

# 实验三 完整性语言实验

16337341 朱志儒

## 实验 3.1 实体完整性实验

### 实验目的

掌握实体完整性的定义和维护方法。

### 实验内容

定义实体完整性，删除实体完整性。能够写出两种方式定义实体完整性的 SQL 语句：

创建表时定义实体完整性、创建表后定义实体完整性。设计 SQL 语句验证完整性约束是否起作用。

### 实验步骤

#### 1) 创建表时定义实体完整性（列级实体完整性）

定义供应商表的实体完整性

```
CREATE TABLE Supplier(  
    supkey INTEGER CONSTRAINT PK_supplier PRIMARY KEY,  
    name CHAR(250),  
    address VARCHAR(400),  
    nationkey INTEGER,  
    phone CHAR(150),  
    acctbal REAL,  
    comment VARCHAR(101));
```

结果：

名称	suppkey
数据类型	int
系统类型	int
主键	True
允许 Null 值	False

## 2) 创建表时定义实体完整性（表级实体完整性）

定义供应商表的实体完整性

```
CREATE TABLE Supplier(  
    suppkey INTEGER,  
    name CHAR(250),  
    address VARCHAR(400),  
    nationkey INTEGER,  
    phone CHAR(150),  
    acctbal REAL,  
    comment VARCHAR(101),  
    CONSTRAINT PK_supplier PRIMARY KEY(suppkey));
```

结果：

名称	suppkey
数据类型	int
系统类型	int
主键	True
允许 Null 值	False

## 3) 创建表后定义实体完整性

定义供应商表

```
CREATE TABLE Supplier(  
    suppkey INTEGER NOT NULL,  
    name CHAR(250),  
    address VARCHAR(400),  
    nationkey INTEGER,  
    phone CHAR(150),  
    acctbal REAL,  
    comment VARCHAR(101),);
```

```
ALTER TABLE Supplier
ADD CONSTRAINT PK_supplier PRIMARY KEY(supkey);
```

结果：

名称	supkey
数据类型	int
系统类型	int
主键	True
允许 Null 值	False

#### 4) 定义实体完整性（主码由多个属性组成）

定义供应关系表的实体完整性

```
CREATE TABLE PartSupp(
    partkey INTEGER,
    supkey INTEGER,
    availqty INTEGER,
    supplycost REAL,
    comment VARCHAR(199),
    PRIMARY KEY(partkey, supkey));
```

结果：

dbo.PartSupp
列
partkey (PK, int, not null)
supkey (PK, int, not null)
availqty (int, null)
supplycost (real, null)
comment (varchar(199), null)

#### 5) 有多个候选码时定义实体完整性

定义国家表的实体完整性，其中 nationkey 和 name 都是候选码，选择 nationkey 作为主码，name 上定义唯一性约束。

```
CREATE TABLE nation(
    nationkey INTEGER CONSTRAINT PK_nation PRIMARY KEY,
    name CHAR(250) UNIQUE,
    regionkey INTEGER,
    comment VARCHAR(152));
```

结果:

dbo.nation
列
nationkey (PK, int, not null)
name (char(250), null)
regionkey (int, null)
comment (varchar(152), null)
键
PK_nation
UQ_nation_72E12F1BADFF3FD0

## 6) 删除实体完整性

删除国家实体的主码

```
ALTER TABLE nation DROP CONSTRAINT PK_nation;
```

结果:

dbo.nation
列
nationkey (int, not null)
name (char(250), null)
regionkey (int, null)
comment (varchar(152), null)
键
UQ_nation_72E12F1BADFF3FD0

## 7) 增加两条相同记录, 验证实体完整性是否起作用

```
INSERT INTO Supplier(supkey, name, address, nationkey, phone, acctbal, comment)
VALUES (11, 'test1', 'test1', 101, '12345678', 0.0, 'test1');
INSERT INTO Supplier(supkey, name, address, nationkey, phone, acctbal, comment)
VALUES (11, 'test2', 'test2', 102, '23456789', 0.0, 'test2');
```

结果:

(1 行受影响)  
消息 2627, 级别 14, 状态 1, 第 57 行  
违反了 PRIMARY KEY 约束“PK\_supplier”。不能在对象“dbo.Supplier”中插入重复键。重复键值为 (11)。  
语句已终止。

只增加了一条记录:

	supkey	name	address	nationkey	phone	acctbal	comment
1	11	test1	test1	101	12345678	0	test1

## 实验总结

通过这次实验，我了解并掌握了实体完整性的定义和维护方法。了解定义列级和表级的实体完整性，掌握了创建表前和创建表后的定义实体完整性的方法，也知道删除实体完整性的方法。

## 实验 3.2 参照完整性实验

### 实验目的

掌握参照完整性的定义和维护方法

### 实验内容

定义参照完整性，定义参照完整性的违约处理，删除参照完整性。写出两种方式定义参照完整性的 SQL 语句：创建表时定义参照完整性、创建表后定义参照完整性。

### 实验步骤

#### 1) 创建表时定义参照完整性

先定义地区表的实体完整性，再定义国家表上的参照完整性。

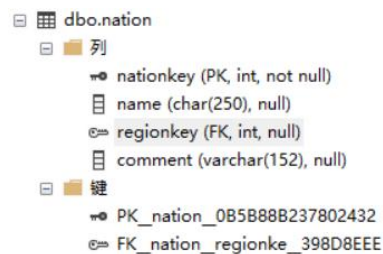
```
CREATE TABLE region(  
    regionkey INTEGER PRIMARY KEY,  
    name CHAR(250),  
    comment VARCHAR(152));  
  
CREATE TABLE nation(  
    nationkey INTEGER PRIMARY KEY,
```

```
name CHAR(250),
regionkey INTEGER REFERENCES region(regionkey),
comment VARCHAR(152));
```

或者

```
CREATE TABLE nation(
    nationkey INTEGER PRIMARY KEY,
    name CHAR(250),
    regionkey INTEGER,
    comment VARCHAR(152),
    CONSTRAINT FK_Nation_regionkey FOREIGN KEY (regionkey) REFERENCES
region(regionkey));
```

结果:

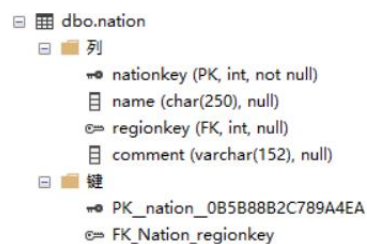


## 2) 创建表后定义参照完整性

定义国家表的参照完整性。

```
CREATE TABLE nation(
    nationkey INTEGER PRIMARY KEY,
    name CHAR(250),
    regionkey INTEGER,
    comment VARCHAR(152));
ALTER TABLE nation
ADD CONSTRAINT FK_Nation_regionkey
FOREIGN KEY(regionkey) REFERENCES region(regionkey);
```

结果:



### 3) 定义参照完整性（外码由多个属性组成）

定义订单项目表的参照完整性。

```
CREATE TABLE Lineitem(  
    orderkey INTEGER REFERENCES Orders(orderkey),  
    partkey INTEGER,  
    supkey INTEGER,  
    linenummer INTEGER,  
    quantity REAL,  
    extendedprice REAL,  
    discount REAL,  
    tax REAL,  
    returnflag CHAR(1),  
    linestatus CHAR(1),  
    shipdate DATE,  
    commitdate DATE,  
    receiptdate DATE,  
    shipinstruct CHAR(250),  
    shipmode CHAR(100),  
    comment VARCHAR(440),  
    PRIMARY KEY(orderkey, linenummer),  
    FOREIGN KEY(partkey, supkey) REFERENCES PartSupp(partkey,  
supkey));
```

结果：

dbo.Lineitem	
列	
orderkey	(PK, FK, int, not null)
partkey	(FK, int, null)
supkey	(FK, int, null)
linenummer	(PK, int, not null)
quantity	(real, null)
extendedprice	(real, null)
discount	(real, null)
tax	(real, null)
returnflag	(char(1), null)
linestatus	(char(1), null)
shipdate	(date, null)
commitdate	(date, null)
receiptdate	(date, null)
shipinstruct	(char(250), null)
shipmode	(char(100), null)
comment	(varchar(440), null)
键	
PK_Lineitem_43FB43B1CFF520FE	
FK_Lineitem_6EF57B66	
FK_Lineitem_orderk_6E01572D	

#### 4) 定义参照完整性的违约处理

定义国家表的参照完整性, 当删除或修改被参照表记录时, 设置参照表中相应记录的值为空值。

```
CREATE TABLE nation(  
    nationkey INTEGER PRIMARY KEY,  
    name CHAR(250),  
    regionkey INTEGER,  
    comment VARCHAR(152),  
    CONSTRAINT FK_nation_regionkey FOREIGN KEY (regionkey) REFERENCES  
region(regionkey)  
    ON DELETE SET NULL  
    ON UPDATE SET NULL);
```

结果:



#### 5) 删除参照完整性

删除国家表的外码。

```
ALTER TABLE nation DROP CONSTRAINT FK_nation_regionkey;
```

结果:



#### 6) 插入一条国家记录, 验证参照完整性是否起作用

```
INSERT INTO nation(nationkey, name, regionkey, comment)  
VALUES(1001, 'nation1', 1001, 'comment1');
```



删除国家表外码前：

消息 547, 级别 16, 状态 0, 第 66 行  
INSERT 语句与 FOREIGN KEY 约束“FK\_nation\_regionkey”)冲突。该冲突发生于数据库“TPCH\_1”, 表“dbo.region”, column 'regionkey'.  
语句已终止。

删除国家表外码后：

	nationkey	name	regionkey	comment
1	1001	nation1	1001	comment1

## 实验总结

通过这次实验，我了解并掌握了参照完整性的定义和维护的方法。实现了在创建表时定义参照完整性和在创建表后定义参照完整性，学习了如何定义参照完整性的违约处理。