

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

*5.5 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)  
);
```



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 C1 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;



修改表中的完整性限制（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);



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



断言

- ❖ **SQL**中，可以使用 **CREATE ASSERTION**语句，通过声明性断言来指定更具一般性的约束。
- ❖ 可以定义涉及多个表的或聚集操作的比较复杂的完整性约束。
- ❖ 断言创建以后，任何对断言中所涉及的关系的操作都会触发关系数据库管理系统对断言的检查，任何使断言不为真值的操作都会被拒绝执行



断言（续）

1. 创建断言的语句格式

CREATE ASSERTION<断言名><CHECK 子句>

- 每个断言都被赋予一个名字
- <CHECK 子句>中的约束条件与WHERE子句的条件表达式类似。



断言（续）

[例5.18] 限制数据库课程最多60名学生选修

```
CREATE ASSERTION ASSE_SC_DB_NUM  
CHECK (60 >= (select count(*)  
      From Course,SC  
      Where SC.Cno=Course.Cno and  
            Course.Cname ='数据库'  
      );
```

触发时机：向SC表中插入元组
违约反应：拒绝执行



断言（续）

[例5.19]限制每一门课程最多60名学生选修

```
CREATE ASSERTION ASSE_SC_CNUM1  
CHECK(60 >= ALL (SELECT count(*)  
      FROM SC  
      GROUP by cno)  
);
```



断言（续）

[例5.20]限制每个学期每一门课程最多**60**名学生选修

首先需要修改**SC**表的模式，增加一个“学期（**TERM**）”属性

ALTER TABLE SC ADD TERM DATE;



断言（续）

[例5.20]限制每个学期每一门课程最多**60**名学生选修

首先需要修改**SC**表的模式，增加一个“学期（**TERM**）”属性

```
ALTER TABLE SC ADD TERM DATE;
```

然后，定义断言：

```
CREATE ASSERTION ASSE_SC_CNUM2
```

```
CHECK(60 >= ALL (SELECT count(*)
```

```
FROM SC
```

```
GROUP by cno, TERM)
```

```
);
```



断言（续）

[例5.20]限制每个学期每一门课程最多**60**名学生选修

首先需要修改**SC**表的模式，增加一个“学期（**TERM**）”属性

如果断言很复杂，则系统检测和维护断言的开销较高，这是在使用断言时应该注意的

```
CREATE ASSERTION ASSE_SC_CNUM2
```

```
CHECK(60 >= ALL (SELECT count(*)
```

```
FROM SC
```

```
GROUP by cno, TERM)
```

```
);
```



断言（续）

2. 删除断言的语句格式为

■ **DROP ASSERTION** <断言名>;



小结

❖ 完整性约束命名子句

■ 命名完整性约束

■ 使用**ALTER TABLE**语句修改完整性约束

- 增加/删除/修改完整性约束

❖ 断言

■ 定义断言

- 用断言定义复杂完整性约束

■ 删除断言



