一．根据员工工资计算其个人所得税，3000元为起征点，超出3000元的部分按照10%的比例征收个人所得税，例如：

员工工资表

|  |  |  |
| --- | --- | --- |
| 员工编号 | 工资 | 个人所得税 |
| 1 | 3100 |  |
| 2 | 3500 |  |
| 3 | 3800 |  |
| **.**  **.**  **.** | **.**  **.**  **.** | **.**  **.**  **.** |

则1号员工个人所得税为10元，2号员工个人所得税为50元，3号员工个人所得税为80元…，请使用游标编写一段Transact-SQL程序段，计算每个员工的个人所得税并更新员工工资表中的个人所得税。

DECLARE @eno int, @salary float, @tax float,

DECLARE etax\_cursor CURSOR FOR

SELECT 员工编号,工资,个人所得税

FROM 员工工资表 ---2分

Open etax\_cursor

FETCH NEXT FROM etax\_cursor

INTO @eno, @salary, @tax ---1分

WHILE @@fetch\_status = 0

BEGIN

select @tax= (@salary-3000)\*0.1

update 员工工资表

set 个人所得税=@tax ---1分

FETCH NEXT FROM etax\_cursor

INTO @eno,@salary, @tax

END

Close etax\_cursor

DEALLOCATE etax\_cursor ---1分

**二、按要求实现下列操作（每题2分，共20分）**

现有关系数据库如下，完成下面题目：

学生(学号，姓名，性别，专业、奖学金)

课程(课程号，名称，学分)

学习(学号，课程号，分数)

1.从学生表中查询“数学”专业的学生的学号，姓名，性别。

SELECT 学号，姓名，性别

FROM 学生

WHERE 专业=‘数学’

2.查询平均成绩大于等于90分的成绩信息，输出列名为学号，平均成绩，并按照平均成绩的升序排序。

SELECT 学号，AVG(分数) AS 平均成绩

FROM 学习

GROUP BY 学号

Having AVG(分数)>=90

ORDER BY AVG(分数)

3．查询没有选修课程的学生的名单。

select \*

from 学生

where 学号 not in (select distinct 学号 from 学习 )

或

select A.\*

from 学生 A left outer join 学习 B

on A.sno = B.sno

where B.cno is null

4．检索没有获得奖学金、同时至少有一门课程成绩在90分以上的学生信息，包括学号、姓名和专业。

SELECT学号，姓名，专业

FROM 学生

WHERE奖学金 is null AND 学号 in (select学号 from学习 where分数>90)

5.检索没有任何一门课程成绩在85分以下的所有学生的信息，包括学号、姓名和专业。

SELECT 学号，姓名，专业

FROM 学生

WHERE not exists

(SELECT \*

FROM 学习

WHERE学习.学号=学生.学号and 分数<85)

6. 将各门课程的选修人数及平均成绩定义为视图V\_AVG，包括课程名称，选修人数和平均成绩。

create view V\_AVG(名称, 选修人数, 平均成绩)

as

select 名称,count(学号),avg(分数)

from 学习,课程

where 学习.课程号=课程.课程号

group by 学习.课程号,名称

7．请将学号为'006'，课程号为'C3'，分数为92的记录插入学习表。

insert into 学习

values('006', 'C3',92)

8．请删除数学专业学生的成绩记录。

delete

from 学习

where 学号 in (select 学号 from 学生 where 专业= ‘数学’)

9．请将高等数学这门课程的成绩加5分。

update 学习

set 分数 = 分数 +5

where 课程号 =

( select 课程号 from 课程 where 名称=’高等数学’)

10．请将学生表的查询权限授予张英，并允许张英传播此权限。

Grant select on 学生

To 张英

With grant option

四、应用题（20分）

某网上书店后台数据库的部分关系模式如下：

**会员**（会员编号，用户名，密码，姓名，地址，邮编，电话，消费额，积分）

**图书**（图书编号，类型名称，图书名称，作者，出版社，出版日期，ISBN，价格）

**订单**（订单编号，会员编号，销售额，订购日期，出货日期）

**订单明细**（订单明细编号，订单编号，图书编号，数量）

1. 创建订单表，订单编号唯一识别一个订单，会员编号为外码。要求销售额大于0。
2. 在会员表的积分列上建立降序索引Index\_point。
3. 查询名称中包含“数据库”的图书的图书名称，作者，出版社和出版日期。
4. 查询每个会员的订购图书的情况，显示用户名、图书名、作者、订购日期。
5. 查询提供销售（图书表中有）但没有销售过（没在订单明细表中出现）的图书名称和出版社。
6. 查询已销售的每种图书的销售数量，显示图书编号、销售数量。
7. 查询订购图书数量最多的用户名及其订购的数量。

8、求书店总的销售额。

9、删除没有销售过的图书。

10、将每位会员的积分增加10分。

创建订单表，订单编号唯一识别一个订单，会员编号为外码。要求订购日期不能大于出货日期。

create table order (

订单编号 char(9) primary key,

会员编号 char(10),

销售额 double check(销售额>0),

订购日期 datetime ,

出货日期 datetime,

foreign key(会员编号) references 会员 (会员编号),

)

在会员表的积分列上建立降序索引Index\_point。

create index Index\_point on 会员(积分 desc)

查询名称中包含“数据库”的图书的图书名称，作者，出版社和出版日期。

select 图书名称，作者，出版社，出版日期

from **图书**

where 图书名称 like ‘%数据库%’

查询每个会员的订购图书的情况，显示用户名、图书名、作者、订购日期。

select 用户名、图书名、作者、订购日期

from**会员 A, 图书 B, 订单 C, 订单明细 D**

**where A.** 会员编号 = C. 会员编号 and

**B.** 图书编号 = D. 图书编号 and

C. 订单编号 = D. 订单编号

查询提供销售（图书表中有）但没有销售过（没在订单明细表中出现）的图书名称和出版社。

select 图书名称、出版社

from 图书

where图书编号 not in (

select distinct 图书编号

from**订单明细**

)

查询已销售的每种图书的销售数量，显示图书编号、销售数量。

select 图书编号,sum(数量)

from **订单明细**

group by图书编号

查询订购图书数量最多的用户名及其订购的数量。

select 用户名, sum(数量)

from**会员 A, 订单 B, 订单明细 C**

where **A.** 会员编号 = B. 会员编号 and

B. 订单编号 = C. 订单编号 and

group by会员编号, 用户名

having sum(数量) >=all(

select sum(数量)

from**会员 A, 订单 B, 订单明细 C**

where **A.** 会员编号 = B. 会员编号 and

B. 订单编号 = C. 订单编号 and

group by会员编号, 用户名

)

求书店总的销售额。

select sum(销售额)

from 订单

删除没有销售过的图书。

delete

from 图书

where 图书编号 not in (

select distinct 图书编号

from**订单明细**

)

将每位会员的积分增加10分

upate 会员

set 积分= 积分+10

**五、应用题（每题2分，共20分）**

某书店后台数据库的部分关系模式如下：

**图书类别**（类别代号，类别名）

**图书**（书号，书名，ISBN，作者，单价，类别代号）

**订单**（订单号，顾客编号，订购日期，出货日期）

**订单明细**（订单号，书号，数量，总价）

按要求实现下列操作：

1.显示ISBN为“9787302163305”、“7560922171”或“9787810097987”的图书的书号、书名和ISBN。

2.显示单价高于40元的图书的书号、书名和单价。

3.在图书表中显示所有没有类别代号的图书的书号和书名。

4.统计有多少图书的价格高于30元。

5.统计每本图书的销售数量总和。

6.显示所有图书的书号、书名和单价以及图书对应的类别代号和类别名。

7.显示书名中包含字符串“ASP”的图书的销售订单号和销售总价。

8.创建一个名为ViewBookSale的视图，该视图包含所有图书的销售信息，显示图书的编号、书名以及销量总册数和销售总金额。

9.向图书表中插入一条图书记录：书号为“9”，书名为“SQL Server 2005实现与维护”, ISBN为“9787302163350”, 作者为“Solid”, 单价为79.00, 类别代号为“CO01”。

10.将书号为1的图书的单价打9折。

**答案：**

1.显示ISBN为“9787302163305”、“7560922171”或“9787810097987”的图书的书号、书名和ISBN。

SELECT 书号, 书名, ISBN

FROM 图书

WHERE ISBN IN('9787302163305', '7560922171', '9787810097987')

2.显示单价高于40元的图书的书号、书名和单价。

SELECT 书号, 书名, 单价

FROM 图书

WHERE 单价>40

3.在图书表中显示所有没有类别代号的图书的书号和书名。

SELECT 书号, 书名

FROM 图书

WHERE 类别代号 IS NULL

4.统计有多少图书的价格高于30元。

SELECT COUNT(\*) AS 图书册数

FROM 图书

WHERE 单价 >30

5.统计每本图书的销售数量总和。

SELECT SUM(数量) AS 销售总册数

FROM 订单明细

GROUP BY 书号

6.显示所有图书的书号、书名和单价以及图书对应的类别代号和类别名。

SELECT b.书号, b.书名, b.单价, c.类别代号, c. 类别名

FROM 图书 b LEFT OUTER JOIN 图书类别 c

ON b.类别代号=c.类别代号

7.显示书名中包含字符串“ASP”的图书的销售订单号和销售总价。

SELECT 订单号, 总价

FROM 订单明细

WHERE 书号 IN ( SELECT 书号

FROM 图书

WHERE 书名 LIKE '%ASP%')

8.创建一个名为ViewBookSale的视图，该视图包含所有图书的销售信息，显示图书的编号、书名以及销量总册数和销售总金额。

CREATE VIEW ViewBookSale

AS

SELECT b.书号, b.书名, SUM(数量) AS 销量总册数, SUM(总价) AS 销售总金额

FROM 图书 b LEFT JOIN订单明细i ON b.书号=i.书号

GROUP BY b.书号, b.书名

9.向图书表中插入一条图书记录：书号为“9”，书名为“SQL Server 2005实现与维护”, ISBN为“9787302163350”, 作者为“Solid”, 单价为79.00, 类别代号为“CO01”。

INSERT INTO 图书

VALUES(‘9’,'SQL Server 2005实现与维护','9787302163350','Solid',79.00,'CO01')

10.将书号为1的图书的单价打9折。

UPDATE 图书

SET 单价=单价\*0.9

WHERE 书号=‘1’

**五、简答题(每题5分，共20分)**

1.（1）创建SQL Server登录账户Sql1，密码为“123456”；（2分）

（2）将Comments表的查询、删除和插入的权限授予数据库用户Sql1和Sql2，并且这两个数据库用户还可以将得到的权限再授予其他人。（3分）

2. 参照第三道应用题所给数据库的部分模式，创建一个函数FunBook，根据用户提供的图书类别名查看相应类别图书的详细信息。

3. 根据第三道应用题所给数据库的部分模式，创建一个触发器TrInsUpd，当向图书表中插入或更新一条记录的类别代号时，新记录的类别代号必须在图书类别表中存在，否则提示类别代号不正确。

**答案：**

1.

CREATE LOGIN Sql1 WITH PASSWORD='123456'

GRANT SELECT,DELETE,INSERT ON Comments TO Sql1,Sql2 WITH GRANT OPTION

2. CREATE FUNCTION FunBook(@categoryName nvarchar(50))

RETURNS TABLE

AS RETURN

(SELECT b.\*

FROM 图书 b JOIN 图书类别 c ON b.类别代号=c.类别代号

WHERE c.类别名=@categoryName

)

**3.** CREATE TRIGGER TrInsUpd ON 图书 AFTER INSERT,UPDATE

AS

IF NOT EXISTS(SELECT \* FROM inserted WHERE 类别代号 IN(

SELECT 类别代号 FROM 图书类别))

BEGIN

PRINT '类别代号不正确！'

ROLLBACK TRANSACTION

END

**六、应用题（每题2分，共20分）**

某书店后台数据库的部分关系模式如下：

**图书类别**（类别代号，类别名）

**图书**（书号，书名，ISBN，作者，单价，类别代号）

**顾客**（顾客编号，姓名，地址，推荐人编号）

推荐人编号表示推荐这名顾客注册的老顾客的编号

**订单**（订单号，顾客编号，订购日期，出货日期）

**订单明细**（订单号，书号，数量，总价）

按要求实现下列操作：

1.在图书表中查看有哪些类别代号。

2.显示类别代号是LA01或者单价低于20元的图书的书号、书名、单价和类别代号。

3.显示所有图书的书号、书名、单价和类别代号，首先按照类别代号升序排序，对于类别代号相同的图书再并按照图书单价降序排。

4.统计类别代号是CO01的图书册数和单价总和。

5.统计销售次数超过5次的图书的书号和销售总册数。

6.显示那些有推荐人的顾客的详细信息以及其推荐人的姓名。

7.显示2009年以后购买过图书的顾客的编号和姓名。

8.创建一个名为ViewComputers的视图，该视图包含所有计算机类的图书详细信息以及类别名。

9.向表图书中除了类别代号以外的所有列中插入数据，书号为“11”，书名为“英语阅读词汇双突破”, ISBN为“7560922171”,作者为“杨建荣”，单价为19.00。

10.删除编号为101的图书记录。

答案：

1.在图书表中查看有哪些类别代号。

SELECT DISTINCT 类别代号

FROM 图书

2.显示类别代号是LA01或者单价低于20元的图书的书号、书名、单价和类别代号。

SELECT 书号, 书名, 单价, 类别代号

FROM 图书

WHERE 类别代号='LA01' OR 单价<20

3.显示所有图书的书号、书名、单价和类别代号，首先按照类别代号升序排序，对于类别代号相同的图书再并按照图书单价降序排。

SELECT 书号, 书名, 单价, 类别代号

FROM 图书

ORDER BY 类别代号, 单价 DESC

4.统计类别代号是CO01的图书册数和单价总和。

SELECT COUNT(书号) AS 图书册数, SUM(单价) AS 单价总和

FROM 图书

WHERE 类别代号='CO01'

5.统计销售次数超过5次的图书的书号和销售总册数。

SELECT 书号, SUM(数量) AS 销售总册数

FROM 订单明细

GROUP BY 书号

HAVING COUNT（订单号）>5

6.显示那些有推荐人的顾客的详细信息以及其推荐人的姓名。

SELECT c1.\*, c2.姓名

FROM 顾客 c1, 顾客 c2

WHERE c1.推荐人编号=c2.顾客编号

7.显示2009年以后购买过图书的顾客的编号和姓名。

SELECT c.顾客编号, c.姓名

FROM 顾客 c

WHERE EXISTS ( SELECT \*

FROM 订单 o

WHERE o.订购日期>='01/01/2009' AND c.顾客编号=o.顾客编号)

8.创建一个名为ViewComputers的视图，该视图包含所有计算机类的图书详细信息以及类别名。

CREATE VIEW ViewComputers

AS

SELECT b.\*, c.类别名

FROM 图书 b JOIN 图书类别 c ON b.类别代号=c.类别代号

WHERE c.类别名='计算机'

9.向表图书中除了类别代号以外的所有列中插入数据，书号为“11”，书名为“英语阅读词汇双突破”, ISBN为“7560922171”,作者为“杨建荣”，单价为19.00。

INSERT INTO 图书(书号,书名,ISBN,作者,单价)

VALUES(‘11’,'英语阅读词汇双突破','7560922171','杨建荣',19.00)

10.删除编号为101的图书记录。

DELETE FROM 图书

WHERE 书号=’101’

**七、简答题(每题5分，共20分)**

1.（1）创建基于登录账户Sql2的数据库用户Sql2，并为该用户指定默认架构Sale。（3分）

（2）把用户U5对SC表的INSERT权限收回。（2分）

2. 参照第三道应用题所给数据库的部分模式，创建一个存储过程PrcSelect，根据用户提供的图书类别名查看相应类别图书的详细信息。

3. 根据第三道应用题所给数据库的部分模式，创建一个函数FunBookSale，该函数根据给定的书号返回该图书销售的数量。

答案：

1.（1）（3分）

CREATE USER Sql2 FROM LOGIN Sql2

WITH DEFAULT\_SCHEMA=Sale

（2）（2分）

REVOKE INSERT ON TABLE SC FROM U5

**2.** CREATE PROCEDURE PrcSelect @categoryName nvarchar(50)

AS

SELECT b.\*

FROM 图书 b JOIN 图书类别 c ON b.类别代号=c.类别代号

WHERE c.类别名=@categoryName

**3.** CREATE FUNCTION FunBookSale (@bookID int)

RETURNS int

AS

BEGIN

DECLARE @quantity int

SELECT @quantity=SUM(quantity)

FROM OrderItems

WHERE bookID=@bookID

RETURN @quantity

END

八、请按要求完成操作（30分）

某校学生-课程数据库部分关系模式如下：

**学生表 Student(Sno,Sname,Sage,Ssex，department)**

**课程表Course(Cno,Cname, Ccredit，Tno)**

**成绩表 SC(Sno,Cno, grade)**

**教师表Teacher(Tno,Tname，tsex)**

1. 数据查询（每题2分，共20分）
2. 查询张姓学生的信息

（2） 查询计算机学院女学生的信息，并按年龄降序排列

（3） 查询选修了项天老师课程的学生信息

（4） 查询至少有一门课程与李燕所选课程相同的学生的学号、姓名

（5） 求各门课程最高成绩、最低成绩、课程号。

1. 求选修了全部课程的学生信息
2. 求1995年前出生的学生信息
3. 查询所有学生的选课情况，包括没有选课的学生
4. 查询两门以上不及格课程的同学的学号及其平均成绩
5. 检索至少选修两门课程的学生学号

2. 请SQL语句完成下列要求（共10分）

（1）创建课程表（2分）

（2） 删除成绩表中成绩为空的记录（2分）

（3） 在课程表的教师编号列上建立降序索引。（2分）

（4）在学生表上创建一个触发器trigger\_delete，当删除学生信息时，将删除的学生保存到oldstudent表中，oldstudent和student表结构相同。（4分）

答案：

数据查询（每题2分，共20分）

1）. select \* from student where sname like '张%'

2） select \* from student where department ='IS' and ssex='女' order by sage desc

3） select \*

from student a,sc b,course c,teacher d

where a.sno = b.sno and

b.cno = c.cno and

c.tno = d.tno and

tname = '项天'

4） select a.sno,sname

from student a,sc b

where a.sno = b.sno and sname <>'李燕' and

cno in(

select distinct cno

from student c,sc d

where c.sno = d.sno and sname = '李燕'

)

5） select cno,max(grade),min(grade)

from sc

grop by cno

6） select \*

from student

where not exists(

select \*

from course

where not exists(

select \*

from sc

where sno = student.sno and

cno = course.cno

)

)

7） select \*

from student

where (year(getdate())-sage)<1995

8） select\*

from student left outer join sc

on student.sno = sc.sno

9） select sno ,avg(grade)

from sc

where sno in (

select sno

from sc

group by sno

having count(case when grade<60 then 1)>=2

)

10） select sno

from sc

group by sno

having count(\*)>=2

2. 请SQL语句完成下列要求（共10分）

1）create table course(

cno varchar(5) primary key,

cname varchar(30),

ccredit float,

tno varchar(5),

foreign key (tno) refrences teacher(tno)

2) delete

from sc

where grade is null

3) create index index\_tno on teacher(tno desc)

4) create trigger trigger\_delete

on student

after delete

as

begin

insert into oldstudent

select \* from deleted

end

九. 已知学生数据库中存放了这样的两张表，一张为毕业生信息表(graduation)，记录毕业生的一些基本信息，一张为学生缴费表(fee)，记录了学生的缴费信息。由于学生毕业，我们要从学生信息表中把毕业生的记录删除，但是如果这个学生欠费的话，则不允许删除这个学生的记录。 (字段名都是中文，可以直接使用) (10分)

graduation (学号 姓名 性别 地址 联系方式)

fee(学号 姓名 已交费用 欠费)

1. 建立一个存储过程pro\_deletestudent在graduation表中删除指定毕业学生的信息，输入参数为学号。
2. 在graduation表上建立一个触发器tr\_checkfee，判断要删除的学生是否欠费，欠费则不允许删除该记录，否则删除该记录

**答案：**

1) 建立一个存储过程pro\_deletestudent在graduation表中删除指定毕业学生的信息，输入参数为学号。

create procedure pro\_deletestudent

@sno char(9)

as

begin

delete from graduation where 学号 = @sno

end

(2) 在graduation表上建立一个触发器tr\_checkfee，判断要删除的学生是否欠费，欠费则不允许删除该记录，否则删除该记录

**create TRIGGER** tr\_checkfee

**on** graduation

**for** delete

**as**

begin

DECLARE @sno char(9);

Select @sno=deleted.sno from **deleted**

if exists(select \* from fee where 学号 = @sno and欠费>0)

rollback

end

十、某书店后台数据库的部分关系模式如下：

**图书类别**（类别代号，类别名）

**图书**（书号，书名，ISBN，作者，单价，类别代号）

**顾客**（顾客编号，姓名，地址，推荐人编号）

推荐人编号表示推荐这名顾客注册的老顾客的编号

**订单**（订单号，顾客编号，订购日期，出货日期）

**订单明细**（订单号，书号，数量，总价）

按要求实现下列操作：

1.使用数据定义语言建立顾客、订单明细两张表的结构（注意添加相应的主外键约束）。（6分）

1. 在图书表中查看有哪些类别代号。（2分）
2. 显示类别代号是LA01或者单价低于20元的图书的书号、书名、单价和类别代号。（2分）
3. 显示所有图书的书号、书名、单价和类别代号，首先按照类别代号升序排序，对于类别代号相同的图书再并按照图书单价降序排。（3分）
4. 统计类别代号是CO01的图书册数和单价总和。（3分）
5. 显示有推荐人的顾客的详细信息以及其推荐人的姓名。（3分）
6. 删除编号为101的图书记录。（2分）
7. 向表图书中除了类别代号以外的所有列中插入数据，书号为“11”，书名为“英语阅读词汇双突破”, ISBN为“7560922171”,作者为“杨建荣”，单价为19.00。（2分）
8. 创建一个视图，要求显示订单编号、书名、ISBN、并且要求书的作者为“王珊”。（3分）
9. 创建一个角色ROLE1，将顾客表的查询、更新、删除的权限授予该角色，并使用该角色对张明、赵强、李峰授权。（4分）

答案：

1.使用数据定义语言建立顾客、订单明细两张表的结构（注意添加相应的主外键约束）。（6分）**顾客**（顾客编号，姓名，地址，推荐人编号）

**订单明细**（订单号，书号，数量，总价）

CREATE TABLE 顾客

(

顾客编号 CHAR(9) PRIMARY KEY,

姓名 CHAR(10),

地址 VARCHAR(20),

推荐人编号 CHAR(9),

FOREIGN KEY (推荐人编号) REFERENCES 顾客(顾客编号)

);

CREATE TABLE 订单明细

(

订单号 CHAR(8),

书号 CHAR(10),

数量 SMALLINT),

总价 NUMERIC(8,2),

PRIMARY (订单号, 书号),

FOREIGN KEY (订单号) REFERENCES 订单(订单号),

FOREIGN KEY (书号) REFERENCES 图书(书号)

);

2.在图书表中查看有哪些类别代号。（2分）

SELECT DISTINCT 类别代号

FROM 图书

3.显示类别代号是LA01或者单价低于20元的图书的书号、书名、单价和类别代号。（2分）

SELECT 书号, 书名, 单价, 类别代号

FROM 图书

WHERE 类别代号='LA01' OR 单价<20

4.显示所有图书的书号、书名、单价和类别代号，首先按照类别代号升序排序，对于类别代号相同的图书再并按照图书单价降序排。（3分）

SELECT 书号, 书名, 单价, 类别代号

FROM 图书

ORDER BY 类别代号, 单价 DESC

5.统计类别代号是CO01的图书册数和单价总和。（3分）

SELECT COUNT(书号) AS 图书册数, SUM(单价) AS 单价总和

FROM 图书

WHERE 类别代号='CO01'

6.显示有推荐人的顾客的详细信息以及其推荐人的姓名。（3分）

SELECT c1.\*, c2.姓名

FROM 顾客 c1, 顾客 c2

WHERE c1.推荐人编号=c2.顾客编号

7.删除编号为101的图书记录。（3分）

DELETE FROM 图书

WHERE 书号=’101’

8.向表图书中除了类别代号以外的所有列中插入数据，书号为“11”，书名为“英语阅读词汇双突破”, ISBN为“7560922171”,作者为“杨建荣”，单价为19.00。（3分）

INSERT INTO 图书(书号,书名,ISBN,作者,单价)

VALUES(‘11’,'英语阅读词汇双突破','7560922171','杨建荣',19.00)

9. CREATE ROLE ROLE1

GRANT SELECT,UPDATE,DELETE

ON TABLE 顾客

TO ROLE1

GRANT ROLE1

TO 张明，赵强，李峰

**十一、综合题**（每题20分，共40分）

1、某服装销售公司拟开发一套服装采购管理系统，以便对服装采购和库存进行管理。经过需求分析和概念设计、逻辑设计阶段最后得到的关系模式包括：

库管员（库管员编号，姓名，级别）Storekeeper（sno,sname,level）

仓库信息（仓库编号，仓库位置，仓库容量）Storehouse(stno,stadress,stvolume)

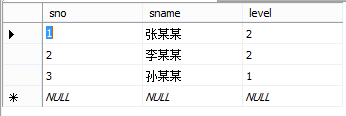
服装（服装编码，服装描述，服装类型，尺码，面料，销售价格）Dress(dno，ddescribe,dtype,dsize,dplus,dprice)供应商（供应商编码，供应商名称，地址，联系电话，企业法人）Supplier(suno,suname,suaddress,sutel,superson)关系模式之间的关联关系为：每个仓库有一个库管员，一个仓库管理员可以管理多个仓库；每种服装有一个供应商，每个供应商提供多种类型的衣服；每种衣服放在同一个仓库里，每个仓库里存放多种类型的衣服。

请完成以下题目：

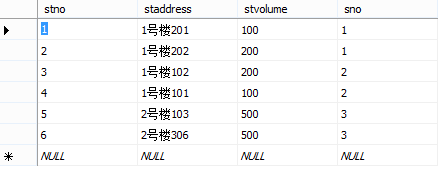
1）请用把以上四个关系模式用SQL创建到数据中（每个创建表的语句2分，共计8分）

2）假设表中已经存在以下数据，请完成a)~e)中SQL语句的编写（每个2分，共12分）

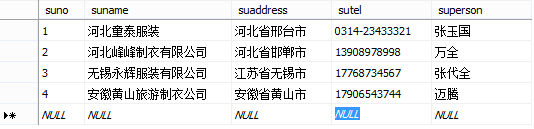
库管表中的数据：



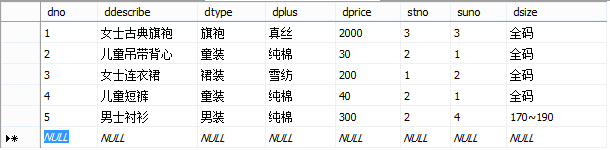
仓库表中的数据：



供应商表中的数据：



服装表中的数据：



1. 请查询存放在1号楼201仓库中中的服装信息
2. 请查询河北童泰服装厂生产的服装信息存储的仓库信息
3. 由于库存销售量上升，现在“女士古典旗袍”的尺码已经不全了，只剩下155~165的号了，请修改该服装的尺码信息
4. 库管员孙某某的离职，现在他的库管工作全部由新来的员工李爽承担，请将李爽的信息插入到库管员表中（李爽的等级是3级），并把原来的孙某某的库房指定给李爽管理。
5. 由于换季，现在女士连衣裙已经下架不再销售，请将服装中的女士连衣裙删除。

2、根据第一题中的需求描述，即某服装销售公司拟开发一套服装采购管理系统，编写以下数据库程序。

1）编写一个自定义函数实现按照某个的库管员查找其所管辖的仓库中的服装的供应商的联系人。（5分）

2）编写一个存储过程，将参数指定的服装信息插入到数据库dress表中。（5分）

3）定义一个触发器，在插入供应商信息的时候检查联系人不能为空值（5分）

4）定义一个游标，实现统计服装信息中价格在500元以内的服装数量。（5分）

答案：

1. 1)创建表的SQL语句：（每个创建表的SQL2分，共8分）

--创建库管员表

create table Storekeeper(

sno int primary key,

sname varchar(20),

level char(2)

)

--创建仓库表

create table Storehouse(

stno int primary key,

staddress varchar(100),

stvolume int,

sno int,

foreign key (sno) references Storekeeper(sno)

)

--创建供应商表

create table Supplier(

suno int primary key,

suname varchar(20),

suaddress varchar(100),

sutel varchar(20),

superson varchar(20)

)

--创建服装表

create table Dress(

dno int primary key,

ddescribe varchar(100),

dtype varchar(20),

dsize varchar(20),

dplus varchar(20),

dprice int,

stno int,

suno int,

foreign key (stno) references Storehouse(stno),

foreign key (suno) references Supplier(suno)

)

2)（共计12分）

a)：select dress.\* from dress,storehouse

where dress.stno=storehouse.stno

and storehouse.staddress='1号楼'(2分)

b)： select distinct storehouse.\* from storehouse,dress,supplier

where dress.stno=storehouse.stno and supplier.suno=dress.suno

and supplier.suname='河北童泰服装'（2分）

c)：update dress set dsize='155~165' where ddescribe='女士古典旗袍'（2分）

d): insert into storekeeper values(4,'李爽',3)（2分）

update storehouse set sno=4 where sno=3（2分）

e)：delete from dress where ddescribe='女士连衣裙'（2分）

2. 1）自定义函数：（5分）

create function fun1(@name varchar(20))

returns varchar(20)

as

begin

return

(select superson from supplier

where suno in(

select suno from dress,Storehouse

where dress.stno=storehouse.stno

and storehouse.sno=(select sno from storekeeper

where sname=@name)

))

end

2）自定义存储过程：（5分）

create proc myproc1

@dno int,

@ddescribe varchar(100),

@dsize varchar(20),

@dplus varchar(20),

@dprice int,

@dtype varchar(20),

@stno int,

@suno int

as

begin

insert into dress values(@dno,@ddescribe,@dsize,

@dplus,@dprice,@dtype,@stno,@suno)

end

3）触发器：（5分）

create trigger trigger1 on supplier

for insert

as

begin

declare @name varchar(20)

select @name=superson from supplier

if(@name is null)

RAISERROR ('联系人的名字不能为null', 16, 10)

ROLLBACK TRANSACTION

End

4）声明游标：（5分）

--声明游标

declare cursor1 cursor

for select dprice from dress

for read only

--打开游标

open cursor1

--定义变量

declare @num int,@price int

set @num=0

--取游标中的数据

fetch next from cursor1 into @price

--循环

while(@@FETCH\_STATUS=0)

begin

if(@price<500)

set @num=@num+1;

fetch next from cursor1 into @price

end

print @num

--关闭游标

close cursor1

--释放游标

deallocate cursor1