

数据库系统实验

实验三：完整性语言

16 级计科教务 2 班

16337327

郑映雪

实验题目

完整性语言实验

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。

实验重点和难点

实验重点：创建表时定义实体完整性。

实验难点：有多个候选码时实体完整性的定义。

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。

实验操作和结果

创建表时定义实体完整性（列级）

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)  
);
```

创建表时定义实体完整性（表级）

定义供应商表的实体完整性。

```
CREATE TABLE Supplier(  
    supkey INTEGER,  
    name CHAR(250),  
    address VARCHAR(400),  
    nationkey INTEGER,  
    phone CHAR(150),  
    acctbal REAL,  
    comment VARCHAR(1010)  
    CONSTRAINT PK_Supplier PRIMARY KEY(supkey)  
);
```

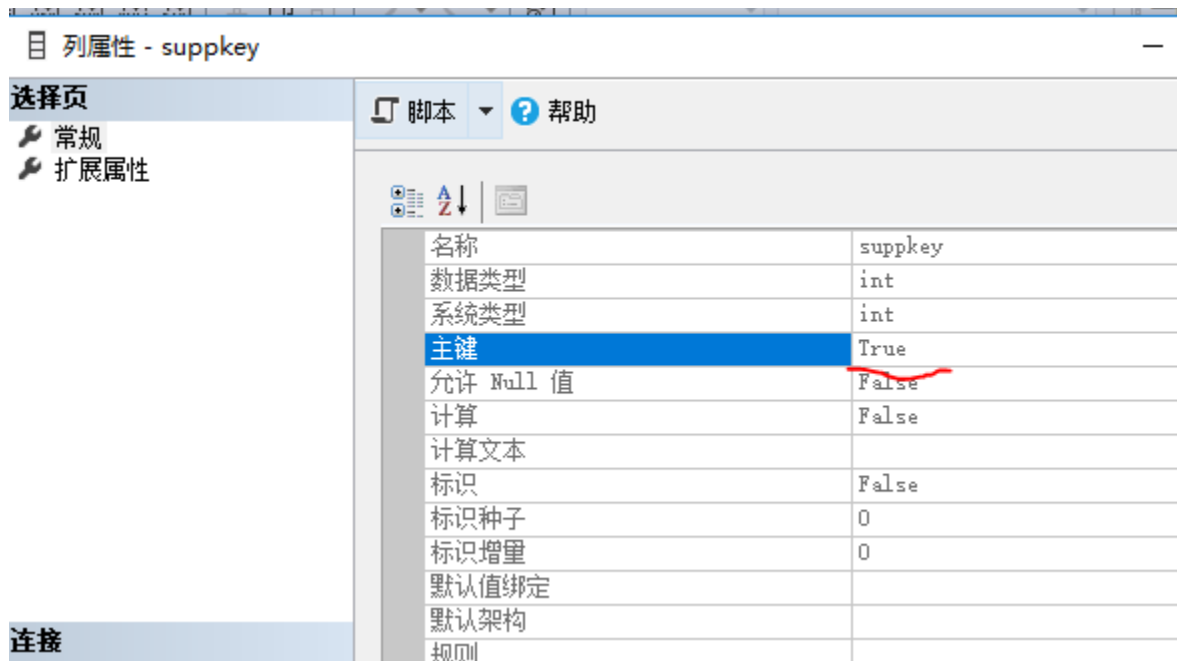
创建表后定义实体完整性

定义供应商表再修改供应商表，增加实体完整性。在 SQLServer 中，不允许定义可以为 NULL 值的键为主键，所以需要另外定义 supkey 为 not null。

```
CREATE TABLE Supplier(  
    name CHAR(250),  
    address VARCHAR(400),  
    nationkey INTEGER,  
    phone CHAR(150),  
    acctbal REAL,  
    comment VARCHAR(1010)
```

```
);
ALTER TABLE Supplier ADD supkey INTEGER NOT NULL default '';
ALTER TABLE Supplier ADD CONSTRAINT PK_Supplier PRIMARY KEY(supkey);
```

实验结果如下：



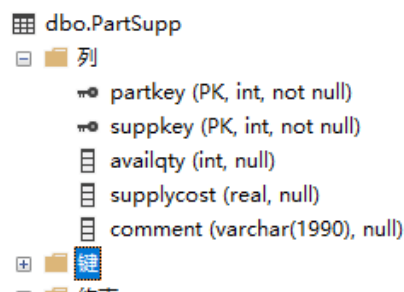
列属性 - supkey	
名称	supkey
数据类型	int
系统类型	int
主键	True
允许 Null 值	False
计算	False
计算文本	
标识	False
标识种子	0
标识增量	0
默认值绑定	
默认架构	
规则	

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(1990),
    PRIMARY KEY(partkey,supkey)
);
```

结果如下：



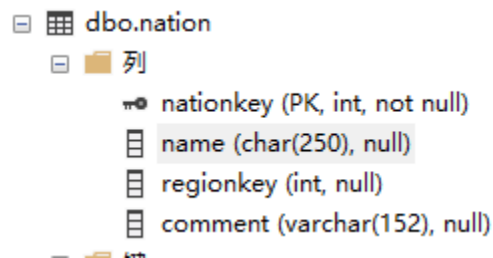
dbo.PartSupp	
列	
partkey (PK, int, not null)	
supkey (PK, int, not null)	
availqty (int, null)	
supplycost (real, null)	
comment (varchar(1990), null)	
键	

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)  
);
```

结果如下：

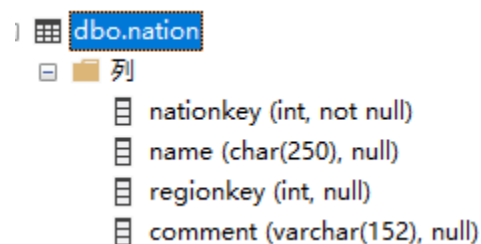


删除实体完整性

删除国家实体的主码。

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

删除后发现 nationkey 的主键标识消失了：



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');
```

作用生效，不允许插入重名的元组：

(1 行受影响)

消息 2627，级别 14，状态 1，第 3 行

违反了 PRIMARY KEY 约束 'PK_Supplier'。不能在对象 'dbo.Supplier' 中插入重复键。
语句已终止。

实验总结

这次实验比较简单，在实验课堂上就完成了。这次实验的设计是很合理的，从多种不同的方法出发定义了同样的完整性，展现了 sql 语句的多样性。但是还是有一些与书上不同的地方，所以我觉得我每次做实验除了熟悉理论课上的一些知识，还对 sql server 语句编写+图形界面的使用有了更多的经验……通过这次实验，我了解到了参照完整性对于数据库设计的重要性，对理论课上的知识有了更深的印象。

实体完整性实验

实验目的

掌握参照完整性的定义和维护方法。

实验重点和难点

实验重点：创建表时定义参照完整性。

实验难点：参照完整性的违约处理定义。

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。

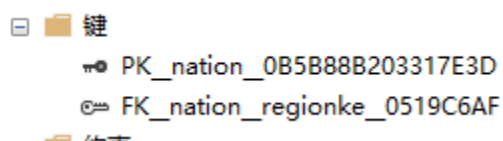
实验操作和结果

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(25),  
    comment VARCHAR(152)  
);  
  
CREATE TABLE nation(  
    nationkey INTEGER PRIMARY KEY,  
    name CHAR (25),  
    regionkey INTEGER REFERENCES region(regionkey),  
    comment VARCHAR(152)  
);
```

结果如下，regionkey 被设置为外键：



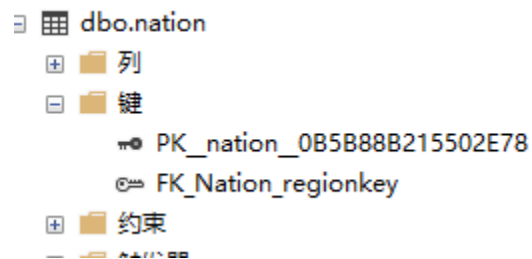
创建表后定义参照完整性

定义国家表的参照完整性。在重新定义之前需要删除之前的表。

```
CREATE TABLE nation(  
    nationkey INTEGER PRIMARY KEY,  
    name CHAR (25),  
    regionkey INTEGER,  
    comment VARCHAR(152)  
);  
  
ALTER TABLE nation
```

```
ADD CONSTRAINT FK_Nation_regionkey
FOREIGN KEY(regionkey) REFERENCES region(regionkey);
```

得到相同的结果如下：



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 PartSupp(
    partkey INTEGER,
    suppkey INTEGER,
    availqty INTEGER,
    supplycost REAL,
    comment VARCHAR (199),
    PRIMARY KEY(partkey,suppkey)
);
CREATE TABLE Lineitem(
    orderkey INTEGER,
    partkey INTEGER,
    suppkey 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(25),
    shipmode CHAR(10),
    comment VARCHAR(44),
    PRIMARY KEY(orderkey,linenummer),
    FOREIGN KEY(partkey,suppkey) REFERENCES PartSupp(partkey,suppkey)
);
```

结果如下，一共有两个外键：

dbo.Lineitem
列
orderkey (PK, int, not null)
partkey (FK, int, null)
suppkey (FK, int, null)

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(25),
  regionkey INTEGER,
  comment VARCHAR (152),
  CONSTRAINT FK_Nation_regionkey FOREIGN KEY(regionkey) REFERENCES
region(regionkey) ON DELETE SET NULL ON UPDATE SET NULL
);
```

结果如下。首先是 region 的定义是 notnull:

dbo.region
列
regionkey (PK, int, not null)

在 nation 中，regionkey 的值被设为可以为 null 值:

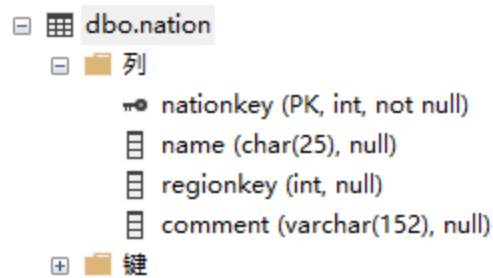
dbo.nation
列
nationkey (PK, int, not null)
name (char(25), null)
regionkey (FK, int, null)
comment (varchar(152), null)

删除参照完整性

删除国家表的外码

```
ALTER TABLE nation DROP CONSTRAINT FK_Nation_regionkey;
```

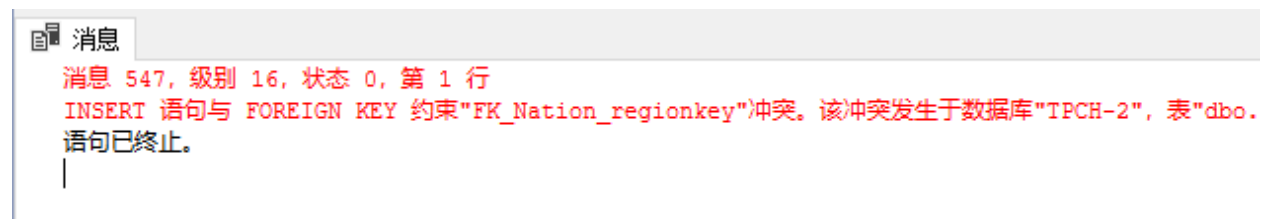
结果如下，可以看到外码属性已经删除:



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');
```

在重新设置好约束之后，插入该记录，结果如下：



可以看到在约束下并不能插入原 region 里没有的数据，完整性已经起了作用。

实验总结

通过这次实验，我对外键的作用理解更加深了。外键用于建立和加强两个表数据之间的链接的一列或多列。通过将保存表中主键值的一列或多列添加到另一个表中，可创建两个表之间的链接。这个列成为第二个表的外键。它的主要目的是控制存储在外键表中的数据和主键表中数据的修改。它是用来控制数据库中数据完整性的。同时，通过这次实验我还了解了多种不同的设置外键的方法，受益匪浅。