

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 完整性约束命名子句

5.7 触发器（通过实验五自学）

第五章 数据库完整性

□ 什么是数据库的完整性

■ 数据的正确性

- 符合现实世界语义，反映了当前实际状况

例：性别只能是男或女；

学生的学号一定是唯一的；

成绩的取值范围是0到100；

■ 数据的相容性

- 数据库同一对象在不同关系表中的数据符合逻辑

例：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

第五章 数据库完整性

□ 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

完整性是阻止**合法用户**通过**合法操作**向数据库加入**不正确**的数据

安全性防范的是**非法用户**和**非法操作**存取数据库中的**正确**数据

□ 完整性控制机制

■ 1.完整性约束条件定义机制

- DBMS应提供定义数据库完整性约束条件，并把它们作为模式的一部分存入数据库中

■ 2.完整性检查机制

- 检查用户发出的操作请求是否违背了完整性约束条件

■ 3.违约处理

- 如果发现用户的操作请求使数据违背了完整性约束条件，则采取一定的动作来保证数据的完整性。

5.1 实体完整性

□ 关系模型的实体完整性

■ CREATE TABLE语句中用**PRIMARY KEY**定义

○ **单属性码**可定义为**列级**或**表级**约束条件

○ **多属性码**只能定义为**表级**约束条件

例1 定义Student表的Sno属性为主码

```
CREATE TABLE Student  
(Sno Char(9) PRIMARY KEY,  
Sname Char(20),  
Sage Smallint,  
Sdept Char(20))
```

列级

```
CREATE TABLE Student  
(Sno Char(9),  
Sname Char(20),  
Sage Smallint,  
Sdept Char(20),  
PRIMARY KEY(Sno))
```

表级

5.1.1 实体完整性定义

例2 CREATE TABLE SC

(Sno Char(9),

Cno Char(4),

Grade Smallint,

只能在表级定义主码

PRIMARY KEY (Sno, Cno))

- 插入或更新主码列时， RDBMS按照实体完整性规则自动进行检查
- 违约反应
 - 检查主码的各个属性是否**为空**，只要有一个为空就拒绝插入或修改
 - 检查主码值是否**唯一**，如果不唯一则拒绝插入或修改

5.2 参照完整性

5.2.1 参照完整性定义

□ **FOREIGN KEY**子句：定义外码列

□ **REFERENCES**子句：外码参照哪个表的主码

Create Table SC (

Sno char(9) ,

Cno char(4) ,

Grade smallint ,

Primary Key (Sno,Cno),

Foreign Key (Sno) **References** Student(Sno),

Foreign Key (Cno) **References** Course(Cno))

5.2.2 参照完整性检查和违约处理

参照完整性中，对被参照表和参照表进行增删改操作时有可能破坏参照完整性，必须进行检查。

表5.1 可能破坏参照完整性的情况及违约处理

被参照表(如Student)	参照表(如SC)	违约处理
	插入元组	拒绝
	修改外码值	拒绝
删除元组		拒绝/级连删除/设置为空值
修改主码值		拒绝/级连修改/设置为空值

拒绝执行策略一般设置为默认策略

5.2.2 参照完整性检查和违约处理

例1 修改Student中学号S02的元组，关系SC中有3个元组的Sno= 'S02'。这时可考虑2种不同的处理策略：

- **级连修改：**将SC中Sno='S02'的所有元组一起修改。
- **拒绝修改：**拒绝此修改操作。

例2 删除专业表中某个元组，即删除了某个专业号。

学生（学号，姓名，性别，专业号，年龄）

专业（专业号，专业名）

- **置空值删除：**将学生关系相应的专业号置空。

[例] 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*/  
)
```

5.3 用户定义的完整性

□ 针对**某一具体应用**的数据必须满足的语义要求。

■ 一、属性上的约束条件

- 列值非空（NOT NULL短语）
- 列值唯一（UNIQUE短语）
- 检查列值是否满足一个条件表达式（CHECK短语）

1、列值非空、唯一

例1：建立学生登记表
Student，要求姓名唯一，
性别非空。

```
CREATE TABLE Student  
(Sno CHAR(9) PRIMARY KEY,  
Sname VARCHAR(20) UNIQUE,  
Ssex CHAR(2) NOT NULL);
```

一、属性上的约束条件

2、用CHECK短语指定列值应该满足的条件

例2： 建立学生登记表Student，要求学号在 90000至99999之间，性别只能是‘男’或‘女’。

CREATE TABLE Student

(Sno CHAR(5) CHECK (Sno BETWEEN 90000 AND 99999),

Sname VARCHAR(20),

Sage SMALLINT,

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

- 当表中插入元组或修改属性值时，RDBMS就检查属性上的约束条件是否被满足，如果不满足则操作被**拒绝执行**。

二、元组上的约束条件

□ **元组级的限制**：可以设置不同属性之间取值的相互约束条件

■ CREATE TABLE时用**CHECK子句**定义元组上的约束条件

例3：建立教师表TEACHER，要求教师的应发工资不超过3000元。应发工资就是实发工资列Sal与扣除项Deduct之和。

```
CREATE TABLE TEACHER
```

```
(Eno    CHAR(4),
```

```
  Ename  VARCHAR(10),
```

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

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

```
  CHECK (Sal + Deduct <=3000))
```

- 往表中插入元组或修改属性值时，RDBMS检查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。

```
insert into TEACHER values('01', '张兰', 2500, 700)
```

% < **2500+700>3000**

消息

消息 547, 级别 16, 状态 0, 第 1 行

INSERT 语句与 CHECK 约束“CK__TEACHER__4E88ABD4”冲突。该冲突发生于数据库“stuData”，表“dbo. TEACHER”语句已终止。

1、完整性约束命名子句

CONSTRAINT<完整性约束条件名>[**PRIMARY KEY**短语|**FOREIGN KEY**短语|**CHECK**短语|**NOT NULL**|**UNIQUE**]

例4：建立学生表Student，要求学号在90000至99999之间。

CREATE TABLE Student3

(Sno CHAR(5)

CONSTRAINT C1 CHECK (Sno BETWEEN 90000 AND 99999),

Sname VARCHAR(20),

Sage SMALLINT)

5.4 完整性约束命名子句

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：修改[例4]student表中的约束条件，要求学号改为在900000-999999之间。

方法：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)

《数据库系统概论》 P173

2题

4题

6题 (**注意理解题目要求**)