Chương 2: DDL
-ĐỊNH NGHĨA DỮ LIỆU

1. Các lệnh làm việc với CSDL

- Tạo CSDL
- Sử dụng CSDL
- Đổi tên
- Xóa CSDL

a. Tạo CSDL

- Cú pháp cơ bản:

  CREATE DATABASE <***database-name***>
  [ON <filespec>]

- Trong đó:
  - <database-name>: Tên CSDL
  - <**filespec**> ::= (**NAME** = logical_file_name,
    FILENAME = os_file_name
    [, **SIZE** = *n* [KB|MB|GB|TB]]
    [, **MAXSIZE** = *max* [KB|MB|GB|TB] | Unlimited]
    [, **FILEGROWTH** = *growth* [KB|MB|GB|TB|%]
)

3

- **<filespec>:**để điều khiển các thuộc tính cho file được tạo
  - Name: chỉ định tên logical cho file
  - Filename: chỉ định tên, đường dẫn file hệ điều hành (file vật lý)
  - Size: chỉ định kích thước của file, tối thiểu là 3MB
  - Maxsize: chỉ định kích thước tối đa lớn nhất mà file có thể phát triển đến, có từ khóa unlimited chỉ định file được phát triển cho đến khi đĩa bị đầy
  - Filegrowth: chỉ định độ tự động gia tăng của file

4

- Ví dụ 1: Tạo CSDL có tên là "Quan_ly_SV"

```
Create Database Quan_ly_SV
```

5

Ví dụ 2:

- *Tạo Database **QLSV** với tập tin dữ liệu chính là Quan_ly_SV**.mdf**, đặt tại thư mục D:\ với dung lượng khởi tạo là 3MB, tối đa là 10MB và độ gia tăng kích thước cho phép là 10%.*

```
Create Database QLSV
On(
    Name = Quan_ly_SV,
    FileName = 'D:\ Quan_ly_SV.mdf ',
    Size = 3MB,
    Maxsize = 10MB,
    FileGrowth = 10%
 )
```

6

b. Sử dụng CSDL:

- **Cú pháp: USE <tên CSDL>;**
- Ví dụ: USE QLSV;

c. Đổi tên CSDL

- **ALTER Database <Tên_CSDL> modify name = <tên mới>**
- Ví dụ: Thay đổi tên CSDL

ALTER Database QLSV modify name= QL_SV;

d. Xóa CSDL: Khi sử dụng lệnh xóa, CSDL sẽ bị xóa khỏi vùng lưu trữ, muốn tạo thì phải thực thi lại lệnh

+ Cú Pháp: DROP DATABASE <tên CSDL>;

+ Ví dụ: DROP Database Quan_ly_SV;

Chú ý: Không dùng cách xóa thông thường để xóa tệp dữ liệu của SQL mà phải dùng lệnh Drop.

7

2. Các lệnh làm việc với Bảng

2.1 Giới thiệu

- Được sử dụng để thao tác với một bảng dữ liệu.

8

2.2 Các ràng buộc khi tạo bảng

- **Ràng buộc toàn vẹn** là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan trong CSDL đều phải thoả mãn ở mọi thời điểm.
- RBTV được áp dụng nhằm đảm bảo tính đúng đắn và hợp lệ của dữ liệu.
- **Ví dụ:**
  - Ngày sinh của sinh viên phải nhỏ hơn ngày nhập học
  - Điểm của sinh viên phải từ 0 đến 10 là một qui định
  - Giới tính của sinh viên chỉ có thể là nam hoặc nữ.
  - Sinh viên đăng ký học với những môn học thuộc khoa mà sinh viên đó học

9

- RBTV có thể áp dụng ở mức cột hoặc mức bảng:
  - RB mức cột: chỉ áp dụng ở mức cột, có thể khai báo ngay hoặc sau khi khai báo cột.
  - RB mức bảng: áp dụng cho toàn bộ bảng, chỉ có thể khai báo sau khi khai báo cột.
- SQL cung cấp 7 công cụ để thực hiện RBTV bao gồm: Default, Not Null, Unique, Primary Key, Foreign Key, Check.

10

Phân loại RBTV

- Ràng buộc toàn vẹn được chia làm 3 loại:
  - Toàn vẹn thực thể: RB thực thể đảm bảo toàn vẹn cho các thực thể được mô hình bởi hệ thống. Ở mức độ đơn giản nhất, sự tồn tại của khóa chính là một RB thực thể nhằm đảm bảo qui tắc 'mỗi thực thể phải là duy nhất'.
  - Toàn vẹn miền: là tập các qui tắc xác định giá trị hợp lệ của thuộc tính.
  - Toàn vẹn tham chiếu: Khi mô hình quan hệ thì cần thiết phải chia các quan hệ để giảm thiểu sự dư thừa và phải thiết lập các khóa ngoại để liên kết các quan hệ. Nếu các liên kết này bị phá vỡ, trường hợp tốt nhất là hệ thống sẽ không còn đáng tin cậy, xấu nhất là không thể sử dụng được nữa. RBTV tham chiếu đảm bảo cho những liên kết này.

11

Các công cụ thực thi RBTV

Kiểu toàn vẹn	Các công cụ SQL Server
Toàn vẹn thực thể	1. Ràng buộc PRIMARY KEY 2. Ràng buộc UNIQUE 3. Ràng buộc IDENTITY
Toàn vẹn miền	1. Ràng buộc DEFAULT 2. Ràng buộc CHECK 3. Ràng buộc NOT NUL
Toàn vẹn tham chiếu	1. Ràng buộc FOREIGN KEY 2. Ràng buộc CHECK

12

2.3 Cú pháp tạo bảng

CREATE TABLE *<table-name>*

(

column1 data-type [RBTV],

[column2 data-type [RBTV],]

...

[columnn data-type [RBTV],]

[Constraint <name_RB> RBTV (column1, column2, …)]

)

Chú ý: nếu RBTV khai báo ngay tại khai báo cột ta gọi là RBTV mức cột, nếu khai báo sau khi khai báo cột (sử dụng Constraint) ta gọi là RBTV mức bảng.

13

- Trong đó,
  - **Table-name:** là tên bảng cần tạo, tuân thủ nguyên tắc định danh, không quá 128 ký tự
  - **Column**: tên cột cần tạo trong bảng, mỗi bảng có ít nhất một cột
  - **Data-type**: xác định kiểu dữ liệu được lưu trữ trong cột, ***Kiểu dữ liệu là thuộc tính bắt buộc***
  - ***RBTV:*** gồm các ràng buộc về khuôn dạng dữ liệu hay các ràng buộc về bảo toàn dữ liệu, có thể: **NOT NULL, NULL, UNIQUE, DEFAULT, PRIMARY KEY, IDENTITY, CHECK**,…
  - ***Constrain:*** dùng khi có nhiều hơn một RBTV cùng loại, đặc biệt là với RBTV khóa chính.

14

Ví dụ:

```
CREATE TABLE NHANVIEN (
   MANV  INT,
   TENNV  VARCHAR(50),
   NS   DATETIME,
   DCHI  VARCHAR(50),
   GT  VARCHAR (3),
   LUONG  INT,
   MAP  INT
)
```

15

*Ràng buộc CHECK

- *Sử dụng nhằm chỉ định điều kiện hợp lệ đối với dữ liệu.*
- Cú pháp:

    <column> <data type> check (dk)

Hoặc:

[CONSTRAINT *tên-ràng-buộc*] **CHECK** (*điều kiện*)

- ***Điều kiện***: là một biểu thức so sánh

16

Ví dụ: RBTV mức cột

```
CREATE TABLE  DIEM (
   MASV  VARCHAR(9)
   MAMH  VARCHAR(50),
   Diem1  INT  CHECK (Diem1 between 0 and 10),
   Diem2  INT  CHECK (Diem2 >=0 and Diem2<=10)
   )
```

17

Ví dụ: RBTV mức bảng

```
CREATE TABLE DIEM (
   MASV   VARCHAR(9),
   MAMH  VARCHAR(50),
   Diem1  INT,
   Diem2  INT,
   constraint  R1  CHECK ( (Diem1>=0 and
   Diem1<=10)  and  ( Diem2 between 0 and 10) )
   )
```

18

*Ràng buộc NOT NULL

- *Bắt buộc phải nhập giá trị cho cột khi thêm dữ liệu vào bảng*
- Cú pháp:

  <column> <type> **NOT NULL**

Ví dụ:

```
CREATE TABLE SINHVIEN (
   MASV   VARCHAR(9)
   TENSV  VARCHAR(50)  NOT NULL,
)
```

19

*Ràng buộc PRIMARY KEY

- Được dùng để định nghĩa khóa chính của bảng
- Cú pháp:

<column> <type> **PRIMARY KEY**

hoặc

[CONSTRAINT tên-ràng-buộc] **PRIMARY KEY**([d/s cột])

- Ví dụ:

```
CREATE TABLE  DIEM (
   MASV   VARCHAR(9)  PRIMARY KEY,
   MAMH   VARCHAR(50),
   Diem1  INT,
   Diem2  INT )
```

20

- Chú ý: khi khóa có hai thuộc tính → Dùng Constraint
- Ví dụ:

```
CREATE TABLE PHANCONG(
    MADA varchar(4),
    MANV varchar(100),
    Constraint PK_Ma PRIMARY KEY( MADA, MANV)
    )
```

21

*Ràng buộc DEFAULT

- Xác định giá trị mặc định ban đầu cho từng cột
- Cú pháp:

<Ten cột> datatype DEFAULT (gt)

Hoặc

[CONSTRAINT tên-ràng-buộc] **DEFAULT** (giá-trị)

- Ví dụ:

```
CREATE TABLE VATTU(
    MAVTU varchar(4) primary key,
    TenVtu varchar(100),
    SoLuong int Default (100)
    )
```

22

- Ví dụ:

```
CREATE TABLE SINHVIEN(
    MASV  varchar(4) primary   key ,
    TenSV  varchar(100),
    GioiTinh  varchar(3)  Default ('Nam')
    )
```


*Ràng buộc UNIQUE

- Dùng khi quy định một cột nào đó cho phép chỉ nhập một giá trị duy nhất cho từng dòng
- Cú pháp:

<column> <type> **UNIQUE**

Hoặc:

[**CONSTRAINT** Tên_RB] **UNIQUE** (DS cột)

- Trong một bảng chỉ có một ràng buộc khoá chính nhưng có thể có nhiều ràng buộc dữ liệu duy nhất

- Ví dụ:

```
CREATE TABLE VATTU
( MAVTU varchar(4) primary key,
 TenVtu varchar(100) UNIQUE )

CREATE  TABLE  VATTU
( MAVTU  varchar(4) not null,
 TenVtu  varchar(100),
Constraint  UN_Vtu  UNIQUE (MaVtu,TenVtu) )
```


*Ràng buộc FOREIGN KEY

- Đảm bảo việc nhập dữ liệu cho một cột thuộc tính nào đó phù hợp tham chiếu tới một bảng quan hệ khác
- Cú pháp:

REFERENCES *tên-bảng-tham-chiếu(d/s cột tham chiếu)*

hoặc

[**CONSTRAINT** *tên-ràng-buộc]* ***FOREIGN KEY****([d/s cột])*
REFERENCES *tên-bảng-tham-chiếu(d/s cột tham chiếu)*
[ON DELETE CASCADE | NO ACTION | SET NULL | SET DEFAULT]
[ON UPDATE CASCADE | NO ACTION | SET NULL | SET DEFAULT]

- Việc định nghĩa ràng buộc Foreign Key gồm:
  - Tên cột và danh sách cột của bảng tham gia vào khóa ngoại
  - Tên bảng được tham chiếu bởi khóa ngoại và danh sách cột được tham chiếu đến trong bảng tham chiếu

- Cách xử lý bản ghi được sử dụng trong các trường hợp:
  - **CASCADE**: tự động xóa (cập nhật) nếu bản ghi được tham chiếu bị xóa (cập nhật)
  - **NO ACTION**: (*mặc định)* nếu bản ghi trong bảng tham chiếu đang được tham chiếu bởi một bản ghi bất kỳ trong bảng được định nghĩa thì bản ghi đó không được phép xóa hoặc cập nhật
  - **SET NULL**: cập nhật lại khóa ngoại của bản ghi thành giá trị NULL (*nếu cột phép nhận giá trị NULL)*
  - **SET DEFAULT**: cập nhật lại khóa ngoại của bản ghi nhận giá trị mặc định (*nếu có giá trị mặc định)*

27

- Ví dụ

```
CREATE TABLE CTHOADON
( SOHD Varchar(3) REFERENCES HOADON(SOHD),
MAVTU Varchar(3) REFERENCES VATTU(MAVTU),
Soluong int,
Dongia float )
CREATE TABLE CTHOADON
(SOHD Varchar(3),
MAVTU Varchar(3),
Soluong int,
Dongia float,
FOREIGN KEY (SOHD) REFERENCES HOADON(SOHD),
FOREIGN KEY (MAVTU) REFERENCES VATTU(MAVTU)
)
```

28

*Ràng buộc IDENTITY

- Cú pháp:

[constraint name] **IDENTITY** [(*start, step*)],

- Trong một bảng chỉ có tối đa một cột được chỉ định làm cột định danh.
- **start:** là số mà SQL Server sử dụng để cấp phát cho mẫu tin đầu tiên. Mặc định là 1.
- **step**: là chỉ số mà SQL Server cộng lên để cấp phát cho từng mẫu tin kế tiếp. Mặc định là 1.

Ví dụ áp dụng

- Tạo CSDL "Quản lý nhân sự"
- Gồm các bảng:
  - **NHAN VIEN** (MaNV, HoTen, GT, NS, QQ, DT, MaPB)
  - **PHONG BAN** (MaPB, TenPB, DienThoai) *
  - **CHUC VU** (MaCV, TenCV, HSPC)
  - **BACLUONG** (MaBL, HSL, HSPC)
  - **DC_PB** (MaPB, DiaChi)
  - **NV_CHUCVU** (MaNV, MaCV, NgayQD)
- Yêu cầu ràng buộc:
  - Giới tính chỉ nhận Nam và Nữ
  - Chức vụ chỉ có "GĐ", "PGD", "TP", "PP", "NV"
  - HSL mặc định là 2.54, HSL từ 2.54 đến 12
  - HSPC tu 0.4 den 1.2
  - NS nhỏ hơn ngày của hệ thống (getdate())
  - Tạo các ràng buộc khóa chính, khóa ngoại và not null

4. Sửa cấu trúc bảng

- Sử dụng câu lệnh **ALTER TABLE**.
- Câu lệnh này cho phép thực hiện được các thao tác sau:
  - *Bổ sung một cột mới vào bảng.*
  - *Xoá một cột khỏi bảng.*
  - *Thay đổi định nghĩa kiểu của một cột trong bảng.*
  - *Xoá bỏ hoặc bổ sung các ràng buộc cho bảng*

31

*Thêm cột mới vào trong bảng

- Cú pháp:

  ALTER TABLE Tên_bảng

  ADD Tên_cột Kiểu_dữ_liệu [RBTV] [,...]

- *Luôn thêm cột mới vào cuối bảng*

- Ví dụ:

  ALTER TABLE VATTU

  ADD Soluong int

32

*Hủy bỏ một cột trong bảng

- Cú pháp

ALTER TABLE Tên_bảng

DROP COLUMN Tên_cột [,...]

- Ví dụ:

ALTER TABLE VATTU

DROP COLUMN TenVtu

33

*Sửa đổi kiểu dữ liệu của cột

- Cú pháp:

ALTER TABLE Tên_bảng

ALTER COLUMN Tên_cột Kiểu_dữ_liệu_mới

- *Kiểu dữ liệu mới phải lơn hơn kiểu dữ liệu cũ đã có*

- Ví dụ:

ALTER TABLE VATTU

ALTER COLUMN TenVtu **Nvarchar(30)**

34

*Thêm ràng buộc cho cột

- Cú pháp:

```
ALTER TABLE Tên_bảng
ADD CONSTRAINT Tên_ràng_buộc Loại_ràng_buộc
```

- Ví dụ:

```
ALTER TABLE VATTU
ADD CONSTRAINT CK_NgayNhap
CHECK ( Ngaynhap <= GetDate() )
```

35

*Hủy ràng buộc đã đặt

- Cú pháp

```
ALTER TABLE Tên_bảng
DROP CONSTRAINT Tên_ràng_buộc
```

- Ví dụ:

```
ALTER TABLE VATTU
DROP CONSTRAINT CK_NgayNhap
```

36

- Lưu ý:
  - Khi thêm cột vào bảng đã có ít nhất 1 bản ghi thì cột mới thêm phải cho phép nhận giá trị NULL hoặc phải có giá trị mặc định
  - Muốn xóa cột có tồn tại ràng buộc/có tham chiếu khóa ngoài thì phải xóa ràng buộc hoặc khóa ngoài
  - Nếu thêm ràng buộc cho bảng đã có dữ liệu mà ràng buộc không thỏa mãn với các dữ liệu đã có thì nó sẽ thêm vào được.

*Bật tắt các ràng buộc

- Cú pháp

ALTER TABLE Tên_bảng

NOCHECK CONSTRAINT ALL | Tên_constraint [,...]

ALTER TABLE Tên_bảng

CHECK CONSTRAINT ALL | Tên_constraint [,...]

*Đổi tên cột

- Cú pháp

EXEC SP_Rename '*Tên_bảng.Tên_cột*', '*Tên_mới*', **'COLUMN'**

- Ví dụ:

EXEC SP_RENAME 'VATTU.MAVTU', 'MAVATTU', 'COLUMN'

39

*Đổi tên bảng

- Cú pháp

EXEC sp_rename 'Tên_bảng', 'Tên_mới'

- Ví dụ:

EXEC SP_RENAME 'VATTU', 'VT'

40

Ví dụ - Thay đổi cấu trúc bảng

- *Thêm cột NGHENGHIEP có kiểu CHAR với độ rộng cho phép 20*
- *Thay đổi độ rộng cột thành 50 ký tự*
- *Thêm ràng buộc NOT NULL với trường NGHENGHIEP*
- *Xóa cột NGHENGHIEP*

5. Xóa bảng

- Để xóa một bảng khỏi CSDL
- Cú pháp:

  DROP TABLE Danh_sách_tên_các_bảng

- Ví dụ:

  DROP TABLE VatTu, CTHoaDon

- Lưu ý:

-Câu lệnh Drop Table không thể thực hiện nếu bảng cần xóa được tham chiếu bởi một **Foreign Key**

-Các ràng buộc, chỉ mục, trigger,.. đều bị xóa, nếu tạo lại bảng thì cũng phải tạo lại các đối tượng này

-Sau khi xóa không thể khôi phục lại bảng và dữ liệu bảng

IV. INDEX

- Index là chỉ mục quan trọng trong CSDL đặc biệt với CSDL lớn.
- Index có thể thiết lập cho 1 hoặc nhiều cột của bảng
- Index được sắp xếp nhằm hỗ trợ việc tìm kiếm, truy vấn dữ liệu một cách nhanh chóng.

CREATE [UNIQUE] [CLUSTERED] [NONCLUSTERED]
INDEX <tên index>
ON <tên bảng>(tên cột,..**)**

-**Unique**: dữ liệu cột Index là duy nhất không lặp lại

-**Clustered**: dữ liệu được sắp xếp vật lý trên ổ đĩa

-**Nonclustered**: dữ liệu được sắp xếp logic, nhanh trong nhập liệu

43

Ví dụ: tạo index trên cột MaNV của bảng Nhân viên

CREATE INDEX ID_MANV ON NHANVIEN(MANV**)**

Xóa INDEX

- DROP INDEX<tên index> ;
- Ví dụ: xóa index vừa thiết lập

DROP INDEX ID_MANV;

44