

数据库系统原理

Database System Principles



四川大学计算机学院

段磊

leiduan@scu.edu.cn

2014.9

第五章 数据库完整性

- 完整性的概念
 - 指数据的**正确性**和**相容性**
- 与安全性的比较
 - 安全性 防止人为恶意破坏数据库和非法存取
 - 完整性 防止不合语义的、不正确的数据进入数据库

DBMS维护完整性的机制

- 定义机制
 - 允许用户定义**实体完整性**、**参照完整性**和**用户定义的完整性**
- 检查机制
 - 在用户的更新语句（事务）开始执行后检查这些操作执行后是否会违背完整性约束条件
- 违约处理机制
 - 如果DBMS检查到用户操作违背了完整性约束条件，就采取一定动作以保证完整性
 - **拒绝** (No Action)
 - **级联** (Cascade)

本章目录

- ➔ ■ 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 完整性约束命名子句
- 5.5 域中的完整性限制
- 5.6 触发器

5.1 实体完整性

- 定义 通过定义主码实现
 - 例1 定义Student的Sno为主码

```
create table student(  
sno char(10),  
sname char(8),  
sgender char(2),  
sage int,  
sdept char(2),  
constraint student_pkey primary key (sno)  
);
```

5.1 实体完整性

- 定义 通过定义主码实现
 - 例1 定义Student的Sno为主码(续)

```
CREATE TABLE student(  
sno char(10),  
sname char(8),  
sgender char(2),  
sage int,  
sdept char(2)  
);
```

```
ALTER TABLE student add constraint student_pkey  
primary key (sno);
```

实体完整性约束的检查 and 违约处理

- 检查主码值是否唯一
 - 不唯一拒绝操作
- 检查主码的各个属性值是否为空
 - 为空拒绝操作
- 主码上自动建立B+树索引
 - 快速检查主码值是否唯一

本章目录

- 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 完整性约束命名子句
- 5.5 域中的完整性限制
- 5.6 触发器

5.2 参照完整性约束

- 定义 通过定义外码实现

- 例3 定义SC表

```
create table sc(  
sno char(10),  
cno char(3),  
grade int,  
constraint pk_sc primary key (sno,cno),  
constraint sc_fkey_sno foreign key (sno) references  
student(sno),  
constraint sc_fkey_cno foreign key (cno) references  
course(cno),  
);
```

参照完整性检查和违约处理

- 破坏参照完整性的情况 参照表和被参照表的增删改操作都可能破坏参照完整性。
 - 参照表插入元组，被参照表没有对应的码值。
 - 如SC表中插入元组，但Sno在Student表中无对应值。
 - 违约处理只能是拒绝
 - 参照表修改外码值，被参照表没有修改后对应的码值。
 - 与插入情况类似，拒绝。

参照完整性检查和违约处理

- **被参照表删除元组**，原参照表的外码值对应的码值找不到。
 - 如删除Student表中的某学生的元组，SC表中改学生选课记录的Sno值再也找不到了。
 - 违约处理可以是 拒绝/级联删除/置空值。
- **被参照表修改主码值**。
 - 如修改学生的学号。SC表中该生对应的Sno值应相应处理。
 - 违约处理可以是 拒绝/级联删除/置空值。
 - 有的系统不允许修改表的主码值。

违约处理策略

- **拒绝操作 (NO ACTION)**
 - 不允许增删改操作
- **级联操作 (CASCADE)**
 - 删除被参照关系元组时，相应删除对应的参照关系的相应元组。
 - 修改被参照关系主码值，相应修改对应的参照关系的相应元组。
- **置空操作 (SET NULL)**
 - 删除被参照关系元组或修改被参照关系主码值时，将对应的参照关系元组的外码值设置为空值。

带有处理策略的FOREIGN KEY子句

- 格式

```
FOREIGN KEY ( column_name [, ... ] )  
REFERENCES reftable [ ( refcolumn [, ...] ) ]  
[ ON DELETE action ] [ ON UPDATE action]
```

带有处理策略的FOREIGN KEY子句

- 示例 带有违约处理的SC表外码说明

```
FOREIGN KEY (Sno) REFERENCES Student(Sno)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY (Cno) REFERENCES Course(Cno)
ON DELETE NO ACTION
ON UPDATE CASCADE
```

本章目录

- 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 完整性约束命名子句
- 5.5 域中的完整性限制
- 5.6 触发器

5.3 用户定义的完整性

- **列级完整性** 涉及单个属性的约束
 - 列值非空 NOT NULL
 - 列值唯一 UNIQUE
 - 列值满足表达式 CHECK
- **表级完整性** 涉及单个或多个属性的约束
 - 列值唯一 UNIQUE
 - 列值满足表达式 CHECK

不允许取空值

- 例5 在定义Student表时，说明Sname不能取空值。

```
CREATE TABLE student(  
sno char(10),  
sname char(8) NOT NULL,  
sgender char(2),  
sage int,  
sdept char(2),  
CONSTRAINT student_pkey PRIMARY KEY (sno)  
);
```

列值唯一

- 例6 创建部门表DEPT，要求部门名称Dname取值唯一，部门编号DeptNo为主码。

```
CREATE TABLE DEPT(  
  DeptNo NUMERIC(2),  
  Dname CHAR(9) UNIQUE,  
  Location CHAR(10),  
  CONSTRAINT DEPT_PKEY PRIMARY KEY (DeptNo)  
);
```

列值唯一

- 也可写成表级约束的形式：

```
CREATE TABLE DEPT(  
  DeptNo NUMERIC(2),  
  Dname CHAR(9),  
  Location CHAR(10),  
  CONSTRAINT DEPT_UKEY_Dname UNIQUE(Dname),  
  CONSTRAINT DEPT_PKEY PRIMARY KEY (DeptNo)  
);
```

CHECK约束

- 说明Student表的Sgender只能取“男”、“女”。
 - CONSTRAINT Student_check_Sgender CHECK (Sgender IN('男','女'));
- 说明SC表的Grade在0到100分之间
 - CONSTRAINT sc_check_grade CHECK (Grade >=0 and Grade <=100)

CHECK约束

- 涉及多个列值之间的比较

```
CREATE TABLE T1(  
  ID char(4),  
  oldvalue int,  
  newvalue int,  
  CONSTRAINT T1_PKEY PRIMARY KEY(ID),  
  CONSTRAINT T1_CHECK CHECK(newvalue>oldvalue)  
);
```

本章目录

- 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 完整性约束命名子句
- 5.5 域中的完整性限制
- 5.6 触发器

5.4 完整性约束命名子句

- 列级完整性约束一般不能显示命名
- 表级完整性约束可显示命名
 - CONSTRAINT子句
 - 方便用ALTER TABLE语句修改约束
 - ADD CONSTRAINT 增加约束
 - DROP CONSTRAINT 删除约束

修改NOT NULL约束

- 修改Student定义，允许Sname为空。
ALTER TABLE Student
ALTER COLUMN Sname DROP NOT NULL;
- 又要修改为不允许空：
ALTER TABLE Student
ALTER COLUMN Sname SET NOT NULL;

本章目录

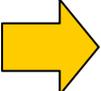
- 5.1 实体完整性
- 5.2 参照完整性
- 5.3 用户定义的完整性
- 5.4 完整性约束命名子句
-  ■ 5.5 域中的完整性限制
- 5.6 触发器

5.5 域中的完整性约束

- **create domain** SQL-92 引入，允许用户创建带有完整性约束的自定义域

```
create domain person_name char(20) not  
null;
```

本章目录

- 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 完整性约束命名子句
- 5.5 域中的完整性限制
-  ■ 5.6 触发器

5.6 触发器 SQL: 1999引入

- 用户定义在关系表上的一类由事件驱动的特殊过程。
 - **事件** 对表的插入、删除、修改
 - **时机** 在对表的插入、删除、修改前或后
 - **条件** 只有当触发条件为真时才执行
 - **类型** 行级触发器 语句级触发器

5.6 触发器

- 例18 为教师表Teacher定义一个完整性规则触发器，规定“教授的工资不得低于4000元，如果低于4000元，自动改为4000元”。
- **create trigger I_U_Sal** after insert or update on *account*
referencing new row as *nrow* //nrow引用新值
for each row //行级
when (*nrow.Job*='教授' and *nrow.Sal* < 4000 //条件
begin
 update Teacher set sal = 4000 where
 Tno=*nrow.Tno*;
end

5.6 触发器

- 例19 完成类似审计功能
- 补充例：warehouse 中4张表
 - inventory(item, level): 库存量
 - minlevel(item, level) : 下限
 - reorder(item, amount): 重新订货量
 - orders(item, amount) : 订货量
 - 当库存量下降到低于下界后自动按照重新订货量订货。

5.6 触发器

```
create trigger reorder-trigger after update of amount on inventory  
referencing old row as orow, new row as nrow  
for each row //新库存少于报警值  
    when nrow.level < = (select level  
        from minlevel  
        where minlevel.item = orow.item)  
    and orow.level > (select level //原库存大于报警值  
        from minlevel  
        where minlevel.item = orow.item)  
begin //触发器动作  
    insert into orders //在订购单中写入要新购的项目和数量  
        (select item, amount  
        from reorder  
        where reorder.item = orow.item);  
end
```

Any Question?



Thank you !