

# 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 触发器

5.7 小结

## 5.4 完整性约束命名子句



### ❖ CONSTRAINT 约束

CONSTRAINT < 完整性约束条件名 >

[ PRIMARY KEY 短语

|FOREIGN KEY 短语

|CHECK 短语]

# 完整性约束命名子句（续）



[例 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）



## ❖ 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

# 完整性约束命名子句（续）



[例 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 触发器

5.7 小结

## 5.5 域中的完整性限制



- ❖ SQL 支持域的概念，并可以用 **CREATE DOMAIN** 语句建立一个域以及该域应该满足的完整性约束条件。

[例 14] 建立一个性别域，并声明性别域的取值范围

```
CREATE DOMAIN GenderDomain CHAR(2)  
CHECK (VALUE IN ('男', '女'));
```

这样 [例 10] 中对 Ssex 的说明可以改写为

Ssex GenderDomain

[例 15] 建立一个性别域 GenderDomain，并对其中的限制命名

```
CREATE DOMAIN GenderDomain CHAR(2)  
CONSTRAINT GD CHECK (VALUE IN ('男', '女'));
```

# 域中的完整性限制（续）



[例 16] 删除域 GenderDomain 的限制条件 GD。

```
ALTER DOMAIN GenderDomain  
DROP CONSTRAINT GD;
```

[例 17] 在域 GenderDomain 上增加限制条件 GDD。

```
ALTER DOMAIN GenderDomain  
ADD CONSTRAINT GDD CHECK (VALUE IN ('1', '0'));
```

- ✓ 通过 [例 16] 和 [例 17]，就把性别的取值范围由 ('男', '女') 改为 ('1', '0')