

# 第五章 数据库完整性

# 数据库完整性

## ●数据的正确性

- ✓是指数据是符合现实世界语义，反映了当前实际状况的

## ●数据的相容性

- ✓是指数据库同一对象在不同关系表中的数据是符合逻辑的

例如，

- 学生的学号必须唯一
- 性别只能是男或女
- 本科学生年龄的取值范围为14~50的整数
- 学生所选的课程必须是学校开设的课程，学生所在的院系必须是学校已成立的院系

# 数据库完整性（续）

数据的完整性和安全性是两个不同概念

- 数据的完整性

- ✓ 防止数据库中存在不符合语义的数据，也就是防止数据库中存在不正确的数据
- ✓ 防范对象：不合语义的、不正确的数据

- 数据的安全性

- ✓ 保护数据库 防止恶意的破坏和非法的存取
- ✓ 防范对象：非法用户和非法操作

# 数据库完整性（续）

为维护数据库的完整性，数据库管理系统必须：

## 1.提供定义完整性约束条件的机制

- 完整性约束条件也称为完整性规则，是数据库中的数据必须满足的语义约束条件
- SQL标准使用了一系列概念来描述完整性，包括关系模型的实体完整性、参照完整性和用户定义完整性
- 这些完整性一般由SQL的数据定义语言语句来实现

# 数据库完整性（续）

## 2.提供完整性检查的方法

- 数据库管理系统中检查数据是否满足完整性约束条件的机制称为完整性检查。
- 一般在INSERT、UPDATE、DELETE语句执行后开始检查，也可以在事务提交时检查

# 数据库完整性（续）

## 3. 违约处理

- 数据库管理系统若发现用户的操作违背了完整性约束条件，就采取一定的动作 (根据应用环境要求确定)
  - 拒绝（NO ACTION）执行该操作
  - 级联（CASCADE）执行其他操作

# 第五章 数据库完整性

---

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名字句

\*5.5 域中的完整性限制

\*5.6 断言

\*5.7 触发器

5.8 小结

# 参照完整性检查和违约处理

## 参照完整性违约处理

### (1) 拒绝 (NO ACTION) 执行

- 不允许该操作执行。该策略一般设置为默认策略

### (2) 级联 (CASCADE) 操作

- 当删除或修改被参照表 (Student) 的一个元组造成了与参照表 (SC) 的不一致, 则删除或修改参照表 (SC) 中的所有造成不一致的元组。

### (3) 设置为空值 (SET-NULL)

- 当删除或修改被参照表 (Student) 的一个元组时造成了不一致, 则将参照表(SC)中的所有造成不一致的元组的对应属性设置为空值。



# 参照完整性检查和违约处理(续)

[例5.4] 显式说明参照完整性的违约处理示例

```
CREATE TABLE SC
```

```
( Sno CHAR(9) NOT NULL, Cno CHAR(4) NOT NULL, Grade  
  SMALLINT,
```

```
PRIMARY KEY(Sno,Cno),
```

```
FOREIGN KEY (Sno) REFERENCES Student(Sno)
```

```
ON DELETE CASCADE      /*级联删除SC表中相应的元组*/
```

```
ON UPDATE CASCADE,     /*级联更新SC表中相应的元组*/
```

```
FOREIGN KEY (Cno) REFERENCES Course(Cno)
```

```
ON DELETE NO ACTION
```

```
/*当删除course 表中的元组造成了与SC表不一致时拒绝删除*/
```

```
ON UPDATE CASCADE
```

```
/*当更新course表中的cno时, 级联更新SC表中相应的元组*/);
```

# 第五章 数据库完整性

---

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名字句

\*5.5 域中的完整性限制

\*5.6 断言

\*5.7 触发器

5.8 小结

## 5.3 用户定义的完整性

用户定义的完整性是：针对某一具体应用的数据必须满足的语义要求。

关系数据库管理系统提供了定义和检验用户定义完整性的机制，不必由应用程序承担。

# CHECK 约束

(1) 用CHECK短语指定列值（属性级别）应该满足的条件

[例5.7] Student表的Ssex只允许取“男”或“女”。

```
CREATE TABLE Student
```

```
( Sno CHAR(9) PRIMARY KEY,
```

```
  Sname CHAR(8) NOT NULL,
```

```
  Ssex CHAR(2) CHECK (Ssex IN ('男','女')) ,
```

```
  /*性别属性Ssex只允许取'男'或'女' */
```

```
  Sage SMALLINT,
```

```
  Sdept CHAR(20));
```

[例5.8] SC表的Grade的值应该在0和100之间。

```
CREATE TABLE SC
```

```
( Sno   CHAR(9) ,
```

```
  Cno   CHAR(4),
```

```
  Grade SMALLINT CHECK (Grade>=0 AND Grade <=100),
```

```
          /*Grade取值范围是0到100*/
```

```
  PRIMARY KEY (Sno,Cno),
```

```
  FOREIGN KEY (Sno) REFERENCES Student(Sno),
```

```
  FOREIGN KEY (Cno) REFERENCES Course(Cno)
```

```
);
```

# CHECK 约束（续）

（2）用CHECK短语指定元组级别（可设置不同属性之间取值互相约束条件）应该满足的条件

[例5.9]当学生的性别是男时，其名字不能以Ms.打头。

```
CREATE TABLE Student
```

```
( Sno CHAR(9),
```

```
  Sname CHAR(8) NOT NULL, Ssex CHAR(2), Sage SMALLINT, Sdept  
  CHAR(20),
```

```
  PRIMARY KEY (Sno),
```

```
  CHECK (Ssex='女' OR Sname NOT LIKE 'Ms.%')
```

```
  /*定义了元组中Sname和 Ssex两个属性值之间的约束条件*/
```

```
);
```

- 性别是女性的元组都能通过该项检查，因为Ssex='女' 成立；
- 当性别是男性时，要通过检查则名字一定不能以Ms.打头

# 属性/元组上的约束条件检查和违约处理

## 属性（或元组）上的约束条件检查和违约处理

- 插入元组或修改属性的值时，关系数据库管理系统检查属性/元组上的约束条件是否被满足
- 如果不满足则操作被拒绝执行

# 第五章 数据库完整性

---

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名字句

\*5.5 域中的完整性限制

\*5.6 断言

\*5.7 触发器

5.8 小结



## 5.4 完整性约束命名子句

### 1.完整性约束命名子句

**CONSTRAINT** <完整性约束条件名><完整性约束条件>

- <完整性约束条件>包括NOT NULL、UNIQUE、PRIMARY KEY短语、FOREIGN KEY短语、CHECK短语等

## 完整性约束命名子句（续）

[例5.10]建立学生登记表Student，要求学号在90000~99999之间，姓名不能取空值，年龄小于30，性别只能是“男”或“女”。

```
CREATE TABLE Student  
( Sno NUMERIC(6)  
  CONSTRAINT C1 CHECK (Sno BETWEEN 90000 AND 99999),  
  Sname CHAR(20)  
  CONSTRAINT C2 NOT NULL,  
  Sage NUMERIC(3)  
  CONSTRAINT C3 CHECK (Sage < 30),  
  Ssex CHAR(2)  
  CONSTRAINT C4 CHECK (Ssex IN ( '男','女')),  
  CONSTRAINT StudentKey PRIMARY KEY(Sno));
```

✓在Student表上建立了5个约束条件，包括主码约束（命名为StudentKey）以及C1、C2、C3、C4四个列级约束。

## 完整性约束命名子句（续）

[例5.11]建立教师表TEACHER，要求每个教师的应发工资不低于3000元。

应发工资是工资列Sal与扣除项Deduct之和。

```
CREATE TABLE TEACHER
```

```
(  Eno  NUMERIC(4) PRIMARY KEY  /*在列级定义主码*/
```

```
   Ename CHAR(10),
```

```
   Job   CHAR(8),
```

```
   Sal   NUMERIC(7,2),
```

```
   Deduct NUMERIC(7,2),
```

```
   Deptno NUMERIC(2),
```

```
CONSTRAINT TEACHERFKKey FOREIGN KEY (Deptno)  
REFERENCES DEPT(Deptno),
```

```
CONSTRAINT C5 CHECK (Sal + Deduct >= 3000)
```

```
);
```

# 完整性约束命名子句（续）

## 2. 修改表中的完整性限制

- 使用ALTER TABLE语句修改表中的完整性限制

[例5.12]去掉例5.10 Student表中对性别的限制。

```
ALTER TABLE Student
```

```
DROP CONSTRAINT C4;
```

# 完整性约束命名子句（续）

[例5.13] 修改表Student中的约束条件，要求学号改为在900000~999999之间，年龄由小于30改为小于40。

- 可以先删除原来的约束条件，再增加新的约束条件

```
ALTER TABLE Student
```

```
DROP CONSTRAINT C1;
```

```
ALTER TABLE Student
```

```
ADD CONSTRAINT C1 CHECK (Sno BETWEEN 900000 AND 999999),
```

```
ALTER TABLE Student
```

```
DROP CONSTRAINT C3;
```

```
ALTER TABLE Student
```

```
ADD CONSTRAINT C3 CHECK(Sage < 40);
```

## 5.8 小结

数据库的完整性是为了保证数据库中存储的数据是正确的

关系数据库管理系统完整性实现的机制

- 完整性约束定义机制
- 完整性检查机制
- 违背完整性约束条件时关系数据库管理系统应采取的动作