第一次：

1. 考虑如下所示的关系数据库。这些关系上适当的主码是什么？

职工（姓名，街道，城市）

工作（姓名，公司名，工资）

公司（公司名，城市）

答：职工的主码：姓名

工作的主码：姓名，公司名

公司的主码：公司名

2. 考虑如下所示的银行数据库。

a.适当的主码是什么？

b.给出你选择的主码，确定适当的外码。

支行（支行名，属在城市，资产）

客户（客户名，街道，城市）

贷款（贷款号，支行名，金额）

贷款人（客户名，贷款号）

账户（账号，支行名，余额）

储户（客户名，账号）

答：支行的主码：支行名

客户的主码：客户名

贷款的主码：贷款号

外码：支行名

贷款人的主码：客户名，贷款号

外码1：客户名 外码2：贷款号

账户的主码：账号

外码：支行名

储户的主码：客户名，账号

外码1：客户名 外码2：账号

3. 考虑教材图2-8所示advisor关系，advisor的主码是s\_id。假设一个学生可以有多位指导教师。那么，s\_id还是advisor的主码吗？如果不是，advisor的主码会是什么呢？

答：假设一个学生可以有多位指导教师。那么s\_id就不再是advisor的主码。advisor的主码是（s\_id,i\_id）

第二次：

1. 写出对应教材图3-18的保险公司数据库所有模式的SQL DDL。在数据类型上做合理的假设，确保声明主码和外码。person(driver\_id,name,address)

答：create table person(

driver\_id varchar2(18) ,

name varchar2(20) not null,

address varchar2(50),

primary key(driver\_id)

);

2. 写出对应教材图3-18的保险公司数据库所有模式的SQL DDL。在数据类型上做合理的假设，确保声明主码和外码。car(license,model,year)

答：create table car(

licence varchar2(10),

model varchar2(20),

year numeric(4),

primary key(licence)

);

3. 写出对应教材图3-18的保险公司数据库所有模式的SQL DDL。在数据类型上做合理的假设，确保声明主码和外码。accident(report\_number,accident\_date,location)

答：create table accident(

report\_number int,

accident\_date date,

location varchar2(50),

primary key(report\_number)

);

4. 写出对应教材图3-18的保险公司数据库所有模式的SQL DDL。在数据类型上做合理的假设，确保声明主码和外码。owns(driver\_id,license)

答：create table owns(

driver\_id varchar2(18),

licence varchar2(10),

primary key(driver\_id,licence),

foreign key(driver\_id) references person,

foreign key(licence) references

);

5. 写出对应教材图3-18的保险公司数据库所有模式的SQL DDL。在数据类型上做合理的假设，确保声明主码和外码。participated(report\_number,license,driver\_id,damage\_amount)

create table participated(

report\_number int,

licence varchar2(10),

driver\_id varchar2(18),

damage\_amount int,

primary key(report\_number,licence),

foreign key(driver\_id) references person,

foreign key(report\_number) references accident,

foreign key(licence) references car

);

第三次：

1.使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

找出Comp.Sci. 系开设的具有3个学分的课程名称。

2. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

找出名叫Einstein的教师所教的所有学生的ID，保证结果中没有重复。

3. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

找出教师的最高工资。

4. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

找出工资最高的所有教师(可能有不止一位教师具有相同的工资)。

5. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

找出2019年秋季开设的每个课程段的选课人数。

6. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

从2019年秋季开设的所有课程段中，找出最多的选课人数。

7. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

找出在2019年秋季拥有最多选课人数的课程段。

1. select title

from course

where dept\_name = 'Comp. Sci.'

and credits = 3

2.

select distinct student.ID

from student,takes,instructor,teaches

where

student.ID=takes.ID AND

instructor.ID=teaches.ID AND

takes.course\_id=teaches.course\_id AND

takes.sec\_id=teaches.sec\_id AND

takes.semester=teaches.semester AND

takes.year=teaches.year AND

instructor.name = 'Einstein';

3.

select max(salary)

from instructor;

4.

select ID, name

from instructor

where salary = (select max(salary) from instructor);

5.

select course\_id, sec\_id, count(ID)

from section natural join takes

where semester = 'Autumn'

and year = 2019

group by course\_id, sec\_id;

6.

select max(enrollment)

from (select count(ID) as enrollment

from section natural join takes

where semester = 'Autumn'

and year = 2019

group by course\_id, sec\_id);

7.

with sec\_enrollment as (

select course\_id, sec\_id, count(ID) as enrollment

from section natural join takes

where semester = 'Autumn'

and year = 2019

group by course\_id, sec\_id)

select course\_id, sec\_id

from sec\_enrollment

where enrollment = (select max(enrollment) from sec\_enrollment);

作业四：

一．1. 根据教材图3-18中的保险公司数据库，其中加下划线的是主码。为这个关系数据库构造出如下sQL查询。

找出2019年其车辆出过交通事故的人员总数。

2. 根据教材图3-18中的保险公司数据库，其中加下划线的是主码。为这个关系数据库构造出如下sQL查询。

找出和"John Smith" 的车有关的交通事故数量

3. 根据教材图3-18中的保险公司数据库，其中加下划线的是主码。为这个关系数据库构造出如下sQL查询。

向数据库中增加一个新的事故，对每个必需的属性可以设定任意值。

4．根据教材图3-18中的保险公司数据库，其中加下划线的是主码。为这个关系数据库构造出如下sQL查询。

对事故报告编号为"AR2197"中的车牌是"AABB2000"的车辆损坏保险费用更新到3000美元。

5. 根据教材图3-18中的保险公司数据库，其中加下划线的是主码。为这个关系数据库构造出如下sQL查询。

删除“John Smith”拥有的马自达车(Mazda)

二．

1. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

找出所有至少选修了一门Comp.Sci.课程的学生姓名，保证结果中没有重复的姓名。

2. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

找出所有没有选修在2019年春季之前开设的任何课程的学生的ID和姓名。

3. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

找出每个系教师的最高工资值。可以假设每个系至少有一位教师。

4. 使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

从前述查询所计算出的每个系最高工资中选出最低值。

答案：

一、1.

select count (distinct name)

from accident, participated, person

where accident.report\_number = participated.report\_number

and participated.driver\_id = person.driver\_id

and date between date ’2019-00-00’ and date ’2019-12-31’

2. select count (\*)

from owns, participated, person

where owns.license = participated.license

and owns.driver\_id = person.driver\_id

and person.name = ’John Smith’

3. insert into accident

values (4007, ’2001-09-01’, ’Berkeley’);

insert into participated

select o.driver\_id, c.license, 4007, 3000

from person p, owns o, car c

where p.name = ’Jones’ and p.driver\_id = o.driver\_id and

o.license = c.license and c.model = ’Toyota’

4. update participated

set damage\_amount=3000

where report\_number='AR2197' and license='AABB2000'

5. delete car

where model = ’Mazda’ and license in

(select license

from person p, owns o

where p.name = ’John Smith’ and p.driver\_id = o.driver\_id)

二、1、select distinct name

from course ,takes,student

where course.dept\_name = 'Comp. Sci.'

and course.course\_id=takes.course\_id

and student.id=takes.id

2、select id,name

from student

where not exists

(select \*

from takes

where year <2019

and student.id=takes.id)

3、select dept\_name,max(salary)

from instructor

group by dept\_name

4、select min(max\_salary)

from (select dept\_name,max(salary) max\_salary

from instructor

group by dept\_name)

作业五

第一题：1、使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

显示所有教师的列表，列出他们的ID、姓名以及所讲授课程段的编号。

对于没有讲授任何课程段的

教师， 确保结果中出现他们的ID与姓名。

2、使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

显示所有教师的列表，列出他们的ID、姓名以及所讲授课程段的数量。

对于没有讲授任何课程段的

教师，确保将课程段数量显示为0。

在你的查询中应该使用外连接，不能使用标量子查询。

3、使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

使用标量子查询，不使用外连接写出上述查询。

第二题：1、使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

显示2020年春季开设的所有课程的列表，包括讲授课程段的教师的姓名。如果一个课程段有不止一位教师讲授，那么有多少位教师，此课程段在结果中就出现多少次。如果一个课程段没有任何教师，它也要出现在结果中，相应的教师名置为“-"。

2、使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

显示所有系的名称，以及该系中教师的总数，不能使用标量子查询。确保正确处理没有教师的系。

3、使用大学模式（参见教材第6版第2章），用SQL写出如下查询。

使用标量子查询，不使用外连接写出上述查询。

第三题：假设给你一个关系grade\_poins(grad\_e,points)，它提供从takes关系中用字母表示的成绩等级到数字表示的得分之间的转换。例如，“A"等级可指定为对应于4分，“A-“对应于3.7分，“B+”对应于3.3分，“B”对应于3分，等等。学生在某门课程(课程段)上所获得的等级分值被定义为该课程段的学分乘以该生得到的成绩等级所对应的数字表示的得分。

给定上述关系和我们的大学模式，用SQL写出下面的每个查询。为简单起见，可以假设没有任何takes元组在grade上取null值。

1、根据ID为12345的学生所选修的所有课程，找出该生所获得的等级分值的总和。

2、找出上述学生等级分值的平均值(GPA)，即用等级分值的总和除以相关课程学分的总和。

3、找出每个学生的ID和等级分值的平均值。

一、1、select ID, name,sec\_id

from instructor natural left outer join teaches；

2、select ID,name, count(sec\_id) as Numberofsetions

from instructor natural left outer join teaches

group by ID,name

3、select ID,name,

(select count(\*) a from teaches T where T.id = I.id) Numberofsetions

from instructor I;

二、1、select course\_id, seC\_id, ID,decode(name, null, '-', name) as name

from (section natural left outer join teaches)

natural left outer join instructor

where semester='Spring' and year= 2020;

2、select dept\_name, count(ID)

from department natural left outer join instructor

group by dept\_name;

3、select dept\_name,

(select count(\*) a from instructor i where i.dept\_name=d.dept\_name ) Numberofinstructors

from department d;

三、

1、(select sum(credits \* points)

from (takes natural join course) natural join grade\_points

whereID = '12345')

union

(select 0

from student

where takes.ID = '12345' and

not exists ( select \* from takes where takes.ID = '12345'))

2、(select sum(credits \* points)/sum(credits) as GPA

from (takes natural join course) natural join grade points

where ID = '12345')

union

(select null as GPA

from student

where takes.ID = '12345' and

not exists ( select \* from takes where takes.ID = '12345'))

3、(select ID, sum(credits \* points)/sum(credits) as GPA

from (takes natural join course) natural join grade points

group by ID)

union

(select ID, null as GPA

from student

where not exists ( select \* from takes where takes.ID = student.ID))

作业六

一、给定上述关系和我们的大学模式，用SQL写出下面的每个查询。为简单起见，可以假设没有任何takes元组在grade上取null值。"

1、给Comp. Sci. 系的每位教师涨10%的工资。

2、删除所有未开设过(即没有出现在section关系中)的课程

3、把每个在tot\_cred属性上取值超过100的学生作为同系的教师插入，工资为10000美元。

4、重新统计每个学生的总学分tot\_cred

答案：1、update instructor

set salary=salary\*1.1

where dept\_name = 'Comp. Sci.';

2、delete from course

where course\_id not in

(select course\_id from section);

3、insert into instructor

select ID, name, dept\_name, 10000

from student

where tot\_cred > 100;

4、update student

set tot\_cred = (select sum(credits)

from takes natural join course

where student.ID=takes.ID

and takes.grade <>'F'

and takes.grade is not null);

二、使用大学模式（参见教材第6版第2章），用SQL写出如下插入、删除和更新语句。

1. 创建一门课程“CS-00”,其名称为“Weekly Seminar”，学分为0。
2. 创建该课程在2019年秋季的一个课程段，sec\_id为1。
3. 让Comp. Sei. 系的每个学生都选修上述课程段。
4. 删除名为Chavez的学生选修上述课程段的信息。
5. 删除课程CS-001。如果在运行此删除语句之前，没有先删除这门课程的授课信息(课程段)，会发生什么事情?
6. 删除课程名称中包含“database”的任意课程的任意课程段所对应的所有takes元组，在课程名的匹配中忽略大小写。

答案：1、insert into course values('CS-001','Weekly Seminar',null,0);

2、insert into section(course\_id,sec\_id,semester,year) values('CS-001',1,'Fall',2019);

3、insert into takes(id,course\_id,sec\_id,semester,year)

select id ,'CS-001',1,'Fall',2019

from student

where dept\_name = 'Comp. Sci.';

4、delete from takes

where course\_id='CS-001'

and sec\_id=1

and id in (select id from student where name='Chavez');

5、delete from course where course\_id='CS-001' ;

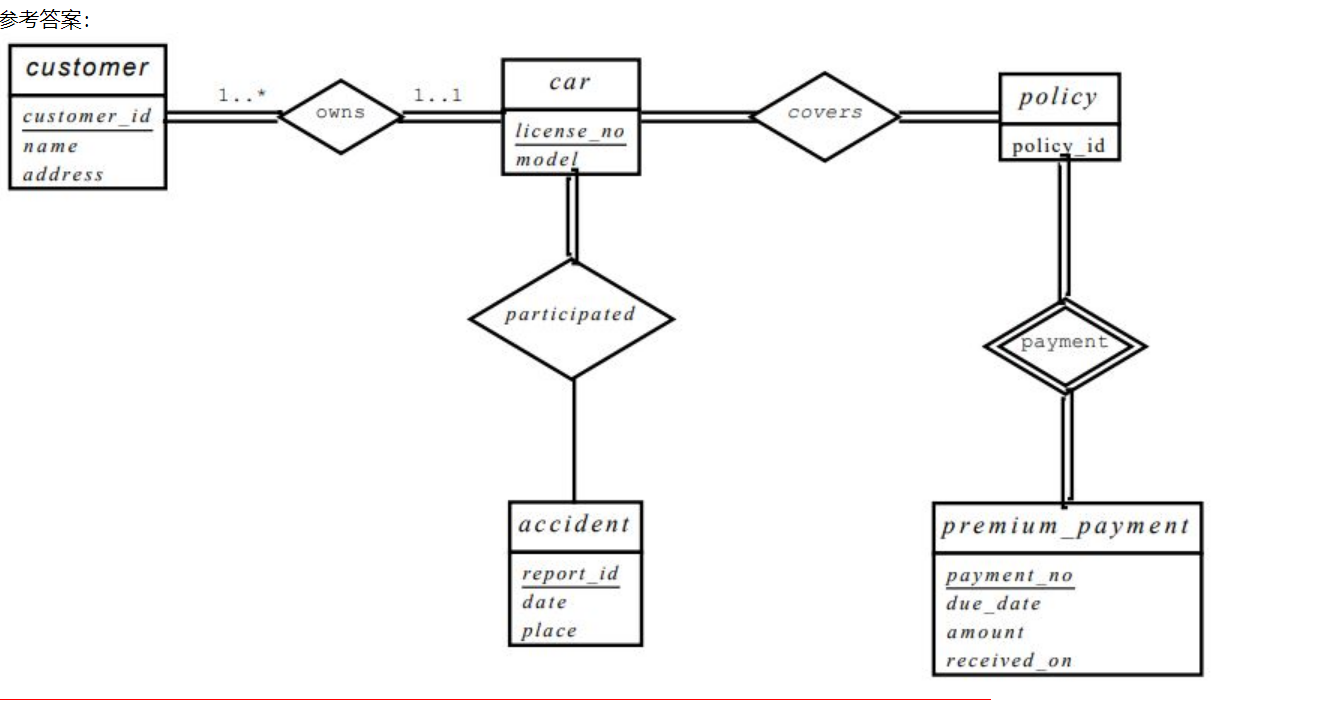
如果在运行此删除语句之前，没有先删除这门课程的授课信息(课程段)，则SECTION表中级联删除course\_id='CS-001'的行，同样，takes与teaches表中也级联删除course\_id='CS-001'的行。（因为section中有约束：foreign key (course\_id) references course (course\_id)on delete cascade，takes与teaches也有类似约束）

6、delete from takes

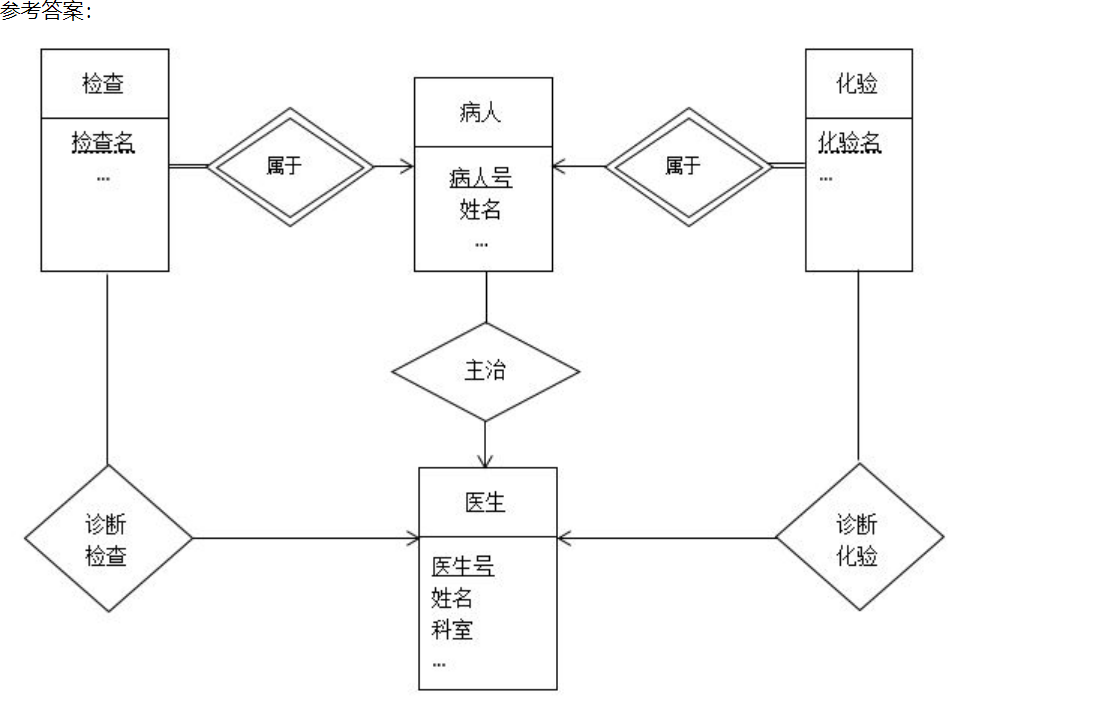
where course\_id in (select course\_id from course where lower(title) like '%database%');

作业七：

1. 为车辆保险公司构建一个E-R图，它的每个客户有一辆或多辆车。每辆车关联零次或任意次事故的记录。每张保险单为一辆或多辆车保险，并与一个或多个保费支付相关联。每次支付只针对特定的一段时间，具有关联的到期日和缴费日。



1. 为医院构建一个包含一组病人和一组医生的E-R图。为每个病人关联一组不同的检查和化验记录。



作业八

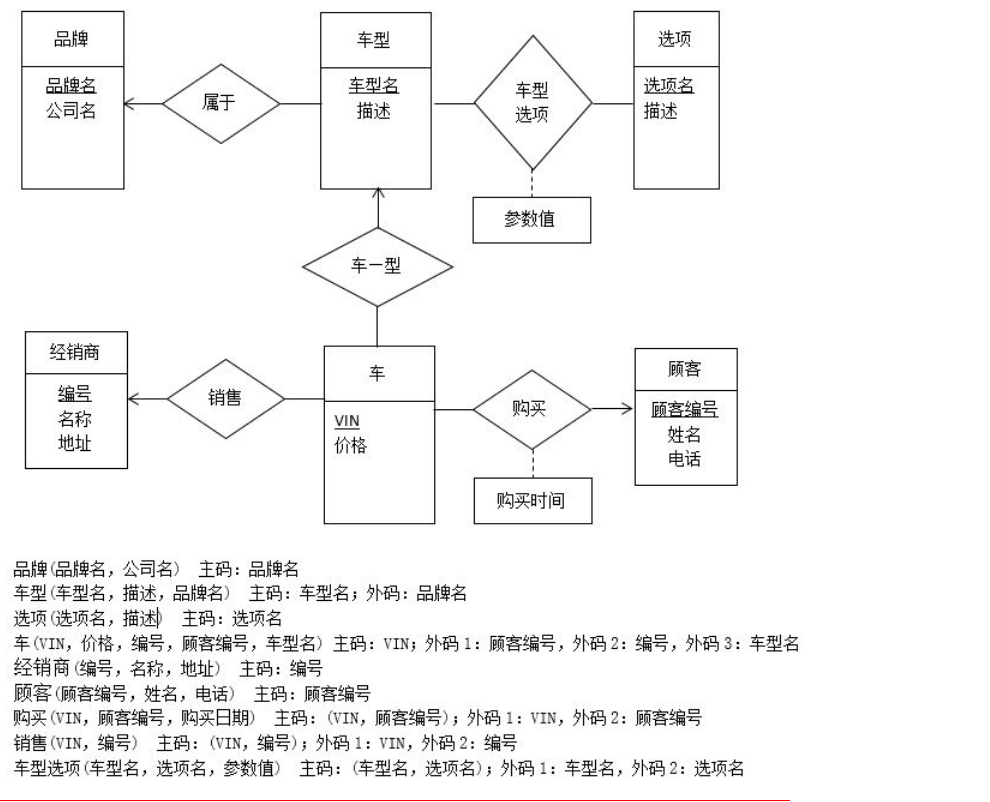
1. 为一个汽车公司设计一个数据库，用于协助它的经销商维护客户记录以及经销商库存，并协助销售人员订购车辆。每辆车由车辆编号(Vehicle Identifcation Number,VIN)唯一标识，每辆单独的车都是公司提供的特定品牌的特定车型(例如，XF是塔塔汽车捷豹品牌的车型)。每个车型都可以有不同的选项，但是一辆车可能只有一些(或没有)可用的选项。数据库需要保存关于车型、品牌、选项的信息，以及每个经销商、顾客和车的信息。

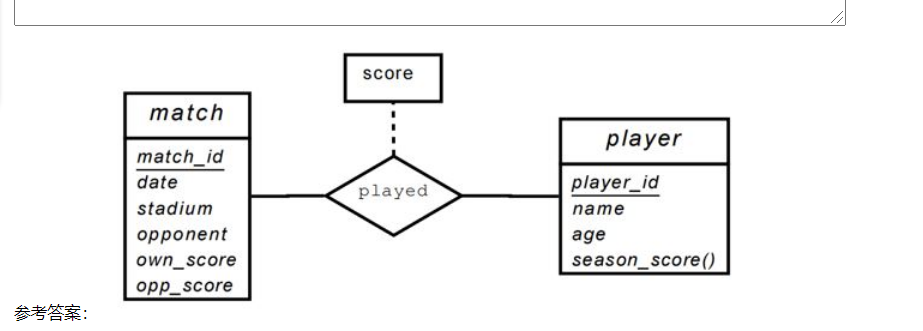
你的设计应该包括E-R图、关系模式的集合，以及包括主码约束和外码约束的一组约束。

请将上述内容表示在一张jpg图片中，然后提交该jpg图片文件。

1. 设计一个E-R图用于跟踪记录你最喜欢的球队的成绩。你应该保存打过的比赛，每场比赛的比分，每场比赛的上场队员以及每个队员在每场比赛中的统计数据。总的统计数据应该被建模成派生属性。

1、



2、

作业九

假设有关系模式R(A,B,C,D,E)，如下函数依赖集F成立:A→BC，CD→E，B→D，E→A

1、如果将模式R分解为：R1(A,B,C)与R2(A,D,E)。证明该分解是无损分解。

2、给出模式R的一个无损连接的BCNF分解。

3、给出模式R的一个无损连接并保持依赖的3NF分解。

4、证明模式R的如下分解不是无损分解:(A,B,C)(C,D,E)

1、如果R1∩R2→ R1或R1 n R2→ R2，则一个分解{R1, R2}是一个无损连接分解。

设R1=(A, B, C)， R2 =(A, D, E)， R1 ∩ R2 = A。因为A→BC => A→ABC，即R1 ∩ R2→R1，因此该分解是无损连接分解。

2、R的候选码是A、E、CD、BC。

B→D中B不是R的超码，因此，R不是BCNF。

根据BCNF分解算法，将R分解为R1(B,D)和R2(A,B,C,E)。

R1上的函数依赖F1={B→D},R1满足BCNF。

R2上的函数依赖F2={A→BCE,E→ABC,BC→AE},R2满足BCNF。

最后，R的BCNF分解为：R1(B,D)与R2(A,B,C,E)。

3、F的正则覆盖Fc={A→BC,CD→E,B→D,E→>A}

R的候选码是A、E、CD、BC。

根据3NF分解算法，将R分解：

A→BC => R1=(A, B, C)，

CD→E => R2=(C, D, E)，

B→D => R3=(B, D)，

E→A => R4=(E,A)，

R1...R4中已经包含了R的候选码，

因此，R的3NF分解为：R1(A,B,C),R2(C,D,E),R3(B,D),R4(A,E)

4、如果R1∩R2→ R1或R1 n R2→ R2，则一个分解{R1, R2}是一个无损连接分解。

设R1=(A,B,C)，R2 =(C,D,E)， R1 ∩ R2 = C。因C→BC与C→DE均不满足，即R1 ∩ R2→R1或R1 n R2→ R2均不满足，因此该分解不是无损连接分解。

作业十

考虑如下关系模式R(A,B.C.D,E,F)上的函数依赖集F:

{A→BCD，BC→DE，B→D，D→A}

1. 计算B的闭包。
2. (使用Armstrong公理)证明AF是超码。
3. 计算上述函数依赖集F的正则覆盖；给出你的推导的步骤并解释。
4. 基于正则覆盖，给出R的一个3NF分解。
5. 利用原始的函数依赖集，给出R的一个BCNF分解。
6. 你能否利用正则覆盖得到与上面的R相同的BCNF分解?

1、设 result=B；

由于B->D,故result=B∪D=BD；

由于D->A,故result=BD∪A=ABD；

由于A->BCD,故result=ABD∪BCD=ABCD；

由于BC->DE,故result=ABCD∪DE=ABCDE;

最终结果：B的闭包为ABCDE

2、由于A->BCD，故A->ABCD(增补率，两边增补A)

由于BC->DE，故ABCD->ABCDE(增补率，两边增补ABCD)

因此A->ABCDE(传递率)

因此AF->ABCDEF(增补率，两边增补F)

因为AF可以推出所有属性，即AF的闭包包含所有属性，故AF是超码

3、首先观察可知无左侧元素相同的可合并的函数依赖，因此查看无关属性。

由于B->D，因此在A->BCD和BC->DE中D均为无关属性(A->B,B->D,故A->D，因此A->BC结合A->D可以推出A->BCD；B->D故BC->D，因此BC->E结合BC->D可以说明BC->DE；因此D为无关属性)，因此式子可以简化为：

A->BC BC->E B->D D->A

由于B->D,D->A,A->BC,因此B->C,因此BC->E中C为无关属性，因此可简化为：

A->BC B->E B->D D->A

左侧为B的有两项，可合并：

A->BC B->DE D->A

发现当前无无关属性，因此此时即为F的正则覆盖

4、由于正则覆盖中没有多余的函数依赖，因此属性集是正则覆盖中其他函数依赖组成的子集，因此这里的三个函数依赖都有自己的关系：R1(A,B,C) R2(B,D,E) R3(D,A) 然后我们发现属性F并不在其中，因为AF是超码且上述的关系中并没有原关系的超码，因此需要添加 R4(A,F)

因此R的一个3NF分解为：R1(A,B,C) R2(B,D,E) R3(D,A) R4(A,F)

5、start: R={A,B,C,D,E,F}

由于A->BCD但A不是超码，故R不满足BCNF,进行分解：

step1: R= R1(A, B, C, D),R2(A, E, F)

由于A->E是F+中的函数依赖，因此R2不满足BCNF,进行分解

step2: R=R1(A, B, C, D),R2(A, F),R3(A, E)

检验可得现在均符合BCNF

6、若想通过正则覆盖得到与上面R相同的BCNF分解，直接进行BCNF分解是不行的，需要通过从正则覆盖推断出原始函数依赖并将其进行BCNF分解才能够得到相同的BCNF分解

作业十一

一、丢失更新(lost update)异常是指如果事务Tj读取了一个数据项，然后另一个事务Ti写该数据项(可能基于先前的读取)，然后Tj写该数据项。于是Ti做的更新丢失了，因为Tj的更新覆盖了Ti写入的值。

1、给出一个表示丢失更新异常的调度实例。

2、给出一个表示丢失更新异常的调度实例，表明在已提交读隔离性级别下该异常也可能存在。

3、解释为什么在可重复读隔离性级别下丢失更新异常不可能发生。

1、R1(A)R2(A)W2(A)W1(A)

或者以表格的形式表示：

T1 T2

-----------------

Read(A)

Read(A)

Write(A)

Write(A)

在上面的调度中，事务T2写入的值由于事务T1的写入而丢失。

2、T1 T2

-----------------

Lock-S(A)

Read(A)

unlock(A)

Lock-X(A)

Read(A)

Write(A)

unlock(A)

commit

Lock-X(A)

Write(A)

unlock(A)

commit

上述调度中的封锁确保了已提交读隔离性的级别。但由事务T2写入的值因T1的写入而丢失。

1. 在可重复读取隔离级别，不会出现丢失更新异常。在可重复读取隔离级别中，事务T1读数据项X，在X上持有一个共享锁，直到事务结束。这使得较新的事务T2要等到T1结束才能够写X的值。这就强制成串行顺序T1、T2，因此T2所写的值不会丢失。

二、考虑下面两个事务:

T1：read(A)

read(B)

if A=O then B:=B+1

write(B)

T2：read(B)

read(A)

if B=O then A:=A+1

write(A)

1. 给事务T1与T2增加加锁、解锁指令，使它们遵从两阶段封锁协议。
2. 这两个事务会引起死锁吗?
3. 设一致性需求为A=0 ∨ B=0，初值是A=B=0。

说明包括这两个事务的每一个串行执行都保持数据库的一致性。

4、设一致性需求为A=0 ∨ B=0，初值是A=B=0。

给出T1和T2的一次并发执行，执行产生不可串行化调度。

5、设一致性需求为A=0 ∨ B=0，初值是A=B=0。

存在产生可串行化调度的T1和T2的并发执行吗?

1、T1：lock-S(A)

read(A)

lock-X(B)

read(B)

if A=O then B:=B+1

write(B)

unlock(A)

unlock(B)

T2：lock-S(B)

read(B)

lock-X(A)

read(A)

if B=O then A:=A+1

write(A)

unlock(B)

unlock(A)

2、执行这些事务可能导致死锁。例如，考虑以下调度：

T1 T2

-----------------

Lock-S(A)

Read(A)

Lock-S(B)

Read(B)

Lock-X(B)

Lock-X(A)

此时调度出现了死锁。

3、T1与T2有2种可能的串行调度：T1、T2，或者T2、T1。

对于调度T1、T2，调度结束时，A=0，B=1

对于调度T2、T1，调度结束时，A=1，B=0

可以看出串行执行会生成两种不同的结果，但都保持了数据库的一致性。

4、T1和T2的如下并发执行，将产生不可串行化调度：

T1 T2

----------------------------

Read(A)

Read(B)

Read(A)

Read(B)

if A=O then B:=B+1

if B=O then A:=A+1

Write(A)

Write(B)

5、没有导致可串行化调度的并行执行。

从第a部分我们知道，一个可串行调度的结果是a = 0∨

B = 0。假设我们从T13开始读(A)。那么当日程安排

结束，无论何时运行T2 B = 1。现在假设

我们在T13完成之前开始执行T14。然后T2读(B)

B的值为0。当T2结束时，A= 1。因此B = 1∧A

= 1→？(A = 0∨B = 0)，类似于从T14开始读(B)。

三、什么是可恢复调度?

假设在一个调度中，Tj读取了Ti写入的数据，Ti在提交前发生故障，我们必须中止Tj以保证事务地原子性。若Tj在Ti出现故障后是可中止的，那么我们就称该调度是可恢复调度。

可恢复调度应满足:对于每个事务Ti和Tj，如果Tj读取了由Ti所写的数据项，则Ti先于Tj提交。