

# 数据库的完整性

浙江农林大学

主讲：刘丽娟



# 数据库完整性

- 数据库的完整性

- 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
      - 等





# 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）



浙江农林大学  
ZHEJIANG A&F UNIVERSITY

## 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
  - 拒绝（NO ACTION）执行该操作
  - 级连（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





## 5.1 实体完整性

### 5.1.1 实体完整性定义

### 5.1.2 实体完整性检查和违约处理





## 5.1.1 实体完整性定义

- 关系模型的实体完整性
  - CREATE TABLE中用PRIMARY KEY定义
- 单属性构成的码有两种说明方法
  - 定义为列级约束条件
  - 定义为表级约束条件
- 对多个属性构成的码只有一种说明方法
  - 定义为表级约束条件



# 实体完整性定义（续）

[例5.1] 将Student表中的Sno属性定义为码

(1) 在列级定义主码

```
CREATE TABLE Student  
( Sno CHAR(9) PRIMARY KEY,  
  Sname CHAR(20) NOT NULL,  
  Ssex CHAR(2),  
  Sage SMALLINT,  
  Sdept CHAR(20)  
);
```





# 实体完整性定义（续）

(2) 在表级定义主码

```
CREATE TABLE Student  
( Sno CHAR(9),  
  Sname CHAR(20) NOT NULL,  
  Ssex CHAR(2),  
  Sage SMALLINT,  
  Sdept CHAR(20),  
  PRIMARY KEY (Sno)  
);
```



# 实体完整性定义（续）

[例5.2] 将SC表中的Sno, Cno属性组定义为码

```
CREATE TABLE SC
```

```
( Sno CHAR(9) NOT NULL,
```

```
  Cno CHAR(4) NOT NULL,
```

```
  Grade SMALLINT,
```

```
  PRIMARY KEY (Sno,Cno)  /*只能在表级定义主码*/
```

```
);
```





## 5.1 实体完整性

### 5.1.1 实体完整性定义

### 5.1.2 实体完整性检查和违约处理



## 5.1.2 实体完整性检查和违约处理

- 插入或对主码列进行更新操作时，关系数据库管理系统按照实体完整性规则自动进行检查。包括：
  - 检查主码值是否唯一，如果不唯一则拒绝插入或修改
  - 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





# 实体完整性检查和违约处理（续）

- 检查记录中主码值是否唯一的一种方法是进行全表扫描
  - 依次判断表中每一条记录的主码值与将插入记录上的主码值（或者修改的新主码值）是否相同

待插入记录				
Key <sub>i</sub>	F2 <sub>i</sub>	F3 <sub>i</sub>	F4 <sub>i</sub>	F5 <sub>i</sub>



# 实体完整性检查和违约处理（续）

## ❖ 表扫描缺点

- 十分耗时

❖ 为避免对基本表进行全表扫描，RDBMS核心一般都在主码上自动建立一个索引

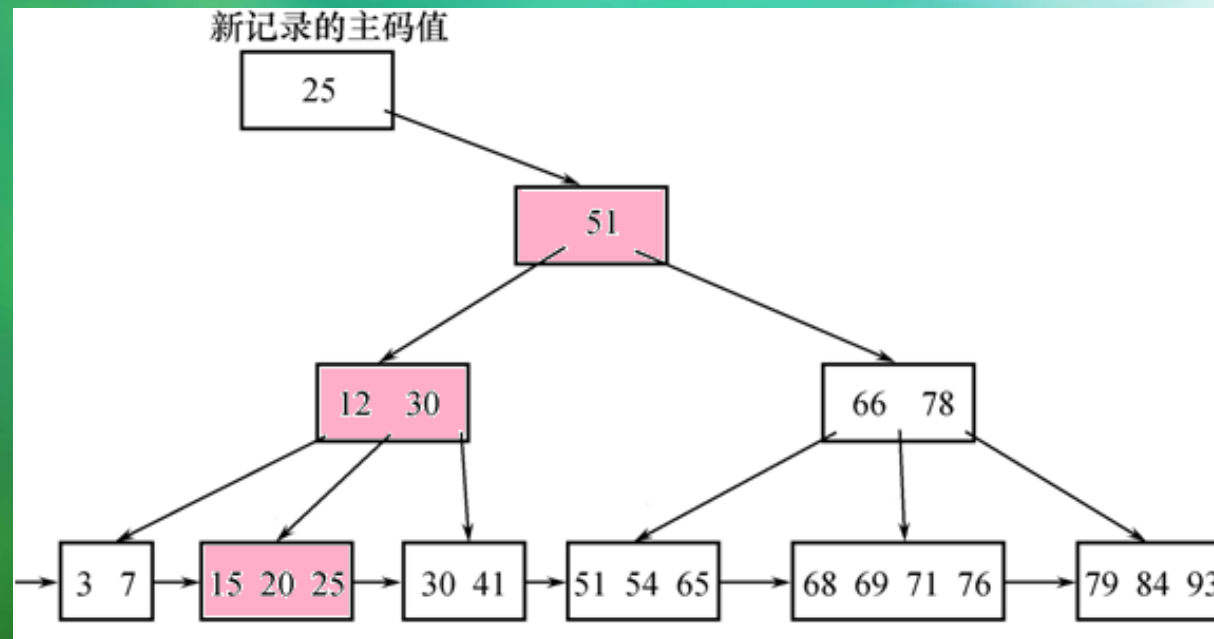


# 实体完整性检查和违约处理（续）



浙江农林大学  
ZHEJIANG A&F UNIVERSITY

- B+树索引



例如，

— 新插入记录的主码值是25

- 通过主码索引，从B+树的根结点开始查找
- 读取3个结点：根结点（51）、中间结点（12 30）、叶结点（15 20 25）
- 该主码值已经存在，不能插入这条记录



# 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







## 5.2 参照完整性

### 5.2.1 参照完整性定义

### 5.2.2 参照完整性检查和违约处理



## 5.2.1 参照完整性定义

- 关系模型的参照完整性定义
  - 在CREATE TABLE中用FOREIGN KEY短语定义哪些列为外码
  - 用REFERENCES短语指明这些外码参照哪些表的主码





## 参照完整性定义（续）

例如，关系SC中（Sno，Cno）是主码。Sno，Cno分别参照Student表的主码和Course表的主码

[例5.3]定义SC中的参照完整性

```
CREATE TABLE SC
( Sno   CHAR(9) NOT NULL,
  Cno   CHAR(4) NOT NULL,
  Grade SMALLINT,
  PRIMARY KEY (Sno, Cno), /*在表级定义实体完整性*/
  FOREIGN KEY (Sno) REFERENCES Student(Sno),
  /*在表级定义参照完整性*/
  FOREIGN KEY (Cno) REFERENCES Course(Cno)
  /*在表级定义参照完整性*/
);
```



## 5.2 参照完整性

### 5.2.1 参照完整性定义

### 5.2.2 参照完整性检查和违约处理





# 参照完整性检查和违约处理

- 一个参照完整性将两个表中的相应元组联系起来
- 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，必须进行检査



## 参照完整性检查和违约处理（续）

- 例如，对表SC和Student有四种可能破坏参照完整性的情况：
  - SC表中增加一个元组，该元组的Sno属性的值在表Student中找不到一个元组，其Sno属性的值与之相等。
  - 修改SC表中的一个元组，修改后该元组的Sno属性的值在表Student中找不到一个元组，其Sno属性的值与之相等。





## 参照完整性检查和违约处理（续）

- 例如，对表SC和Student有四种可能破坏参照完整性的情况（续）：
  - 从Student表中删除一个元组，造成SC表中某些元组的Sno属性的值在表Student中找不到一个元组，其Sno属性的值与之相等。
  - 修改Student表中一个元组的Sno属性，造成SC表中某些元组的Sno属性的值在表Student中找不到一个元组，其Sno属性的值与之相等。





# 参照完整性检查和违约处理（续）

表5.1 可能破坏参照完整性的情况及违约处理

被参照表（例如Student）	参照表（例如SC）	违约处理
可能破坏参照完整性 ←	插入元组	拒绝
可能破坏参照完整性 ←	修改外码值	拒绝
删除元组 →	可能破坏参照完整性	拒绝/级连删除/设置为空值
修改主码值 →	可能破坏参照完整性	拒绝/级连修改/设置为空值





# 参照完整性检查和违约处理（续）

## • 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的一个元组时造成了不一致，则将参照表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）

外码

例如，有下面2个关系

学生（学号，姓名，性别，专业号，年龄）

专业（专业号，专业名）

- 假设专业表中某个元组被删除，专业号为12
- 按照设置为空值的策略，就要把学生表中专业号=12的所有元组的专业号设置为空值
- 对应语义：某个专业删除了，该专业的所有学生专业未定，等待重新分配专业





# 参照完整性检查和违约处理（续）

- 对于参照完整性，除了应该定义外码，还应定义外码列是否允许空值
- 参见爱课程网数据库系统概论5.2节动画《参照完整性》



# 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 [dbo].[sc1](  
[snum] [nchar](12) NOT NULL,  
[cnum] [nchar](12) NOT NULL,  
[score] [tinyint] NULL,  
primary key(snum,cnum),  
foreign key (snum) references s(snum)  
on delete cascade on update cascade,  
foreign key (cnum) references c(cnum)  
on delete no action on update cascade,  
)  
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*/  
on delete no action: /*当删除c表中的元组造成了与SC表不一致时拒绝  
删除*/
```





# 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





## 5.3 用户定义的完整性

### 5.3.1 属性上的约束条件

### 5.3.2 元组上的约束条件



# 1. 属性上约束条件的定义

- CREATE TABLE时定义属性上的约束条件
  - 列值非空（NOT NULL）
  - 列值唯一（UNIQUE）
  - 检查列值是否满足一个条件表达式（CHECK）





# 属性上约束条件的定义（续）

## （1）不允许取空值

[例5.5] 在定义SC表时，说明Sno、Cno、Grade属性不允许取空值。

```
CREATE TABLE SC  
( Sno CHAR(9) NOT NULL,  
  Cno CHAR(4) NOT NULL,  
  Grade SMALLINT NOT NULL,  
  PRIMARY KEY (Sno, Cno),
```

...

/\* 如果在表级定义实体完整性，隐含了Sno，Cno不允许取空值，则在列级不允许取空值的定义可以不写 \*/

```
);
```



# 属性上约束条件的定义（续）

## （2）列值唯一

[例5.6]建立部门表DEPT，要求部门名称Dname列取值唯一，  
部门编号Deptno列为主码

```
CREATE TABLE DEPT
```

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

```
  Dname CHAR(9) UNIQUE NOT NULL,
```

```
                /*要求Dname列值唯一, 并且不能取空值*/
```

```
  Location CHAR(10),
```

```
  PRIMARY KEY (Deptno)
```

```
);
```





# 属性上约束条件的定义（续）

## （3）用CHECK短语指定列值应该满足的条件

[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)
);
```



# 属性上约束条件的定义（续）

[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)
```

```
);
```





## 2. 属性上的约束条件检查和违约处理

- 属性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上的约束条件是否被满足
  - 如果不满足则操作被拒绝执行



## 5.3 用户定义的完整性

### 5.3.1 属性上的约束条件

### 5.3.2 元组上的约束条件





# 1. 元组上约束条件的定义

- 在CREATE TABLE时可以用CHECK短语定义元组上的约束条件，即元组级的限制
- 同属性值限制相比，元组级的限制可以设置不同属性之间的取值的相互约束条件



# 元组上约束条件的定义（续）

[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





## 2. 元组上约束条件检查和违约处理

- 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



浙江农林大学  
ZHEJIANG A&F UNIVERSITY

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)  
);
```

- ✓ 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 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;
```





# 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);
```



# 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



## 5.8 小结



浙江农林大学  
ZHEJIANG A&F UNIVERSITY

- 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