**数据库实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验人：** | **凌国明** | | **学号：** | **21307077** | **日 期：** | **2023年10月26日** |
| **院（系）：** | **计算机学院** | | | **专业（班级）：** | **计算机科学与技术** | |
| **联系方式：** |  | | |  |  | |
| **实验题目：** | | **3.2 参照完整性实验** | | | | |

* + - 1. **实验目的**

掌握参照完整性的定义和维护方法。

* + - 1. **实验内容和要求**

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。

* + - 1. **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。

* + - 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 T32\_region(

regionkey INTEGER PRIMARY KEY,

name CHAR(25) ,

comment VARCHAR(152) ) ;

CREATE TABLE T32\_nation(

nationkey INTEGER PRIMARY KEY,

name CHAR(25) ,

regionkey INTEGER REFERENCES T32\_Region(regionkey) ,/\*列级参照完整性\*/

comment VARCHAR(152) ) ;

验证截图如下：

或者

CREATE TABLE T32\_nation2(

nationkey INTEGER PRIMARY KEY,

name CHAR(25) ,

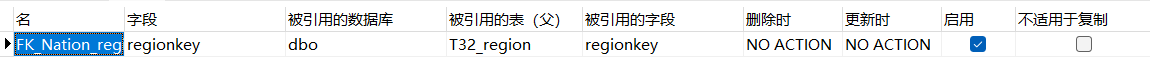
regionkey INTEGER,

comment VARCHAR(152),

CONSTRAINT FK\_Nation\_regionkey FOREIGN KEY(regionkey) REFERENCES T32\_region(regionkey) /\*表级参照完整性\*/

) ;

验证截图如下：



2. 创建表后定义参照完整性

定义国家表的参照完整性。创建表后用ALTER TABLE语句修改完整性限制。

CREATE TABLE T32\_nation3(

nationkey INTEGER PRIMARY KEY,

name CHAR(25) ,

regionkey INTEGER,

comment VARCHAR(152)

) ;

ALTER TABLE T32\_nation3

ADD CONSTRAINT FK\_Nation\_regionkey3 FOREIGN KEY(regionkey) REFERENCES T32\_region(regionkey)

验证截图如下：



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。（partkey，suppkey）是被参照关系PartSupp的主码，同时是参照关系LineItem的外码。

CREATE TABLE T32\_PartSupp(

partkey INTEGER,

suppkey INTEGER,

availqty INTEGER,

supplycost REAL,

comment VARCHAR(199),

PRIMARY KEY(partkey, suppkey) );

CREATE TABLE T32\_Lineitem(

orderkey INTEGER REFERENCES Orders(orderkey) ,

partkey INTEGER REFERENCES Part(partkey),

suppkey INTEGER REFERENCES Supplier(suppkey) ,

linenumber 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, linenumber) ,

FOREIGN KEY(partkey, suppkey) REFERENCES T32\_PartSupp(partkey, suppkey) ) ;

验证截图如下：



4. 定义参照完整性的违反约束

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 T32\_nation4(

nationkey INTEGER PRIMARY KEY,

name CHAR(25) ,

regionkey INTEGER,

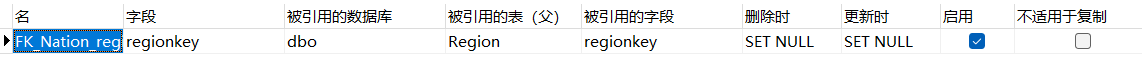
comment VARCHAR(152),

CONSTRAINT FK\_Nation\_regionkey4 FOREIGN KEY(regionkey) REFERENCES Region(regionkey)

ON DELETE SET NULL ON UPDATE SET NULL

);

验证截图如下：

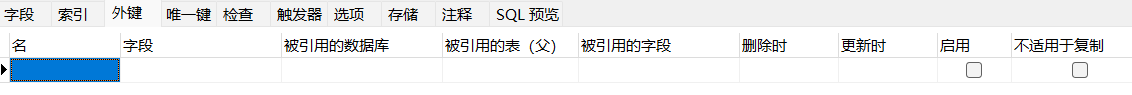


5. 删除参照完整性。

删除国家表的外码：

ALTER TABLE T32\_nation4 DROP CONSTRAINT FK\_Nation\_regionkey4;

验证截图如下：



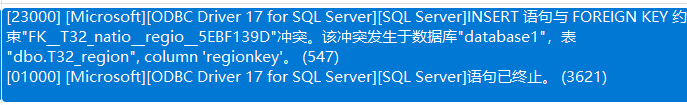
6. 插入一条国家记录，验证参照完整性是否起作用

/\*插入一条国家记录,如果'1001'号地区记录不存在,违反参照完整性约束。\*/

INSERT INTO T32\_Nation(nationkey, name, regionkey, comment)

VALUES(1001, 'nation 1', 1001, 'comment 1') ;

验证截图如下：违反完整性约束，拒绝执行



输入以下信息，先使'1001'号地区记录存在，再插入国家记录，不违法约束，成功执行。

INSERT INTO T32\_region(regionkey, name, comment)

VALUES(1001, 'region 1', 'comment 1') ;

INSERT INTO T32\_Nation(nationkey, name, regionkey, comment)

VALUES(1001, 'nation 1', 1001, 'comment 1') ;

* + - 1. **与实验结果相关的文件**

无

* + - 1. **思考题**

对于自引用表，例如课程表(课程号、课程名、先修课程号，学分)中的先修课程号引用该表的课程号，请完成如下任务：

(1)写出课程表上的实体完整性和参照完整性。

答：

主键为course\_id，pre\_course\_id外键引用（自引用）Course的course\_id

CREATE TABLE Course(

course\_id INTEGER CONSTRAINT c PRIMARY KEY,

course\_name VARCHAR(20),

pre\_course\_id INTEGER REFERENCES Course(course\_id),

credit INTEGER

);

(2)在考虑实体完整性约束的情况下，试举出几种录入课程数据的方法。

答：

没有先修课程的课程，可以直接插入

INSERT INTO Course(course\_id, course\_name, pre\_course\_id, credit)

VALUES(123, ‘abc’, NULL, 4);

有先修课程的课程，要先保证先修课程已被录入。可以采用递归的方式，递归地寻找先修课程：如果一个课程的先修课程没有在库中，则先录入这个先修课程，如果这个先修课程的先修课程没有在库中，递归地执行相同操作。

* + - 1. **实验总结**

本次实验深化了我的理论和实际应用对于参照完整性的理解。参照完整性是确保外键值的有效性，从而在不同表之间维护数据的逻辑一致性的关键因素。在实验中，我探索了两种不同的方式来定义外键约束，一种是在创建表时直接定义外键约束，另一种是在创建表后通过ALTER TABLE语句添加外键约束。

通过实际编写SQL语句并验证数据操作，我更加深入地理解了参照完整性的重要性和操作方式。特别是在设置参照完整性违约行为为SET NULL的实践中，我学会了如何在保持数据完整性的同时，处理可能出现的数据删除或更新问题。在插入记录时，我也遇到了违反参照完整性约束的情况，这提醒我在执行数据插入前必须确保所有的外键约束都得到满足。这个实践强调了在进行数据库操作之前进行逻辑检查的重要性，以确保数据的一致性和完整性。

最后，通过删除外键约束的操作，我学到了如何灵活地管理数据库的结构变化，以适应可能的需求变动。这对于数据库的维护和演化至关重要，使数据库能够适应不断变化的业务需求。

总之，这次实验不仅增强了我的理论知识，还提供了实际操作的机会，让我更加深入地理解了参照完整性的概念和实际应用。这将对我今后在数据库设计和管理方面的工作产生积极影响。