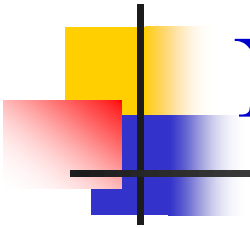
BÀI 3

TOÀN VẸN DỮ LIỆU



Nội dung

1. Giới thiệu Toàn vẹn dữ liệu
2. Hàm và biểu thức trong T-SQL
3. Ngôn ngữ DDL
4. Ngôn ngữ DML và DCL
5. Thực thi câu lệnh T-SQL

Giới thiệu 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 không đúng mà đã được lưu trữ trong CSDL thì gọi là vi phạm TVDL.
- Các loại ràng buộc toàn vẹm: Not Null, Default, Identity, Constraints, Rule, Triggers, Indexs.
- Định nghĩa tàng buộc:

  Create Table…: Định nghĩa trong lúc thiết kế.

  Alter Table…: Định nghĩa trong khi hiệu chỉnh bảng.
- Kiểm tra /xem các toàn vẹn dữ liệu:

  Sp_HelpConstraint <Tên Table>
- Xóa toàn vẹn dữ liệu:

  ALTER TABLE <TenTable>

  DROP CONSTANT <Ten Constrant>

3

Định nghĩa NULL/NOT NULL

- Giá trị 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.
- Ví dụ: số phone của khách hàng hiện tại chưa có, nhưng có thể sẽ được bổ sung này. Số phone sẽ có giá trị là null.
- Để kiểm tra giá trị null trong lệnh truy vấn, phải dùng toán tử IS NULL hay IS NOT NULL trong mệnh đề WHERE.
- Mặc định các cột hay kiểu dữ liệu của người dùng thường không có giá trị NULL.

4

Định nghĩa NULL/NOT NULL

- **Ví dụ:**

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

Tạo bảng - CREATE TABLE

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

- Tạo giá trị gia tăng duy nhất cho 1 cột, và cột này thường được dùng khoá chính cho bảng.
- 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.
- *Seed: là giá trị đầu tiên được tạo*.
- *Increment:là bước tăng để tạo ra giá trị kế tiếp*.
- Giá trị mặc định thường là (1,1).
- Giá trị của cột Identity sẽ tự động tăng.

Tạo bảng - CREATE TABLE

Cú pháp : Tạo cột có giá trị phát sinh tự động

```
CREATE TABLE <Table_Name>
(<Column_Name>  <Data_Type>
INDENTITY(seed[, Increment]) NOT NULL….)
```

Bật chế độ chèn dữ liệu cho cột Identity cho bảng

```
SET IDENTITY_INSERT <TenTable> ON
```

Ví dụ

```
CREATE TABLE Nhacungcap
(MaNCC int Identity NOT NULL Primary key, TenNCC Varchar(25), Diachi Varchar(40))
Set Identity_Insert Nhacungcap ON
INSERT Nhacungcap (MaNCC, TenNCC, Diachi) VALUES (2,‘Minh’, ‘Go Vap’)
```

Tạo bảng - CREATE TABLE

Tạo ràng buộc Default

- Cú pháp: DEFAULT *constant_expression*
- Default dùng để xác định giá trị "sẵn trước" 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.
- *constant_expression: c*hỉ có giá trị hằng như chuỗi ký tự, hàm hệ thống, hay giá trị NULL.

CREATE TABLE <TableName>
(<Column_Name> <DataType>
DEFAULT (<expresion>))

ALTER TABLE *tablename*
ADD [CONSTRAINT *constraintname*]
DEFAULT *expression* FOR *columnname*

Tạo bảng - CREATE TABLE

Ví dụ 1

```
CREATE TABLE HoaDon
(MaHD char(5), LoaiHD Char(1) DEFAULT 'X', NgayLap
DateTime NOT NULL)
```

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

Hay

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

Sử dụng defaults

- Sau khi được tạo DEFAULT, 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 được vào cột của bảng hay kiểu dữ liệu người dùng.

  sp_unbindefault object_name
  [, FUTUREONLY]

Sử dụng defaults

- **Ví dụ default và cột của bảng**

```
USE pubs
GO
CREATE DEFAULT phonedflt AS 'unknown'
GO
sp_bindefault phonedflt, 'authors.phone'
GO
sp_unbindefault 'authors.phone'
GO
DROP DEFAULT phonedflt
```

Sử dụng defaults

- **Ví dụ default và kiểu dữ liệu người dùng**

  sp_addType typCity, ‘char(15)’

  CREATE DEFAULT defCity AS 'Oakland'

  sp_bindefault defCity, 'typCity'

  sp_binddefault ‘defCity’, ‘customer.cCity’

- **Ví dụ 3**

```
CREATE TABLE jobs
(       job_id smallint IDENTITY(1,1) ,
        job_desc varchar(50) NOT NULL
        DEFAULT 'New Position - title not formalized yet'
)
```

Tạo bảng - CREATE TABLE

Xoá Default - mặc định

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

Hay

ALTER TABLE <TenTable>
DROP CONSTRAINT <TenDefault>

- Lệnh drop có thể xóa cùng lúc nhiều default
- Ví dụ:

DROP DEFAULT phonedflt
DROP DEFAULT Ngay_DF

Hay

ALTER TABLE Hoadon
DROP CONSTRAINT Ngay_DF

Ràng buộc Check

Cú pháp:

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

- *Qui định nhập dữ liệu phải thỏa mãn điều kiện của biểu thức check_logic.*
- *Check_logic:* biểu thức với các toán tử số học, toán tử quan

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

Ràng buộc Check

- **Ví dụ 1:**

  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‘))

- **Ví dụ 2:**

  ALTER TABLE Nhanvien

  ADD CONSTRAINT NV_HSPC

  CHECK (HSPC>=0.1 AND HSPC<0.5)

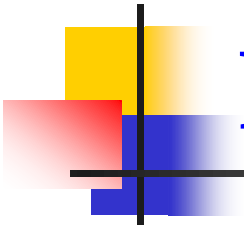
Ràng buộc Check

```
CREATE TABLE Orders (
OrderID int IDENTITY (1, 1) NOT NULL,
CustomerID nchar (5) CHECK (CustomerID LIKE '[A-Z][A-Z][A-Z][A-Z][A-Z]'),
EmployeeID int NULL, OrderDate datetime NULL
CHECK (OrderDate BETWEEN '01/01/70' AND GETDATE()),
RequiredDate datetime NULL, ShipVia int NULL
CHECK (ShipVia IN (1, 2, 3, 4)),
Freight money NULL CHECK (Freight>=0),
ShipCountry nvarchar (15),
CHECK (RequiredDate>OrderDate))
```

Ràng buộc Check

- Ví dụ 4:

```
CREATE TABLE PHANCONG(
        ma_nvien CHAR(9) NOT NULL,
        soda INT NOT NULL,
        thoigian DECIMAL(3,1) NOT NULL,
        PRIMARY KEY (ma_nvien, soda),
        FOREIGN KEY (ma_nvien) REFERENCES NHANVIEN),
        FOREIGN KEY (soda) REFERENCES DEAN(mada),
        CHECK (thoigian ≥ 0))
```



Rule

- Định nghĩa các qui tắc hợp lệ mà có thể kết buộc vào các cột của bảng hay các kiểu dữ liệu do người dùng định nghĩa.
- Rule được tạo nên chính nó trước khi kết buộc vào đối tượng khác
- Định nghĩa 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]

Rule

- Ví dụ:

```
CREATE RULE ActiveDate AS
        @Date Between ’01/01/70’ AND Getdate()
sp_bindrule ActiveDate, ‘Orders.OrderDate’
```

- Chú ý:
  - Futureonly chỉ định các cột tồn tại sẵn mà có dùng kiểu dữ liệu này thì không thể kế thừa Rule mới. Chỉ sử dụng với kiểu dữ liệu, cột thì không.

Các ràng buộc - Constraints

< 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* **)**

}

Ràng buộc Primary Key

- SQL Server tự động tạo một chỉ mục cho bảng ứng với các cột tham gia ràng buộc Primary key.
- Mỗi bảng chỉ có thể có duy nhất 1 ràng buộc primary key.
- Ràng buộc Primary key gồm một hay nhiều cột dùng để nhận diện các record, giá trị của primary key không được phép trùng nhau và không chứa giá trị Null.
- Chỉ mục sẽ được tự động tạo ra khi có khai báo 1 ràng buộc primary key.
- Chỉ mục do primary key tạo ra mặc định thường là clustered

Ràng buộc Primary Key

```
CREATE TABLE TableName
   (columname datatype [,…],[CONSTRAINT constraint_name]
   PRIMARY KEY [CLUSTERED|NONCLUSTERED]
   {(column [ASC |DESC][,…,n])}
   [WITH FILLFACTOR = fillfactor]
   [ON {filegroup|DEFAULT}]
```

```
ALTER TABLE TableName
   ADD [CONSTRAINT constraint_name]
   PRIMARY KEY {(column [ASC |DESC][,…,n])}
   [ON {filegroup|DEFAULT}]
```

Ràng buộc Primary Key

- **Ví dụ 1**

```
CREATE TABLE jobs
(
        job_id smallint PRIMARY KEY
        CLUSTERED NOT NULL,
        job_desc varchar(50) NOT NULL
        DEFAULT 'New Position - title not
        formalized yet'
)
```

Ràng buộc Primary Key

Ví dụ 2: Định nghĩa mức cột

```
CREATE TABLE Events (
EventID int NOT NULL PRIMARY KEY,
EventTitle nvarchar (100) NULL ,
EventDescription ntext NULL , ……. )
```

Ví dụ 3: Định nghĩa mức bảng

```
CREATE TABLE Orders (
OrderID int IDENTITY (1, 1) NOT NULL,
CustomerID nchar (5), …….
PRIMARY KEY NONCLUSTERED (OrderID) WITH
   FILLFACTOR=90 )
```

Ràng buộc Primary Key

Ví dụ 4:

```
CREATE TABLE Table3 (
    col1 int NOT NULL,
    col2 varchar (100) )
```

Thêm ràng buộc khóa chính

```
ALTER TABLE Table3
    ADD CONSTRAINT Table3_PK
    PRIMARY KEY (Col1)
    EXEC Sp_helpconstraint Table3
```

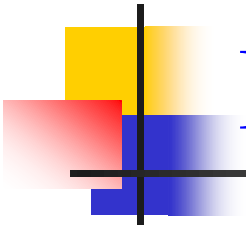
Ràng buộc Primary Key

Xóa một Primary key Constraint

```
ALTER TABLE Table3
    DROP CONSTRAINT Table3_PK
```

Lưu ý:

- Không thể xóa một Primary key constraint nếu nó được tham chiếu bởi Foreign key của một bảng khác, muốn xóa phải xóa Foreign key trước



Ràng buộc Unique

- Dùng để đảm bảo không có giá trị trùng ở các cột.
- Một cột hay sự kết hợp giữa các cột vốn không phải là khóa chính.
- Chấp nhận một hàng chứa giá trị Null.
- Một bảng có thể có nhiều Unique constraint.

So sánh Unique và Primary key

- Ràng buộc Primary key gồm một hay nhiều cột dùng để nhận diện các record, giá trị của primary key không được phép trùng nhau và không chứa giá trị Null.
- Ràng buộc UNIQUE được dùng cho các cột không phải là primary key.
- Ràng buộc UNIQUE tương tự như PRIMARY KEY nhưng nó cho phép 1 hàng được quyền có giá trị NULL
- Một bảng có thể có nhiều ràng buộc unique nhưng chỉ có 1 ràng buộc primary key mà thôi.
- Chỉ mục do primary key tạo ra mặc định thường là clustered
- Chỉ mục do unique tạo ra mặc định thường là nonclustered

Ràng buộc Primary Key

```
CREATE TABLE TableName
   (columname datatype [,…],[CONSTRAINT constraint_name]
   UNIQUE [CLUSTERED|NONCLUSTERED]
   {(column [ASC |DESC][,…,n])}
   [WITH FILLFACTOR = fillfactor]
   [ON {filegroup|DEFAULT}]
```

```
ALTER TABLE TableName
   ADD [CONSTRAINT constraint_name]
   UNIQUE {(column [ASC |DESC][,…,n])}
   [ON {filegroup|DEFAULT}]
```

Ràng buộc Primary Key

- **Ví dụ 1**

```
CREATE TABLE jobs
(
        job_id smallint UNIQUE
        CLUSTERED NOT NULL,
        job_desc varchar(50) NOT NULL
        DEFAULT 'New Position - title not
        formalized yet'
)
```

Ràng buộc Primary Key

Ví dụ 2: Định nghĩa mức cột

```
CREATE TABLE Events (
EventID int NOT NULL UNIQUE,
EventTitle nvarchar (100) NULL ,
EventDescription ntext NULL , ……. )
```

Ví dụ 3: Định nghĩa mức bảng

```
CREATE TABLE Orders (
OrderID int IDENTITY (1, 1) NOT NULL,
CustomerID nchar (5), …….
UNIQUE NONCLUSTERED (OrderID) WITH
   FILLFACTOR=90 )
```

Ràng buộc Primary Key

Ví dụ 4:

```
CREATE TABLE Table3 (
    col1 int NOT NULL,
    col2 varchar (100) )
```

Thêm ràng buộc unique

```
ALTER TABLE Table3
    ADD col3 CONSTRAINT Table3_Unique UNIQUE
    EXEC Sp_helpconstraint Table3
```

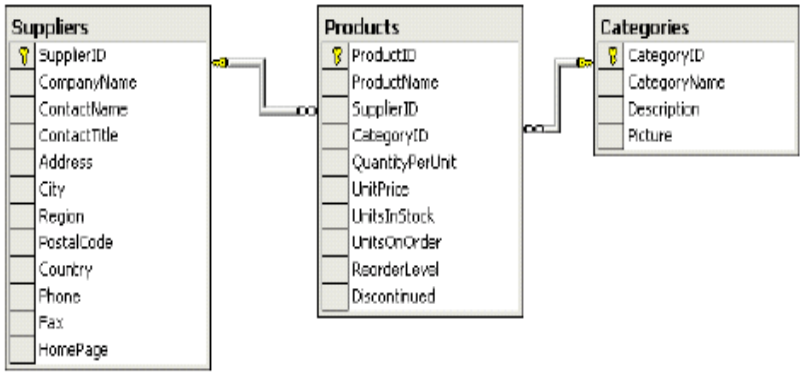
Ràng buộc Primary Key

Xóa một Unique Constraint

```
ALTER TABLE Table3
    DROP CONSTRAINT Table3_Unique
```

Ràng buộc Foreign key

- Quan hệ chỉ có thể được tạo ra giữa các bảng trong cùng 1 CSDL và trên cùng 1 server.
- Khoá ngoại chỉ có thể tham chiếu đến một cột sau trong bảng chính:
  - Là 1 cột hay 1 phần của khoá
  - Là cột có ràng buộc unique
  - Là cột có chỉ mục unique



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

Ràng buộc Foreign key

Định nghĩa FOREIGN KEY CONSTRAIT khi tạo bảng

CREATE TABLE TableName

(columnName datatype [,…],

[CONSTRAINT constraintName]

FOREIGN KEY[(column[,..n])]

REFERENCES *ref_table* [**(** **r***ef_column [,..n])]***)**]

[ON DELETE { CASCADE | NO ACTION }]

[ON UPDATE { CASCADE | NO ACTION }]

[NOT FOR REPLICATION]

Ràng buộc Foreign key

- ON UPDATE|DELETE {CASCADE | NO ACTION}
- Xác định hành động cần phải thực hiện cho 1 hàng trong bảng đang tạo nếu hàng đó có quan hệ tham chiếu và hàng tham chiếu bị xoá khỏi bảng chính. Mặc định là NO ACTION.
- CASCADE: dùng để xác định là hàng sẽ bị cập nhật/xoá khỏi bảng tham chiếu nếu hàng đó bị cập nhật/xóa khỏi bảng chính
- NO ACTION: SQL Server sẽ đưa ra thông báo lỗi và việc xoá hàng trên bảng chính sẽ bị từ chối.

Ràng buộc Foreign key

- **Ví dụ 1**

```
CREATE TABLE PhongBan
   (        Mapb int,
            TenPb varchar(30),
            MaTp int REFERENCES Nhanvien(Manv)
   )
```

Ràng buộc Foreign key

- **Ví dụ 2**

```
CREATE TABLE NHANVIEN (
        manv    CHAR(9) NOT NULL,
        honv VARCHAR(15)    NOT NULL,
        tennv VARCHAR(15)   NOT NULL,
        ngsinh DATETIME, diachi VARCHAR(30),
        phai CHAR(1), ma_nql CHAR(9),
        phg INT  NOT NULL,
        CONSTRAINT Nv_PK PRIMARY KEY (manv),
        CONSTRAINT Nv_fk FOREIGN KEY (phg)
        REFERENCES PHONGBAN(mapb))
```

Ràng buộc Foreign key

Định nghĩa FOREIGN KEY CONSTRAIT khi bảng đã tồn tại

ALTER TABLE TableName

[WITH CHECH | WITH NOCHECK] ADD

[CONSTRAINT constraintName]

FOREIGN KEY[(column[,..n])]

REFERENCES *ref_table* [**(** r*ef_column [,..n])]***)**]

[ON DELETE { CASCADE | NO ACTION }]

[ON UPDATE { CASCADE | NO ACTION }]

[NOT FOR REPLICATION]

Ràng buộc Foreign key

- WITH CHECK: trước khi tạo ràng buộc, SQL Server sẽ kiểm tra dữ liệu hiện có vi phạm ràng buộc hay không, nếu có sẽ không tạo constraint.
- WITH NOCHECK: tạo constraint mà không cần kiểm tra dữ liệu hiện có có vi phạm ràng buộc hay không.

Ràng buộc Foreign key

- **Ví dụ 1**

```
ALTER TABLE Nhanvien
    ADD CONSTRAINT Cv_FK Foreign key (Macv)
    REFERENCES Chucvu(Macv)
)
```

Các mức ràng buộc

- Có thể tạo ràng buộc theo 2 mức :
  - Mức cột (Column level)
  - Mức bảng (Table level)
- Ràng buộc mức bảng:

< table_constraint >::= [CONSTRAINT *constraint_name*]
{ [{ PRIMARY KEY | UNIQUE }[CLUSTERED | NONCLUSTERED]
{ **(** *column* [ASC | DESC] [**,**...*n*] **)** }]
| FOREIGN KEY [**(** *column* [**,**...*n*] **)**]
EFERENCES *ref_table* [**(** *ref_column* [**,**...*n*] **)**]
[ON DELETE { CASCADE | NO ACTION }]
[ON UPDATE { CASCADE | NO ACTION }]
[NOT FOR REPLICATION]
| CHECK [NOT FOR REPLICATION]
(*search_conditions* **)**}

Các mức ràng buộc

- **Ví dụ về ràng buộc mức bảng**
- Tạo 1 ràng buộc khoá chính ở mức bảng

```
CREATE TABLE cthoadon
(    sohd int NOT NULL,
     MaHang char(4) NOT NULL,
     SoLuong int NOT NULL,
     DonGia money,
     CONSTRAINT pk_ctHoadon primary key
     clustered (sohd,MaHang)
)
```

Thủ tục lưu trữ hệ thống

sp_help- System stored procedure

- Để kiểm tra xem bảng đã được tạo hay chưa?

sp_help table_name

- Để kiểm tra xem kiểu dữ liệu của người dùng đã được tạo hay chưa?

sp_help datatype_name

Xem Constraints

- **Viewing Constraints**
  - Sp_helpConstraint Events
- **Verify constraints by inserting data**
  - INSERT Events DEFAULT VALUES

	EventID	EventType	EventTitle	EventDescription	EventLanguage
1	1	Party	NULL	NULL	NULL

EventDate	EventEndDate	EventCreator
2004-02-15 01:28:00	2004-02-16 01:28:00	sa

  - SELECT * FROM Events

Bài tập

Example

a) Tạo Table có khóa chính

CREATE TABLE KhachHang

(Makh char(5), Tenkh Varchar(40), DiaChi Varchar(50), DienThoai Nvarchar(10) CONSTRAINT Makh_pk Primary key(Makh))

b) Tạo Table có khóa ngoại

CREATE TABLE HoaDon

(Mahd Char(5), NgayLap Datetime, Makh Char(5) CONSTRAINT Mahd_pk Primary key(Mahd)

CONSTRAINT Makh_fk Foreign key References KhachHang (Makh))

Modifyling Table_Defining Constraints

Example

a) ALTER TABLE Sanpham

ADD CONSTRAINT Masp_pk Primary key(Masp)

b) ALTER TABLE ChiTietHoaDon

ADD CONSTRAINT Masp_Mahd_pk Primary key(Mahd,Masp)

c) ALTER TABLE ChiTietHoaDon

ADD CONSTRAINT Masp_fk Foregin key (Masp) References Sanpham(Masp)

d) ALTER TABLE ChiTietHoaDon

ADD CONSTRAINT Mahd_fk Foregin key(Mahd) References HoaDon(Mahd)

Xóa Constraints

- **Viewing Constraints**

  sp_helpconstraint *tablename*

- **Dropping Constraints**

  ALTER TABLE *tablename*
  DROP [CONSTRAINT] *constraintname*

- **Disabling Constraints**

  ALTER TABLE *tablename*
  NOCHECK CONSTRAINT {ALL | *constraintname* [,...]}

Cập nhập nội dung Table

Cú pháp: Thêm dòng

```
INSERT [INTO] <table_name> VALUES <values>
```

Cú pháp: Thay đổi dữ liệu các dòng

```
UPDATE <table_name>
SET <column_name = value>
WHERE <condition>
```

Cú pháp: Xóa dòng

```
DELETE FROM <table_name> WHERE <condition>
```

Xem Tables

Cú pháp: Xem thông tin Table

```
sp_help <table_name>
```

Cú pháp: Xem dữ liệu Table

```
SELECT <select_list> FROM <table_name>
```

Xóa tables

Cú pháp

```
DROP TABLE <Table_Name>
```

Ví dụ

```
DROP TABLE Airlines_Master
```