> 邓语苏 2021级本科生
>
> 计算机科学与技术系
>
> 2024/4/18
>
> 数据库实验三

实验3.1 数据更新
- 实验内容
- 实验步骤

实验3.2
- 实验内容
- 实验步骤

实验3.3
- 实验内容
- 实验步骤

实验3.4
- 实验内容
- 实验步骤

实验3.5
- 实验内容
- 实验步骤

遇到的错误以及解决方案

实验3.1 数据更新

实验内容

(1) 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。

(2) 使用SELECT INTO语句，产生一个新表并插入数据。

(3) 使用UPDATE语句可以修改制定表中满足WHERE子句条件的元组，有三种修改的方式：修改某个元组的值；修改多个元组的值；带子查询的修改语句。

(4) 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。

实验步骤

1. 使用SQL语句向STUDENTS表中插入元组（编号：12345678 名字：LiMing EMAIL: LM@gmail.com 年级：2002）。

```
insert into STUDENTS(sid,sname,email,grade)values('12345678','LiMing','LM@gmail.com',2002);
```

验证结果：

```
select * from STUDENTS
```

结果 消息

	sid	sname	email	grade
1	12345678	LiMing	LM@gmail.com	2002
2	800001216	gfxrgs	hhce4@qhldj.gov	1992
3	800002933	vnbqzsvv	pvhxd41@zqur.org	2002
4	800005753	waqcj	hlhq0h8@jdba.gov	1992

2. 对每个课程，求学生的选课人数和学生的最高成绩，并把结果存入数据库。使用SELECT INTO 和 INSERT INTO 两种方法实现。
   - SELECT INTO

     需要**注意**的是：temp为一个新建表，因此需要为其列命名(count(distinct sid) as ...)

```
select count(distinct sid) as count_students,max(score) as max_score,cid
into temp
from CHOICES
group by cid
```

     验证结果：

```
select * from temp
```

结果 消息

	count_students	max_score	cid
1	5825	99	10008
2	5889	99	10019
3	5812	99	10018
4	5945	99	10040
5	5937	99	10011
6	5886	99	10028

   - INSERT INTO

     新建一个临时表用来存放结果

```
create table temp2(
count_students int,
max_score int,
cid int
);
```

     插入子查询结果

```sql
insert into temp2(count_students,max_score,cid)
select count(distinct sid) as count_students,max(score) as max_score,cid
from CHOICES
group by cid
```

验证结果

```sql
select * from temp2
```

结果 消息

	count_students	max_score	cid
1	5825	99	10008
2	5889	99	10019
3	5812	99	10018
4	5945	99	10040
5	5937	99	10011

3. 在STUDENTS表中使用SQL语句将姓名为LiMing.的学生的EMAIL改为LM@qq.com。

更新STUDENT表

```sql
update STUDENTS
set email='LM@qq.com'
where sname='LiMing'
```

验证结果

```sql
select sname,email
from STUDENTS
where sname='LiMing'
```

	sname	email
1	LiMing	LM@qq.com

4. 在TEACHERS表中使用SQL语句将所有教师的工资翻倍。

将所有教师工资翻倍

```sql
update TEACHERS
set salary=salary*2;
```

验证结果：

```sql
select tid,tname,salary
from TEACHERS;
```

结果 消息

	tid	tname	salary
1	199999999	abcd	3000
2	200003125	fqmmyi	4428
3	200005322	udjom	1373
4	200009423	jhoulvtr	3786
5	200010493	xzgkflko	3832
6	200011899	lwujzim	NULL
7	200016731	nfgrod	4319
8	200017039	qibsy	4679
9	200023671	zfuxiahq	4177

	tid	tname	salary
1	199999999	abcd	6000
2	200003125	fqmmyi	8856
3	200005322	udjom	2746
4	200009423	jhoulvtr	7572
5	200010493	xzgkflko	7664
6	200011899	lwujzim	NULL
7	200016731	nfgrod	8638
8	200017039	qibsy	9358
9	200023671	zfuxiahq	8354

5. 将姓名为waqcj的学生的课程C++的成绩加10分。

```
UPDATE CHOICES
SET score = score + 10
WHERE sid IN (SELECT sid FROM STUDENTS WHERE sname = 'waqcj')
AND cid IN (SELECT cid FROM COURSES WHERE cname = 'C++');
```

验证结果：

结果 消息

	sname	score	cname
1	waqcj	66	c++

	sname	score	cname
1	waqcj	76	c++

6. 在STUDENTS表中使用SQL语句删除姓名为LiMing的学生信息。

```
delete from STUDENTS
where sname='LiMing'
```

验证结果：

	sid	sname	email	grade
1	12345678	LiMing	LM@qq.com	2002

	sid	sname	email	grade

7. 删除所有选修课程C的选课记录。

```
delete from CHOICES
where cid in(select cid from COURSES where cname='C')
```

验证结果

结果 消息

	no	sid	tid	cid	score
1	500019253	826800279	254743764	10042	52
2	500026725	828743446	264652927	10042	85
3	500055883	841348939	212523885	10042	94
4	500077819	832815482	241749288	10042	92
5	500103431	878941549	268548571	10042	73
6	500115088	800575453	213781494	10042	78
7	500119959	893003636	214156099	10042	88
8	500130427	857435877	201045902	10042	52

	no	sid	tid	cid	score

8. 对COURSES表做删去时间>80的元组的操作，讨论该删除操作所受到的约束。

```
delete from COURSES
where hour>80
```

验证结果：

结果 消息

	cid	cname	hour
1	10001	database	96
2	10002	operating system	88

	cid	cname	hour

删除操作可能受到的约束：

1. 外键约束

   首先hour并非primary key，也就不可能作为外键约束中的主表，不存在外键约束。

   若hour为primary key，在外键约束中作为主表，则删除后，可能会造成应用hour的发生引用错误（引用的值已经不存在），违反了引用完整性。

2. 级联删除约束

   如果涉及了级联删除约束，删除一个表可能会造成多个表的删除，影响了数据完整性

实验3.2

实验内容

(1) 使用CREATE VIEW命令建立视图。

(2) 对视图进行查询

(3) 更新视图

(4) 使用DROP VIEW命令删除视图。

实验步骤

1. 建立薪水大于3000的教师的视图t_view，并要求进行修改和插入操作时仍需保证该视图只有薪水大于3000的教师信息。

   加入check option保证

```
create view t_view as
select *
from TEACHERS
where salary>3000
with check option
```

   验证结果：

   向TEACHERS语句中插入工资小于3000的信息测试

```
select *
from t_view
where salary<=3000

insert into TEACHERS(tid,tname,email,salary)values('12345','测试2','123@gmail.com',2000)

select *
from t_view
where salary<=3000
```

结果 消息

	tid	tname	email	salary

	tid	tname	email	salary

2. 在视图t_view中查询邮件地址为xibl@izd.edu的教师的相关信息。

```
select *
from t_view
where email='xibl@izd.edu'
```

	tid	tname	email	salary
1	200026895	nuyoq	xibl@izd.edu	10660

3. 向视图t_view中插入一个新的教师记录，其中教师编号为199999998，姓名为abc，邮件地址为abc@def.com，薪水为5000。

```
insert into t_view(tid,tname,email,salary)values('199999998','abc','abc@def.com',5000)
```

结果 消息

	tid	tname	email	salary
1	199999998	abc	abc@def.com	5000

4. 在视图t_view中将编号为200010493的教师的薪水改为6000。

```
update t_view
set salary=6000
where tid='200010493'
```

结果 消息

	tid	tname	email	salary
1	200010493	xzgkflko	zcf_m@def.com	7664

	tid	tname	email	salary
1	200010493	xzgkflko	zcf_m@def.com	6000

5. 删除视图t_view。

```
drop view t_view
```

实验3.3

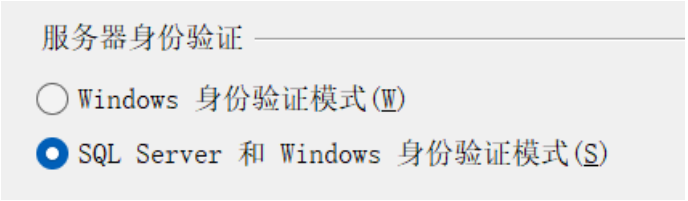
实验内容

由系统提供一定的方式让用户标识自己的名字和身份。

尽量不要用中文用户名。且L1不应该在School下，应该在整个下。

实验步骤

1. 在SSMS中，设置SQL Server的安全认证模式。



2. 在SSMS中建立一个名为“L1”的登陆用户、School数据库用户。

   建立一个名为L1，密码为123456的登录名

```
exec sp_addlogin 'L1','123456'
```

   切换至School数据库，建立U1用户

```
use school
exec sp_adduser 'L1','U1'
```

3. 在SSMS中取消“L1”这个用户。

```
exec sp_droplogin 'L1'
```

4. 学习【参考命令】中的几个命令，了解利用命令实现上述步骤。

实验3.4

实验内容

定义用户权限，将用户权限登记到数据字典中，并对用户进行合法权限检查。

实验步骤

1. 在SSMS中建立一个名为“L1”的登陆用户、School数据库的用户。参见实验3.3的试验步骤2

   在3.3的步骤2已完成

2. 使用查询验证“L1”这个用户名是否具有对学生表的SELECT权限。

   切换至用户U1

```
exec as user='U1'
```

   查看当前操作用户

```
select user
```

   使用查询验证

```
select *
from STUDENTS
```

```
(1 行受影响)
消息 229, 级别 14, 状态 5, 第 176 行
拒绝了对对象 'STUDENTS' (数据库 'School', 架构 'dbo')的 SELECT 权限。
```

3. 将School数据库的操作权限赋予数据库用户L1。

   切换回管理员权限

```
revert
```

   将select权限赋给U1

```
grant select on CHOICES to U1
grant select on COURSES to U1
grant select on STUDENTS to U1
grant select on TEACHERS to U1
```

   验证结果

	sid	sname	email	grade
1	800001216	gfxrgs	hhce4@qhldj.gov	1992
2	800002933	vnbqzsvv	pvhxd41@zqur.org	2002
3	800005753	waqcj	hlhq0h8@jdba.gov	1992
4	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992
5	800006941	ogvmu	62sfbd@lrt.gov	1995
6	800007595	uxqqbkjn	cr8g@zrvgt.edu	1997

实验3.5

实验内容

在视图机制上定义用户权限，将用户权限登记到数据字典中，并对用户进行合法权限检查。

实验步骤

1. 在数据库School上创建用户“U1”，具体操作参见实验3.3中的试验步骤2

   在3.3的步骤2已完成

2. 新建查询，用管理员身份登陆数据库。在choices表上创建视图ch_view，并显示其内容（选课课程号为10005）。

```
create view ch_view as
select *
from CHOICES
where cid='10005'
```

	no	sid	tid	cid	score
1	500003262	826310502	267846042	10005	90
2	500029445	880275978	245622362	10005	91
3	500030717	883794999	220037007	10005	82
4	500039740	852880400	230467893	10005	83
5	500056410	812844702	233198108	10005	74

3. 在视图ch_view上给用户U1赋予INSERT的权限。

```
grant insert on ch_view to U1
```

   验证结果：

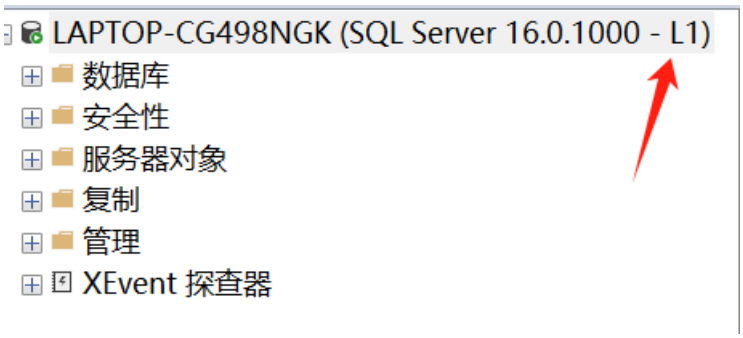
```
exec as user='U1'
insert into
ch_view(no,sid,tid,cid,score)values(999999,'12345','12345','12345',99)
```

4. 将视图ch_view上score列的权限赋予用户U1。

```
grant all privileges on ch_view(score) to U1
```

5. 以登录名L1登陆查询分析器，对ch_view进行查询操作。

   以用户L1登陆查询分析器




```
select * from ch_view
```

	no	sid	tid	cid	score
1	500003262	826310502	267846042	10005	90
2	500029445	880275978	245622362	10005	91
3	500030717	883794999	220037007	10005	82
4	500039740	852880400	230467893	10005	83

6. 以用户L1登陆查询分析器, 对no为500127998的学生的成绩进行修改，改为90分。

结果 消息

	no	score
1	500127998	81

	no	score
1	500127998	90

7. 收回对用户U1对视图ch_view查询权限的授权

```
revoke select on ch_view from U1
```

遇到的错误以及解决方案

1. 关于外键建立

   在建立外键关系时

```
alter table temp
add constraint FK3 foreign key(cid) references COURSES(cid)
```

   出现了如下错误

消息

```
消息 547，级别 16，状态 0，第 103 行
ALTER TABLE 语句与 FOREIGN KEY 约束"FK3"冲突。该冲突发生于数据库
```

   - 首先排除了约束重复的问题，运行`exec sp_helpconstraint temp`，发现temp无约束

	Object Name
1	temp

   - 考虑是否是COURSES(cid)不为Primary key。发现为PK，满足建立外键关系的条件。

dbo.COURSES
- 列
  - cid (PK, char(10), not null)
  - cname (char(30), not null)
  - hour (int, null)

   - 最后考虑是否temp中的cid有部分不在COURSES(cid)中，运行

```sql
select *
from temp
where not exists(
select *
from COURSES
where temp.cid=COURSES.cid
)
```

发现会输出结果，因此是这个错误造成的无法建立外键关系。

2. 对CHOICES表的插入错误

CHOICES表有外键约束，其tid的值必须为主表TEACHERS(tid)中存在的值。

constraint_keys
tid
REFERENCES School.dbo.TEACHERS (tid)

constraint_type	constraint_name	delete_action	update_action	status_enabled
FOREIGN KEY	FK_CHOICES_TEACHERS	No Action	No Action	Enabled

若插入主表TEACER(tid)中不存在的值会违反完整性约束。