

第八章：数据库完整性

8.1 实体完整性

实体完整性定义

➤ 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

[例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)
);
```

(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)
);
```

[例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*/
);
```

实体完整性检查和违约处理

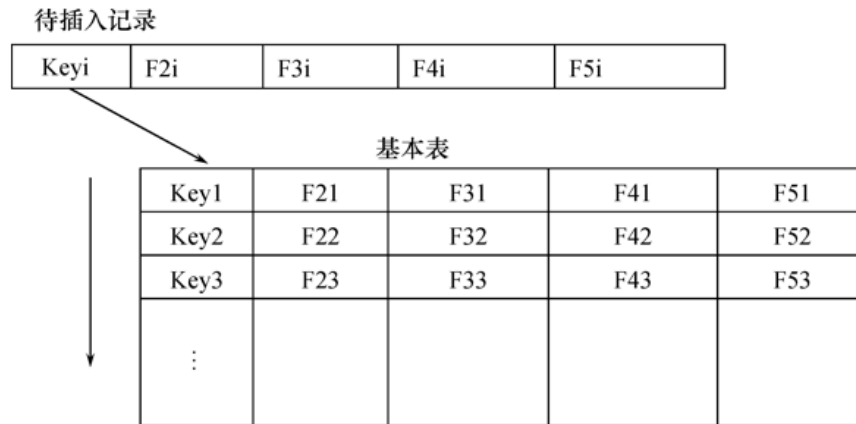
➤ 实体完整性检查和违约处理

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

检查记录中主码值是否唯一的一种方法是进行**全表扫描**

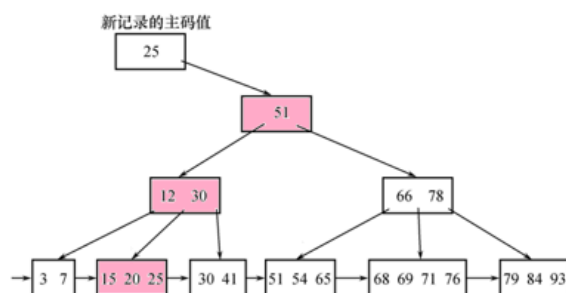
- 依次判断表中每一条记录的主码值与将插入记录上的主码值（或者修改的新主码值）是否相同



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**



• 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

8.2 参照完整性

参照完整性定义

➤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

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*/
);
```

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，必须进行检查

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。
- 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。

表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。

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

[例4] 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*/
);
```