**数据库实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验人：** | **凌国明** | | **学号：** | **21307077** | **日 期：** | **2023年11月2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：** | | **3.3 用户自定义完整性实验** | | | | |

* + - 1. **实验目的**

掌握用户自定义完整性的定义和维护方法。

* + - 1. **实验内容和要求**

针对具体应用语义，选择NULL/NOT NULL、DEFAULT、UNIQUE、CHECK等， 定义属性上的约束条件。

* + - 1. **实验重点和难点**

实验重点：NULL/NOT NULL， DEFAULT。

实验难点：CHECK

* + - 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) 定义属性NULL/NOT NULL约束

定义地区表各属性的NULL/NOT NULL属性。

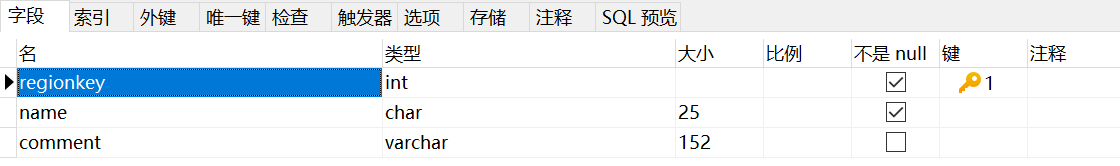
CREATE TABLE T33\_region(

regionkey INTEGER NOT NULL PRIMARY KEY,

name CHAR(25) NOT NULL,

comment VARCHAR(152) NULL)

验证截图如下：



(2) 定义属性DEFAULT约束

定义国家表的regionkey的缺省属性值为0值, 表示其他地区。

CREATE TABLE T33\_nation(

nationkey INTEGER PRIMARY KEY,

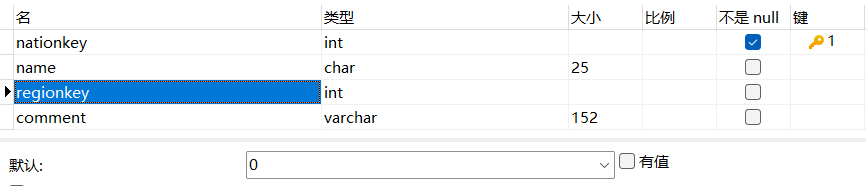
name CHAR(25) ,

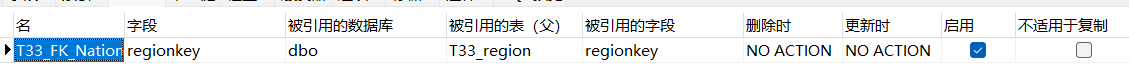
regionkey INTEGER DEFAULT 0,

comment VARCHAR(152) ,

CONSTRAINT T33\_FK\_Nation\_regionkey FOREIGN KEY(regionkey) REFERENCES T33\_Region(regionkey) ) ;

验证截图如下：





(3) 定义属性UNIQUE约束

定义国家表的名称属性必须唯一的完整性约束。

CREATE TABLE T33\_nation2(

nationkey INTEGER PRIMARY KEY,

name CHAR(25) UNIQUE,

regionkey INTEGER,

comment VARCHAR(152) ) ;

验证截图如下：



(4) 使用CHECK

使用CHECK定义订单项目表中某些属性应该满足的约束

CREATE TABLE T33\_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) ,

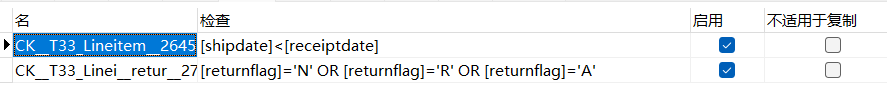
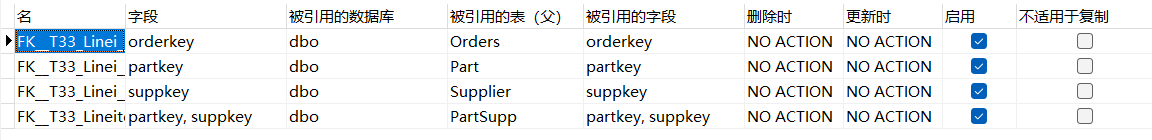
PRIMARY KEY(orderkey, linenumber) ,

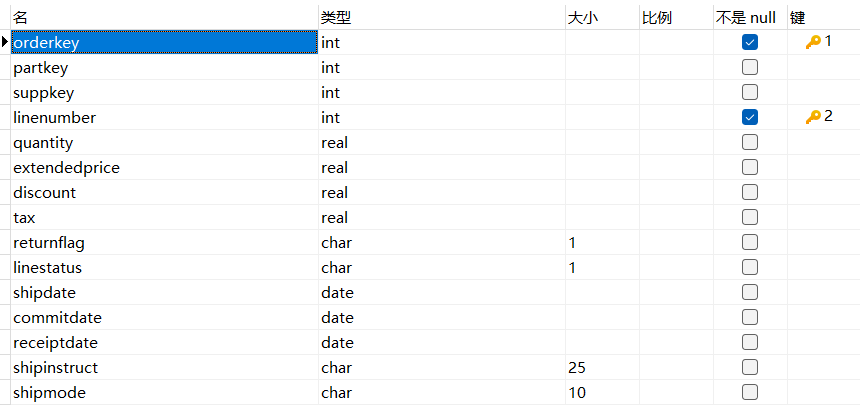
FOREIGN KEY(partkey, suppkey) REFERENCES PartSupp(partkey, suppkey) ,

CHECK(shipdate<receiptdate) ,/\*装运日期<签收日期\*/

CHECK(returnflag IN('A', 'R', 'N')));/\*退货标记为A或R或N\*/

验证截图如下：



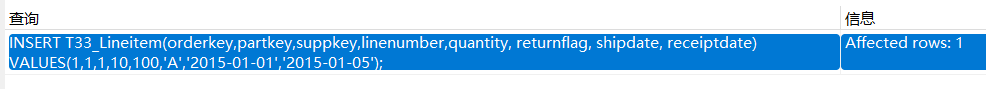
(5) 修改Lineitem的一条记录验证是否违反CHECK约束

先插入一条记录：

INSERT T33\_Lineitem(orderkey,partkey,suppkey,linenumber,quantity, returnflag, shipdate, receiptdate)

VALUES(1,1,1,10,100,'A','2015-01-01','2015-01-05');

验证截图如下：



然后修改这条记录：

UPDATE T33\_Lineitem SET shipdate='2015-01-05', receiptdate='2015-01-01'

WHERE orderkey= 1 AND linenumber=10;

验证截图如下：



* + - 1. **与实验结果相关的文件**

无

* + - 1. **思考题**

(1)请分析哪些完整性约束只针对单个属性，哪些完整性约束可以针对多个属性?哪些只针对一个表，哪些针对多个表?

答：

只针对单个属性的完整性约束：NOT NULL 约束、DEFAULT 约束、 可针对多个属性的完整性约束：CHECK、UNIQUE、主键约束、外键约束。

外键约束针对多个表，上述其他约束只针对一个表

(2)对表中某一列数据类型进行修改时，要修改的列是否必须为空列?

答：不一定，但是要注意约束和索引：如果该列是主键、外键、参与索引或有特定的约束（如 CHECK），在更改数据类型前，可能需要先删除这些约束和索引。而且如果列中已经包含数据，那么在转换为新的数据类型时，现有数据必须能够有效地转换为新类型。

* + - 1. **实验总结**

本次实验深化了我对用户自定义数据完整性的理解和应用。我通过定义和操作 SQL Server 中的不同数据完整性约束，如NULL/NOT NULL、DEFAULT、UNIQUE、和CHECK，重点在于掌握如何根据业务逻辑的需求来施加和维护数据库的数据完整性。

实验的核心是通过实际操作，更深入地理解了基础约束，如NOT NULL和DEFAULT。同时，通过应用CHECK约束，解决了如何在保证数据质量的前提下，实现复杂的业务规则验证。特别是在CHECK约束的设置和测试中，实验展示了如何强制执行逻辑规则，例如确保发货日期必须早于接收日期，这突出了在数据库设计和管理中，如何有效利用约束来实施复杂业务规则的重要性。

在实际实验过程中，我还了解了当违反这些约束时，数据库会如何响应，以及如何通过错误消息来诊断和修正问题。通过这些实践，我不仅学会了如何定义数据完整性约束，还理解了如何在确保这些约束有效执行的同时，处理可能出现的数据完整性冲突。这对于建立和维护健壮的数据库系统至关重要，确保数据的一致性和准确性。这次实验为我加强数据库管理和设计方面的知识和技能，提供了宝贵的实践经验。