê danh sách tất cả các mặt hàng (Products).
2. Liệt kê danh sách tất cả các mặt hàng (Products). Thông tin bao gồm ProductID, ProductName, UnitPrice.
3. Liệt kê danh sách các nhân viên (Employees). Thông tin bao gồm EmployeeID, EmployeeName, Phone, Age. Trong đó EmployeeName được ghép từ LastName và FirstName; Age là tuổi được tính dựa trên năm hiện hành (GetDate()) và năm sinh.
4. Liệt kê danh sách các khách hàng (Customers) mà người đại diện có ContactTitle bắt đầu bằng chữ 'O'. Thông tin bao gồm CustomerID, CompanyName, ContactName, ContactTitle, City, Phone.
5. Liệt kê danh sách khách hàng (Customers) ở thành phố LonDon, Boise và Paris.
6. Liệt kê danh sách khách hàng (Customers) có tên bắt đầu bằng chữ V mà ở thành phố Lyon.
7. Liệt kê danh sách các khách hàng (Customers) không có số fax.
8. Liệt kê danh sách các khách hàng (Customers) có số Fax.
9. Liệt kê danh sách nhân viên (Employees) có năm sinh ≤ 1960
10. Liệt kê danh sách các sản phẩm (Products) có chứa chữ 'Boxes' trong cột QuantityPerUnit.
11. Liệt kê danh sách các mặt hàng (Products) có đơn giá (Unitprice) lớn hơn 10 và nhỏ hơn 15.
12. Liệt kê danh sách các mặt hàng (Products) có số lượng tồn nhỏ hơn 5.
13. Liệt kê danh sách các mặt hàng (Products) ứng với tiền tồn vốn. Thông tin bao gồm ProductId, ProductName, Unitprice, UnitsInStock, Total. Trong đó $Total = UnitsInStock * Unitprice$. Được sắp xếp theo Total giảm dần.
14. Hiển thị thông tin OrderID, OrderDate, CustomerID, EmployeeID của 2 hóa đơn có mã OrderID là '10248' và '10250'
15. Liệt kê chi tiết của hóa đơn có OrderID là '10248'. Thông tin gồm OrderID, ProductID, Quantity, Unitprice, Discount, TotalLine = $Quantity * unitPrice * (1 - Discount)$
16. Liệt kê danh sách các hóa đơn (orders) có OrderDate được lập trong tháng 9 năm 1996. Được sắp xếp theo mã khách hàng, cùng mã khách hàng sắp xếp theo ngày lập hóa đơn giảm dần.

17. Liệt kê danh sách các hóa đơn (Orders) được lập trong quý 4 năm 1997. Thông tin gồm OrderID, OrderDate, CustomerID, EmployeeID. Được sắp xếp theo tháng của ngày lập hóa đơn.
18. Liệt kê danh sách các hóa đơn (Orders) được lập trong ngày thứ 7 và chủ nhật của tháng 12 năm 1997. Thông tin gồm OrderID, OrderDate, Customerid, EmployeeID, WeekDayOfOrdate (Ngày thứ mấy trong tuần).
19. Liệt kê danh sách 5 customers có city có ký tự bắt đầu ‘M’.
20. Liệt kê danh sách 2 employees có tuổi lớn nhất. Thông tin bao gồm EmployeeID, EmployeeName, Age. Trong đó, EmployeeName được ghép từ LastName và FirstName; Age là tuổi.

TUẦN 6 (3 tiết)

BÀI TẬP 2: LỆNH SELECT – TRUY VẤN CÓ KẾT NỐI

1. Hiển thị thông tin về hóa đơn có mã ‘10248’, bao gồm: OrderID, OrderDate, CustomerID, EmployeeID, ProductID, Quantity, Unitprice, Discount.
2. Liệt kê các khách hàng có lập hóa đơn trong tháng 7/1997 và 9/1997. Thông tin gồm CustomerID, CompanyName, Address, OrderID, Orderdate. Được sắp xếp theo CustomerID, cùng CustomerID thì sắp xếp theo OrderDate giảm dần.
3. Liệt kê danh sách các mặt hàng xuất bán vào ngày 19/7/1996. Thông tin gồm : ProductID, ProductName, OrderID, OrderDate, Quantity.
4. Liệt kê danh sách các mặt hàng từ nhà cung cấp (supplier) có mã 1,3,6 và đã xuất bán trong quý 2 năm 1997. Thông tin gồm : ProductID, ProductName, SupplierID, OrderID, Quantity. Được sắp xếp theo mã nhà cung cấp (SupplierID), cùng mã nhà cung cấp thì sắp xếp theo ProductID.
5. Liệt kê danh sách các mặt hàng có đơn giá bán bằng đơn giá mua.
6. Danh sách các mặt hàng bán trong ngày thứ 7 và chủ nhật của tháng 12 năm 1996, thông tin gồm ProductID, ProductName, OrderID, OrderDate, CustomerID, Unitprice, Quantity, ToTal= Quantity*UnitPrice. Được sắp xếp theo ProductID, cùng ProductID thì sắp xếp theo Quantity giảm dần.
7. Liệt kê danh sách các nhân viên đã lập hóa đơn trong tháng 7 của năm 1996. Thông tin gồm : EmployeeID, EmployeeName, OrderID, Orderdate.
8. Liệt kê danh sách các hóa đơn do nhân viên có Lastname là ‘Fuller’ lập. Thông tin gồm : OrderID, Orderdate, ProductID, Quantity, Unitprice.

9. Liệt kê chi tiết bán hàng của mỗi nhân viên theo từng hóa đơn trong năm 1996. Thông tin gồm: EmployeeID, EmployName, OrderID, Orderdate, ProductID, quantity, unitprice, ToTalLine=quantity*unitprice.
10. Danh sách các đơn hàng sẽ được giao trong các thứ 7 của tháng 12 năm 1996.
11. Liệt kê danh sách các nhân viên chưa lập hóa đơn (dùng LEFT JOIN/RIGHT JOIN).
12. Liệt kê danh sách các sản phẩm chưa bán được (dùng LEFT JOIN/RIGHT JOIN).
13. Liệt kê danh sách các khách hàng chưa mua hàng lần nào (dùng LEFT JOIN/RIGHT JOIN).

TUẦN 7 (3 TIẾT)

BÀI TẬP 3: LỆNH SELECT – TRUY VẤN GOM NHÓM

1. Liệt kê danh sách các orders ứng với tổng tiền của từng hóa đơn. Thông tin bao gồm OrderID, OrderDate, Total. Trong đó Total là Sum của Quantity * Unitprice, kết nhóm theo OrderID.
2. Liệt kê danh sách các orders mà địa chỉ nhận hàng ở thành phố ‘Madrid’ (Shipcity). Thông tin bao gồm OrderID, OrderDate, Total. Trong đó Total là tổng trị giá hóa đơn, kết nhóm theo OrderID.
3. Viết các truy vấn để thống kê số lượng các hóa đơn :
 - Trong mỗi năm. Thông tin hiển thị : Year , CoutOfOrders ?
 - Trong mỗi tháng/năm . Thông tin hiển thị : Year , Month, CoutOfOrders ?
 - Trong mỗi tháng/năm và ứng với mỗi nhân viên. Thông tin hiển thị : Year, Month, EmployeeID, CoutOfOrders ?
4. Cho biết mỗi Employee đã lập bao nhiêu hóa đơn. Thông tin gồm EmployeeID, EmployName, CountOfOrder. Trong đó CountOfOrder là tổng số hóa đơn của từng employee. EmployName được ghép từ LastName và FirstName.
5. Cho biết mỗi Employee đã lập được bao nhiêu hóa đơn, ứng với tổng tiền các hóa đơn tương ứng. Thông tin gồm EmployeeID, EmployName, CountOfOrder , Total.
6. Liệt kê bảng lương của mỗi Employee theo từng tháng trong năm 1996 gồm EmployeeID, EmployName, Month_Salary, Salary = sum(quantity*unitprice)*10%. Được sắp xếp theo Month_Salary, cùng Month_Salary thì sắp xếp theo Salary giảm dần.
7. Tính tổng số hóa đơn và tổng tiền các hóa đơn của mỗi nhân viên đã bán trong tháng 3/1997, có tổng tiền >4000. Thông tin gồm EmployeeID, LastName, FirstName, CountofOrder, Total.

8. Liệt kê danh sách các customer ứng với tổng số hoá đơn, tổng tiền các hoá đơn, mà các hóa đơn được lập từ 31/12/1996 đến 1/1/1998 và tổng tiền các hóa đơn >20000. Thông tin được sắp xếp theo CustomerID, cùng mã thì sắp xếp theo tổng tiền giảm dần.
9. Liệt kê danh sách các customer ứng với tổng tiền của các hóa đơn ở từng tháng. Thông tin bao gồm CustomerID, CompanyName, Month_Year, Total. Trong đó Month_year là tháng và năm lập hóa đơn, Total là tổng của Unitprice* Quantity.
10. Liệt kê danh sách các nhóm hàng (category) có tổng số lượng tồn (UnitsInStock) lớn hơn 300, đơn giá trung bình nhỏ hơn 25. Thông tin bao gồm CategoryID, CategoryName, Total_UnitsInStock, Average_Unitprice.
11. Liệt kê danh sách các nhóm hàng (category) có tổng số mặt hàng (product) nhỏ hơn 10. Thông tin kết quả bao gồm CategoryID, CategoryName, CountOfProducts. Được sắp xếp theo CategoryName, cùng CategoryName thì sắp theo CountOfProducts giảm dần.
12. Liệt kê danh sách các Product bán trong quý 1 năm 1998 có tổng số lượng bán ra >200, thông tin gồm [ProductID], [ProductName], SumofQuatity
13. Cho biết Employee nào bán được nhiều tiền nhất trong tháng 7 năm 1997
14. Liệt kê danh sách 3 Customer có nhiều đơn hàng nhất của năm 1996.
15. Liệt kê danh sách các Products có tổng số lượng lập hóa đơn lớn nhất. Thông tin gồm ProductID, ProductName, CountOfOrders.

TUẦN 8 (3 TIẾT)

BÀI TẬP 4: LỆNH SELECT – TRUY VẤN LỒNG NHAU

1. Liệt kê các product có đơn giá mua lớn hơn đơn giá mua trung bình của tất cả các product.
2. Liệt kê các product có đơn giá mua lớn hơn đơn giá mua nhỏ nhất của tất cả các product.
3. Liệt kê các product có đơn giá bán lớn hơn đơn giá bán trung bình của các product. Thông tin gồm ProductID, ProductName, OrderID, Orderdate, Unitprice .
4. Liệt kê các product có đơn giá bán lớn hơn đơn giá bán trung bình của các product có ProductName bắt đầu là ‘N’.
5. Cho biết những sản phẩm có tên bắt đầu bằng ‘T’ và có đơn giá bán lớn hơn đơn giá bán của (tất cả) những sản phẩm có tên bắt đầu bằng chữ ‘V’.
6. Cho biết sản phẩm nào có đơn giá bán cao nhất trong số những sản phẩm có đơn vị tính có chứa chữ ‘box’ .
7. Liệt kê các product có tổng số lượng bán (Quantity) trong năm 1998 lớn hơn tổng số lượng bán trong năm 1998 của mặt hàng có mã 71

8. Thực hiện :

- Thống kê tổng số lượng bán ứng với mỗi mặt hàng thuộc nhóm hàng có CategoryID là 4. Thông tin : ProductID, QuantityTotal (tập A)
- Thống kê tổng số lượng bán ứng với mỗi mặt hàng thuộc nhóm hàng khác 4 . Thông tin : ProductID, QuantityTotal (tập B)
- Dựa vào 2 truy vấn trên : Liệt kê danh sách các mặt hàng trong tập A có QuantityTotal lớn hơn tất cả QuantityTotal của tập B

9. Danh sách các Product có tổng số lượng bán được lớn nhất trong năm 1998

Lưu ý : Có nhiều phương án thực hiện các truy vấn sau (dùng JOIN hoặc subquery). Hãy đưa ra phương án sử dụng subquery.

10. Danh sách các products đã có khách hàng mua hàng (tức là ProductID có trong [Order Details]). Thông tin bao gồm ProductID, ProductName, Unitprice
11. Danh sách các hóa đơn của những khách hàng ở thành phố LonDon và Madrid.
12. Liệt kê các sản phẩm có trên 20 đơn hàng trong quý 3 năm 1998, thông tin gồm ProductID, ProductName.
13. Liệt kê danh sách các sản phẩm chưa bán được trong tháng 6 năm 1996
14. Liệt kê danh sách các Employes không lập hóa đơn vào ngày hôm nay
15. Liệt kê danh sách các Customers chưa mua hàng trong năm 1997
16. Tìm tất cả các Customers mua các sản phẩm có tên bắt đầu bằng chữ T trong tháng 7 năm 1997
17. Liệt kê danh sách các khách hàng mua các hóa đơn mà các hóa đơn này chỉ mua những sản phẩm có mã ≥ 3
18. Tìm các Customer chưa từng lập hóa đơn (viết bằng ba cách: dùng NOT EXISTS, dùng LEFT JOIN, dùng NOT IN)
19. Bạn hãy mô tả kết quả của các câu truy vấn sau ?

```
Select ProductID, ProductName, UnitPrice From [Products]
Where Unitprice > ALL (Select Unitprice from [Products] where
ProductName like 'N%')
```

```
Select ProductId, ProductName, UnitPrice From [Products]
Where Unitprice > ANY (Select Unitprice from [Products] where
ProductName like 'N%')
```

```
Select ProductId, ProductName, UnitPrice from [Products]
Where Unitprice = ANY (Select Unitprice from [Products] where
```

ProductName like 'N%')

Select ProductId, ProductName, UnitPrice from [Products]
 Where ProductName like 'N%' and
 Unitprice>=ALL (Select Unitprice from [Products] where
 ProductName like 'N%')

BÀI TẬP 5: LỆNH SELECT – CÁC LOẠI TRUY VẤN KHÁC

- Sử dụng Select và Union để “hợp” tập dữ liệu lấy từ bảng Customers và Employees. Thông tin gồm CodeID, Name, Address, Phone. Trong đó CodeID là CustomerID/EmployeeID, Name là Companyname/LastName + FirstName, Phone là Homephone.
- Dùng lệnh SELECT...INTO tạo bảng HDKH_71997 chứa thông tin về các khách hàng gồm : CustomerID, CompanyName, Address, Total =sum(quantity*Unitprice) , với total là tổng tiền khách hàng đã mua trong tháng 7 năm 1997.
- Dùng lệnh SELECT...INTO tạo bảng LuongNV chứa dữ liệu về nhân viên gồm : EmployeeID, Name = LastName + FirstName, Address, Total =10%*sum(quantity*Unitprice) , với Total là tổng lương của nhân viên trong tháng 12 năm 1996.
- Dùng lệnh SELECT...INTO tạo bảng Ger_USA chứa thông tin về các hóa đơn xuất bán trong quý 1 năm 1998 với địa chỉ nhận hàng thuộc các quốc gia (ShipCountry) là 'Germany' và 'USA', do công ty vận chuyển 'Speedy Express' thực hiện.
- Pivot Query

Tạo bảng dbo.HoaDonBanHang có cấu trúc sau

```
CREATE TABLE dbo.HoaDonBanHang
(
    orderid INT NOT NULL,
    orderdate DATE NOT NULL,
    empid INT NOT NULL,
    custid VARCHAR(5) NOT NULL,
    qty INT NOT NULL,
    CONSTRAINT PK_Orders PRIMARY KEY(orderid)
)
```

Chèn dữ liệu vào bảng

```
(30001, '20070802', 3, 'A', 10),
(10001, '20071224', 2, 'A', 12),
(10005, '20071224', 1, 'B', 20),
(40001, '20080109', 2, 'A', 40),
```

(10006, '20080118', 1, 'C', 14),
 (20001, '20080212', 2, 'B', 12),
 (40005, '20090212', 3, 'A', 10),
 (20002, '20090216', 1, 'C', 20),
 (30003, '20090418', 2, 'B', 15),
 (30004, '20070418', 3, 'C', 22),
 (30007, '20090907', 3, 'D', 30)

- a) Tính tổng Qty cho mỗi nhân viên. Thông tin gồm empid, custid
- b) Tạo bảng Pivot có dạng sau

empid	A	B	C	D
1	NULL	20	34	NULL
2	52	27	NULL	NULL
3	20	NULL	22	30

Gợi ý:

```
SELECT empid, A, B, C, D
FROM (SELECT empid, custid, qty
      FROM dbo.Orders) AS D
PIVOT(SUM(qty) FOR custid IN(A, B, C, D)) AS P;
```

- c) Tạo 1 query lấy dữ liệu từ bảng dbo.HoaDonBanHang trả về số hóa đơn đã lập của nhân viên employee trong mỗi năm.
- d) Tạo bảng pivot hiển thị số đơn đặt hàng được thực hiện bởi nhân viên có mã 1, 3, 4, 8, 9.

BÀI TẬP TUẦN 9**Số tiết: 3****Mục tiêu:****☞ Thêm, sửa, xóa dữ liệu nâng cao**

- Insert dữ liệu bằng những cách khác
- Thực hiện Update, Delete bằng cú pháp 2 của lệnh hoặc dùng subquery

☞ Ôn tập**BÀI TẬP 1: INSERT dữ liệu****1. Dùng lệnh Insert...Select...**

Lệnh **INSERT ... SELECT ...** cho phép **nhập dữ liệu** vào một bảng (bảng đích) bằng cách lấy (truy vấn) dữ liệu từ bảng đã tồn tại (bảng nguồn). Bảng đích và bảng nguồn có thể nằm trong cùng CSDL hay thuộc những CSDL khác nhau.

Lưu ý : tập dữ liệu lấy từ bảng nguồn phải phù hợp cấu trúc, kiểu dữ liệu và ràng buộc dữ liệu trên bảng đích.

Áp dụng lệnh **INSERT ... SELECT ...** để nhập dữ liệu vào các bảng trong csdl QLBH dựa trên dữ liệu trong csdl Northwind ? Kiểm tra kết quả sau mỗi lần thực hiện ?

- a. Insert dữ liệu vào bảng KhachHang trong QLBH với dữ liệu nguồn là bảng Customers trong NorthWind.
- b. Insert dữ liệu vào bảng Sanpham trong QLBH. Dữ liệu nguồn là các sản phẩm có SupplierID từ 4 đến 29 trong bảng Northwind.dbo.Products
- c. Insert dữ liệu vào bảng HoaDon trong QLBH. Dữ liệu nguồn là các hóa đơn có OrderID nằm trong khoảng 10248 đến 10350 trong NorthWind.dbo.[Orders]
- d. Insert dữ liệu vào bảng CT_HoaDon trong QLBH. Dữ liệu nguồn là các chi tiết hóa đơn có OderID nằm trong khoảng 10248 đến 10350 trong NorthWind.dbo.[Order Detail]

2. Dùng công cụ SQL Server Import/Export Data Wizard

Cung cấp chức năng cho phép **nhập/xuất dữ liệu** từ SQL Server. Với chức năng import, ta có thể nhập dữ liệu vào CSDL từ các nguồn như

- Các cơ sở dữ liệu trong SQL Server, Oracle, MySQL, PostgreSQL, ...
- Text file, Excel và Access file
- ...

Tương tự, với chức năng export , ta có thể xuất dữ liệu từ SQL Server Database ra các dạng dữ liệu nêu trên.

Sử dụng công cụ Import/Export Wizard thực hiện các yêu cầu sau, kiểm tra kết quả sau mỗi lần thực hiện ?

- 1) Export toàn bộ dữ liệu trong bảng Customers trong NorthWind thành file Khachhang.txt.
- 2) Import : Danh sách các khách hàng có trong tập tin Khachhang.txt vào bảng KhachHang2 trong QLBH
- 3) Import : Các sản phẩm có SupplierID là 1 hoặc 2 hoặc 3 ở bảng Products trong NorthWind vào bảng SanPham trong QLBH. Lưu ý chỉ chọn những cột có tương ứng trong bảng SanPham.
- 4) Import : Các nhà cung cấp có Country là USA ở bảng Suppliers trong NorthWind vào bảng NhaCungCap trong QLBH. Lưu ý: chỉ chọn những cột có tương ứng trong bảng NhaCungCap.
- 5) Export : Dữ liệu của các bảng Products, Orders, Order Details trong bảng NorthWind vào tập tin QLHH.MDB. Lưu ý: Tập tin QLHH.MDB phải tồn tại trên đĩa trước khi thực hiện Export.
- 6) Export : Dữ liệu các bảng Products, Suppliers trong NorthWind thành tập tin SP_NCC.XLS
- 7) Export : Các khách hàng có City là LonDon từ bảng Customers trong NorthWind ra thành tập tin KH_london.TXT.
- 8) Export : Danh sách các sản phẩm ở Products trong NorthWind thành tập tin SanPham.TXT, thông tin cần lấy bao gồm ProductID, ProductName, QuantityPerUnit, Unitprice.

BÀI TẬP 2: LỆNH UPDATE

Cú pháp 2: **UPDATE <table_name>**
SET <column_name> = value , [, ...]
FROM <table_name> [, ...]
WHERE <condition>

Sử dụng cú pháp 2 hoặc dùng subquery thực hiện các yêu cầu sau :

1. Cập nhật chiết khấu 0.1 cho các mặt hàng trong các hóa đơn xuất bán vào ngày ‘1/1/1997’
2. Cập nhật đơn giá bán 17.5 cho mặt hàng có mã 11 trong các hóa đơn xuất bán vào tháng 2 năm 1997
3. Cập nhật giá bán các sản phẩm trong bảng [Order Details] bằng với đơn giá mua trong bảng [Products] của các sản phẩm được cung cấp từ nhà cung cấp có mã là 4 hay 7 và xuất bán trong tháng 4 năm 1997

4. Cập nhật tăng phí vận chuyển (Freight) lên 20% cho những hóa đơn có tổng trị giá hóa đơn ≥ 10000 và xuất bán trong tháng 1/1997
5. Thêm 1 cột vào bảng Customers lưu thông tin về loại thành viên : Member97 varchar(3) . Cập nhật cột Member97 là ‘VIP’ cho những khách hàng có tổng trị giá các đơn hàng trong năm 1997 từ 50000 trở lên.

BÀI TẬP 3: LỆNH DELETE

Cú pháp 2: **DELETE FROM <table_name>**
FROM <table_name> [, ...]
WHERE <condition>

Sử dụng cú pháp 2 hoặc dùng subquery thực hiện các yêu cầu sau :

HD : Các lệnh sau sẽ xóa những dòng dữ liệu trong [Order Details], là chi tiết của các hóa đơn bán cho khách hàng có mã ‘SANTG’ . Có thể sử dụng lệnh SELECT ... INTO ... để sao lưu những dòng này trước khi thực hiện lệnh xóa.

1. Xóa các dòng trong [Order Details] có ProductID 24, là “chi tiết của hóa đơn” xuất bán cho khách hàng có mã ‘SANTG’
2. Xóa các dòng trong [Order Details] có ProductID 35, là “chi tiết của hóa đơn” xuất bán trong năm 1998 cho khách hàng có mã ‘SANTG’
3. Thực hiện xóa tất cả các dòng trong [Order Details] là “chi tiết của các hóa đơn” bán cho khách hàng có mã ‘SANTG’

BÀI TẬP TUẦN 10**Số tiết: 3****Mục tiêu:**

- View**
- Index**

1) View

1. Tạo view **vw_Products_Info** hiển thị danh sách các sản phẩm từ bảng Products và bảng Categories. Thông tin bao gồm CategoryName, Description, ProductName, QuantityPerUnit, UnitPrice, UnitsInStock ? Thực hiện truy vấn dữ liệu từ View ?
Xem lại cú pháp lệnh tạo view ? Thực hiện truy vấn dữ liệu từ View ?
2. Tạo view **vw_CustomerTotals** hiển thị tổng tiền các hóa đơn của mỗi khách hàng theo tháng và theo năm. Thông tin gồm CustomerID, YEAR(OrderDate) AS Year, MONTH(OrderDate) AS Month, SUM(UnitPrice*Quantity) ?
Xem lại cú pháp lệnh tạo view ? Thực hiện truy vấn dữ liệu từ View ?
3. Tạo view hiển thị tổng số lượng sản phẩm bán được của mỗi nhân viên theo từng năm. Thông tin gồm EmployeeID, OrderYear, SumOfQuantity. Yêu cầu : người dùng không xem được cú pháp lệnh tạo view ?
Xem lại cú pháp lệnh tạo view ? Thực hiện truy vấn dữ liệu từ View ?
4. Tạo view **ListCustomer_view** chứa danh sách các khách hàng có trên 5 hóa đơn đặt hàng từ năm 1997 đến 1998, thông tin gồm mã khách hàng (CustomerID) , họ tên (CompanyName), Số hóa đơn (CountOfOrders) ?
Thực hiện truy vấn dữ liệu từ View ?
5. Tạo view **ListProduct_view** chứa danh sách những sản phẩm thuộc nhóm hàng ‘Beverages’ và ‘Seafood’ có tổng số lượng bán trong mỗi năm trên 30 sản phẩm, thông tin gồm CategoryName, ProductName, Year, SumOfQuantity ?
Thực hiện truy vấn dữ liệu từ View ?
6. Tạo view **vw_OrderSummary** với từ khóa WITH ENCRYPTION gồm OrderYear (năm của ngày lập hóa đơn), OrderMonth (tháng của ngày lập hóa đơn), OrderTotal (tổng tiền sum(UnitPrice*Quantity)) ?
Thực hiện truy vấn dữ liệu từ View ?
Viết lệnh để thấy công dụng của từ khóa trên ?
7. Tạo view **vwProducts** với từ khóa WITH SCHEMABINDING gồm ProductID, ProductName, Discount ?
Thực hiện truy vấn dữ liệu từ View ?

Thực hiện xóa cột Discount trong bảng Products. Có xóa được không?
Vì sao?

8. Tạo view **vw_Customer** với với từ khóa WITH CHECK OPTION chỉ chứa các khách hàng ở thành phố London và Madrid, thông tin gồm: CustomerID, CompanyName, City.
 - a. Chèn thêm một khách hàng mới không ở thành phố London và Madrid thông qua view vừa tạo. Có chèn được không? Giải thích ?
 - b. Chèn thêm một khách hàng mới ở thành phố London và một khách hàng mới ở thành phố Madrid. Dùng câu lệnh select trên bảng Customers để xem kết quả ?
9. Tạo 3 bảng lần lượt có tên là KhangHang_Bac, KhachHang_Trung, KhachHang_Nam, dùng để lưu danh sách các khách hàng ở ba miền, có cấu trúc như sau: MaKh, TenKH, DiaChi, KhuVuc. Trong đó,

KhachHang_Bac có một Check Constraint là KhuVuc là 'Bac Bo'

KhachHang_Nam có một Check Constraint là KhuVuc là 'Nam Bo'

KhachHang_Trung có một Check Constraint là KhuVuc là 'Trung Bo'

 Khoá chính là MaKH và KhuVuc.
 Tạo một partition view từ ba bảng trên, sau đó chèn mẫu tin tùy ý thông qua view. Kiểm tra xem mẫu tin được lưu vào bảng nào khi thêm/sửa/xóa dữ liệu vào view?
10. Lần lượt tạo các view sau, đặt tên tùy ý, sau đó thực hiện truy vấn dữ liệu từ view.
 - Danh sách các sản phẩm có chữ 'Boxes' trong DonViTinh.
 - Danh sách các sản phẩm có đơn giá <10.
 - Các sản phẩm có đơn giá gốc lớn hơn hay bằng đơn giá gốc trung bình.
 - Danh sách các khách hàng ứng với các hóa đơn được lập. Thông tin gồm CustomerID, CompanyName, OrderID, Orderdate.

Có thể INSERT, UPDATE, DELETE dữ liệu thông qua view nào trong các view trên ? Hãy Insert/Update/Delete thử dữ liệu tùy ý.

2) Index

1. Sử dụng Northwind database

- a. Xem thông tin các chỉ mục hiện hành trên bảng Orders bằng lệnh sp_helpindex ?
- b. SQL Server tìm kiếm dựa trên index như thế nào ?
 Thực thi các truy vấn sau trên Orders và xem thông tin trên cửa sổ Execution Plan

```
select *
from orders
where orderid = 10375
go
select *
```

```
from orders
where customerid = 'vinet'
go
select *
from orders
where customerid like 'v%'
go
select orderid, customerid
from orders
where customerid like 'b%'
go
select customerid
from orders
where customerid like 't%'
```

2. Sử dụng QLBH database, thực hiện :

- a. Xem thông tin các chỉ mục hiện hành trên bảng Hoadon ?
- b. Tạo chỉ mục dạng NONCLUSTERED cho bảng Hoadon với cột làm chỉ mục là MaKH ? Xem thông tin về chỉ mục vừa tạo ? Chỉ mục này hỗ trợ cho câu truy vấn nào , hãy viết và thực thi câu truy vấn để kiểm chứng ?

----- *Hết* -----