

实验 8 数据库完整性

一、目的和要求

- (1) 掌握完整性约束的概念、实现方法和重要性
- (2) 了解表级约束、元组级约束、属性级约束的概念和区别
- (3) 掌握 SQL Server 中声明数据完整性和过程数据完整性的实现方法
- (4) 掌握 SQL Server 中规则的创建方法及调用系统存储过程将规则绑定到某个字段的方法

二、背景知识

(1) 数据库完整性概念：

数据库完整性就是确保数据库中的数据的一致性和正确性。SQL Server 提供了相应的组件以实现数据库的完整性。数据库完整性就是要防止数据库中存在不符合语义的数据，防止数据库中存在不正确的数据。为了保证数据库的完整性，DBMS 提供了定义、检查和控制数据完整性的机制，并能把用户定义的数据库完整性约束条件作为模式的一部分存入数据库中。它使用约束、默认、规则和触发器 4 种方法定义和实施数据库完整性功能。

(2) 数据完整性约束的分类：

数据完整性约束可以分为 3 类：①表级约束：若干元组间以及关系之间联系的数据约束。例如：选课表中，每个人最多能选 10 门课；学生表中，学生的学号必须唯一；选课表中的学号和课程号必须在学生表和课程表中存在。②元组级约束：同一个元组属性之间必须满足的约束条件。如学生表中年龄属性的值应该等于当前日期减去出生日期。③属性级约束：针对列的类型、取值范围、精度、排序等而制定的约束条件。例如：性别只能是‘男’或‘女’。

SQL Server 2005 支持以下 6 种约束即：主键约束(primary key)、唯一性约束(unique)、检查约束(check)、默认值约束(default)、外部键约束(foreign key)和空值(NULL)约束。其描述如表 8.1 所示。

在 SQL Server 2005 中，最快捷的方法通过在定义表时用 CREATE TABLE 语句加约束子句的方法定义约束，当然也可以通过 SQL Server Management Studio 图形界面定义约束

(3) 规则

规则(Rule)是数据库对存储在表的列或用户自定义数据类型中的值的规定和限制，规则只有绑定到列或者用户定义数据类型时才起作用。规则是作为一个独立的数据库对象存在，表中每列或者每个用户定义数据类型只能和一个规则绑定。

规则与 CHECK KEY 约束具有相同的功能，对表中的列进行约束检查。它可以使用多种方式来完成对输入数据的检验，可以使用函数返回值进行验证，也可以使用关键字

BETWEEN、LIKE 和 IN 完成对输入数据的检查。

表 8.1 SQL Server 提供的约束

约束类型	完整性功能描述
PRIMARY KEY	指定主码，确保主码值不重复，并不允许主码内的任何列中输入 NULL，一个表只能有一个 PRIMARY KEY 约束
FOREIGN KEY	主要用于在不同的表之间建立约束，以实现不同表之间的参照完整性。FOREIGN KEY 约束的列或者为 NULL，或者为被参照列的值。
UNIQUE	唯一性约束，对表中的列集进行约束，其值或者为空，或者在整个表中是唯一的
CHECK	指定某个列或列组可以接受值的范围，或指定数据应满足的条件，来强制实现用户要求的域完整性。可以为每列设置多个 CHECK KEY 约束
NOT NULL	非空约束，指定某个列不能为 NULL
DEFAULT	插入数据时，如果没有明确提供列值，则用缺省值作为该列的值

在 SQL Server 中可以通过 CREATE RULE 语句创建规则。其语法格式如下：

CREATE RULE rulename AS condition_expression

其中各参数含义如下：

rulename：规则的名称

condition_expression：规则的条件，可以是 SQL 语句中 WHERE 子句中任何有效的表达式。

当将规则绑定到列或者用户定义的数据类型时，规则将指定可以插入到列中的可接受的值。规则作为数据库的一个独立对象存在，表中每列或者每个用户定义的数据类型只能和一个规则绑定。绑定规则可以通过 sp_bindrule 存储过程实现。

三、实验内容

通过查询编辑器，设置 XSGl 数据库为当前数据库，利用 CREATE TABLE 语句，实现数据库的完整性约束定义，以及利用 SQL Server Management Studio 创建规则，并与相应的列进行绑定。

四、实验步骤

1. 在 XSGl 数据库中，创建一个名为 yanshi_student 的表，指定 SNO 为主码，且设置属性级约束、元组级约束和表级约束。

```
CREATE TABLE yanshi_student
( SNO char (8) ,
  SNAME char (10) NOT NULL ,
  AGE smallint NULL,
  SEX char (2) NULL DEFAULT '男' /*SEX 缺省值为男 */ ,
```

```

DNO char(4) NOT NULL ,
BIRTHDAY datetime NULL ,
PRIMARY KEY (Sno) /*在表级定义主码，实现实体完整性*/,
CHECK(SEX IN ('男','女')) /*性别属性 SEX 只允许取'男'或'女'*/,
CHECK(SNO LIKE '[1-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
/*SNO 只能为 8 位数字，且不能以 0 开头*/ ,
CHECK((SUBSTRING(SNO,5,1)='3' AND DNO='0003')
OR(SUBSTRING(SNO,5,1)='2' AND DNO='0002')
OR (SUBSTRING(SNO,5,1)='1' AND DNO='0001'))
/*定义了元组中 SNO 和 DNO 两个属性值之间的约束条件，SNO 的第 5 位只能为
1, 2, 3, 且当第 5 位为 1 时，DNO 只能为 0001，第 5 位为 2 时，DNO 只能为 0002，第 5
位为 3 时，DNO 只能为 0003*/
)

```

请同学们用下列 SQL 语句调试，并观察其现象：

- (1) INSERT yanshi_student(SNO,SNAME,DNO) values('x0001','li','0001')
- (2) INSERT yanshi_student(SNO,SNAME,DNO) values('00001','li','0001')
- (3) INSERT yanshi_student(SNO,SNAME,DNO) values('10001','li','0001')
- (4) INSERT yanshi_student(SNO,SNAME,DNO) values('20073001','li','0001')
- (5) INSERT yanshi_student(SNO,SNAME,DNO) values('20073001','li','0003')

2. 在 XSGl 数据库中，创建一个名为 yanshi_course 的表，指定 CNO 为主码。

```

CREATE TABLE yanshi_course (
    CNO char(6) ,
    CNAME char(30) NOT NULL ,
    TNAME char(10) NULL ,
    CREDIT float NULL ,
    ROOM char(30) NULL ,
    PRIMARY KEY (CNO)
)

```

请同学们用下列 SQL 语句调试，并观察其现象：

- (1) INSERT yanshi_course(CNO,CNAME) values('222201','软件技术基础')
- (2) INSERT yanshi_course(CNO,CNAME) values('c43002','可视化程序设计与用户界面')
- (3) INSERT yanshi_course(CNO,CNAME) values('133804','工程训练')

3. 在 XSGl 数据库中，创建一个名为 yanshi_sc 的表，指定 SNO、CNO 为主码，并且 SNO 参照 yanshi_student 表中的 SNO，CNO 参照 yanshi_course 表中的 CNO。

```

CREATE TABLE yanshi_sc(

```

```

SNO char(8) ,
CNO char(6) ,
GRADE float NULL,
PRIMARY KEY(SNO,CNO),
FOREIGN KEY(SNO) references yanshi_student(SNO),
FOREIGN KEY(CNO) references yanshi_course(CNO),
CHECK (GRADE BETWEEN 0 AND 100),
)

```

请同学们用下列 SQL 语句调试，并观察其现象：

- (1) INSERT yanshi_sc(SNO,CNO,GRADE) VALUES('20073001','222201',81)
- (2) INSERT yanshi_sc(SNO,CNO,GRADE) VALUES ('20073001','133804',67)
- (3) INSERT yanshi_sc(SNO,CNO,GRADE) VALUES('20073001','222203',70)

4. 在 XSGE 数据库中，创建一个规则，限定绑定列的输入值在 0.5 和 5 之间，并使用存储过程 sp_bindrule 将其绑定到 yanshi_course 表中的 CREDIT 列中，限定学分只能在 0.5 和 5 之间。

```

CREATE RULE rule1 AS @c1 BETWEEN 0.5 AND 5
EXEC sp_bindrule 'rule1','yanshi_course.CREDIT'

```

请同学们用下列 SQL 语句调试，并观察其现象：

- (1) INSERT yanshi_course(CNO,CNAME,CREDIT) values('133805','工程训练',6)
- (2) INSERT yanshi_course(CNO,CNAME,CREDIT) values('133805','工程训练',2)

5. 解除 rule1 规则到 yanshi_course.CREDIT 的绑定，并删除该规则。

- (1) EXEC sp_unbindrule 'yanshi_course.CREDIT'
- (2) DROP RULE rule1

思考题

- (1) 创建一个表 yanshi_depart，包含 DNO、DNAME、DADDRESS 和 DEAN 四个属性，DNO 由 4 个数字字符构成，且为该表的主码，DNAME 由 30 个字符构成，DADDRESS 由 5 个字符构成，DEAN 由 10 个字符构成。
- (2) 为例 1 中的 yanshi_student 表建立外键“DNO”，参考表 yanshi_depart 的“DNO”列。
- (3) 为表 yanshi_depart 的 DNAME 建立一个规则：@dname IN ('机电学院','信息学院','工商学院')。
- (4) 为表 yanshi_depart 的 DADDRESS 建立一个规则：@daddress LIKE '[A-Z][1-9][1-9][1-9][1-9]'，限定 DADDRESS 的值只能由字母开头，后跟 4 个数字。
- (5) 删除第 3 小题所建立的规则。