一. 实验目的

1. 熟悉SQL的**数据更新操作。**
2. 能够使用SQL语句对数据库进行**数据插入、更新、删除操作。**

二. 实验内容

在本次实验中,主要的内容是如何使用SQL语句对数据进行更新。

- 使用 INSERT INTO语句插入数据,包括插入一个元组或将子查询的结果插入到数据库中两种方式。
- 使用 SELECT INTO语句,产生一个新表并插入数据。
- 使用 UPDATE语句可以修改指定表中满足 WHERE子句条件的元组,有三种修改的方式:修改某一个元组的值,修改多个元组的值,带子查询的修改语句。
- 使用 DELETE语句删除数据:删除某一个元组的值,删除多个元组的值,带子查询的删除语句。

三. 实验结果

1. **向 STUDENTS表插入编号是“800022222”且姓名是“ WangLan”的元组。**

```
INSERT
INTO STUDENTS
VALUES ('800022222','WangLan',NULL,NULL)
```

结果如下:

查询添加后信息

```
SELECT * FROM STUDENTS WHERE sid = '800022222'
```

结果 消息

	sid	sname	email	grade
1	800022222	WangLan	NULL	NULL

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 1 行

2. **向 TEACHERS表插入元组(“200001000”,“LXL”,“s4zrck@pew,net”,“3024”)。**

```
INSERT  INTO TEACHERS  VALUES ('200001000','LXL','s4zrck@pew.net','3024')
```

结果如下:

消息

(1 行受影响)

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 0 行

查询添加后信息

```
SELECT *
FROM TEACHERS
WHERE tid = '200001000'
```

结果 消息

	tid	tname	email	salary
1	200001000	LXL	s4zrck@pew.net	3024

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 1 行

3. **将 TEACHERS表中编号为“200010493”的老师工资改为4000。**

```
UPDATE TEACHERS
SET salary = 4000
WHERE tid = '200010493'
```

结果如下：

消息

(1 行受影响)

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 0 行

查询修改后信息

```
SELECT *
FROM TEACHERS
WHERE tid = '200010493'
```

结果 消息

	tid	tname	email	salary
1	200010493	xzgkflko	zcf_m@def.com	4000

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 1 行

4. **将 TEACHERS表中所有工资小于2500的老师工资改为2500。**

```
UPDATE TEACHERS
SET salary = 2500
WHERE salary < 2500
```

结果如下：

消息

(5814 行受影响)

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 0 行

查询修改后信息

```
SELECT *
FROM TEACHERS
WHERE salary <= ALL(
    SELECT salary
    FROM TEACHERS
    WHERE salary IS NOT NULL
)
```

结果 消息

	tid	tname	email	salary
1	200005322	udjom	gd5l8@ppmlf.gov	2500
2	200050683	bmwdtty	uw__6@ozpe.edu	2500
3	200066763	btiagykw	dxd09@vxumi.net	2500
4	200067473	uakfqeog	hbcqo5q@tqgxc.net	2500
5	200120595	pejyw	zu9ee@zrjls.org	2500
6	200126787	zssildg	my4d3c@ozkc.org	2500
7	200132776	rdhznvtcl	kug3q@fyot.net	2500
8	200144197	uvcoff	gni8t7@zag.org	2500
9	200164470	pclrbeuwx	dn5b@czr.org	2500
10	200171064	ptmzib	fmr9co7@fjtk.org	2500

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 5815 行

5. **将编号为“200016731”的老师讲授的课程全部改成由姓名为“rnupx”的老师讲授**

```
UPDATE CHOICES
SET tid = (
    SELECT tid
    FROM TEACHERS
    WHERE tname = 'rnupx'
    )
WHERE tid = '200016731'
```

结果如下：

消息

(23 行受影响)

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 0 行

查询修改后tid=200016731的老师讲授的课程

```
SELECT tid ,cid
FROM CHOICES
WHERE tid = '200016731'
```

结果 消息

tid	cid

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School2 00:00:00 0 行

6. **更新编号“800071780”的学生年级为“2001”。**

```
UPDATE STUDENTS
SET grade = '2001'
WHERE sid = '800071780'
```

结果如下：

消息

(1 行受影响)

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 0 行

查询修改后学生信息

```
SELECT *
FROM STUDENTS
WHERE sid = '800071780'
```

结果 消息

	sid	sname	email	grade
1	800071780	iscmwink	htoactp@aof.gov	2001

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 1 行

7. **删除没有学生选修的课程。**

先查询一下有多少课程没人选修用于验证答案

```
SELECT COURSES.cname as'课程名称'
FROM COURSES
WHERE COURSES.cid NOT IN(
SELECT DISTINCT cid
FROM CHOICES
)
```

结果 消息

课程名称

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 0 行

```
DELETE
FROM COURSES
WHERE COURSES.cid NOT IN(
    SELECT DISTINCT cid
    FROM CHOICES
)
```

结果如下：

消息

(0 行受影响)

查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School | 00:00:00 | 0 行

8. **删除年级高于1998的学生信息。**

```
DELETE
FROM CHOICES
WHERE sid IN(
    SELECT sid
    FROM STUDENTS
    WHERE grade < '1998'
)
DELETE
FROM STUDENTS
WHERE grade < '1998'
```

结果如下：

消息

(139750 行受影响)

(46600 行受影响)

查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School2 | 00:00:01 | 0 行

查询一下删除结果

```
SELECT sid
FROM STUDENTS
WHERE grade < '1998'
```

结果 消息

sid

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 0 行

9. **删除没有选修课程的学生信息。**

先查询一下有多少学生没有选修课程

```
SELECT *
FROM STUDENTS
WHERE sid NOT IN (
    SELECT DISTINCT sid
    FROM CHOICES
)
```

结果 消息

	sid	sname	email	grade
1	800022222	WangLan	NULL	NULL

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School1 00:00:00 1 行

```
DELETE
FROM STUDENTS
WHERE sid NOT IN (
    SELECT DISTINCT sid
    FROM CHOICES
)
```

结果如下：

```
消息
(1 行受影响)
查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School1 | 00:00:00 | 0 行
```

10. **删除成绩不及格的选课记录。**

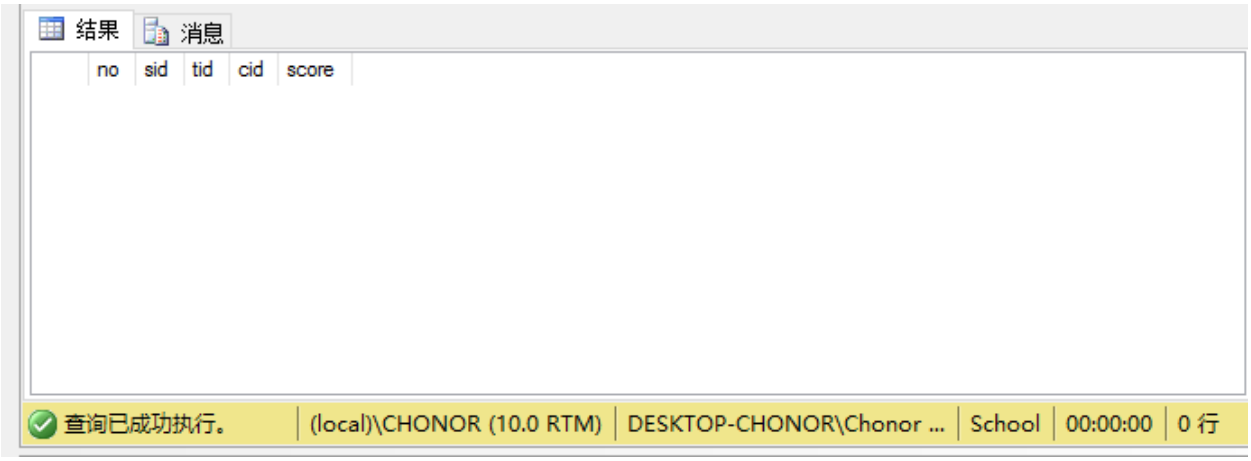
```sql
DELETE
FROM CHOICES
WHERE score < 60
```

结果如下：

```
消息
(19199 行受影响)
查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School2 | 00:00:00 | 0 行
```

验证一下结果

```sql
SELECT *
FROM CHOICES
WHERE score <= ALL(
    SELECT score
    FROM CHOICES
)
```



四. 实验感想

这次实验主要是增加、修改和删除。增加和修改都比较简单，增加只需要INSERT一下然后key对应就行了，插入其实就是就是查找然后SET。删除比较麻烦，因为一旦代码写错那么很有可能会出现误删的情况，同时因为会出现外键报错的情况，所以删除的时候可能需要先删除一张表的一部分才能删除我们想要的表，主要还是要理解一下外键约束的问题，理解之后就能正确的删除我们想要的部分了，删除之前最好选用SELECT看看我们要删除多少东西对比一下比较好。