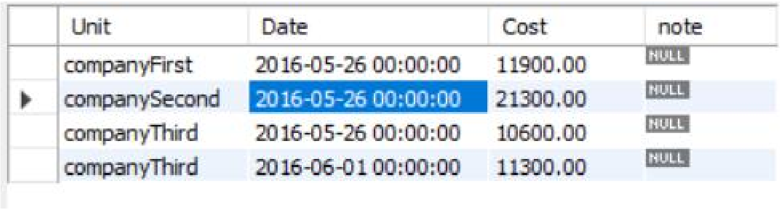
1．完成以下操作：

⑴ 向在实验二中所定义的数据表增加“备注”列，其数据类型为字符型，并查看新增列的值。

```
alter table sumcostall add column note varchar(16);
select *
from sumcostall;
```

运行结果：

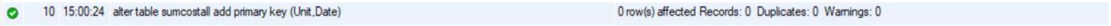


	Unit	Date	Cost	note
	companyFirst	2016-05-26 00:00:00	11900.00	NULL
▶	companySecond	2016-05-26 00:00:00	21300.00	NULL
	companyThird	2016-05-26 00:00:00	10600.00	NULL
	companyThird	2016-06-01 00:00:00	11300.00	NULL

图 1

⑵ 对上述数据表增加主码约束条件，并观察在数据表中存在数据的情况下主码约束是否创建成功，然后再次执行实验二中实验内容3-⑵的操作，并观察记录执行结果。

```
alter table sumcostall
        add primary key (Unit,Date);
select *
from sumcostall;
```



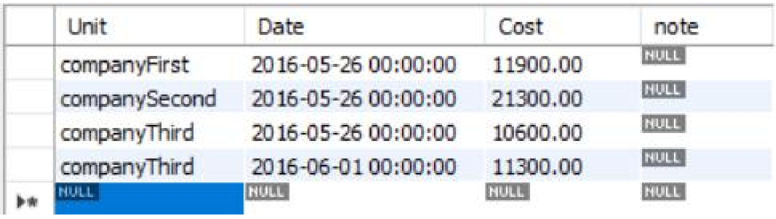
```
10  15:00:24  alter table sumcostall add primary key (Unit,Date)        0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

图 2

创建成功

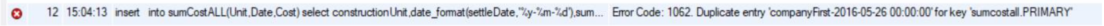
```
into sumCostALL(Unit,Date,Cost)
select constructionUnit,date_format(settleDate,'%y-%m-%d'),sum(allCost)
from costTable
group by constructionUnit,settleDate;
select *
from sumcostall;
```

运行结果：



	Unit	Date	Cost	note
	companyFirst	2016-05-26 00:00:00	11900.00	NULL
	companySecond	2016-05-26 00:00:00	21300.00	NULL
	companyThird	2016-05-26 00:00:00	10600.00	NULL
	companyThird	2016-06-01 00:00:00	11300.00	NULL
▶*	NULL	NULL	NULL	NULL

图 3



```
12  15:04:13  insert  into sumCostALL(Unit,Date,Cost) select constructionUnit,date_format(settleDate,'%y-%m-%d'),sum...  Error Code: 1062. Duplicate entry 'companyFirst-2016-05-26 00:00:00' for key 'sumcostall.PRIMARY'
```

图 4

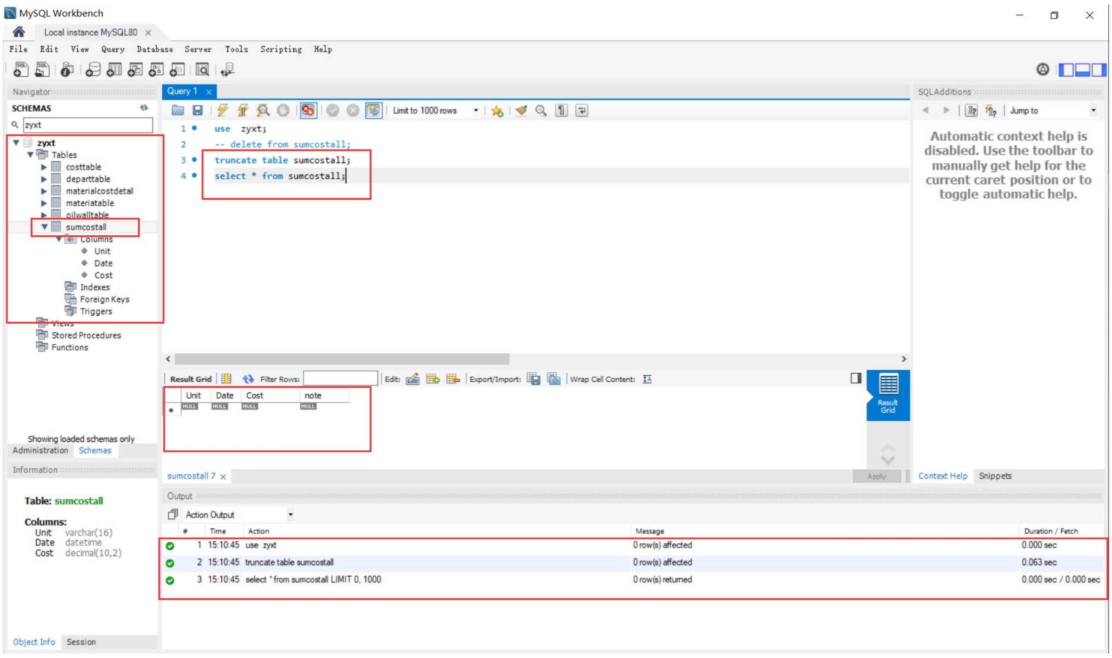
⑶ 删除上述数据表中的数据，然后再删除该数据表，对这两个操作进行比较

删除表中数据：代码

```
truncate table sumcostall;
```

select * from sumcostall;

结果：



```
use  zyxt;
-- delete from sumcostall;
truncate table sumcostall;
select * from sumcostall;
```

Unit	Date	Cost	note
NULL	NULL	NULL	NULL

Table: sumcostall

Columns:
Unit varchar(16)
Date datetime
Cost decimal(10,2)

```
#  Time      Action                                  Message           Duration / Fetch
1  15:10:45  use  zyxt                               0 row(s) affected 0.000 sec
2  15:10:45  truncate table sumcostall               0 row(s) affected 0.063 sec
3  15:10:45  select * from sumcostall LIMIT 0, 1000  0 row(s) returned 0.000 sec / 0.000 sec
```

图 5

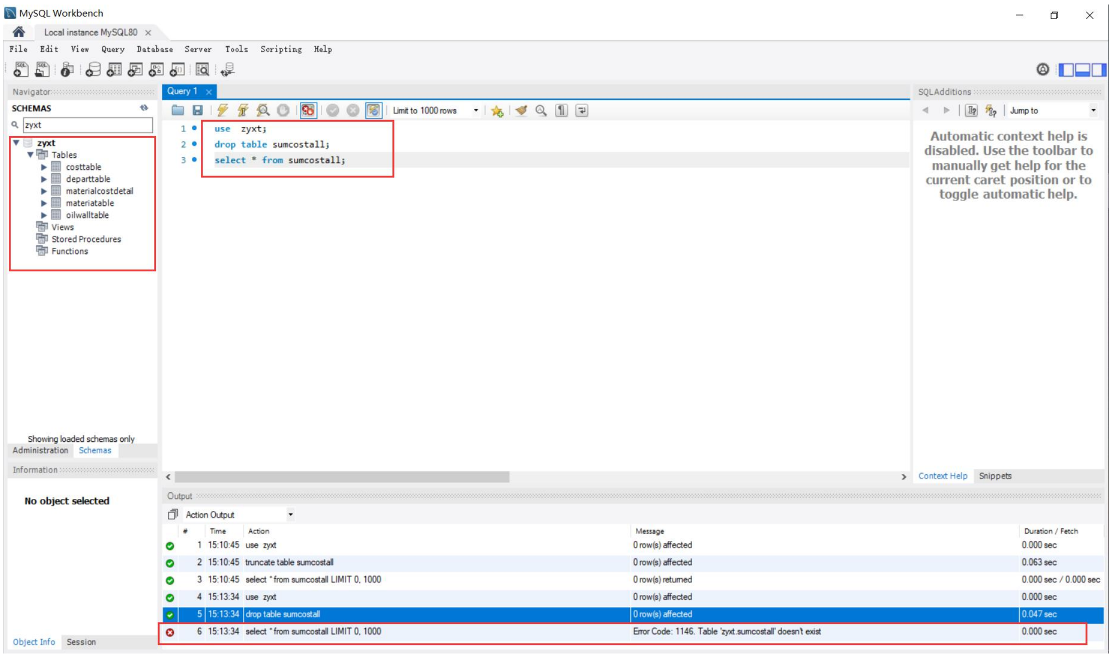
此时查询表仍然能被查询到，但是表中的数据全部为空。

删除表：

drop table sumcostall;

select * from sumcostall;

运行结果：



```
use  zyxt;
drop table sumcostall;
select * from sumcostall;
```

```
#  Time      Action                                  Message                                            Duration / Fetch
1  15:10:45  use  zyxt                               0 row(s) affected                                  0.000 sec
2  15:10:45  truncate table sumcostall               0 row(s) affected                                  0.063 sec
3  15:10:45  select * from sumcostall LIMIT 0, 1000  0 row(s) returned                                  0.000 sec / 0.000 sec
4  15:13:34  use  zyxt                               0 row(s) affected                                  0.000 sec
5  15:13:34  drop table sumcostall                   0 row(s) affected                                  0.047 sec
6  15:13:34  select * from sumcostall LIMIT 0, 1000  Error Code: 1146. Table 'zyxt.sumcostall' doesn't exist  0.000 sec
```

图 6

由图中可以看出，表 sumcostall 消失，并且查询语句 “select * from sumcostall” 报错，报错语句为“Error Code: 1146. Table 'zyxt.sumcostall' doesn't exist”（表 sumcostall 不存在）。

由此可见当删除表中数据时，表本身没有被删除，保留表的结构，被删除的只是表中的数据。而删除数据表时，基本表在数据库中被删除，查询时为表不存在。

2．完成以下任务：

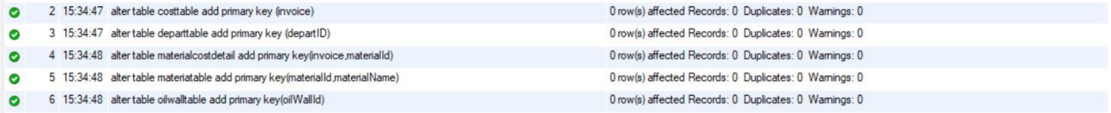
⑴ 对实验一中所定义的 5 个数据表增加主码约束条件，并观察在数据表中存在数据的情况下主码约束是否创建成功，然后执行以下 2 个操作，观察并记录实体完整性的检查和违约处理。

① insert into **材料消耗表** values('zy2020001','wm004',100)② insert into **材料消耗表** values('zy2020002',NULL,200)

为数据表添加约束代码 ：

```
alter table costtable
        add primary key (invoice);
alter table departtable
        add primary key (departID);
alter table materialcostdetail
        add primary key(invoice,materialId);
alter table materiatable
        add primary key(materialId,materialName);
alter table oilwalltable
        add primary key(oilWallId);
```

结果：



```
✓  2  15:34:47  alter table costtable add primary key (invoice)                        0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
✓  3  15:34:47  alter table departtable add primary key (departID)                     0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
✓  4  15:34:48  alter table materialcostdetail add primary key(invoice,materialId)     0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
✓  5  15:34:48  alter table materiatable add primary key(materialId,materialName)      0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
✓  6  15:34:48  alter table oilwalltable add primary key(oilWallId)                    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

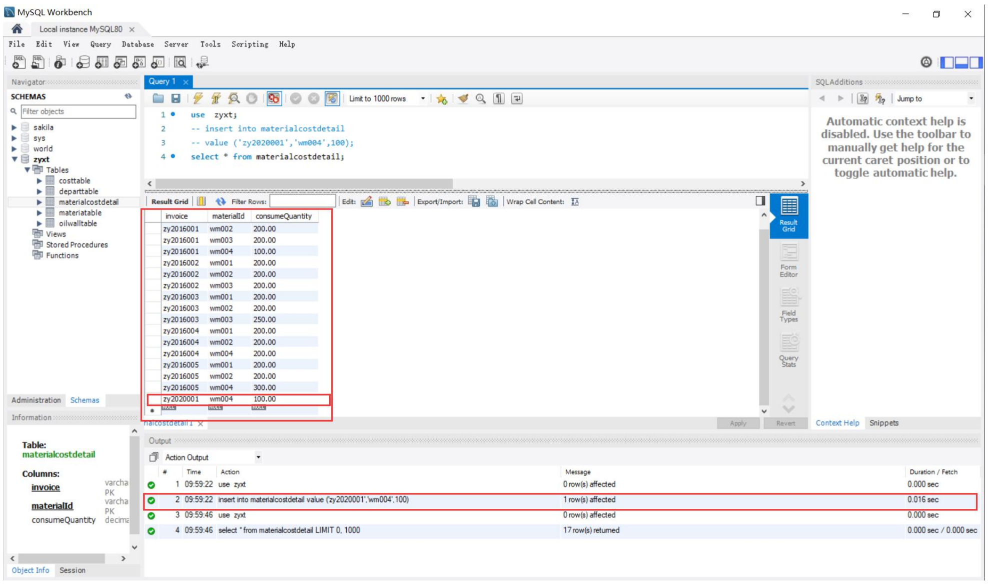
图 7

①代码：

insert into materialcostdetail value ('zy2020001','wm004',100);

select * from materialcostdetail;

结果：



```
use  zyxt;
-- insert into materialcostdetail
-- value ('zy2020001','wm004',100);
select * from materialcostdetail;
```

invoice	materialId	consumeQuantity
zy2016001	wm002	200.00
zy2016001	wm003	200.00
zy2016001	wm004	100.00
zy2016002	wm001	200.00
zy2016002	wm002	200.00
zy2016002	wm003	200.00
zy2016003	wm001	200.00
zy2016003	wm002	200.00
zy2016003	wm003	250.00
zy2016004	wm001	200.00
zy2016004	wm002	200.00
zy2016004	wm004	200.00
zy2016005	wm001	200.00
zy2016005	wm002	200.00
zy2016005	wm004	300.00
zy2020001	wm004	100.00
NULL	NULL	NULL

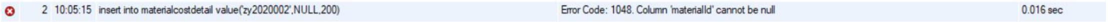
```
   #  Time      Action                                                         Message               Duration / Fetch
✓  1  09:59:22  use  zyxt                                                      0 row(s) affected     0.000 sec
✓  2  09:59:22  insert into materialcostdetail value ('zy2020001','wm004',100) 1 row(s) affected     0.016 sec
✓  3  09:59:46  use  zyxt                                                      0 row(s) affected     0.000 sec
✓  4  09:59:46  select * from materialcostdetail LIMIT 0, 1000                 17 row(s) returned    0.000 sec / 0.000 sec
```

图 8

②代码：

insert into materialcostdetail value('zy2020002',NULL,200);

结果：



```
2  10:05:15  insert into materialcostdetail value('zy2020002',NULL,200)    Error Code: 1048. Column 'materialId' cannot be null    0.016 sec
```

图 9

报错信息：Error Code: 1048. Column 'materialId' cannot be null

由于将表 materialcostdetail 的主码定义为 primary key(invoice,materialId)，所以 materialId 不能为空。

⑵ 对实验一中所定义的 5 个数据表增加相应的参照完整性约束，

并观察在数据表中存在数据的情况下参照完整性约束是否创建成功，

然后执行以下操作，观察并记录参照完整性的检查和违约处理。

① 将（y007 油井 112203002）插入到油水井表。

② insert into **材料消耗表** values('zy2020007','wm006',100)

③ 将单位表中的（112202002 采油二矿二队）删除，查看油水

井表和**成本表**中的数据有何变化。

④ 将材料表中的（wm004 材料四 袋 10）修改为（wm04 材料

四 袋 10）。

⑤ 撤销上述成功的更新操作。

表 1 参照关系表

被参照表	参照表	属性
departTable	oilWallTable	departID
departTable	costTable	departID
oilWallTable	costTable	oilWall
materiaTable	materialCostDetail	materialId

添加参照完整性约束代码：

alter table oilwalltable add foreign key(departID) references departtable(departID);

alter table costtable add foreign key(departmentId) references departtable(departID);

alter table costtable add foreign key(oilWall) references oilwalltable(oilWallId);

alter table materialcostdetail add foreign key(materialId) references materiatable(materialId);

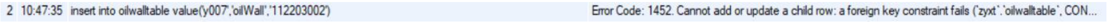
①

代码：

insert into oilwalltable

value('y007','oilWall','112203002');

结果：



```
2  10:47:35  insert into oilwalltable value('y007','oilWall','112203002')    Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('zyxt'.'oilwalltable', CON...
```

图 10

报错信息：Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`zyxt`.`oilwalltable`, CONSTRAINT `oilwalltable_ibfk_1` FOREIGN KEY (`departID`) REFERENCES `departtable` (`departID`))

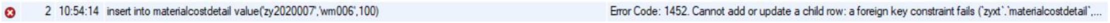
②：

代码：

insert into materialcostdetail

value('zy2020007','wm006',100);

结果：



```
2 10:54:14 insert into materialcostdetail value('zy2020007','wm006',100)    Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('zyxt'.'materialcostdetail',...
```

图 11

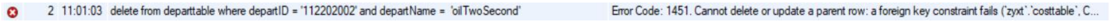
报错信息：Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`zyxt`.`materialcostdetail`, CONSTRAINT `materialcostdetail_ibfk_1` FOREIGN KEY (`materialId`) REFERENCES `materiatable` (`materialId`))

③

代码：

```
delete from departtable
where departID = '112202002' and departName =    'oilTwoSecond';
```

结果：



```
2 11:01:03 delete from departtable where departID = '112202002' and departName = 'oilTwoSecond'    Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('zyxt'.'costtable', C...
```

图 12

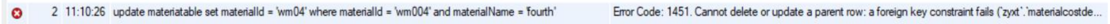
报错信息：Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`zyxt`.`costtable`, CONSTRAINT `costtable_ibfk_1` FOREIGN KEY (`departmentId`) REFERENCES `departtable` (`departID`))

④

代码：

```
update materiatable
set materialId = 'wm04'
where materialId = 'wm004' and materialName = 'fourth';
```

结果：



```
2 11:10:26 update materiatable set materialId = 'wm04' where materialId = 'wm004' and materialName = 'fourth'    Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('zyxt'.'materialcostde...
```

图 13

报错信息：Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`zyxt`.`materialcostdetail`, CONSTRAINT `materialcostdetail_ibfk_1` FOREIGN KEY (`materialId`) REFERENCES `materiatable` (`materialId`))

⑤由于上述操作均未成功，没有需要撤销的成功操作。

⑶ 对实验一中所定义的 5 个数据表按以下要求增加相应的完整性约束条件，并观察在数据表中存在数据的情况下完整性约束是否创建成功。

① 单位表的单位名称不能取空值、且取值唯一。

② 油水井表的井别只允许取“油井”或“水井”，单位代码不能取空值。

③ 材料表的名称不能取空值、且取值唯一，计量单位不能取空值。

④ **材料消耗表**的消耗数量不能取空值。

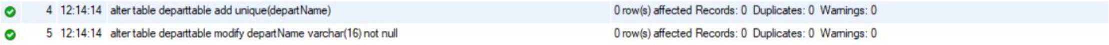
⑤ 对**成本表**根据实际应用的要求定义适当的用户定义的完整性约束条件

①

代码：

```
alter table departtable add unique(departName) ;
alter table departtable modify departName varchar(16) not null;
```

结果：



```
4  12:14:14  alter table departtable add unique(depart Name)                      0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
5  12:14:14  alter table departtable modify depart Name varchar(16) not null      0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

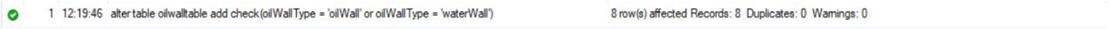
图 14

②

代码：

```
alter table oilwalltable add check(oilWallType = 'oilWall' or oilWallType = 'waterWall');
```

结果：



```
1  12:19:46  alter table oilwalltable add check(oilWallType = 'oilWall' or oilWallType = 'waterWall')      8 row(s) affected Records: 8  Duplicates: 0  Warnings: 0
```

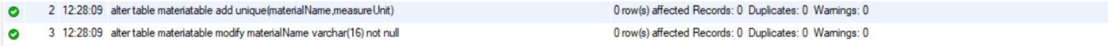
图 15

③

代码：

```
alter table materiatable add unique(materialName,measureUnit);
alter table materiatable modify materialName varchar(16) not null;
```

结果：



```
2  12:28:09  alter table materiatable add unique(materialName,measure Unit)          0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
3  12:28:09  alter table materiatable modify materialName varchar(16) not null      0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

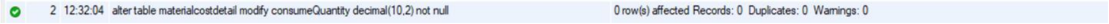
图 16

④

代码：

```
alter table materialcostdetail modify consumeQuantity decimal(10,2) not null;
```

结果：



```
2  12:32:04  alter table materialcostdetail modify consumeQuantity decimal(10,2) not null      0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

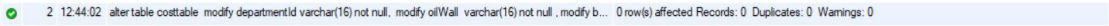
图 17

⑤

代码：

```
alter table costtable
modify departmentId varchar(16) not null,
modify oilWall    varchar(16) not null ,
modify budgetAmount decimal(10,2) not null,
modify budgetPerson varchar(16) not null,
modify budgetDate datetime not null;
```

结果：



```
2  12:44:02  alter table costtable  modify departmentId varchar(16) not null,  modify oilWall  varchar(16) not null , modify b...  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

图 18

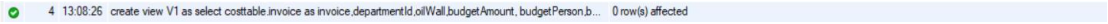
3．完成以下操作：

⑴ 定义视图 V1，用于保存成本表和材料消耗表的全部列。

代码：

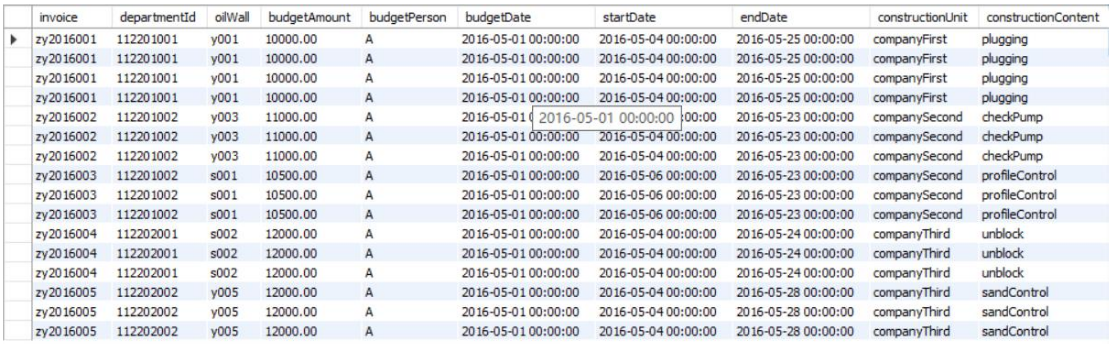
```
create view V1
as
select costtable.invoice as invoice,departmentId,oilWall,budgetAmount,
budgetPerson,budgetDate,startDate,endDate,constructionUnit,
constructionContent,materialCost,presonCost,equipmentCost
otherCost,allCost,settlePreson,settleDate,recordAmount,recordPerson,
recordDate,materialId,consumeQuantity
from    costtable,materialcostdetail
where costtable.invoice = materialcostdetail.invoice;
select *
from V1;
```

结果：



```
4  13:08:26  create view V1 as select costtable.invoice as invoice,departmentId,oilWall,budgetAmount, budgetPerson,b...  0 row(s) affected
```

图 19



invoice	departmentId	oilWall	budgetAmount	budgetPerson	budgetDate	startDate	endDate	constructionUnit	constructionContent
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-25 00:00:00	companyFirst	plugging
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-25 00:00:00	companyFirst	plugging
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-25 00:00:00	companyFirst	plugging
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-25 00:00:00	companyFirst	plugging
zy2016002	112201002	y003	11000.00	A	2016-05-01 2016-05-01 00:00:00	:00:00	2016-05-23 00:00:00	companySecond	checkPump
zy2016002	112201002	y003	11000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-23 00:00:00	companySecond	checkPump
zy2016002	112201002	y003	11000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-23 00:00:00	companySecond	checkPump
zy2016003	112201002	s001	10500.00	A	2016-05-01 00:00:00	2016-05-06 00:00:00	2016-05-23 00:00:00	companySecond	profileControl
zy2016003	112201002	s001	10500.00	A	2016-05-01 00:00:00	2016-05-06 00:00:00	2016-05-23 00:00:00	companySecond	profileControl
zy2016003	112201002	s001	10500.00	A	2016-05-01 00:00:00	2016-05-06 00:00:00	2016-05-23 00:00:00	companySecond	profileControl
zy2016004	112202001	s002	12000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-24 00:00:00	companyThird	unblock
zy2016004	112202001	s002	12000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-24 00:00:00	companyThird	unblock
zy2016004	112202001	s002	12000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-24 00:00:00	companyThird	unblock
zy2016005	112202002	y005	12000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-28 00:00:00	companyThird	sandControl
zy2016005	112202002	y005	12000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-28 00:00:00	companyThird	sandControl
zy2016005	112202002	y005	12000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-28 00:00:00	companyThird	sandControl

图 20

⑵ 查询上面定义的视图 V1，可任意组合查询条件，构造出 2 个查询。

代码：

①

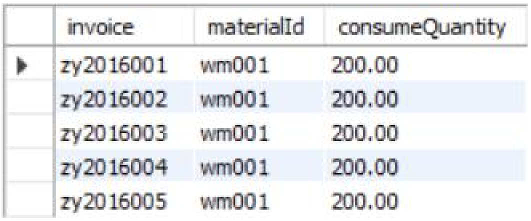
```
select invoice,materialId,consumeQuantity
from V1
where materialId = 'wm001';
```

②

```
select *
from V1
where invoice = 'zy2016001';
```

结果：

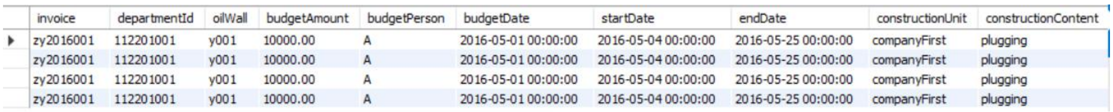
①



invoice	materialId	consumeQuantity
zy2016001	wm001	200.00
zy2016002	wm001	200.00
zy2016003	wm001	200.00
zy2016004	wm001	200.00
zy2016005	wm001	200.00

图 21

②



invoice	departmentId	oilWall	budgetAmount	budgetPerson	budgetDate	startDate	endDate	constructionUnit	constructionContent
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-25 00:00:00	companyFirst	plugging
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-25 00:00:00	companyFirst	plugging
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-25 00:00:00	companyFirst	plugging
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-25 00:00:00	companyFirst	plugging

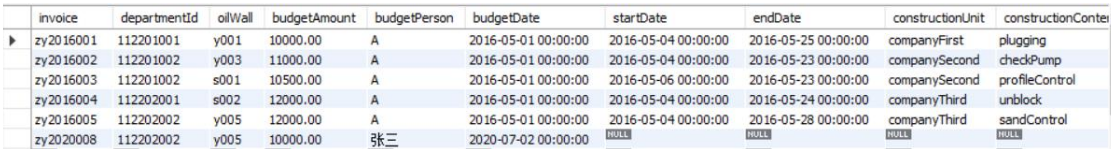
图 22

⑶ 定义一个反映成本表预算状态的视图 V2，并向该视图插入('zy2020008','112202002','y005',10000,'张三', '2020-07-02')，查看成本表的数据有何变化。

代码：

```
create view V2
as
select invoice,departmentId,oilWall,budgetAmount,budgetPerson,budgetDate
from costtable;
insert into V2
value ('zy2020008','112202002','y005',10000,'张三', '2020-07-02');
select *
from V2;
```

结果：



invoice	departmentId	oilWall	budgetAmount	budgetPerson	budgetDate	startDate	endDate	constructionUnit	constructionConte
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-25 00:00:00	companyFirst	plugging
zy2016002	112201002	y003	11000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-23 00:00:00	companySecond	checkPump
zy2016003	112201002	s001	10500.00	A	2016-05-01 00:00:00	2016-05-06 00:00:00	2016-05-23 00:00:00	companySecond	profileControl
zy2016004	112202001	s002	12000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-24 00:00:00	companyThird	unblock
zy2016005	112202002	y005	12000.00	A	2016-05-01 00:00:00	2016-05-04 00:00:00	2016-05-28 00:00:00	companyThird	sandControl
zy2020008	112202002	y005	10000.00	张三	2020-07-02 00:00:00	NULL	NULL	NULL	NULL

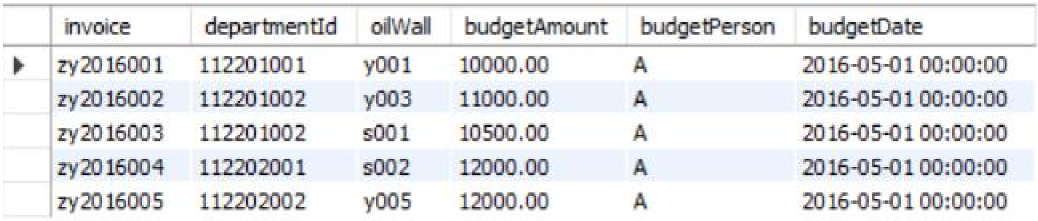
图 23

⑷ 撤销上述成功的更新操作。

代码：

```
delete
from costtable
where invoice    = 'zy2020008' and budgetPerson = '张三';
select *
from V2;
```

结果：



invoice	departmentId	oilWall	budgetAmount	budgetPerson	budgetDate
zy2016001	112201001	y001	10000.00	A	2016-05-01 00:00:00
zy2016002	112201002	y003	11000.00	A	2016-05-01 00:00:00
zy2016003	112201002	s001	10500.00	A	2016-05-01 00:00:00
zy2016004	112202001	s002	12000.00	A	2016-05-01 00:00:00
zy2016005	112202002	y005	12000.00	A	2016-05-01 00:00:00

图 24