《数据库技术实验》报告

|  |  |  |
| --- | --- | --- |
| **第 七 次实验** | **日期：2022-11-16** | **得分：** |
| **学号：** | **姓名：*Steven*** | **专业：智能科学与技术** |

一、实验目的

1. 掌握实现实体完整性和参照完整性的SQL语句

2. 了解实现用户定义完整性及触发器的SQL语句

二、实验内容

（一）实验题

1. 在Student\_Course数据库上，为Student、Course、SC表定义实体完整性、参照完整性（包括违约处理）、用户定义完整性（包括列非空、列值唯一），并验证这些完整性约束是否起效。

2. 在TCP\_H数据库上，设计一个完整性约束，采用触发器实现该完整性约束，并验证该完整性约束是否起效。

（二）思考题

1. 在Student\_Course数据库中，如果Student表删除元组或修改sno值，导致SC表违反参照完整性，可以如何设置违约处理？

2. 在Student\_Course数据库中，Course表的cpno列参照了Course表的cno列，应该如何录入Course表中的数据？

三、实验过程及实验结果

1. 既然为了验证，便首先将所有已有的约束删除，重新建立并验证。

(1) 清空数据库中的所有约束

删除所有约束后，表间关系如图所示（该图使用DataGrip自动生成）：

图示, 表格

描述已自动生成

（既无外键依赖，也无表内主键、索引，此时就是三个独立的表）

(2) 依次定义实体完整性、参照完整性、用户定义完整性

|  |
| --- |
| *-- 定义实体完整性*  use student\_course;ALTER TABLE student  ADD PRIMARY KEY (sno); ALTER TABLE course  ADD PRIMARY KEY (cno); ALTER TABLE sc  ADD PRIMARY KEY (sno, cno); |
| *-- 定义参照完整性*  ALTER TABLE sc  ADD FOREIGN KEY (sno) REFERENCES student (sno) ON UPDATE CASCADE;  ALTER TABLE sc  ADD FOREIGN KEY (cno) REFERENCES course (cno) ON UPDATE CASCADE;  ALTER TABLE course  ADD FOREIGN KEY (cpno) REFERENCES course (cno) ON UPDATE CASCADE; |
| *-- 定义用户定义完整性*  *-- 课程号不能为null，且应该是唯一的* ALTER TABLE course  MODIFY cno char(11) NOT NULL UNIQUE ; *-- 学分不能为null* ALTER TABLE course  MODIFY ccredit SMALLINT NOT NULL; *-- 学号不能为null，且应该是唯一的* ALTER TABLE student  MODIFY sno char(11) NOT NULL UNIQUE ; *-- 年龄不能为null* ALTER TABLE student  MODIFY sage smallint NOT NULL ; *-- 所属的系部不能为null* ALTER TABLE student  MODIFY sdept char(20) NOT NULL ; |

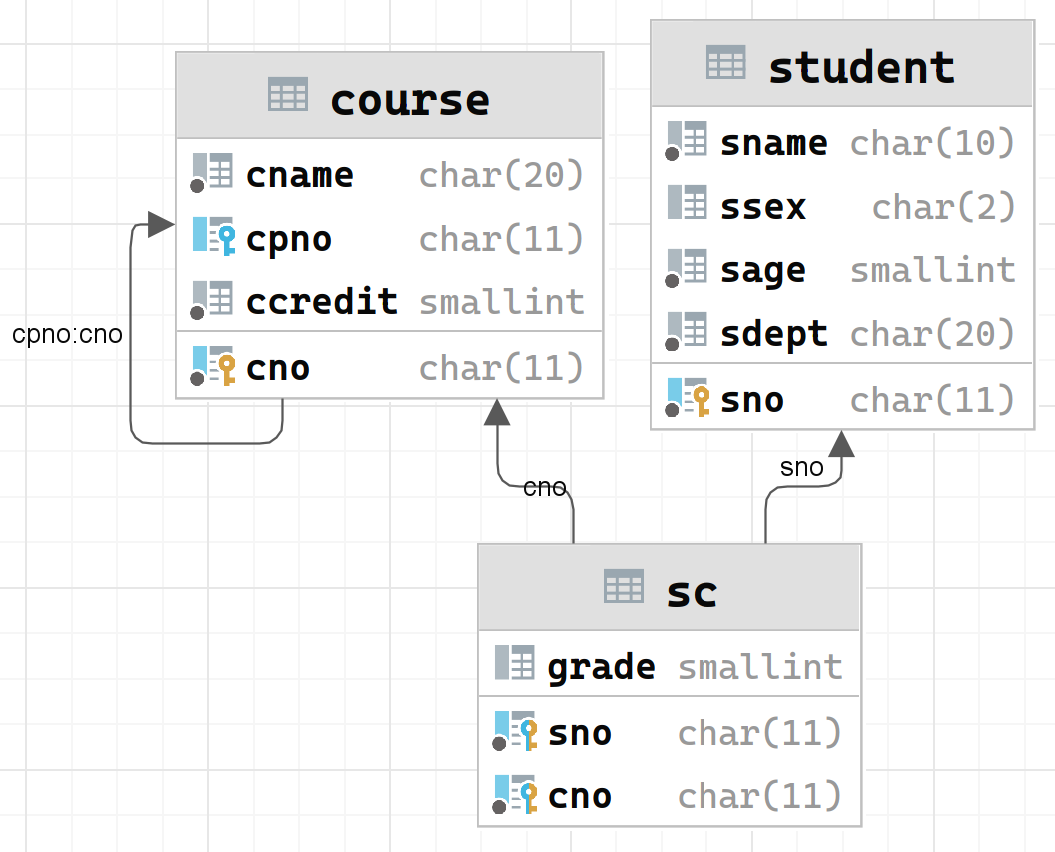
* 实体完整性：设置能确定单个对象的最小属性集
* 参照完整性：主要考虑到以下两点

逻辑上sc.sno参照student.sno，sc.cno参照course.cno，course.cpno参照course.cno，所以需要建立三组外键。

在违约处理上，设置了ON UPDATE CASCADE、ON DELETE NO ACTION。这样当出现输入错误或选课信息变动等情况时，只需要修改被参照表，sc中也会进行对应的变化；当被参照表删除数据时，为尽量避免误删，设置此时不允许操作。

* 用户定义完整性：考虑到部分数据的通常特征，对其额外设置一些约束。

(3) 完整性约束建立完成



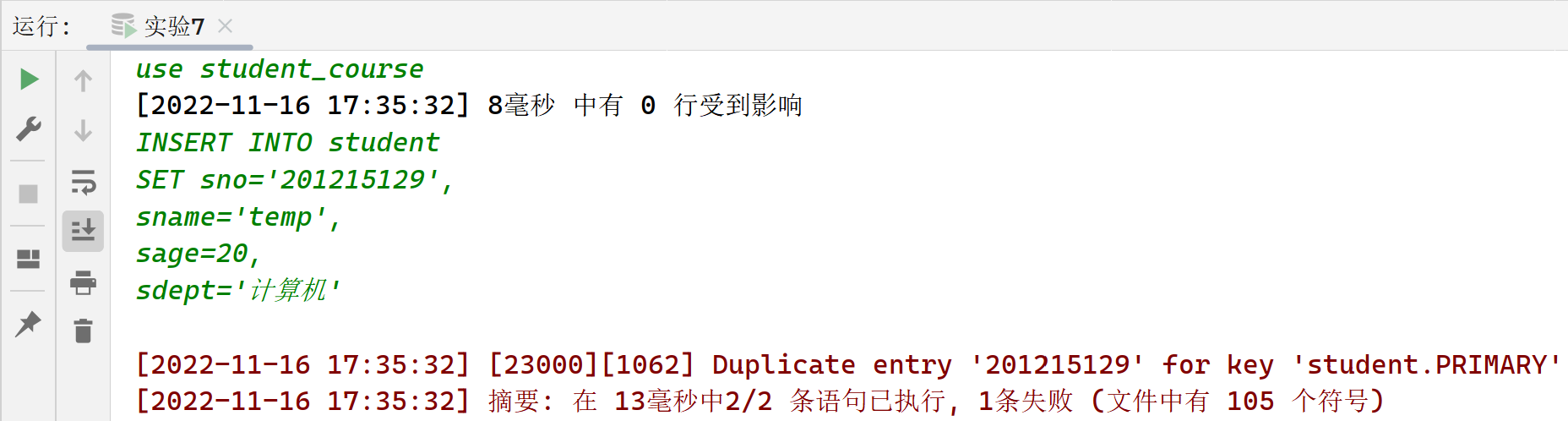
（此时完整性约束建立完成，如图所属）

(4) 测试完整性约束是否起作用

① 实体完整性：

表格

描述已自动生成



（student中已有赵新的sno=201215129，此时再次插入相同的主码的对象，插入失败，提示Duplicate entry，说明实体完整性生效）

② 参照完整性：

图形用户界面, 表格

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

（cs.sno的被依赖列student.sno中没有’test’，根据参照完整性，这一行无法插入，说明参照完整性生效）

③ 用户定义完整性：

图形用户界面

描述已自动生成

文本

描述已自动生成

（新插入的数据并不违反course的实体完整性和参照完整性，但是ccredit=null违背了用户定义的完整性，无法插入，说明用户定义完整性生效）

2. 在TCP\_H数据库上，设计一个完整性约束，采用触发器实现该完整性约束，并验证该完整性约束是否起效。

图示

描述已自动生成

(1) 根据该图，添加主键和外键结果如下：

表格

描述已自动生成

(2) 设计触发器

设计个较为简单的触发器：对于supplier，在修改后，检测其phone的格式是否正确，不正确则设置为null

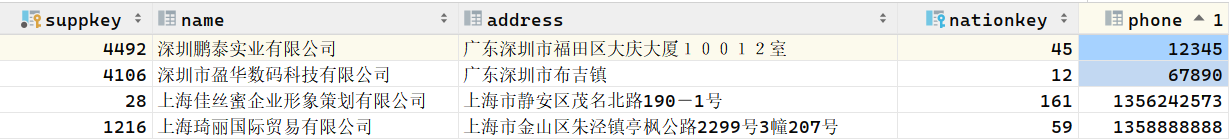
|  |
| --- |
| DELIMITER $$ CREATE TRIGGER check\_phone\_format  BEFORE UPDATE  ON supplier  FOR EACH ROW BEGIN  IF *CONVERT*(NEW.phone, char) NOT REGEXP '^1\d{10}$' THEN  SET NEW.phone = null;  END IF; END$$ DELIMITER ; |

(3) 测试

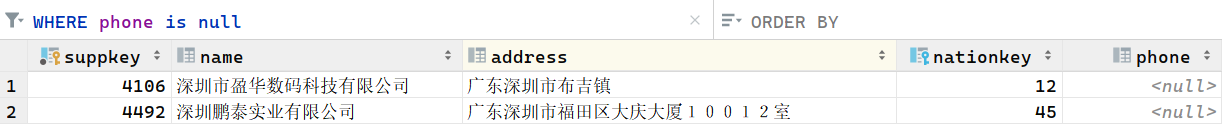
图形用户界面, 文本

描述已自动生成

此时根据phone降序排列，可以看到前7个对象的phone都是不符合规则的，若更新后的phone仍然不符合规则，则会将其设为null。



为了测试Trigger是否起作用，特意将两个phone设置的规则不符合正则表达式。



提交完成后，这两个对应的phone都被自动设为了null，说明触发器在起作用。

四、思考题

1. 在Student\_Course数据库中，如果Student表删除元组或修改sno值，导致SC表违反参照完整性，可以如何设置违约处理？

一共有5种违约处理方式，分别是NO\_ACTION（不允许进行处理）、RESTRICT（限制操作）、SET\_NULL（将依赖表中对应项设为null）、SET\_DEFAULT（将依赖表中对应项设为该列的默认值）、CASCADE（级联，即依赖表同步进行修改）

2. 在Student\_Course数据库中，Course表的cpno列参照了Course表的cno列，应该如何录入Course表中的数据？

先输入cpno为null的所有课，然后逐步输入cpno已存在的项，否则将会由于参照完整性无法输入。