lab3 实验报告

PB13011079 杨智

实验环境:

操作系统: mac osx 10.10
dbms: mysql 5.7

实验要求:

1、PL/SQL实验
（1）创建三个表：Student(S#, Sname, age), Course(C#, cname, credit), SC(s#, c#, score)，其中SC的S#和C#都是外键，分别引用Student表的S#和Course表的C#。请首先在各自表中插入若干条记录，然后用存储过程实现修改指定学生的学号。
（2）创建两个表：系表：Dept(D#, Dname, S_Count) ，其中S_count是每个系的学生人数；学生表：Stu(S#, Sname, age, D#) 其中D#是引用Dept(D#)的外键。请用触发器实现S_count和学生表中实际人数的一致性。

实验说明:

part(1)

根据实验要求，可以创建如下三个表:

```
CREATE TABLE IF NOT EXISTS STUDENT (
        SID INT(11) PRIMARY KEY AUTO_INCREMENT,
        SNAME VARCHAR(20) NOT NULL,
        AGE INT(10) NOT NULL
        );

CREATE TABLE IF NOT EXISTS COURSE (
        CID INT(11) PRIMARY KEY AUTO_INCREMENT,
        CNAME VARCHAR(20) NOT NULL,
        CREADIT INT(10) NOT NULL
        );

CREATE TABLE IF NOT EXISTS SC (
        SID INT(11) NOT NULL,
        CID INT(11) NOT NULL,
        SCORE INT(10) NOT NULL,
        FOREIGN KEY FO_S (SID) REFERENCES STUDENT(SID) ON DELETE CASCADE ON
UPDATE CASCADE,
        FOREIGN KEY FO_C (CID) REFERENCES COURSE(CID) ON DELETE CASCADE ON
UPDATE CASCADE
        );
```

其中SC的SID和CID都是外键，分别引用Student表的SID和Course表的CID。

根据实验要求，再创建一个存储过程：

```
DELIMITER $$
CREATE PROCEDURE CHANGE_SID(IN OLD_SID INT, IN NEW_SID INT)
BEGIN
        UPDATE STUDENT SET SID = NEW_SID WHERE SID = OLD_SID;
END;
$$
DELIMITER ;
```

注：在mysql中，parsing的时候遇到分号就会判断为语句结束，所以这里要先用“DELIMITER $$”把语句结束符更换为“$$”，然后再CREATE PROCEDURE ……

part(2)

根据实验要求，创建两个表：

```
CREATE TABLE IF NOT EXISTS DEPT (
        DID INT(11) PRIMARY KEY AUTO_INCREMENT,
        DNAME VARCHAR(20) NOT NULL,
        S_COUNT INT(11) NOT NULL
        );

CREATE TABLE IF NOT EXISTS STU (
        SID INT(11) PRIMARY KEY AUTO_INCREMENT,
        SNAME VARCHAR(20) NOT NULL,
        AGE INT (10) NOT NULL,
        DID INT (11) NOT NULL,
        FOREIGN KEY FO_D (DID) REFERENCES DEPT(DID) ON DELETE CASCADE ON
UPDATE CASCADE
```

```
    );
```

然后创建三个触发器，分别对应insert、update、delete操作：

```
DROP TRIGGER IF EXISTS INSERT_S_COUNT;

DELIMITER $$
CREATE TRIGGER INSERT_S_COUNT AFTER INSERT ON STU FOR EACH ROW
BEGIN
    UPDATE DEPT SET S_COUNT = S_COUNT + 1 WHERE NEW.DID = DID;
END
$$
DELIMITER ;

DROP TRIGGER IF EXISTS UPDATE_S_COUNT;

DELIMITER $$
CREATE TRIGGER UPDATE_S_COUNT AFTER UPDATE ON STU FOR EACH ROW
BEGIN
    UPDATE DEPT SET S_COUNT = S_COUNT + 1 WHERE NEW.DID = DID;
    UPDATE DEPT SET S_COUNT = S_COUNT - 1 WHERE OLD.DID = DID;
END
$$
DELIMITER ;

DROP TRIGGER IF EXISTS DELETE_S_COUNT;

DELIMITER $$
CREATE TRIGGER DELETE_S_COUNT AFTER UPDATE ON STU FOR EACH ROW
BEGIN
    UPDATE DEPT SET S_COUNT = S_COUNT - 1 WHERE OLD.DID = DID;
END
$$
DELIMITER ;
```

实验结果：

把part(1)的所有SQL语句和测试语句都写到part1.sql中：

```
DROP DATABASE LAB3;
CREATE DATABASE LAB3;
USE LAB3;

CREATE TABLE IF NOT EXISTS STUDENT (
    SID INT(11) PRIMARY KEY AUTO_INCREMENT,
```

```
	SNAME VARCHAR(20) NOT NULL,
	AGE INT(10) NOT NULL
	);

CREATE TABLE IF NOT EXISTS COURSE (
	CID INT(11) PRIMARY KEY AUTO_INCREMENT,
	CNAME VARCHAR(20) NOT NULL,
	CREADIT INT(10) NOT NULL
	);

CREATE TABLE IF NOT EXISTS SC (
	SID INT(11) NOT NULL,
	CID INT(11) NOT NULL,
	SCORE INT(10) NOT NULL,
	FOREIGN KEY FO_S (SID) REFERENCES STUDENT(SID) ON DELETE CASCADE ON
UPDATE CASCADE,
	FOREIGN KEY FO_C (CID) REFERENCES COURSE(CID) ON DELETE CASCADE ON
UPDATE CASCADE
	);

INSERT INTO STUDENT VALUES
	(NULL, 'AAA', 17),
	(NULL, 'BBB', 18),
	(NULL, 'CCC', 19);

INSERT INTO COURSE VALUES
	(NULL, 'MATH', 17),
	(NULL, 'PHYSICS', 18);

INSERT INTO SC VALUES
	(1, 1, 100),
	(1, 2, 99),
	(2, 1, 98),
	(2, 2, 97),
	(3, 1, 96),
	(3, 2, 95);

SELECT * FROM STUDENT;
SELECT * FROM COURSE;
SELECT * FROM SC;

/*
INSERT INTO SC VALUES
	(4, 1, 88);
*/

DELIMITER $$
CREATE PROCEDURE CHANGE_SID(IN OLD_SID INT, IN NEW_SID INT)
BEGIN
	UPDATE STUDENT SET SID = NEW_SID WHERE SID = OLD_SID;
END;
$$
DELIMITER ;

CALL CHANGE_SID(3, 9);
```

SELECT * FROM STUDENT;
SELECT * FROM COURSE;
SELECT * FROM SC;

测试结果如下：

```
mysql> source ./part1.sql
Query OK, 2 rows affected (0.12 sec)

Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

+-----+-------+-----+
| SID | SNAME | AGE |
+-----+-------+-----+
|   1 | AAA   |  17 |
|   2 | BBB   |  18 |
|   3 | CCC   |  19 |
+-----+-------+-----+
3 rows in set (0.00 sec)

+-----+---------+---------+
| CID | CNAME   | CREADIT |
+-----+---------+---------+
|   1 | MATH    |      17 |
|   2 | PHYSICS |      18 |
+-----+---------+---------+
2 rows in set (0.00 sec)

+-----+-----+-------+
```

```
| SID | CID | SCORE |
+-----+-----+-------+
|   1 |   1 |   100 |
|   1 |   2 |    99 |
|   2 |   1 |    98 |
|   2 |   2 |    97 |
|   3 |   1 |    96 |
|   3 |   2 |    95 |
+-----+-----+-------+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

+-----+-------+-----+
| SID | SNAME | AGE |
+-----+-------+-----+
|   1 | AAA   |  17 |
|   2 | BBB   |  18 |
|   9 | CCC   |  19 |
+-----+-------+-----+
3 rows in set (0.00 sec)

+-----+---------+---------+
| CID | CNAME   | CREADIT |
+-----+---------+---------+
|   1 | MATH    |      17 |
|   2 | PHYSICS |      18 |
+-----+---------+---------+
2 rows in set (0.00 sec)

+-----+-----+-------+
| SID | CID | SCORE |
+-----+-----+-------+
|   1 |   1 |   100 |
|   1 |   2 |    99 |
|   2 |   1 |    98 |
|   2 |   2 |    97 |
|   9 |   1 |    96 |
|   9 |   2 |    95 |
+-----+-----+-------+
6 rows in set (0.00 sec)
```

把part(2)的所有SQL语句和测试语句都写到part2.sql中：

```
DROP DATABASE LAB3;
CREATE DATABASE LAB3;
USE LAB3;

CREATE TABLE IF NOT EXISTS DEPT (
	DID INT(11) PRIMARY KEY AUTO_INCREMENT,
	DNAME VARCHAR(20) NOT NULL,
	S_COUNT INT(11) NOT NULL
	);

CREATE TABLE IF NOT EXISTS STU (
	SID INT(11) PRIMARY KEY AUTO_INCREMENT,
	SNAME VARCHAR(20) NOT NULL,
	AGE INT (10) NOT NULL,
	DID INT (11) NOT NULL,
	FOREIGN KEY FO_D (DID) REFERENCES DEPT(DID) ON DELETE CASCADE ON
UPDATE CASCADE
	);

DROP TRIGGER IF EXISTS INSERT_S_COUNT;

DELIMITER $$
CREATE TRIGGER INSERT_S_COUNT AFTER INSERT ON STU FOR EACH ROW
BEGIN
	UPDATE DEPT SET S_COUNT = S_COUNT + 1 WHERE NEW.DID = DID;
END
$$
DELIMITER ;

DROP TRIGGER IF EXISTS UPDATE_S_COUNT;

DELIMITER $$
CREATE TRIGGER UPDATE_S_COUNT AFTER UPDATE ON STU FOR EACH ROW
BEGIN
	UPDATE DEPT SET S_COUNT = S_COUNT + 1 WHERE NEW.DID = DID;
	UPDATE DEPT SET S_COUNT = S_COUNT - 1 WHERE OLD.DID = DID;
END
$$
DELIMITER ;

DROP TRIGGER IF EXISTS DELETE_S_COUNT;

DELIMITER $$
CREATE TRIGGER DELETE_S_COUNT AFTER UPDATE ON STU FOR EACH ROW
BEGIN
	UPDATE DEPT SET S_COUNT = S_COUNT - 1 WHERE OLD.DID = DID;
END
$$
DELIMITER ;

INSERT INTO DEPT VALUES
	(NULL, 'MATH', 0),
	(NULL, 'CHINESE', 0),
	(NULL, 'PHYSICS', 0);
```

```
INSERT INTO STU VALUES
        (NULL, 'STEVE', '18', 2),
        (NULL, 'DAVE', '18', 2);

SELECT * FROM STU;
SELECT * FROM DEPT;

UPDATE STU SET DID = 1 WHERE SID = 2;

SELECT * FROM STU;
SELECT * FROM DEPT;
```

测试结果如下:

```
mysql> source ./part2.sql
Query OK, 3 rows affected (0.02 sec)

Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

+-----+-------+-----+-----+
| SID | SNAME | AGE | DID |
+-----+-------+-----+-----+
|   1 | STEVE |  18 |   2 |
|   2 | DAVE  |  18 |   2 |
+-----+-------+-----+-----+
2 rows in set (0.00 sec)
```

```
+-----+---------+---------+
| DID | DNAME   | S_COUNT |
+-----+---------+---------+
|   1 | MATH    |       0 |
|   2 | CHINESE |       2 |
|   3 | PHYSICS |       0 |
+-----+---------+---------+
3 rows in set (0.00 sec)
```

```
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
+-----+-------+-----+-----+
| SID | SNAME | AGE | DID |
+-----+-------+-----+-----+
|   1 | STEVE |  18 |   2 |
|   2 | DAVE  |  18 |   1 |
+-----+-------+-----+-----+
2 rows in set (0.00 sec)
```

```
+-----+---------+---------+
| DID | DNAME   | S_COUNT |
+-----+---------+---------+
|   1 | MATH    |       1 |
|   2 | CHINESE |       0 |
|   3 | PHYSICS |       0 |
+-----+---------+---------+
3 rows in set (0.00 sec)
```

可见，实验结果是正确的。

实验总结：

这个实验没什么难度，只要熟悉了mysql的语法就好。