

Chương 2 (Phần 3)

Tạo và quản lý Table

Giáo trình & Tài liệu tham khảo:

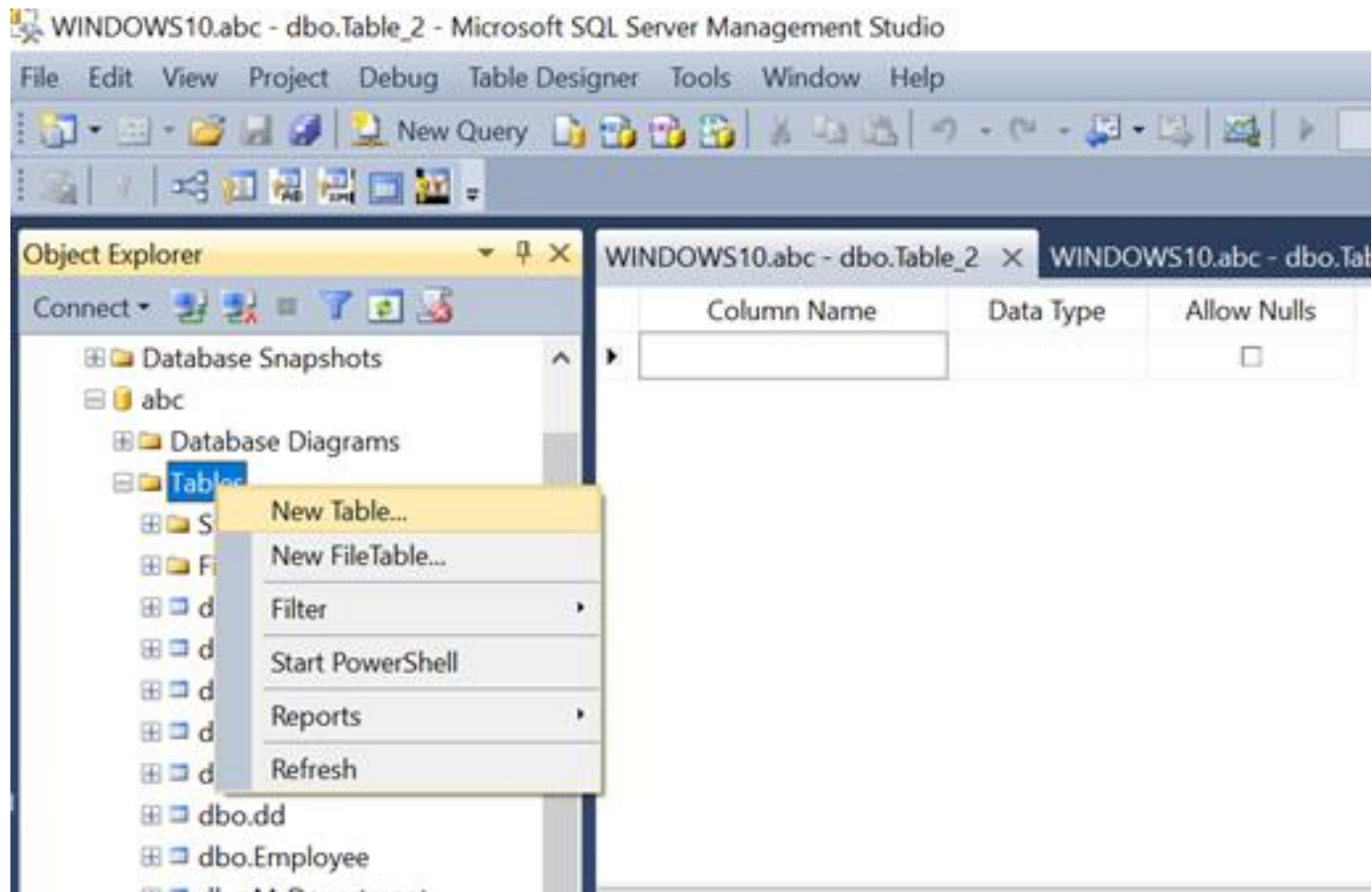
1. Microsoft SQL Server 2008 R2 Unleashed, Ray Rankins, Paul Bertucci, Chris Gallelli, Alex T. Silverstein, 2011, Pearson Education, Inc

2. <https://docs.microsoft.com/>

Nội dung

- Tạo (định nghĩa) table
- Thay đổi cấu trúc/xóa table
- Kiểu dữ liệu cho cột
- Chỉ định vị trí lưu trữ cho table
- Thiết lập các ràng buộc
- Partitioned Tables
- Temporary Tables
- Hai công cụ : SSMS và T-SQL

Tạo table bằng SSMS



Các lệnh T-SQL về table

- Các lệnh về table
 - CREATE TABLE (tạo bảng)
 - DROP TABLE (xóa bảng)
 - ALTER TABLE (sửa bảng)
 - Xem thông tin về table

Ôn lại : CSDL chương 4 về SQL

Cú pháp cơ bản

```
CREATE TABLE <Table_Name>  
( <Column_Name> <Data_Type>,....)
```

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

Kiểu dữ liệu

- Tham khảo
- <https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15>
- <https://www.sqlshack.com/an-overview-of-sql-server-data-types/>

Data type	Range	Storage
bigint	-2^{63} (-9,223,372,036,854,775,808) to $2^{63}-1$ (9,223,372,036,854,775,807)	8 Bytes
int	-2^{31} (-2,147,483,648) to $2^{31}-1$ (2,147,483,647)	4 Bytes
smallint	-2^{15} (-32,768) to $2^{15}-1$ (32,767)	2 Bytes
tinyint	0 to 255	1 Byte

Data Type	Lower Range	Upper Range	Storage	Remarks
Char(n)	0 characters	8000 characters	N bytes	1. It provides a fixed-width character data type.
Varchar(n)	0 characters	8000 characters	n bytes + 2 bytes	1.It is a variable length character data type. 2.N defines the string size.
Nchar	0 characters	4000 characters	2 times n bytes	It is a Unicode string of fixed width.
Nvarchar	0 chars	4000 Chars	2 times n bytes	Nvarchar is a Unicode string of variable width.

If you use **char** or **varchar**, we recommend to:

- Use **char** when the sizes of the column data entries are consistent.
- Use **varchar** when the sizes of the column data entries vary considerably.
- Use **varchar(max)** when the sizes of the column data entries vary considerably, and the string length might exceed 8,000 bytes.

Ràng buộc toàn vẹn

- Các loại ràng buộc
 - Domain integrity
 - Entity Integrity
 - Referential Integrity
- HAI phương thức cài đặt ràng buộc
 - Khai báo qua constraints, rules/default objects
 - Tạo mã code kiểm tra qua store procedure, trigger hay mã code trong applications

Các lệnh T-SQL về ràng buộc

- Khai báo các ràng buộc toàn vẹn
 - Primary key, foreign key
 - Unique constraint
 - Default constraint
 - Check constraint
 - NULL/NOT NULL
- Sử dụng rule/default object
- Các ràng buộc đặc biệt (*chưa học*)
 - Sử dụng trigger, store procedure để tạo
- Các lệnh về ràng buộc
 - Liệt kê các ràng buộc
 - Thêm/Xóa một ràng buộc của table
 - Vô hiệu hóa ràng buộc

Ôn lại : CSDL chương 4 về SQL

Các lệnh xem thông tin

- Xem thông tin một table

`sp_help <tableName>`

- Liệt kê các ràng buộc trong table

`sp_helpConstraint <tableName>`

- Xem thông tin một ràng buộc

`sp_help <constraintName>`

Ví dụ 1

```
CREATE TABLE NhaCungCap (  
MaNCC int identity NOT NULL Primary key,  
TenNCC varChar(25)  
)
```

IDENTITY [(*seed* , *increment*)]

Giá trị của cột là tự động phát sinh, gia tăng duy nhất, với thiết lập ban đầu *seed*, và bước tăng *increment*. Mặc định (1,1).

Cột có thiết lập Identity thường là cột làm khoá chính của table.

Giá trị được gán thường là các kiểu dữ liệu sau: tinyint, smallint, int, bigint, decimal(p,0), hay numeric(p,0).

Trong mỗi bảng chỉ cho phép 1 cột là identity mà thôi

Ví dụ 2

```
CREATE TABLE CT_hoadon  
(  sohd int NOT NULL,  
   MaHang char(5) NOT NULL,  
   SoLuong int NOT NULL,  
   DonGia money,  
   ThanhTien AS SoLuong*DonGia  
)
```

Cột tính toán : Giá trị của cột tự động phát sinh dựa theo biểu thức

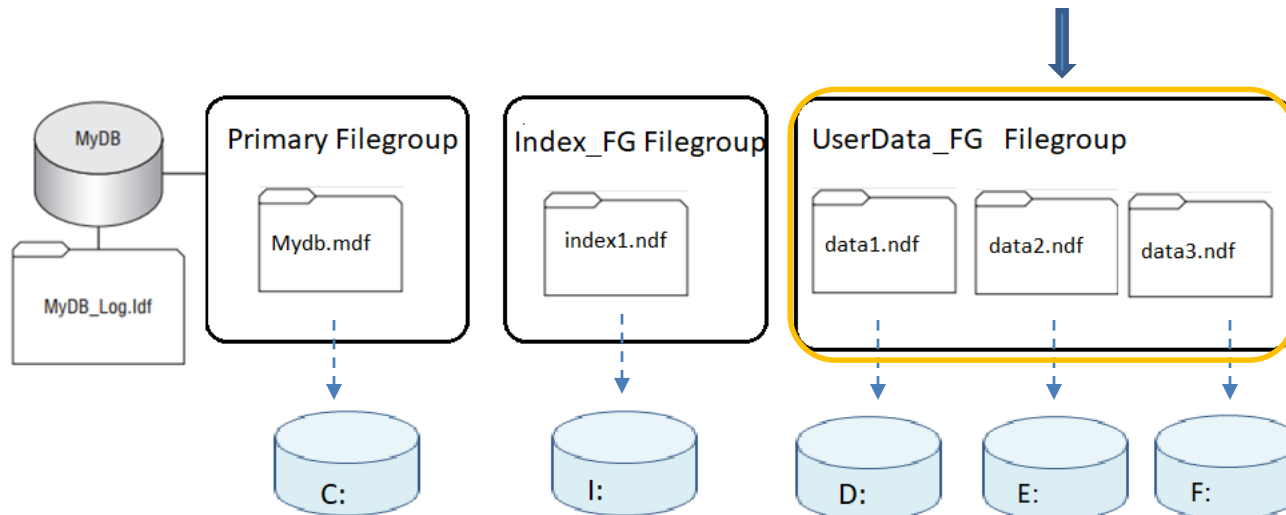
Khai báo cột tính toán : *column_name AS computed_column_expression*

Insert / update trên table có cột tính toán ?

Ví dụ 3

- Tạo và đặt table vào filegroup khác với *default filegroup*

```
CREATE TABLE CT_hoadon  
(  
    Sohd int NOT NULL,  
    MaHang char(5) NOT NULL,  
    SoLuong int NOT NULL,  
    DonGia money,  
    ThanhTien AS SoLuong*DonGia  
) ON [UserData_FG]
```



Ví dụ 4a


- Tạo table

```
CREATE TABLE NHANVIEN (  
    MANV CHAR(9) NOT NULL,  
    HOTEN VARCHAR(30) NOT NULL,  
    PHAI CHAR(3),  
    PHG INT      )
```

Ví dụ 4b

- Tạo table và bổ sung khai báo ràng buộc

```
CREATE TABLE NHANVIEN (  
    MANV CHAR(9) NOT NULL,  
    HOTEN VARCHAR(30) NOT NULL,  
    PHAI CHAR(3),  
    PHG INT );  
  
ALTER TABLE NHANVIEN ADD CONSTRAINT nv_PK PRIMARY KEY(MANV) ;  
ALTER TABLE NHANVIEN ADD LUONG INT CHECK (LUONG >1000);  
sp_help NHANVIEN ;  
ALTER TABLE NHANVIEN DROP <CHECK_constraint_name> ;  
ALTER TABLE NHANVIEN ADD CONSTRAINT CHECK (LUONG >5000);
```



Ví dụ 5

- Khai báo ràng buộc PK, FK, Check trong lệnh tạo table

```
CREATE TABLE NHANVIEN (  
    MANV CHAR(9) not null primary key ,  
    HOTEN VARCHAR(30),  
    PHAI CHAR(3) check (PHAI like 'nam' or PHAI like 'nu'),  
    PHG INT references PHONGBAN(MAPB)      )
```

Hoặc

```
CREATE TABLE NHANVIEN (  
    MANV CHAR(9) not null CONSTRAINT nv_pk primary key ,  
    HOTEN VARCHAR(30),  
    PHAI CHAR(3) CONSTRAINT nv_chk_phai check (PHAI like 'nam' or PHAI like 'nu'),  
    PHG INT CONSTRAINT nv_fk_pb references PHONGBAN(MAPB)      )
```

Ví dụ 6

- Khai báo ràng buộc PK, FK, Check trong lệnh Tạo table

```
CREATE TABLE NHANVIEN (  
    MANV CHAR(9) not null,  
    HOTEN VARCHAR(30),  
    PHAI CHAR(3),  
    PHG INT ,  
CONSTRAINT nv_pk primary key (MANV),  
CONSTRAINT nv_fk_pb Foreign key(PHG) references PHONGBAN(MAPB),  
CONSTRAINT nv_chk_phai check (PHAI like 'nam' or PHAI like 'nu')    )
```

Ví dụ 7

- Khai báo ràng buộc PK, FK, check trong lệnh Tạo table

```
CREATE TABLE PHANCONG (  
    MANV CHAR(9) NOT NULL,  
    SODA INT NOT NULL,  
    THOIGIAN DECIMAL(3,1) NOT NULL,  
    PRIMARY KEY (MANV, SODA),  
    FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV),  
    FOREIGN KEY (SODA) REFERENCES DEAN(MADA),  
    CHECK (thoigian ≥ 0) )
```

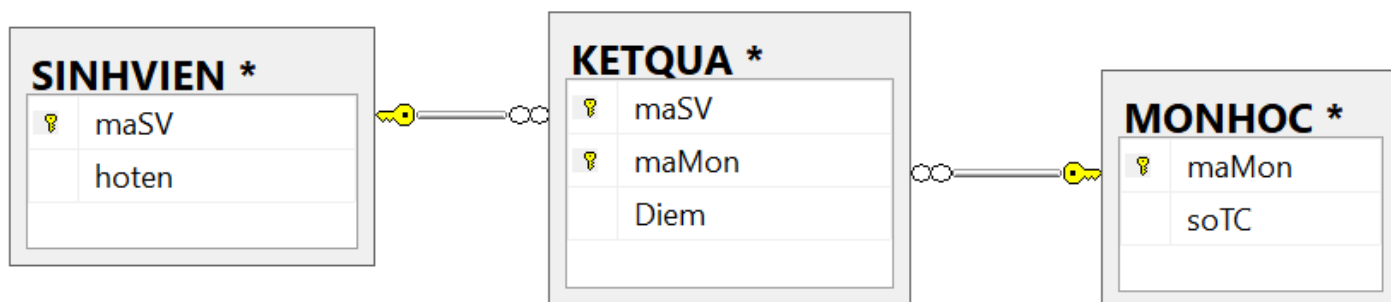
Ví dụ 8

- Khai báo ràng buộc unique , default

```
CREATE TABLE PHONGBAN (  
    MAPHG INT NOT NULL PRIMARY KEY,  
    TENPB VARCHAR(20) UNIQUE,  
    TRPHG CHAR(9),  
    NG_NHANCHUC DATETIME DEFAULT (GETDATE())  
)
```

Ví dụ 9

- Viết lệnh tạo 3 table với các PK và FK như hình



- Nhập dữ liệu

WINDOWS10.abc - dbo.SINHVIEN		
	maSV	hoten
	1	Hoàng Văn Minh
	2	Nguyễn Anh
	3	Trần Thị Mai

WINDOWS10.abc - dbo.KETQUA			
	maSV	maMon	Diem
	1	cs001	10
	2	cs001	7
	3	cs001	5
	1	is012	8
	2	is012	9
	3	is012	6

WINDOWS10.abc - dbo.MONHOC		
	maMon	soTC
	cs001	45
	is012	30
	it015	45

Ví dụ 9 (tiếp theo)

Ràng buộc khóa ngoại (FK) trên KETQUA ?

```
CREATE TABLE KETQUA (  
    MASV int NOT NULL,  
    MAMON char(5) NOT NULL,  
    DIEM int ,  
    PRIMARY KEY (MASV, MAMON),  
    FOREIGN KEY (MASV) REFERENCES SINHVIEN(MASV),  
    FOREIGN KEY (MAMON) REFERENCES MONHOC(MAMON) )
```

MASV trong KETQUA phải có giá trị mà tồn tại trong SINHVIEN(MASV)

MAMON trong KETQUA phải có giá trị mà tồn tại trong MONHOC(MAMON)

FK constraints trên KETQUA sẽ kiểm tra khi có thao tác :

- Xóa/ Sửa dữ liệu trên SINHVIEN, MONHoc
- Sửa/thêm dữ liệu trên bảng KETQUA

Ví dụ 9 (tiếp theo)

- Xóa dữ liệu từ bảng SINHVIEN , hoặc MONHOC

```
DELETE FROM monhoc  
WHERE maMon = 'CS001'
```

Không xóa được .

Vì xóa dòng 'cs001' sẽ khiến cho 3 dòng trong KETQUA vi phạm ràng buộc FK

```
DELETE FROM monhoc  
WHERE maMon = 'it015'
```

Xóa được .

Vì xóa dòng 'it015' sẽ không làm cho dòng nào trong KETQUA vi phạm ràng buộc FK

Ví dụ 9 (tiếp theo)

- Dùng thiết lập cascade cho ràng buộc khóa ngoại

```
CREATE TABLE KETQUA (  
    MASV int NOT NULL,  
    MAMON char(5) NOT NULL,  
    DIEM int ,  
    PRIMARY KEY (MASV, MAMON),  
    FOREIGN KEY (MASV) REFERENCES SINHVIEN(MASV)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
    FOREIGN KEY (MAMON) REFERENCES MONHOC(MAMON)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE )
```



Khi UPDATE / DELETE các record trong MONHOC, SINHVIEN sẽ update/delete tương ứng các record trong KETQUA để đảm bảo ràng buộc FK

Ví dụ 9 (tiếp theo)

- `sp_helpconstraint ketqua`

Object Name							
1	ketqua						
	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1	FOREIGN KEY	FK_KETQUA_MONHOC	No Action	No Action	Enabled	Is_For_Replication	maMon
2							REFERENCES abc.dbo.MONHOC (maMon)
3	FOREIGN KEY	FK_KETQUA_SINHVIEN	No Action	No Action	Enabled	Is_For_Replication	maSV
4							REFERENCES abc.dbo.SINHVIEN (maSV)
5	PRIMARY KEY (cl...	PK_KETQUA	(n/a)	(n/a)	(n/a)	(n/a)	maSV, maMon



Object Name							
1	ketqua						
	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1	FOREIGN KEY	FK_KETQUA_MONHOC	Cascade	Cascade	Enabled	Is_For_Replication	maMon
2							REFERENCES abc.dbo.MONHOC (maMon)
3	FOREIGN KEY	FK_KETQUA_SINHVIEN	Cascade	Cascade	Enabled	Is_For_Replication	maSV
4							REFERENCES abc.dbo.SINHVIEN (maSV)
5	PRIMARY KEY (clustered)	PK_KETQUA	(n/a)	(n/a)	(n/a)	(n/a)	maSV, maMon

Ví dụ 10

Rule

- Được quản lý như một object trong database, cung cấp một cách khác ngoài check constraint để thiết lập ràng buộc miền giá trị
- Ưu điểm : một rule được tạo ra có thể được gán buộc (áp dụng) cho nhiều cột hay nhiều user-defined data types. Rule thuận lợi trong tình huống một cột xuất hiện ở nhiều bảng trong database
- Dữ liệu đang tồn tại vào thời điểm gán buộc rule thì sẽ không bị ảnh hưởng
- Ví dụ

```
CREATE RULE phone_rule AS  
@phone LIKE '([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]'
```

```
sp_bindrule phone_rule, 'Person.PersonPhone.PhoneNumber'
```

Ví dụ 11

Default

- Được quản lý như một object trong database, cung cấp một cách khác ngoài default constraint để hỗ trợ nhập giá trị cho cột trong trường hợp không có giá trị cho cột khi thực hiện insert
- Ví dụ

```
CREATE DEFAULT password_df AS 'defaultpw'
```

```
sp_bindefault password_df, 'person.password.PasswordSalt'
```

Ví dụ 12

- Disable ràng buộc

```
ALTER TABLE Sales.Customer  
NOCHECK CONSTRAINT FK_Cust_SalesTerritory_TerritoryID
```

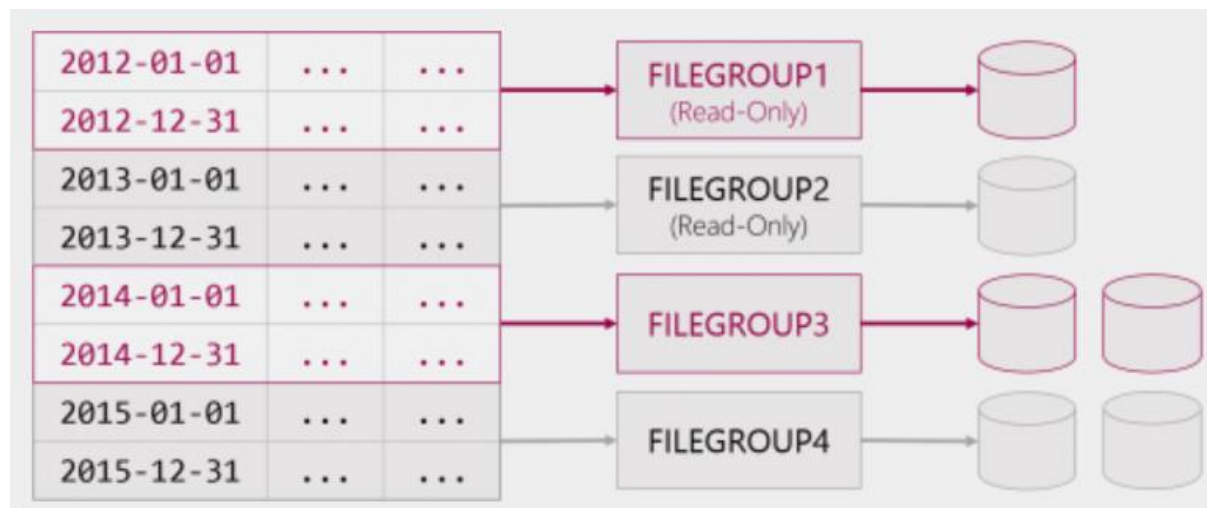
--- thao tác trên data

--- re-enable ràng buộc

```
ALTER TABLE Sales.Customer  
CHECK CONSTRAINT FK_Cust_SalesTerritory_TerritoryID
```

Partitioned table

- Phân vùng table là kỹ thuật cho phép lưu trữ data trong table thành từng phần (subset) , mỗi phần được truy suất và duy trì riêng biệt
- Table được phân vùng mức vật lý, và hoàn toàn là một logic table với end user



Partitioned table

- Tham khảo : <https://viblo.asia/p/partition-table-sql-server-MLzGOzbRvpq>
- Partitioned table cho phép quản trị dễ dàng hơn và cải thiện hiệu năng khi thực hiện backup, loading, restore và truy vấn dữ liệu

Temporary Tables

- Bảng tạm được chứa trong CSDL TempDb và được xóa một cách tự động khi không còn sử dụng nữa
- Bảng tạm có nhiều đặc tính giống với permanent tables, và được dùng để chứa data trung gian
- Bảng tạm được tạo với tên bắt đầu là dấu # hay ##
- Bảng tạm được xóa tường minh, hoặc được xóa tự động khi phiên làm việc tạo ra nó, hay thủ tục tạo ra nó kết thúc

Temporary Tables

- Bảng tạm local
 - Tên bắt đầu #
 - Chỉ hiện hữu với phiên làm việc hiện hành_đã tạo table tạm
- Bảng tạm global
 - Tên bắt đầu ##
 - Có thể được truy suất bởi tất cả các kết nối hiện hành tới DB

Temporary Tables

Ví dụ :

```
CREATE TABLE ##MyLocalTemTable  
( ID INT Primary key,  
  ColA Varchar(30) NULL )
```

Hoặc

```
SELECT *  
INTO #Employee2  
FROM Employee
```

Tóm tắt

Bài tập

- Đọc hiểu và ứng dụng được các ví dụ trong chương này ?