



HỌC VIỆN KỸ THUẬT MẬT MÃ

KHOA CÔNG NGHỆ THÔNG TIN

Bài giảng: Các ràng buộc toàn vẹn



“Tri thức là chìa khóa mở cánh cửa tương lai”



Ràng buộc toàn vẹn (integrity constraints)

➤ Khái niệm

RBTV 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 dùng để duy trì tính nhất quán của dữ liệu, đảm bảo dữ liệu phù hợp với các quy định theo yêu cầu của bài toán. Ràng buộc giúp ngăn chặn dữ liệu không hợp lệ và chỉ cho phép dữ liệu hợp lệ được lưu vào database

➤ Ví dụ

- Ngày sinh của sinh viên phải nhỏ hơn ngày nhập học
- Điểm của sinh viên phải từ 0 đến 10
- 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



Phân loại ràng buộc toàn vẹn

***RBTV** có thể chia thành hai loại:*

- 1. Ràng buộc mức cột*
- 2. Ràng buộc mức bảng*

	Ràng buộc mức cột	Ràng buộc mức bảng
Ý nghĩa	Chỉ áp dụng ở mức cột	Có thể áp dụng ở một hoặc nhiều cột trong bảng
Phạm vi khai báo	Ngay sau khi khai báo cột (sau kiểu dữ liệu và phía trước dấu phẩy)	Khai báo sau khi đã khai báo sau cột cuối cùng của bảng và phía trước dấu đóng ngoặc)



Phân loại ràng buộc toàn vẹn

	Ràng buộc mức cột	Ràng buộc mức bảng
Các loại ràng buộc	NOT NULL, PRIMARY KEY, UNIQUE, FOREIGN KEY, CHECK, DEFAULT	PRIMARY KEY, UNIQUE, FOREIGN KEY, CHECK, DEFAULT
Khai báo	<ul style="list-style-type: none">- RB không đặt tên- RB có đặt tên	<ul style="list-style-type: none">- RB có đặt tên



Các loại ràng buộc toàn vẹn khi tạo bảng

Default

Not
null

Unique

Primary
Key

Foreign
Key

Check



Cơ sở dữ liệu sử dụng:

Sinhvien(MaSV, hotenSV, gioitinh, ngaysinh, quequan, lop)

Monhoc(MaMH, TenMH, DVHT)

Ketqua(MaSV, maMH, Diem)



Cách khai báo ràng buộc toàn vẹn

➤ Cú pháp

1. Khai báo ở mức cột

- ✓ Khai báo không đặt tên cho ràng buộc

<Tencot> <kieudulieu> <Loairangbuoc>

- ✓ Khai báo đặt tên cho ràng buộc

*<Tencot> <kieudulieu> constraint <tenrangbuoc>
<Loairangbuoc>*

2. Khai báo ở mức bảng

constraint <tenrangbuoc> <Loairangbuoc> (dscot)



Khi nào nên khai báo ràng buộc ở mức cột, mức bảng?

- 1. Khi ràng cột chỉ có tác động vào một cột nào đó thì nên dùng ràng buộc mức cột để tường minh*
- 2. Khi ràng buộc các cột trong bảng vào với nhau thì dùng ràng buộc mức bảng*
- 3. Khi đã tạo xong bảng, bắt buộc phải dùng ràng buộc mức bảng để thêm ràng buộc*



Tại sao nên đặt tên cho ràng buộc toàn vẹn

1. Lý do quan trọng nhất là sự vi phạm ràng buộc trong SQL Server
→ Có thể nhanh chóng xác định và sửa lỗi
2. Dễ dàng sửa đổi hoặc loại bỏ các ràng buộc



Thêm ràng buộc mới vào bảng

➤ Ý nghĩa

- ✓ Dùng để thêm ràng buộc (constraint) mới vào bảng đã được tạo sẵn
- ✓ Sử dụng từ khóa Alter table

➤ Cú pháp:

ALTER TABLE <tenbang>

ADD CONSTRAINT <tenrangbuoc> <loairangbuoc> (dscot)



DEFAULT

➤ Ý nghĩa

DEFAULT là ràng buộc mặc định. Khi nhập dữ liệu cho bảng mà cột đó không được cung cấp giá trị thì giá trị mặc định sẽ được sử dụng

Cách 1: Không đặt tên cho ràng buộc

```
create table MONHOC  
(  
MaMH char(4),  
TenMH nvarchar(40) default N'Giao trinh',  
DVHT tinyint  
)
```

Cách 2: Đặt tên cho ràng buộc

```
create table MONHOC  
(  
MaMH char(4),  
TenMH nvarchar(40) constraint default_MH default N'Tên môn học',  
DVHT tinyint,  
)
```



DEFAULT

Tạo ràng buộc sau khi tạo bảng

Tên bảng

```
alter table MONHOC add constraint default_MH default 'tên  
môn học' for TenMH
```

Tên ràng buộc

Tên cột muốn tạo ràng buộc

Chú ý: Không thể tạo ràng buộc mức bảng cho Default trong định nghĩa Bảng



CHECK

➤ Ý nghĩa

CHECK là ràng buộc kiểm tra. Yêu cầu cột tương ứng phải thỏa mãn một biểu thức logic.

Khai báo ràng buộc CHECK ở mức cột:

```
create table KETQUA  
(  
  MaSV char(3),  
  MaMH char(4),  
  Diem float check(diem>=0  
and diem <=10)  
)
```

```
create table KETQUA  
(  
  MaSV char(3),  
  MaMH char(4),  
  Diem float constraint C1  
check(diem>=0 and diem <=10)  
)
```



CHECK

Khai báo ràng buộc CHECK ở mức bảng:

-- khi muốn đặt tên cho ràng buộc và định nghĩa ràng buộc CHECK trên nhiều cột

+ Cách 1:

```
CREATE TABLE Persons (  
  ID int,  
  Name varchar(255),  
  Age int,  
  City varchar(255),  
  CONSTRAINT CHK_Person CHECK (Age>=18 AND City='Sandnes')  
)
```

+ Cách 2:

```
CREATE TABLE Persons (  
  ID int,  
  Name varchar(255),  
  Age int,  
  City varchar(255))
```

```
Alter Table Persons Add CONSTRAINT CHK_Person CHECK (Age>=18 AND  
City='Sandnes')
```



NOT NULL

➤ Thế nào là NULL ?

Giá trị NULL trong SQL là khái niệm được sử dụng để biểu diễn một giá trị bị khuyết thiếu. Một giá trị NULL trong một bảng là một giá trị trong một trường và bị bỏ trống.

➤ Ví dụ

```
create table SINHVIEN
(
  MaSV char(3),
  HotenSV nvarchar(30) not null,
  GioiTinh nchar(7),
  NgaySinh date not null,
  QueQuan nvarchar(20),
  Lop nvarchar(5)
)
```



UNIQUE

➤ Ý nghĩa

UNIQUE là ràng buộc duy nhất, yêu cầu các giá trị trong cột hoặc một tập hợp cột phải khác nhau, chắc chắn rằng không có các giá trị trùng lặp được nhập vào.

➤ Lưu ý

- ✓ Cột đặt ràng buộc UNIQUE có thể lưu trữ giá trị NULL nhưng chỉ lưu trữ được duy nhất một lần;
- ✓ Có thể tạo nhiều ràng buộc duy nhất trên cùng một bảng;
- ✓ Về mặt logic, UNIQUE + NOT NULL ~ PRIMARY KEY.



UNIQUE

➤ Ví dụ 1

```
create table SINHVIEN
(
  MaSV char(3),
  HotenSV nvarchar(30) not null,
  GioiTinh nchar(7) check (Gioitinh = N'Nam' or Gioitinh = N'Nữ'),
  Email varchar(50) unique,
  NgaySinh date not null,
  QueQuan nvarchar(20),
  Lop nvarchar(5),
)
```

Khai báo ràng buộc Unique ở mức nào ?

Câu hỏi: Làm thế nào để khai báo ràng buộc Unique ở mức cột có đặt tên

```
Email varchar(50) constraint Unique_SV unique
```



UNIQUE

- Ví dụ 2 Tạo ràng buộc UNIQUE trên hai cột của bảng

```
create table SINHVIEN
(
  MaSV char(3),
  HotenSV nvarchar(30) not null,
  GioiTinh nchar(7) check (Gioitinh = N'Nam' or Gioitinh = N'Nữ'),
  Email varchar(50),
  Phone varchar(11),
  NgaySinh date not null,
  QueQuan nvarchar(20),
  Lop nvarchar(5),
  constraint Unique_SV unique (Email,Phone)
)
```

←
Đây là ràng buộc mức cột hay mức bảng?



UNIQUE

➤ **Ví dụ 3** Cho bảng *SINHVIEN* sau:

```
create table SINHVIEN(  
MaSV char(3),  
HotenSV nvarchar(30) not null,  
GioiTinh nchar(7) check (Gioitinh = N'Nam' or Gioitinh = N'Nữ'),  
Email varchar(50),  
Phone tinyint,  
NgaySinh date not null,  
QueQuan nvarchar(20),  
Lop nvarchar(5),  
)
```

Yêu cầu: ***Tạo ràng buộc Unique cho hai cột Email và Phone***

```
alter table sinhvien add constraint Unique_SV  
unique(Email,Phone)
```



Khoá chính (PRIMARY KEY)

➤ Ý nghĩa

Primary key là khóa chính của bảng, là định danh duy nhất cho mỗi bản ghi trong bảng. Nó đòi hỏi cột (hoặc các cột) tạo thành khóa chính phải thỏa mãn hai điều kiện:

1. Chứa các giá trị không rỗng (không NULL)
2. Chỉ tồn tại dữ liệu duy nhất, không trùng lặp trong toàn bảng

➤ Lưu ý

- ✓ Mỗi bảng chỉ cho phép tối đa một khóa chính và theo nguyên tắc thiết kế, mỗi bảng đều cần có khóa chính
- ✓ Ngoài ra, thông qua khóa chính của bảng, mà các bảng khác có thể tham chiếu tới bằng khóa ngoại (FOREIGN KEY).



Cách tạo khóa chính

Ví dụ 1

```
create table SINHVIEN
(
  MaSV char(3) primary key,
  HotenSV nvarchar(30) not null,
  GioiTinh nchar(7) check (Gioitinh = N'Nam' or Gioitinh = N'Nữ'),
  NgaySinh date not null,
  QueQuan nvarchar(20),
  Lop nvarchar(5),
  Email varchar(50) unique
)
```



Cách tạo khóa chính

Yêu cầu: Cho Bảng *Ketqua*(*MaSV*, *MaMH*, *Diem*). Làm thế nào để tạo được khóa chính của bảng?

```
create table KETQUA  
(  
  MaSV char(3) primary key,  
  MaMH char(4) primary key,  
  Diem float check(diem >= 0  
and diem <= 10)  
)
```

```
create table KETQUA  
(  
  MaSV char(3),  
  MaMH char(4),  
  Diem float check(diem >= 0  
and diem <= 10),  
  constraint PK_KQ primary  
  key (MaSV, MaMH)  
)
```

Tên khóa chính

Hai trường cấu thành khóa chính



Cách tạo khóa chính

Yêu cầu: Tạo bảng *Ketqua*(*MaSV*, *MaMH*, *Diem*)

```
create table KETQUA  
(  
  MaSV char(3),  
  MaMH char(4),  
  Diem float check(diem>=0 and diem <=10)  
)
```

Làm thế nào để tạo được khóa chính sau
khi đã tạo bảng thành công?

```
alter table KETQUA  
add constraint PK_KQ primary key (MaSV, MaMH)
```



Khóa ngoại (FOREIGN KEY)

➤ Ý nghĩa

Foreign key là khóa ngoại được sử dụng để nối các bảng lại với nhau. Nó còn gọi là khóa tham chiếu.

Khóa ngoại đảm bảo việc nhập dữ liệu cho một cột (hoặc một số cột) thuộc tính nào đó từ một bảng phù hợp tham chiếu tới **khóa chính** ở một bảng khác.

➤ Điều kiện để tạo khóa ngoại

- ✓ Tham chiếu đến khóa chính của bảng khác, đảm bảo các ràng buộc UNIQUE, NOT NULL tương tự khóa chính.
- ✓ Khóa ngoại phải có cùng kiểu dữ liệu, cùng số lượng trường có sắp xếp tương ứng khóa chính.



Cách tạo khóa ngoại

➤ Tạo khóa ngoại trong lúc tạo bảng

```
create table KETQUA
```

Bảng chứa khóa ngoại

```
(  
MaSV char(3) foreign key references SINHVIEN
```

```
MaMH char(4) foreign key references MONHOC ,
```

*Bảng chứa
khóa chính*

```
Diem tinyint check(diem >= 0 and diem <= 10),
```

```
constraint PK_KQ primary key (MaSV, MaMH)
```

Tạo khóa chính

➤ Tạo khóa ngoại sau khi tạo bảng

```
alter table KETQUA add constraint FK_SV foreign key (MaSV) references  
SINHVIEN(MaSV)
```

```
alter table KETQUA add constraint FK_MH foreign key (MaMH) references  
MONHOC(MaMH)
```

*Bảng chứa khóa
chính*

Bảng chứa khóa ngoại

Tên khóa ngoại

Cột ràng buộc



Chú ý khi làm việc với khóa ngoại

1. Trường giá trị của bảng chứa khóa ngoại phải được lấy từ trường tương ứng của bảng chứa khóa chính

Ví dụ *Sinhvien(MaSV, hotenSV, gioitinh, ngaysinh, quequan,lop)*
Monhoc(MaMH, tenMH,DVHT)
Ketqua(MaSV,maMH,Diem)

2. Khi muốn xóa bảng hoặc dữ liệu ở bảng chứa khóa chính, phải xóa bảng hoặc dữ liệu ở bảng chứa khóa ngoại trước

Ví dụ

Nếu MaSV đã xuất hiện trong Kết quả (nghĩa là sinh viên đã có điểm thì không thể xóa bản ghi có MaSV tương ứng trong bảng Sinh viên được)



Xóa ràng buộc

➤ Cú pháp

`ALTER TABLE <tenbang> DROP CONSTRAINT <tenrangbuoc>`

➤ Ví dụ

```
alter table KETQUA drop constraint PK_KQ  
alter table SINHVIEN drop constraint Unique_SV
```

**Ràng buộc không
đặt tên???**



Thuộc tính IDENTITY

Ví dụ

```
create table SINHVIEN
(
  MaSV int identity primary key,
  HotenSV nvarchar(30) not null,
  GioiTinh nchar(7) check (Gioitinh = N'Nam' or Gioitinh = N'Nữ'),
  NgaySinh date not null,
  QueQuan nvarchar(20),
  Lop nvarchar(5),
  Email varchar(50) unique
)
```



Thuộc tính IDENTITY

Ý nghĩa

Identity cho phép tự động tạo ra một dãy số liên tục tăng dần và không trùng lặp. Trong rất nhiều trường hợp, cột IDENTITY cũng được dùng luôn làm khóa chính cho bảng.

Cú pháp

IDENTITY(*seed*, *increment*)

Trong đó *seed* là giá trị khởi tạo

increment là giá trị tăng cho mỗi lần

Khi không khởi tạo thì seed và increment đều có giá trị bằng 1

Ví dụ

ID int IDENTITY(1000,5)

ID int IDENTITY(-10,-5)



Một vài lưu ý với IDENTITY

- ✓ SQL Server không duy trì tính liên tục của dãy số IDENTITY, khi DELETE bản ghi thì khoảng trống sẽ xuất hiện. Ở lần INSERT sau đó một giá trị mới sẽ được dùng, thay vì giá trị bị xóa được dùng lại.
- ✓ Không thể thay đổi một cột có sẵn để nó trở thành cột IDENTITY, hoặc bỏ thuộc tính IDENTITY của nó. Khi edit bảng trong Management Studio và sửa một cột thành IDENTITY hoặc ngược lại, ở phía sau thực ra SQL Server xóa bảng và tạo một bảng mới.
- ✓ Khi cần INSERT vào bảng, có thể bỏ qua cột IDENTITY trong danh sách các cột, và giá trị của cột IDENTITY sẽ được tự động sinh ra (đó là mục đích chính khi tạo cột này)



Một vài lưu ý với IDENTITY

- ✓ Bản thân thuộc tính IDENTITY không đảm bảo các giá trị trong cột là duy nhất bởi vì có thể đặt lại seed với giá trị tùy ý (kể cả giá trị đã đặt trước đó) -> Khi cần đảm bảo tính duy nhất cần phải đặt nó là PRIMARY KEY hoặc là UNIQUE

Ví dụ

-- cho trường hợp muốn seed sắp tới là 1

```
DBCC CHECKIDENT('Tenbang', RESEED, 0)
```

-- cho trường hợp muốn seed sắp tới là 1001

```
DBCC CHECKIDENT('Tenbang', RESEED, 1000)
```

- ✓ Một bảng chỉ được phép có tối đa một cột IDENTITY



Bài tập thực hành

Yêu cầu:

1. Tạo cơ sở dữ liệu QLSV gồm 3 bảng

Sinhvien(*MaSV*, *hotenSV*, *gioitinh*, *ngaysinh*, *quequan*, *lop*)

Monhoc(*MaMH*, *TenMH*, *DVHT*)

Ketqua(*MaSV*, *maMH*, *Diem*)

2. Kiểu dữ liệu, RBTV tự đề xuất

3. Thêm cột Email vào bảng sinh viên, xóa cột lop, sửa kiểu dữ liệu cho trường Gioitinh

4. Tạo ràng buộc giữa các bảng (Ràng buộc khóa ngoại)

