

# Transact-SQL (T-SQL) - Cơ bản

1

## Nội dung

- Giới thiệu T-SQL
- Tạo cơ sở dữ liệu
- Thao tác với bảng
- Ràng buộc toàn vẹn
- Biến
- Các cấu trúc lập trình

2

## T-SQL

- Ngôn ngữ cơ bản trong SQL Server
  - Mở rộng ngôn ngữ SQL bằng cách thêm một số cấu trúc lập trình.
- T-SQL là ngôn ngữ đơn giản
  - Các biến cục bộ
  - Cấu trúc rẽ nhánh, lặp
  - Các toán tử, các hàm
  - Không có các câu lệnh để tạo giao diện người dùng
  - Không có các thao tác xuất nhập tập tin
- Tham khảo **Books Online!**

3

| Convention       | Used for   |
|------------------|--|
| UPPERCASE        | Transact-SQL keywords.   |
| <i>italic</i>    | User-supplied parameters of Transact-SQL syntax.   |
| <b>bold</b>      | Database names, table names, column names, index names, stored procedures, utilities, data type names, and text that must be typed exactly as shown.   |
| <u>underline</u> | Indicates the default value applied when the clause that contains the underlined value is omitted from the statement.  |
| (vertical bar)   | Separates syntax items enclosed in brackets or braces. You can use only one of the items.  |
| [ ] (brackets)   | Optional syntax items. Do not type the brackets.   |
| { } (braces)     | Required syntax items. Do not type the braces.   |
| [...n]           | Indicates the preceding item can be repeated <i>n</i> number of times. The occurrences are separated by commas.  |
| [...n]           | Indicates the preceding item can be repeated <i>n</i> number of times. The occurrences are separated by blanks.  |
| [.]              | Optional Transact-SQL statement terminator. Do not type the brackets.  |
| <label> ::=      | The name for a block of syntax. This convention is used to group and label sections of lengthy syntax or a unit of syntax that can be used in more than one location within a statement. Each location in which the block of syntax can be used is indicated with the label enclosed in chevrons: <label>. |

## Tạo CSDL với T-SQL (1)

**Cú pháp:**  
CREATE DATABASE *database\_name*  
[ ON  
    [ < filespec > [ ,...n ]  
    [ , < filegroup > [ ,...n ] ]  
    [ LOG ON { < filespec > [ ,...n ] }  
    [ COLLATE *collation\_name* ]  
    [ FOR LOAD | FOR ATTACH ]  
    < filespec > ::=  
    [ PRIMARY ]  
    { ( [ NAME = *logical\_file\_name* , ]  
      FILENAME = '*os\_file\_name*'  
      [ , SIZE = *size* ]  
      [ , MAXSIZE = { *max\_size* | UNLIMITED } ]  
      [ , FILEGROWTH = *growth\_increment* ] } [ ,...n ]  
    < filegroup > ::=  
    FILEGROUP *filegroup\_name* < filespec > [ ,...n ]

5

## Tạo CSDL với T-SQL (2)

Ví dụ: Tạo một CSDL đơn giản:

```
CREATE DATABASE Sales_1
```

6

## Tạo CSDL với T-SQL (3)

Ví dụ 2: Tạo một CSDL có mô tả chi tiết

```
CREATE DATABASE Sales_2
ON
( NAME = Sales_dat, FILENAME = 'c:\program
files\microsoft sql
server\mssql\data\saledat.mdf',
SIZE = 10, MAXSIZE = 50, FILEGROWTH = 5 )
LOG ON
( NAME = 'Sales_log',
FILENAME = 'c:\program files\microsoft sql
server\mssql\data\salelog.ldf',
SIZE = 5MB, MAXSIZE = 25MB, FILEGROWTH = 5MB )
GO
```

7

## Chỉnh sửa CSDL với T-SQL (1)

Cú pháp:

```
ALTER DATABASE database
{ ADD FILE <filespec> [,...n] [ TO FILEGROUP filegroup_name ]
ADD LOG FILE <filespec> [,...n]
REMOVE FILE logical_file_name
ADD FILEGROUP filegroup_name
REMOVE FILEGROUP filegroup_name
MODIFY FILE <filespec>
MODIFY NAME = new_dbname
MODIFY FILEGROUP filegroup_name {filegroup_property | NAME
= new_filegroup_name }
SET <optionspec> [,...n] [ WITH <termination> ]
COLLATE <collation_name> }
```

8

## Chỉnh sửa CSDL với T-SQL (2)

Ví dụ:

--Tạo CSDL Test1

```
CREATE DATABASE Test1 ON
( NAME = Test1dat1,
FILENAME = 'c:\Program Files\Microsoft SQL
Server\MSSQL\Data\t1dat1.mdf',
SIZE = 5MB, MAXSIZE = 100MB, FILEGROWTH = 5MB )
```

GO

--Chỉnh sửa CSDL Test1

```
ALTER DATABASE Test1
ADD FILE
( NAME = Test1dat2,
FILENAME = 'c:\Program Files\Microsoft SQL
Server\MSSQL\Data\t1dat2.mdf',
SIZE = 5MB, MAXSIZE = 100MB, FILEGROWTH = 5MB )
```

9

## Xóa CSDL với T-SQL (2)

Cú pháp: DROP DATABASE <DATABASE NAME>

Ví dụ: DROP DATABASE NHANSU

10

## Tạo Bảng bằng T-SQL (1)

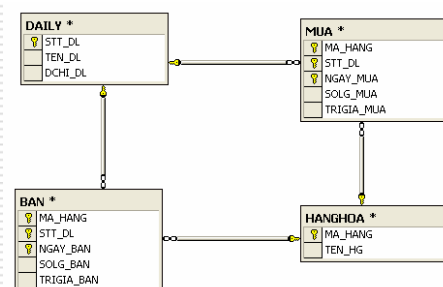
▪ Cú pháp:

CREATE TABLE <table\_name>

```
( <column1 datatype [ Null | Not Null ] > ,
[ <column2 datatype [ Null | Not Null ] > ,
... ]
)
```

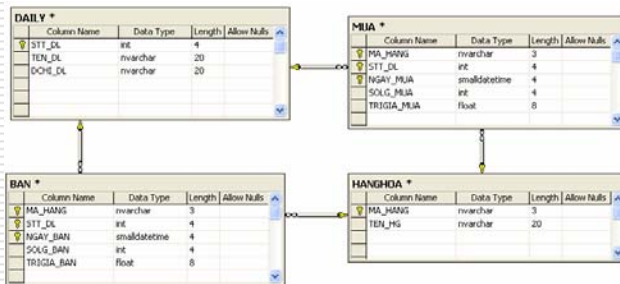
11

## Ví dụ: Cho Diagram của CSDL Hàng hóa



12

## Diagram chi tiết



13

## Ví dụ tạo bảng với T-SQL (1)

--Tạo CSDL HANGHOA-----

```
CREATE DATABASE HANGHOA
```

--Mở CSDL HANGHOA-----

```
USE HANGHOA
```

-- Tạo bảng Đại lý-----

```
CREATE TABLE DAILY (
    STT_DL int NOT NULL ,
    TEN_DL nvarchar (20) NOT NULL ,
    DCHI_DL nvarchar (20) NOT NULL )
```

-- Tạo bảng Hàng hóa-----

```
CREATE TABLE HANGHOA (
    MA_HANG nvarchar (3) NOT NULL ,
    TEN_HG nvarchar (20) NOT NULL )
```

## Ví dụ tạo bảng với T-SQL (2)

--Tạo bảng BÁN-----

```
CREATE TABLE BAN (
    MA_HANG nvarchar(3) NOT NULL ,
    STT_DL int NOT NULL ,
    NGÀY_BAN smalldatetime NOT NULL,
    SOLG_BAN int NOT NULL ,
    TRIGIA_BAN float NOT NULL )
```

--Tạo bảng MUA-----

```
CREATE TABLE MUA (
    MA_HANG nvarchar(3) NOT NULL ,
    STT_DL int NOT NULL ,
    NGÀY_MUA smalldatetime NOT NULL ,
    SOLG_MUA int NOT NULL ,
    TRIGIA_MUA float NOT NULL )
```

15

## Chỉnh sửa bảng

- Các bảng sau khi tạo ra có thể được chỉnh sửa lại.
- Các lựa chọn chỉnh sửa có thể:
  - Các trường, kiểu dữ liệu, ràng buộc,...
  - Thêm các trường, xóa bớt trường,...
  - Tên bảng

16

## Chỉnh sửa bảng với T-SQL

- Cú pháp:
 

```
ALTER TABLE <Table Name>
    {[Alter colum <Column Name> <New data type>] |
    ADD [Column Name <Data type>] |
    DROP COLUMN <Column Name>}
```
- Ví dụ:
 

```
ALTER TABLE HANGHOA
    ADD DVT CHAR (15)
```

17

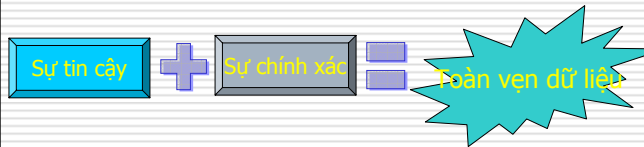
## Xóa bảng T-SQL

- Cú pháp: Drop table <Table Name>
- Ví dụ:
 

```
Drop table HANGHOA
```

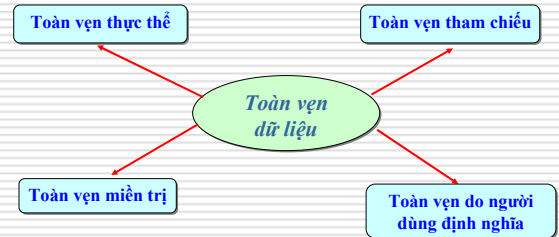
18

## Toàn vẹn dữ liệu



19

## Toàn vẹn dữ liệu



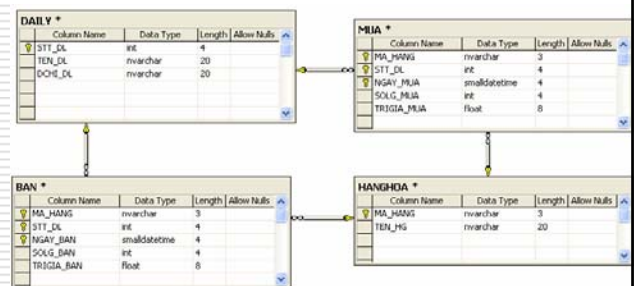
20

## Một số công cụ cho phép đặt RBTV

| Integrity type         | SQL Server Tools  |
|------------------------|---|
| Entity Integrity       | 1. PRIMARY KEY constraint<br>2. UNIQUE constraint<br>3. IDENTITY columns                          |
| Domain Integrity       | 1. DEFAULT definition<br>2. FOREIGN KEY constraint<br>3. CHECK constraint<br>4. NOT NULL property |
| Referential Integrity  | 1. FOREIGN KEY constraint<br>2. CHECK constraint  |
| User-defined Integrity | 1. Rules<br>2. Stored Procedures<br>3. Triggers   |

21

## Diagram chi tiết CSDL HANGHOA



22

## Tạo khóa chính với T-SQL (Primary Key)

- Tạo khóa chính khi tạo bảng  

```
CREATE TABLE DAILY
(
    STT_DL int NOT NULL PRIMARY KEY,
    TEN_DL nvarchar (20) NOT NULL,
    DCHI_DL nvarchar (20) NOT NULL
)
```
- Tạo khóa chính với EM

23

## Tạo khóa chính với T-SQL (Primary Key)

- Tạo khóa chính khi bảng đã tồn tại  

```
--Tạo khóa chính cho bảng DAILY
ALTER TABLE DAILY ADD CONSTRAINT PK_DAILY
PRIMARY KEY (STT_DL)

--Tạo khóa chính cho bảng MUA
ALTER TABLE MUA ADD CONSTRAINT PK_MUA
PRIMARY KEY (MA_HANG, STT_DL, NGÀY_MUA)
```

24

## Tạo khóa ngoại với T-SQL (Forgein Key)

- Tạo khóa ngoại khi tạo bảng

```
CREATE TABLE MUA
(
    MA_HANG nvarchar(3) NOT NULL,
    STT_DL int NOT NULL,
    NGAY_MUA smalldatetime NOT NULL,
    SOLG_MUA int NOT NULL,
    TRIGIA_MUA float NOT NULL,
    CONSTRAINT PK_MUA PRIMARY KEY(MA_HANG, STT_DL, NGAY_MUA),
    CONSTRAINT FK_MUA_HANGHOA FOREIGN KEY(MA_HANG)
        REFERENCES HANGHOA(MA_HANG) ON UPDATE CASCADE,
    CONSTRAINT FK_MUA_DAILY FOREIGN KEY(STT_DL)
        REFERENCES DAILY(STT_DL) ON UPDATE CASCADE
)
```

25

## Tạo khóa ngoại với T-SQL (Forgein Key)

- Tạo khóa ngoại khi tạo bảng

```
CREATE TABLE BAN
(
    MA_HANG nvarchar(3) NOT NULL,
    STT_DL int NOT NULL,
    NGAY_BAN smalldatetime NOT NULL,
    SOLG_BAN int NOT NULL,
    TRIGIA_BAN float NOT NULL,
    CONSTRAINT PK_BAN PRIMARY KEY(MA_HANG, STT_DL, NGAY_BAN),
    CONSTRAINT FK_BAN_HANGHOA FOREIGN KEY(MA_HANG)
        REFERENCES HANGHOA(MA_HANG) ON UPDATE CASCADE,
    CONSTRAINT FK_BAN_DAILY FOREIGN KEY(STT_DL)
        REFERENCES DAILY(STT_DL) ON UPDATE CASCADE
)
```

26

## Tạo khóa ngoại (Forgein Key) với T-SQL

- Tạo khóa ngoại khi bảng đã tồn tại

```
ALTER TABLE BAN ADD
    CONSTRAINT FK_BAN_DAILY FOREIGN KEY(STT_DL)
        REFERENCES DAILY(STT_DL) ON UPDATE CASCADE,
    CONSTRAINT FK_BAN_HANGHOA FOREIGN KEY(MA_HANG)
        REFERENCES HANGHOA(MA_HANG)
            ON UPDATE CASCADE
```

27

## Tạo khóa ràng buộc duy nhất (Unique Constraint) với T-SQL

- Tạo ràng buộc duy nhất khi tạo bảng

```
CREATE TABLE DAILY
(
    STT_DL int NOT NULL PRIMARY KEY,
    TEN_DL nvarchar(20) NOT NULL,
    DCHI_DL nvarchar(20) NOT NULL,
    SO_DT nvarchar(15) NOT NULL UNIQUE
)
```

28

## Tạo khóa ràng buộc duy nhất (Unique Constraint) với T-SQL

- Tạo ràng buộc duy nhất khi bảng đã tồn tại

```
CREATE TABLE DAILY
(
    STT_DL int NOT NULL PRIMARY KEY,
    TEN_DL nvarchar(20) NOT NULL,
    DCHI_DL nvarchar(20) NOT NULL,
    SO_DT nvarchar(15) NOT NULL
)
-----
ALTER TABLE DAILY ADD CONSTRAINT AK_DAILY
    UNIQUE(SO_DT)
```

29

## Tạo ràng buộc kiểm tra (Check Constraint) với T-SQL

Tạo Check Constraint khi bảng đã tồn tại

```
CREATE TABLE BAN (
    MA_HANG nvarchar(3) NOT NULL,
    STT_DL int NOT NULL,
    NGAY_BAN smalldatetime NOT NULL,
    SOLG_BAN int NOT NULL,
    TRIGIA_BAN float NOT NULL)

---Create Check Constraint-----
ALTER TABLE BAN ADD CONSTRAINT CHECK_BAN
    CHECK(NGAY_BAN <= GETDATE())
```

30

## Tạo ràng buộc giá trị mặc định (Default Constraint) với T-SQL

Tạo Default Constraint khi bảng đã tồn tại

```
CREATE TABLE BAN (  
    MA_HANG nvarchar(3) NOT NULL ,  
    STT_DL int NOT NULL ,  
    NGÀY_BAN smalldatetime NOT NULL ,  
    SOLG_BAN int NOT NULL ,  
    TRIGIA_BAN float NOT NULL)
```

---Create Default Constraint---

```
ALTER TABLE BAN ADD CONSTRAINT DEFAULT_BAN  
    DEFAULT GETDATE() FOR NGÀY_BAN
```

31

## Tạo ràng buộc giá trị mặc định (Default Constraint) với T-SQL

Tạo Default Constraint khi tạo bảng

```
CREATE TABLE BAN  
(  
    MA_HANG nvarchar(3) NOT NULL ,  
    STT_DL int NOT NULL ,  
    NGÀY_BAN smalldatetime NOT NULL  
        DEFAULT GETDATE(),  
    SOLG_BAN int NOT NULL ,  
    TRIGIA_BAN float NOT NULL  
)
```

32

## Biến (1)

□ Khai báo:

```
declare @variable_name datatype
```

□ Gán trị cho biến:

```
select @variable = expression
```

```
[from ... [where ...]]
```

Hoặc

```
set @variable = expression
```

□ Truy xuất giá trị:

```
select @variable
```

33

## Biến (2)

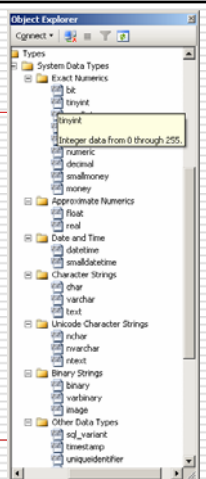
```
DECLARE @TEN varchar(30),  
        @SOLG int
```

```
SELECT @TEN=ProductName,  
        @SOLG=QuantityPerUnit  
FROM Products  
WHERE ProductID=13
```

```
PRINT @TEN +' '+ @SOLG
```

34

## Kiểu dữ liệu



## Các biến toàn cục

Biến toàn cục bắt đầu bằng 2 ký tự @.

Ta có thể truy xuất giá trị của các biến này bằng SELECT

Ví dụ: SELECT @@VERSION AS

SQL\_SERVER\_VERSION

36

## Danh sách biến toàn cục (1)

| Các biến       | Ý nghĩa  |
|----------------|--|
| @@CONNECTIONS  | Số các kết nối đến máy chủ từ lần khởi động cuối.  |
| @@CPU_BUSY     | Số ms hệ thống đã xử lý từ khi SQL Server được khởi động.                                |
| @@CURSOR_ROWS  | Số bản ghi trong cursor mở gần nhất.   |
| @@DATEFIRST    | Giá trị hiện tại của tham số trong lệnh SET DATEFIRST quyết định ngày đầu tiên của tuần. |
| @@ERROR        | Mã lỗi của lỗi xảy ra gần nhất   |
| @@FETCH_STATUS | 0 nếu trạng thái lần truy xuất cuối thành công.<br>-1 nếu có lỗi                         |

37

## Danh sách các biến toàn cục (2)

| Các biến           | Ý nghĩa   |
|--------------------|---|
| @@IDENTITY:        | Giá trị identity gần nhất được sinh ra            |
| @@LANGUAGE:        | Tên của ngôn ngữ đang được sử dụng.               |
| @@MAX_CONNECTIONS: | Số kết nối tối đa có thể.                         |
| @@ROWCOUNT         | Số bản ghi bị tác động bởi câu lệnh SQL gần nhất. |
| @@SERVERNAME:      | Tên của máy chủ                                   |
| @@SERVICENAME:     | Tên của dịch vụ SQL trên máy chủ                  |

38

## Thuộc tính Identity

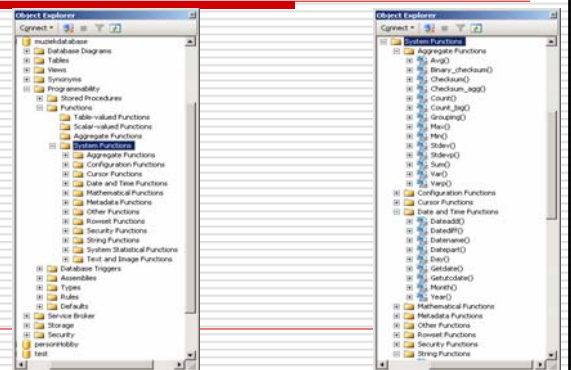
- Tạo cột tự tăng trong 1 bảng
- Cú pháp:  
IDENTITY [(seed, increment)]

```
CREATE TABLE new_employees (
    emp_id int primary key IDENTITY(1,1),
    fname varchar(20),
    middle char(1),
    lname varchar(30)
)
```

```
INSERT new_employees (fname, middle, lname)
VALUES ('Karin', 'F', 'Josephs')
```

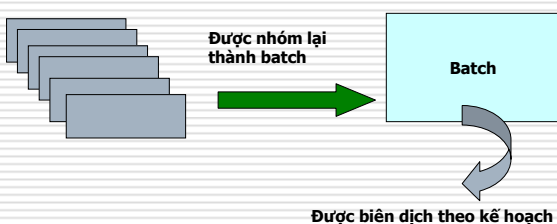
39

## Các hàm hệ thống



## Lô (Batch)

Các lệnh SQL



41

## Xử lý theo lô

- Là quá trình trong đó một tập lệnh được xử lý cùng một khoảng thời gian.

```
Use Pubs
Select * from authors
Update authors
set phone= '890 451-7366'
where au_lname= 'White'
Go
```

→ Lệnh báo hiệu kết thúc lô

42

## Cấu trúc điều kiện

```
IF Boolean_expression
{Transact-sql_statement | statement_block}
[ELSE [Boolean_expression]
{Transact-sql_statement | statement_block}]
```

```
statement_block:
BEGIN
Transact-SQL_statements
END
```

43

## Thí dụ cấu trúc điều kiện (1)

```
CREATE TABLE product
( prod_nr      int              NOT NULL
  CONSTRAINT pk_product PRIMARY KEY (prod_nr),
  name         varchar(30)     NOT NULL,
  price        money           NOT NULL,
  type         varchar(30)     NOT NULL
)
```

```
INSERT product (prod_nr, name, price, type)
VALUES (1, 'tv', 500, 'electronics')
```

```
INSERT product (prod_nr, name, price, type)
VALUES (2, 'radio', 100, 'electronics')
```

```
INSERT product (prod_nr, name, price, type)
VALUES (3, 'ball', 100, 'sport')
```

44

## Thí dụ cấu trúc điều kiện (2)

### □ Thí dụ 1

- Khai báo 1 biến và gán trị là 100
- Hiển thị thông điệp "Giá trị trung bình của sản phẩm điện tử (electronic) là lớn hơn 100' nếu điều này là đúng trong bảng product.

### □ Thí dụ 2

- Hiển thị thông điệp: 'Có radio trong kho' nếu điều này là đúng trong bảng product

45

## Cấu trúc lựa chọn

### Cú pháp:

```
CASE expression
WHEN expression1 THEN expression1
[
[WHEN expression2 THEN expression2]
[...]]
[ELSE expression]
END
```

46

## Vòng lặp

```
WHILE Boolean_expression
{Transact-sql_statement | statement_block}
```

Thí dụ: Cập nhật giá của tất cả các sản phẩm lên 5% cho tới khi giá trung bình các sản phẩm lớn hơn 400.

47

## Bài tập

- Cho một số tài khoản (account number) gồm 9 chữ số chẳng hạn 972428577.

Viết một đoạn mã T-SQL kiểm tra số tài khoản này có hợp lệ theo nguyên tắc kiểm tra chia dư 11 ('modulo11-check')

Thí dụ:

972428577 là hợp lệ vì:

$(9*9 + 8*7 + 7*2 + 6*4 + 5*2 + 4*8 + 3*5 + 2*7 + 1*7) \% 11 = 0$

- Viết chương trình đổi thứ trong tuần từ tiếng Anh sang tiếng Việt (dùng CASE)

48