

Chapter 4.ER模型的转换

2018年6月18日 15:29

主键: primary key

作用: 用于唯一地标识表中的某一条记录, 确保每列的原子性。

形式: 主键可以由一个字段, 也可以由多个字段组成, 分别成为单字段主键或多字段主键

其他: 建立主键会生成唯一索引, 一个表只能建一个主键。

外键: foreign key

作用: 保持数据一致性, 完整性, 主要目的是使两张表形成关联, 并控制存储在外键表中的数据。

形式: 建立外键的表为“从表”, 外键依赖的表为“主表”, 主表是从表的“外表”

其他: 外键只能引用外表中的列的值! 一个表只能建一个外键。

建立外键的列, 必须是外表的主键, 并且字段类型必须一致。

1. 实体集 (如创建表)

```
CREATE TABLE Employees
```

```
(ssn CHAR(11),
```

```
name CHAR(20),
```

```
lot INTEGER,
```

```
PRIMARY KEY (ssn))
```

2. 多对多联系的关系表

- 所有参与联系的实体主关键字要作为外键存在, 且所有外键构成该实体集的主关键字

- 联系集本身属性: 一般属性

- 例1:

```
CREATE TABLE Works_in (
```

```
ssn CHAR(11),
```

did INTEGER,
since DATE,
PRIMARY KEY (ssn, did),
FOREIGN KEY (ssn) REFERENCES Employees,
FOREIGN KEY (did) REFERENCES Departments)

• 例2:

```
CREATE TABLE Work-In3 (  
    ssn CHAR(11), did integer,  
    from date, to date,  
    PRIMARY KEY(ssn, did, from, to)  
    FOREIGN KEY (ssn) REFERENCES Employees (ssn) ,  
    ON DELETE NO ACTION,  
    FOREIGN KEY (did) REFERENCES Departments,  
    FOREIGN KEY (from, to) REFERENCES DURATION)
```

On delete no action指定如果试图删除某一行，而该行的键被其他表的现有行中的外键所引用，则产生错误并回滚 **DELETE**语句。

3. 一对多联系

- 一对多的联系要以**多数参与的实体主键**作为主键或者将联系和多数实体**合并为一个关系表**（因为主键相同）
 - 独立关系表：多产生一个表，且需要两次连接
 - 合并成一个关系表：如果不是全参与会浪费空间（**全参与时不浪费空间**）

4. 几种情况

- **参加约束**（全参与）：声明时后加not null
- **弱实体集**：
 - 首先是**全参与约束**
 - 其次要将**部分码**和依赖实体的**主键**一起作为主码
 - 对于依赖实体的主键要声明**级联删除**。

```
CREATE TABLE Dep_Policy (  
    pname CHAR(20),  
    age INTEGER,
```

```
CREATE TABLE Dep_Policy (  
    pname CHAR(20),  
    age INTEGER,  
    cost REAL,  
    ssn CHAR(11) NOT NULL,  
    PRIMARY KEY (pname, ssn),  
    FOREIGN KEY (ssn) REFERENCES Employees,  
    ON DELETE CASCADE)
```

- 类层次 (ISA联系) :

Overlap约束: 两个子类之间是否允许包含同一实体

Covering约束: 子类中所有实体是否包含超类中所有实体

- 方法一: 子类引入超类的主键做外键 (同时也是主键), 并标注级联删除。(不包含超类的全部属性)
 - 查询子类的其他属性需要连接, 自己的属性不需要连接
 - 可以存储非子类的实体
- 方法二: 仅仅有两个关系, 包含超类的全部属性, 除主关键字约束外无其他约束
 - 不能存储非子类实体
 - 只需要在一个关系上进行, 无需其他连接操作
 - 查询所有雇员属性需要在两个关系上进行操作