



I. Tóm tắt bài thực hành

1. Yêu cầu lý thuyết

Sinh viên đã được trang bị kiến thức:

- Các khái niệm về database, table...
- Các ràng buộc toàn vẹn: khoá chính, khoá ngoại...

2. Nội dung

❖ Tìm hiểu công cụ SQL Server Management Studio

- Cách kết nối CSDL
- Cách tạo database, table...
- Cách chỉnh sửa table: thêm/xóa/sửa thuộc tính, ràng buộc...

❖ Thực hiện được các bài tập sau

- Tạo CSDL Quản lí bán hàng (bài tập TH/Trang 3), Quản lí giáo vụ (bài tập TH/Trang 11).
- Thêm/Xóa/Sửa thuộc tính cho các bảng

❖ Một số lưu ý

- Cách đặt tên CSDL, tên quan hệ....
- Các kiểu dữ liệu trong SQL: int, char, varchar...

II. Giới thiệu và hướng dẫn sử dụng Microsoft SQL Server

Sinh viên xem nội dung file [HuongDanSuDungMSSQL2012.pdf](#)

III. Giới thiệu các kiểu dữ liệu

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

1. Số chính xác, không sai số (Exact numerics)

❖ Số nguyên

- *bigint*: khoảng giá trị từ -2^{63} đến $2^{63} - 1$ tức từ -9.223.372.036.854.775.808 đến 9.223.372.036.854.775,807.
- *int*: khoảng giá trị từ -2^{31} đến $2^{31} - 1$ tức từ -2.147.483.648 đến 2.147.483.647.
- *smallint*: khoảng giá trị từ -2^{15} đến 2^{15} tức từ -32.768 đến 32.767.
- *tinyint*: khoảng giá trị từ 0 đến 255.
- *bit*: chứa giá trị 0 hoặc 1.

❖ **Số thập phân**

- *decimal*: có giá trị từ $-10^{38} + 1$ đến $10^{38} - 1$.
- *numeric*: tương đương với *decimal*.

❖ **Kiểu tiền tệ**

- *money*: khoảng giá trị từ -2^{63} đến 2^{63} tức từ -922.337.203.685.477,5808 đến 922.337.203.685.477,5807.
- *smallmoney*: khoảng giá trị từ -214.748,3648 đến 214.748,3647

2. Số xấp xỉ

- ❖ *float*: độ chính xác từ $1.79E + 308$ đến $1.79E + 308$.
- ❖ *real*: độ chính xác từ $-3.40E + 38$ đến $3.40E + 38$.

3. Kiểu ngày tháng

- ❖ *datetime*: khoảng giá trị từ 01/01/1753 đến 31/12/9999.
- ❖ *smalldatetime*: khoảng giá trị từ 01/01/1900 đến 06/06/2079.

✚ Khi nhập dữ liệu, dữ liệu ngày tháng được đặt trong cặp ngoặc đơn và theo đúng format 'yyyy-mm-dd'. Ví dụ: '2014-12-27'

4. Kiểu chuỗi không có ký tự Unicode

Là kiểu ký tự, chứa chữ cái, ký hiệu, số, gồm những kiểu dữ liệu sau

- ❖ *char*: không chứa Unicode, chiều dài tối đa là 8.000 ký tự.
- ❖ *varchar*: không chứa Unicode, kích thước trong CSDL sẽ là kích thước thực dữ liệu hiện có, không tính theo kích thước đặt trước, chiều dài tối đa là 8.000 ký tự.
- ❖ *text*: không chứa Unicode, chiều dài tối đa là $2^{31} - 1$ tương đương với 2.147.483.647 ký tự

✚ Khi nhập dữ liệu, dữ liệu kiểu chuỗi không có ký tự Unicode được đặt trong cặp ngoặc đơn. Ví dụ: 'Chuoi khong co ky tu Unicode'

5. Kiểu chuỗi có ký tự Unicode

- ❖ *nchar*: có chứa Unicode, chiều dài tối đa là 4.000 ký tự
- ❖ *nvarchar*: có chứa Unicode, kích thước trong CSDL sẽ là kích thước thực dữ liệu hiện có, không tính theo kích thước đặt trước, chiều dài tối đa là 4.000 ký tự
- ❖ *ntext*: có chứa Unicode, chiều dài tối đa là $2^{30} - 1$ tức 1.073.741.823 ký tự.

✚ Khi nhập dữ liệu, dữ liệu kiểu chuỗi có ký tự Unicode được đặt trong cặp ngoặc đơn và có ký tự N ở trước cặp ngoặc. Ví dụ: N'Chuỗi có ký tự Unicode'

6. Kiểu chuỗi nhị phân

Là kiểu dữ liệu chứa dạng số ở hệ hexa, gồm 3 kiểu dữ liệu *binary*, *varbinary*, *image*

- ❖ *binary*: chiều dài tối đa là 8.000 byte
- ❖ *varbinary*: chiều dài tối đa là 8.000 byte
- ❖ *image*: chiều dài tối đa là 231 -1 tương đương với 2.147.483.647 byte

7. Các kiểu dữ liệu khác

- ❖ *sql_variant*: là kiểu dữ liệu xác định theo kiểu dữ liệu khác, một cột dữ liệu được định nghĩa dữ liệu kiểu này có thể lưu trữ nhiều dữ liệu có kiểu khác nhau trong cùng một bảng. Ví dụ có thể lưu trữ nhiều kiểu dữ liệu int, binary, char, nhưng không chứa dữ liệu kiểu text, ntext, image, timestamp, sql_variant.
 - ❖ *timestamp*: là kiểu dữ liệu có kích thước 8 byte, lưu trữ dạng số nhị phân do hệ thống tự sinh ra, mỗi giá trị timestamp trong CSDL là duy nhất.
- Ngoài ra SQL còn hỗ trợ một số kiểu dữ liệu khác như: *cursor*, *table*, và *uniqueidentifier*.

IV. Các lệnh tạo database, table, ràng buộc toàn vẹn

1. Câu lệnh tạo database

Để tạo một database mới ta sử dụng cú pháp

```
CREATE DATABASE database_name
```

Sau khi tạo mới một database ta cần dùng lệnh USE để chỉ định database sẽ thao tác trên đó. Cú pháp như sau

```
USE database_name
```

2. Câu lệnh tạo table

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

SQL	CSDL
Bảng	Quan hệ
Cột	Thuộc tính
Dòng	Bộ

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

Cú pháp như sau

```
CREATE TABLE table_name
(
    column_name data_type [integrity_constraint],
    column_name data_type [integrity_constraint],
    ...
    [integrity_constraint]
)
```

3. Các lệnh tạo ràng buộc toàn vẹn

Việc ràng buộc toàn vẹn trong SQL Server được chia làm 2 loại chính

- ❖ Loại đơn giản: sử dụng CONSTRAINT để mô tả

- ❖ Loại phức tạp: sử dụng TRIGGER để *thực hiện*

Trong khuôn khổ nội dung của bài thực hành này sinh viên chỉ tìm hiểu loại đơn giản.

Các loại ràng buộc đơn giản

- ❖ Kiểm tra duy nhất
PRIMARY KEY, UNIQUE
- ❖ Kiểm tra rỗng/khác rỗng
NULL/NOT NULL
- ❖ Kiểm tra tồn tại
FOREIGN KEY ... REFERENCES ...
- ❖ Kiểm tra miền giá trị
CHECK, DEFAULT

Các nguyên tắc chung

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

Cú pháp khai báo các mức constraint

- ❖ Constraint mức cột
column_def [CONSTRAINT constraint_name] constraint_type
- ❖ Constraint mức bảng
column_def,
...
[CONSTRAINT constraint_name] constraint_type parameters

Bảng sau mô tả quy định về khai báo các constraint

Tên	Mức cột	Mức bảng	Ý nghĩa
NULL	x		Cho phép cột tồn tại giá trị NULL
NOT NULL	x		Không cho phép cột tồn tại giá trị NULL
UNIQUE	x	x	Không cho phép cột tồn tại giá trị trùng nhau
PRIMARY KEY	x	x	Khóa chính của bảng
FOREIGN KEY	x	x	Khóa ngoại của bảng
CHECK	x	x	Quy định miền giá trị của dữ liệu cột
DEFAULT	x		Quy định giá trị mặc định của cột

Cú pháp chi tiết từng loại constraint

[CONSTRAINT ten_constraint]

```

PRIMARY KEY (danh_sach_cot_khoa_chinh)
hoặc UNIQUE (danh_sach_cot)
hoặc FOREIGN KEY (danh_sach_cot_khoa_ngoai)
REFERENCES bang_tham_chieu (ds_cot_tham_chieu)
hoặc CHECK (bieu_thuc_luan_ly)
hoặc DEFAULT gia_tri_mac_dinh FOR Ten_cot

```

V. Giới thiệu các lệnh `sp_help`, `sp_tables`, `sp_columns`, `sp_pkeys`, `sp_fkeys`, `sp_helpconstraint`, `sp_helpsql`...

VI. Các lệnh drop, alter

1. Lệnh DROP

Lệnh DROP được dùng để xóa một đối tượng trong SQL Server.

- ❖ Để xóa database ta sử dụng cú pháp

```
DROP DATABASE database_name
```

- ❖ Để xóa nhiều database một lúc

```
DROP DATABASE db_name1, db_name2, ..., db_nameN
```

- ❖ Để xóa table ta sử dụng cú pháp

```
DROP TABLE table_name
```

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

- ❖ Để xóa một bảng View ta sử dụng cú pháp

```
DROP VIEW view_name
```

- ❖ Để xóa một Procedure ta sử dụng cú pháp

```
DROP [ PROCEDURE | PROC ] procedure_name
```

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

2. Lệnh ALTER

SQL hỗ trợ ALTER TABLE cho phép chúng ta thêm, xóa hay sửa các thành phần của table.

- ❖ Để thêm cột vào table ta sử dụng cú pháp

```
ALTER TABLE table_name
ADD column_name datatype
```

Trong đó:

- *table_name*: là tên table cần thêm cột
- *column_name* : là tên cột sẽ được thêm vào *table_name*
- *datatype*: là kiểu dữ liệu của cột đó, ví dụ *int*, *varchar*, *nvarchar*, *decimal*...

- ❖ Để xóa cột khỏi table ta sử dụng cú pháp

```
ALTER TABLE table_name
DROP column_name
```


Trong đó:

- *table_name*: là tên table chứa cột sẽ bị xóa
 - *column_name* : là tên cột sẽ bị xóa khỏi *table_name*
- ❖ Để thay đổi kiểu dữ liệu của cột ta sử dụng cú pháp

```
ALTER TABLE table_name  
ALTER COLUMN column_name datatype
```

Trong đó:

- *table_name*: là tên table có cột cần thay đổi kiểu dữ liệu
- *column_name* : là tên cột sẽ được thay đổi kiểu dữ liệu
- *datatype*: là kiểu dữ liệu của cột đó, ví dụ *int*, *varchar*, *nvarchar*, *decimal*...

- ❖ Để thêm một RBTV ta sử dụng cú pháp

```
ALTER TABLE  
ADD [CONSTRAINT constraint_name] ...
```

Ta khai báo constraint sau từ khóa ADD tùy vào cú pháp chi tiết của từng loại constraint được đề cập ở phần trước.

- ❖ Để xóa một RBTV ta sử dụng cú pháp

```
ALTER TABLE  
DROP CONSTRAINT constraint_name
```

VII. Bài tập yêu cầu

1. Bài tập 1

Sinh viên cài đặt hoàn chỉnh bằng các câu lệnh SQL cho 2 CSDL QuanLyBanHang (Phần I, câu 1 bài tập thực hành trang 3) và QuanLyGiaoVu (Phần I, câu 1 bài tập thực hành trang 11).

2. Bài tập 2

Sinh viên hoàn thành Phần I bài tập QuanLyBanHang từ câu 2 đến câu 10.

3. Bài tập 3

Sinh viên hoàn thành Phần I bài tập QuanLyGiaoVu từ câu 2 đến câu 8.

~ HẾT ~