数据库完整性

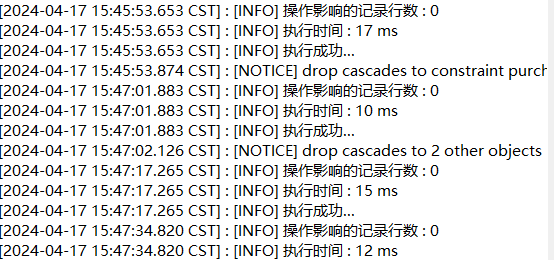
1. 完全删除已经创建的四张表books, customers, purchases, pricing以及相关视图; 要注意删除的顺序！

**drop** **table** books **cascade** ;

**drop** **table** customers **cascade** ;

**drop** **table** purchases **cascade** ;

**drop** **table** pricing **cascade** ;



2. 重新定义四张表(属性参照lab2-createdb.sql)：

定义新的books表，其所拥有的属性与属性类型与原有的

books表一致

以列级完整性的方式定义属性bookId为主码，同时在定

义的时候必须提供约束名(books\_pk)

定义属性title为” 非空” ，同时在定义的时候必须提

供约束名(books\_title)

**create** **table** Books(

bookId **number**(9,0) **constraint** books\_pk **PRIMARY** **KEY**,

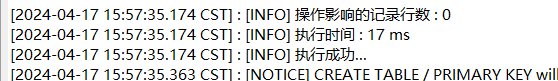
title **varchar**(80) **CONSTRAINT** books\_title **not** **null**,

author **varchar**(40),

**year** **integer**,

category **varchar**(15)

);



定义新的customers表，其所拥有的属性与属性类型与原有的

customers表一致

以表级完整性的方式定义属性cid为主码

以列级完整性的方式约束age不能小于18并且不能超过80，同时

在定义的时候必须提供约束名(customers\_age)

**create** **table** customers(

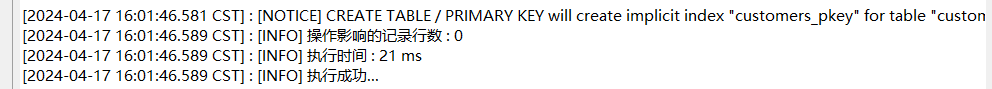
cid **number**(9,0),

cname **char**(40),

age **INTEGER** **constraint** customers\_age **check**(age**>**18**and** age **<=**80),

**PRIMARY** **key** (cid)

);



定义新的purchases表，其所拥有的属性与属性类型与原有的

purchases表一致

⦁ 定义属性cid与bookId为主码，同时在定义的时候必须提供约束

名(purchases\_pk)

⦁ 定义属性bookId为外码，它参照Books表的主码。当删除Books

表中的某一本书时，级联删除与它相关的购买记录

⦁ 定义属性cid为外码，它参照customers表，同时在定义的时候

必须提供约束名(purchases\_fk)

**create** **table** purchases (

cid **number**(9,0),

bookId **number**(9,0),

pdate **DATE**,

pprice **number**(8,2),

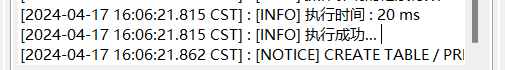
**CONSTRAINT** purchases\_pk **primary** **key** (cid, bookId),

**foreign** **key** (bookId) **references** books

**on** **delete** **CASCADE**,

**CONSTRAINT** purchases\_fk **FOREIGN** **key** (cid) **REFERENCES** customers

);



⦁ 定义新的pricing表，其所拥有的属性与属性类型与原有

的pricing表一致

⦁ 以列级完整性的方式定义属性booId为主码，同时在定义

的时候必须提供约束名(pricing\_pk)

⦁ 定义约束(price\_check)，要求当书的format是

’paperback’时，当前价格不能超过50

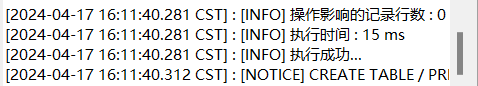
**create** **table** pricing (

bookId **number**(9,0) **CONSTRAINT** pricing\_pk **PRIMARY** **KEY**,

format **varchar**(15) **constraint** price\_check **check**(format!='paperback' **or** price **<=**50),

price **number**(9,2)

);



3.增加或修改完整性约束：

针对books表，要求category的取值只能属于以下集合

('textbook','politics','romance','cooking')

**alter** **table** books **add** **constraint** category\_cons **check**(

category **in** ('textbook','politics','romance','cooking'));

将customers表中age的约束放宽，上限放宽到85。约束

名不变。

**alter** **table** customers

**drop** **constraint** customers\_age ;

**alter** **table** customers

**add** **constraint** customers\_age **check**( age**>**18 **and** age**<=**85);

针对表pricing，将约束名price\_check修改为

price\_check\_paper

x

**alter** **table** pricing

**drop** **constraint** price\_check;

**alter** **table** pricing

**add** **CONSTRAINT** price\_check\_paper **check**(format!='paperback' **or** price **<=**50);

4.定义如下的一张表：

′ Score(Sno, Chinese, Math, English, Total)

Sno是主码，类型是长度为9的定长字符串

Chinese, Math, English分别表示三门课成绩，类型为

整数，并且三门课的成绩取值范围是0-100，也可以无成

绩(null)

Total为总成绩

**create** **table** Score(

Sno **char**(9) **primary** **key**,

Chinese **int**

**constraint** Chineses\_check **check**((Chinese **BETWEEN** 0 **and** 100) **or** Chinese **is** **null**),

Math **int**

**constraint** Math\_check **check**((Math **BETWEEN** 0 **and** 100) **or** Math **is** **null**),

English **int**

**constraint** English\_checek **check**((English **BETWEEN** 0 **and** 100 )**or** English **is** **null**),

Total **int**

);

如果三门课均无成绩(null)，则无总成绩(null)

如果有某一门或两门课无成绩，则总成绩为有成绩的课

程成绩之和

用触发器来实现对total的值的判断

**CREATE** **OR** **REPLACE** **FUNCTION** cal\_func()

**RETURNS** **TRIGGER** **AS**

$$

**BEGIN**

-- 初始化Total为NULL，如果所有课程成绩都是NULL，则保持为NULL

NEW.Total **:=** **NULL**;

**IF** NEW.Chinese **IS** **NOT** **NULL** **THEN**

NEW.Total **:=** **COALESCE**(NEW.Total, 0) **+** NEW.Chinese;

**END** **IF**;

**IF** NEW.Math **IS** **NOT** **NULL** **THEN**

NEW.Total **:=** **COALESCE**(NEW.Total, 0) **+** NEW.Math;

**END** **IF**;

**IF** NEW.English **IS** **NOT** **NULL** **THEN**

NEW.Total **:=** **COALESCE**(NEW.Total, 0) **+** NEW.English;

**END** **IF**;

-- 如果此时Total仍然为NULL，说明所有课程成绩都是NULL，保持Total为NULL

-- 如果Total不是NULL，说明至少有一个课程成绩是非空的

**RETURN** NEW;

**END**;

$$

**LANGUAGE** PLPGSQL;

5．创建触发器函数：

**CREATE** **OR** **REPLACE** **FUNCTION** tri\_insert\_func()

**RETURNS** **TRIGGER** **AS**

$$

**DECLARE**

**BEGIN**

**INSERT** **INTO** test\_trigger\_des\_tbl

**VALUES**(NEW.id1, NEW.id2, NEW.id3);

**RETURN** NEW;

**END**

$$ **LANGUAGE** PLPGSQL;

6. 创建INSERT触发器：

定义一个Before行级触发器，当向Score表插入新元

组或修改Score表中某门课的成绩后，确保Total的取

值正好是三门课成绩总和

**CREATE** **TRIGGER** insert\_trigger

**BEFORE** **INSERT** **OR** **UPDATE** **ON** score

**FOR** **EACH** **ROW**

**EXECUTE** **PROCEDURE** cal\_func();

7.向score表分别插入如下元组：

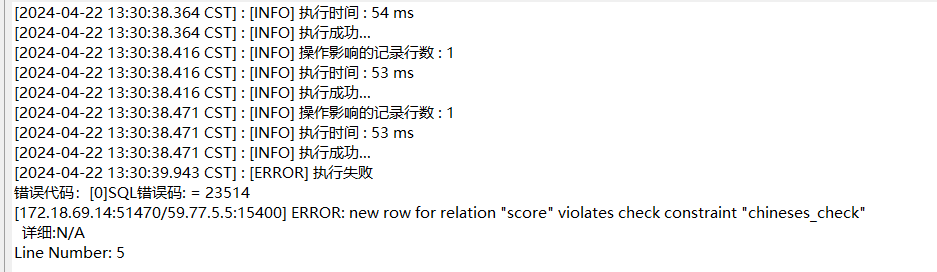
**INSERT** **INTO** score **VALUES**('201515121',90,80,70,250);

**INSERT** **INTO** score **VALUES**('201515122',90,**null**,70,150);

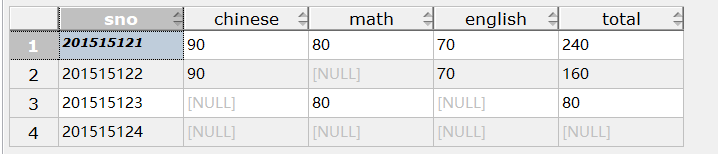
**INSERT** **INTO** score **VALUES**('201515123',**null**,80,**null**,**null**);

**INSERT** **INTO** score **VALUES**('201515124',**null**,**null**,**null**,100);

**INSERT** **INTO** score **VALUES**('201515125',110,90,80,280);



结果有一个插入失败，因为其语文成绩不满足语文成绩的约束条件



成功实现了通过触发器在插入时自动将total设置成对应的成绩和或null

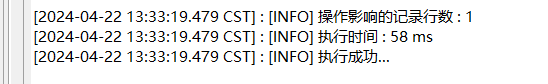
将score表中sno为‘ 201515124’ 的english属性值改

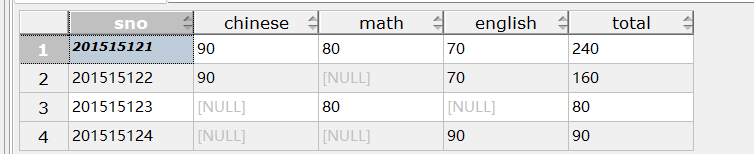
为90，结果如何？

**UPDATE** score

**SET** English**=**90

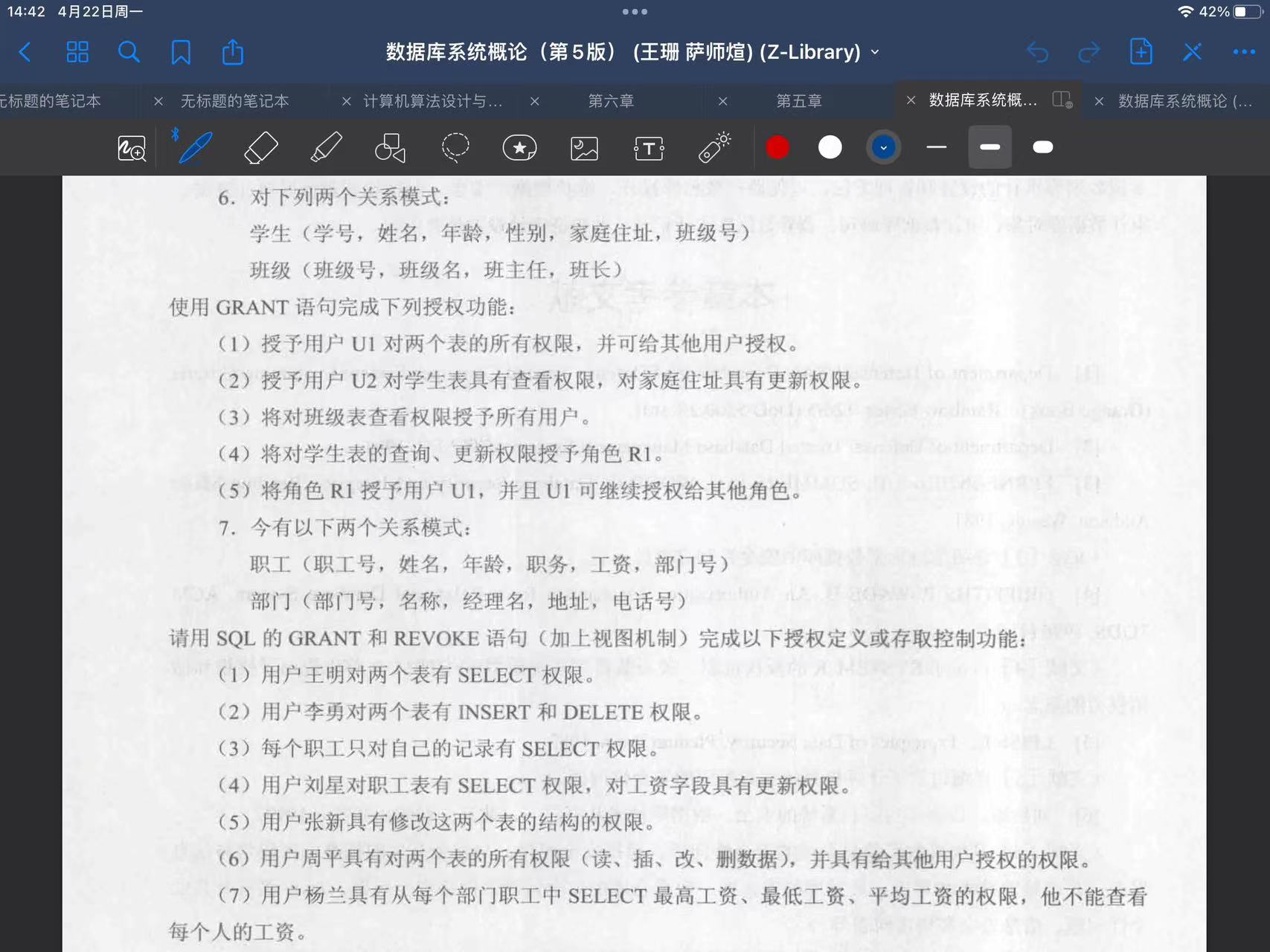
**WHERE** sno**=**'201515124';





运行成功，且该学生的total成绩也被对应更新

数据库安全性



Pdf上第五题，对应旧版课本第六题

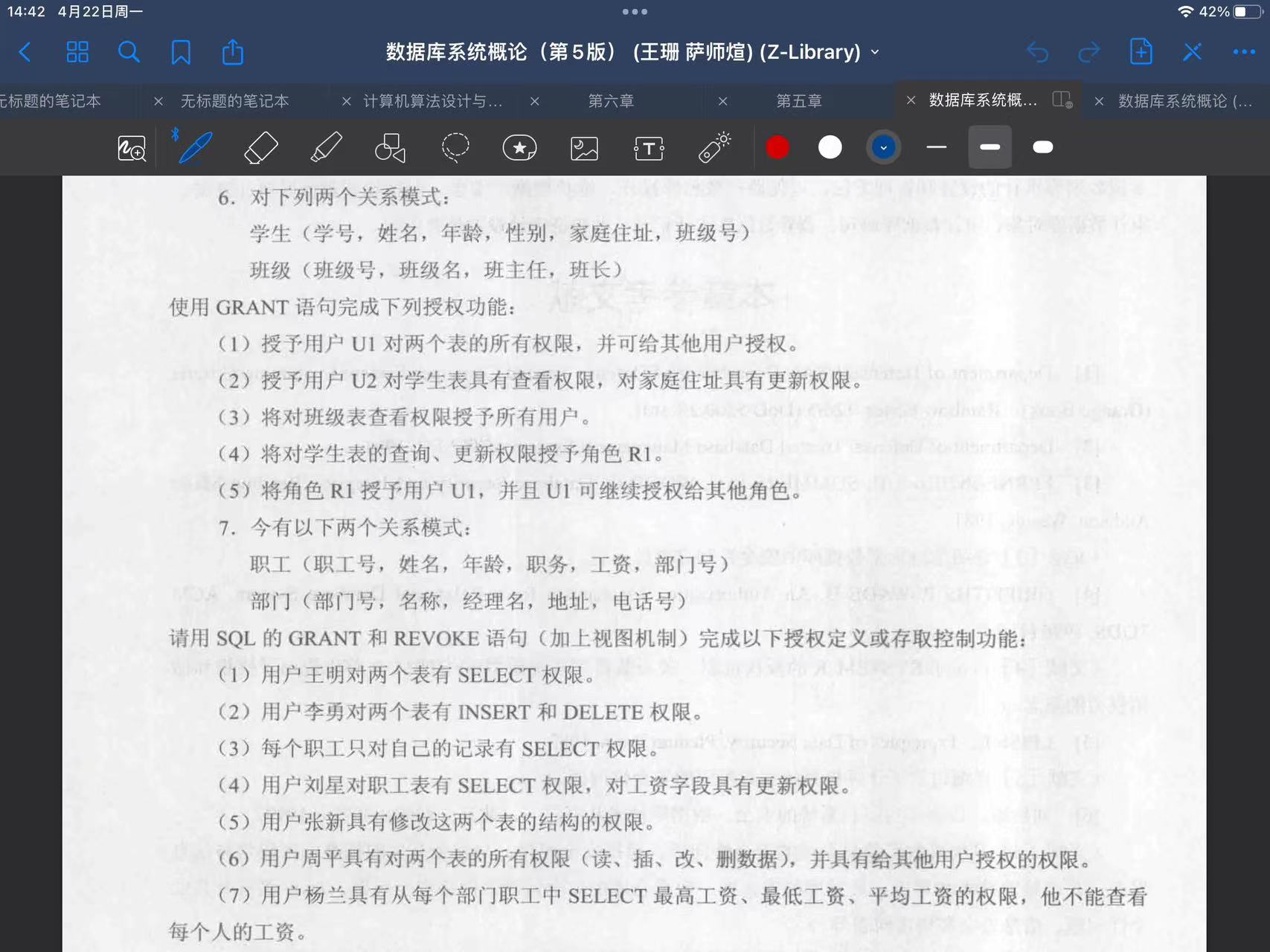
**1.grant** **all** **privileges** **on** **table** 学生，班级 **to** U1 **with** **grant** **option**;

**2.grant** **select** ,**update**(家庭住址) **on** **table** 学生 **to** U2;

**3.grant** **select** **on** **table** 班级 **to** public;

**4.grant** **select** ,**update** **on** **table** 学生 **to** R1;

**5.grant** R1 **to** U1 **with** **admin** **option**;



6 除(3) 对应 旧课本第七题

1.**grant** **select** **on** **table** 职工，部门 **to** 王明;

2.**grant** **insert** **delete** **on** **table** 职工，部门 **to** 李勇;

4.**grant** **select** ,**update**(工资） **on** **table** 职工 **to** 刘星;

5.**grant** **alter** **table** **on** **table** 职工，部门 **to** 张新;

6.**grant** **all** **privileges** **on** **table** 职工，部门 **to** 周平 **with** **grant** **option**;

7.**create** **view** deptsalary **AS**

**select** 部门，名称,MAX(工资）,MIN(工资），AVG（工资）

**from** 职工，部门

**where** 职工.部门号**=**部门.部门号

**group** **by** 职工.部门号;

**grant** **select** **on** **table** 部门工资 **to** 杨兰;

7 针对题6中的两个表

′ (1) 取消王明对部门表的SELECT权限

′ (2) 取消李勇对两个表的INSERT权限

′ (3) 取消刘星对职工表的工资字段的更新权限

′ (4) 撤销周平对两个表的所有权限以及他转授给其他

用户的权限

对应旧课本第八题

1.**revoke** **select** **on** **table** 部门 **from** 王明;

2.**revoke** **insert** **on** **table** 职工，部门 **from** 李勇;

3.**revoke** **update**(工资) **on** **table** 职工 **from** 刘星;

4.**revoke** **all** **privileges** **on** **table** 部门，职工 **from** 周平 **cascade**;