**数据库实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实 验 人：** | **伍建霖** | | **学 号：** | **20337251** | **日 期：** | **2022.10.26** |
| **院（系）：** | **计算机学院** | | | **专业（班级）：** | **网络空间安全** | |
| **联系方式：** | **13692259882** | | |  |  | |
| **实验题目：** | | **3.1 实体完整性实验** | | | | |

* + - 1. **实验目的**

掌握实体完整性的定义和维护方法。

* + - 1. **实验内容和要求**

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。

* + - 1. **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。

* + - 1. **实验工具**

MySQL、SQL Server、Navicat

* + - 1. **实验过程**
* 在SQL Server环境下，执行以下操作

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 Test\_Supplier1(

suppkey INTEGER CONSTRAINT PK\_supplier PRIMARY KEY,

name CHAR(25) ,

address VARCHAR(40) ,

nationkey INTEGER,

phone CHAR(15) ,

acctbal REAL,

comment VARCHAR(101)

) ;

截图如下：



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 Test\_Supplier2(

suppkey INTEGER ,

name CHAR(25) ,

address VARCHAR(40) ,

nationkey INTEGER,

phone CHAR(15) ,

acctbal REAL,

comment VARCHAR(101),

CONSTRAINT PK\_supplier2 PRIMARY KEY (suppkey)

) ;

截图如下：



3. 创建表后定义实体完整性

定义供应商表的实体完整性。

CREATE TABLE Test\_Supplier3(

suppkey INTEGER not NULL,

name CHAR(25) ,

address VARCHAR(40) ,

nationkey INTEGER,

phone CHAR(15) ,

acctbal REAL,

comment VARCHAR(101) );

ALTER TABLE Test\_Supplier3 /\*再修改供应商表,增加实体完整性\*/

ADD CONSTRAINT PK\_Supplir3 PRIMARY KEY(suppkey);

在SQL Server中，只能将NOT NULL的列设置为主码，所以在创建表时suppkey列不允许空值。

截图如下：



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。主码由partkey和suppkey组成，实体完整性需定义在表级。

CREATE TABLE Test\_PartSupp(

partkey INTEGER,

suppkey INTEGER,

availqty INTEGER,

supplycost REAL,

comment VARCHAR(199) ,

PRIMARY KEY(partkey, suppkey) );

/\*主码由多个属性组成,实体完整性必须定义在表级\*/

截图如下：



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 Test\_nation(

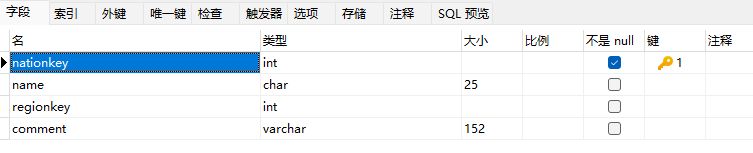
nationkey INTEGER CONSTRAINT PK\_nation PRIMARY KEY,

name CHAR(25) UNIQUE,

regionkey INTEGER,

comment VARCHAR(152) ) ;

截图如下：



6. 删除实体完整性

删除国家实体的主码。

ALTER TABLE Test\_nation DROP CONSTRAINT PK\_nation;

截图如下：



7. 增加两条相同记录，验证实体完整性是否起作用。

/\*插入两条主码相同的记录就会违反实体完整性约束\*/

插入第一条记录，主码值为11。

INSERT INTO Test\_Supplier1(suppkey, name, address, nationkey, phone, acctbal, comment)

VALUES(11, 'test 1', 'test 1', 101, '12345678', 0.0, 'test 1');

插入结果如下：



插入两条主码相同的记录会违反实体完整性约束，接下来插入第二条记录，主码值为11。

INSERT INTO Test\_Supplier1(suppkey, name, address, nationkey, phone, acctbal, comment)

VALUES(11, 'test 2', 'test 2', 102, '23456789', 0.0, 'test 2');

违反实体完整性约束，拒绝执行。

截图如下：



* + - 1. **与实验结果相关的文件**

无

* + - 1. **思考题**

（1）所有列级完整性约束都可以改写为表级完整性约束，而表级完整性约束不一定能改写成列级完整性约束。请举例说明。

答：多属性主码就只能通过表级完整性约束定义，不能改写成列级完整性约束定义。

* + - 1. **实验总结**