

## 一、简答题

1. 简述物理数据独立性，数据库如何实现物理数据独立性。
2. 简述空值（NULL）的含义，空值在参与运算时有哪些特点。
3. 对于以下关系 R，指出是否存在多值依赖  $C \twoheadrightarrow H$ ，并说明原因。

C	T	H	S	G
c <sub>1</sub>	t <sub>1</sub>	h <sub>1</sub>	s <sub>1</sub>	g <sub>1</sub>
c <sub>1</sub>	t <sub>1</sub>	h <sub>2</sub>	s <sub>1</sub>	g <sub>1</sub>
c <sub>1</sub>	t <sub>1</sub>	h <sub>1</sub>	s <sub>2</sub>	g <sub>2</sub>
c <sub>1</sub>	t <sub>1</sub>	h <sub>2</sub>	s <sub>1</sub>	g <sub>2</sub>

4. 简述带有锁转换的两阶段封锁协议，并证明两阶段封锁协议能够保证事务集合的可串行化。
5. 简述基于日志和检查点的数据库恢复的基本步骤。
6. 考虑以下两个事务 T<sub>1</sub> 和 T<sub>2</sub>，给出 T<sub>1</sub> 和 T<sub>2</sub> 的一次并发执行，执行产生不可串行化调度。  
(请将调度写在下面表格中，只关注 read 和 write 操作即可)

T<sub>1</sub> : read(X);  
       read(Y);  
       Y = Y + 10;  
       write(Y);

T<sub>2</sub> : read(Y);  
       read(X);  
       X = X + 10;  
       write(X);

T <sub>1</sub>	T <sub>2</sub>

## 二、综合题

### (一) 数据库设计 (每小题 5 分，共 10 分)

某医院的管理需求如下：

医院有若干科室，每个科室有若干医生、护士和病房；每个医生负责多名病人的治疗工作，每个护士负责多名病人的护理工作；每个病房有若干病人。

对于科室，关注的信息包括科室编号和科室名称；对于病房，关注的信息包括病房编号、病房名称；对于医生关注的信息包括医生编号、医生姓名；对于护士关注的信息包括护士编号、护士姓名；对于病人关注的信息包括病人编号，病人姓名和联系人，联系人关注联系人姓名

和联系人的电话号码，每个病人要求提供两个以上的联系人。

要求：

1. 用 E-R 模型表示上述需求。

2. 将 E-R 模型转换成相应的关系模式。

$sno, cno \rightarrow tno$

$tno \rightarrow tname, cno$

$F_c = sno \rightarrow sname$

$cno \rightarrow cname$

## (二) 规范化

对于下列关系模式 R 和函数依赖集合 F：

$R(sno, sname, cno, cname, tno, tname)$ ,  $F = \{ sno \rightarrow sname, cno \rightarrow cname, tno \rightarrow (tname, cno), (sno, cno) \rightarrow (tno, tname) \}$ , 完成下列任务（要求写出主要计算步骤）。

1. 将 R 分解为属于 3NF 并且保持函数依赖和无损连接的关系模式。

2. 将 R 分解为属于 BCNF 并且无损连接的关系模式。

$(sno, sname)$

$(cno, cname)$

$(sno, tno)$

$(tno, tname, cno)$

## (三) 证明题（每小题 4 分，共 8 分）

1. 给定关系模式 R，其上的函数依赖集 F，及其分解  $\{R_1, R_2\}$ ，证明如果  $R_1 \cap R_2 \rightarrow R_1$  属于  $F^+$ ，则  $R_1$  和  $R_2$  是 R 的无损分解。

2. 针对以下关于函数依赖规则：已知  $\alpha \rightarrow \beta$ ,  $\gamma \rightarrow \beta$ ，则  $\alpha \rightarrow \gamma$ 。请给出一个关系 r，r 满足  $\alpha \rightarrow \beta$  和  $\gamma \rightarrow \beta$ ，但是不满足  $\alpha \rightarrow \gamma$ ，从而证明上述规则不成立。

## 三、运算题

有关系模式  $S(sno, sname, dno, sex, dorm, monitor)$ ,  $C(cno, ctitle, credit)$ ,  $SC(sno, cno, score)$ ,  $D(dno, dname)$ 。关系 S、C、SC 和 D 分别表示学生信息、课程信息、学生选课情况和学院信息。其属性分别表示如下：sno—学生编号，sname—学生姓名，dno—学院编号，sex—性别，dorm—宿舍，monitor—班长，cno—课程编号，ctitle—课程名称，credit—课程学分，score—成绩，dname—学院名称。关系模式的实例如下：

S

sno	sname	dno	sex	dorm	monitor
S <sub>1</sub>	赵晨	d <sub>1</sub>	M	101	S <sub>1</sub>
S <sub>2</sub>	许亮	d <sub>2</sub>	M	101	
S <sub>3</sub>	肖莹	d <sub>1</sub>	F	201	S <sub>1</sub>
S <sub>4</sub>	陈静	d <sub>3</sub>	F	201	
S <sub>5</sub>	王勤勤	d <sub>1</sub>	M	101	S <sub>1</sub>

C

cno	ctitle	credit
C <sub>1</sub>	计算机组成原理	4
C <sub>2</sub>	离散数学	3
C <sub>3</sub>	数据库系统	4
C <sub>4</sub>	计算机网络	4

SC

sno	cno	score
S <sub>1</sub>	C <sub>1</sub>	80
S <sub>1</sub>	C <sub>2</sub>	75
S <sub>1</sub>	C <sub>3</sub>	45
S <sub>2</sub>	C <sub>2</sub>	70
S <sub>3</sub>	C <sub>1</sub>	90
S <sub>3</sub>	C <sub>2</sub>	60

D

dno	dname
d <sub>1</sub>	计算机学院
d <sub>2</sub>	物理学院
d <sub>3</sub>	化学学院
d <sub>4</sub>	生命科学学院

(一) 针对上述实例, 分别给出关系代数表达式 (1 至 4 题)、表达式计算结果 (5 至 6 题)、元组关系演算表达式 (7 至 8 题) (每小题 3 分, 共 24 分)

1. 列出所有男生的姓名。
2. 列出每个学生的学号以及所选课程的门数。
3. 列出没有学生的学院的名称。
4. 列出学生 S<sub>1</sub> 的成绩大于 60 分的课程名称。

$$5. \pi_{a.sno}(\sigma_{a.score < b.score \wedge b.sno = 's2' \wedge a.cno = 'c2' \wedge b.cno = 'c2'}(\rho_a(SC) \times \rho_b(SC)))$$

$$6. \pi_{sname}(\pi_{sno, cno}(SC) \div \pi_{cno}(\sigma_{sno = 's_3'}(SC)) \bowtie S)$$

7. 列出课程号为  $c_1$  的课程名称。

8. 列出没有学生的学院的编号。

(二) 对于上述关系模式，用 SQL 完成下列操作（只写出 SQL 即可，不需要执行结果，每小题 3 分，共 24 分）

1. 查询物理学院所有学生姓名，按照学生姓名降序排序。

2. 查询没有选修“ $c_2$ ”课程的学生学号和姓名。

3. 查询与自己的班长住在同一个宿舍的学生学号和姓名。

4. 查询获得“离散数学”课程最高分的学生学号和姓名。

5. 查询学生人数最多的学院编号和学院名称。

6. 查询被所有学生都选修的课程编号和课程名称

7. 如果学生某门课的考试成绩低于该门课的平均成绩，则将该学生该门课的考试成绩提高 5%。

8. 查询平均成绩在 85 分以上的课程编号、课程名称和平均成绩，按照课程编号升序排序。

## 一、简答题

1.

物理数据独立性是指数据库物理结构变化时不必修改现有的应用程序。

物理数据独立性是通过 DBMS 的模式/内模式映像来实现的。当数据的物理存储结构改变时, 由 DBMS 对模式/内模式映像进行修改, 可以使模式保持不变, 从而使应用程序也不必改变, 从而保证了物理数据独立性。

2 .

空值 NULL 表示“不知道”或者“不存在”的含义。

空值参与算术运算结果均为 NULL ;空值参与比较运算, 结果为 UNKNOWN。

3 .

答：不存在。能够举出违背多值依赖定义的反例，例如，关系中存在  $(c_1, t_1, h_1, s_2, g_2)$ ，但是不存在  $(c_1, t_1, h_2, s_2, g_2)$ 。

4 .

带有锁转换的两阶段锁协议要求每个事务分成两个阶段提出加锁/锁升级和解锁/锁降级申请：

增长阶段：事务可以获得封锁，或者将共享锁升级为排他锁，不能释放锁

缩减阶段：事务可以释放锁，或者将排他锁降级为共享锁，不能获得新锁

采用反证法，假定两阶段封锁协议不能保证可串行化，则存在一个遵循两阶段封锁协议的事务集合  $T_0, T_1, \dots, T_{n-1}$  产生了一个非串行的调度，那么在其优先图中会有一个环。不失一般性，假定优先图中的环是  $T_0 T_1 T_2 \dots T_{n-1} T_0$ ，让  $a_i$  对应事务  $T_i$  获得到最后一个锁的时间（即事务  $T_i$  的封锁点）。因而对于所有的事务  $T_i < T_j$ ，都有  $a_i < a_j$ ，对于这个环，我们就有  $a_0 < a_1 < a_2 \dots < a_{n-1} < a_0$ ，存在矛盾，因此假设不成立。

所以，两阶段封锁协议能够保证事务集合的可串行化。

5 .

在基于日志和检查点的数据库恢复过程中，只需要考虑在检查点开始时的活跃事务，以及在检查点之后开始的事务。

- (1) 从日志末端向后扫描，查找到最近的检查点记录<checkpoint L>；
  - (2) 对于在 L 中的事务，以及在检查点之后开始的事务，进行重做或撤销；
- 其中，对于有 start 和 commit/abort 标志的事务，进行重做，对于有 start 但是没有 commit/abort 标志的，进行撤销。

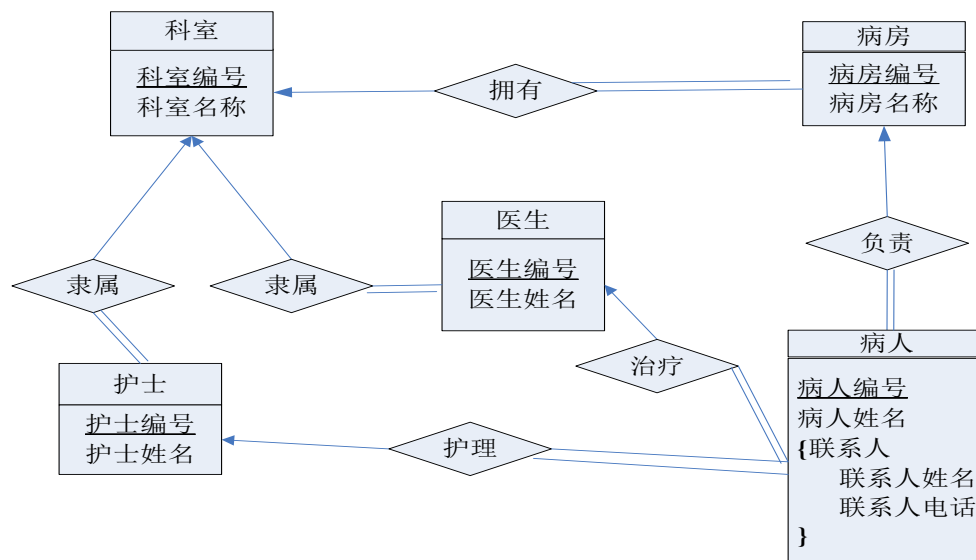
在检查点之前提交或者中止的事务，对数据库的修改已经输出到了稳定存储器上。

6 .

T <sub>1</sub>	T <sub>2</sub>
read(X)	
read(Y)	
	read(Y)
	read(X)
write(Y)	
	write(X);

## 二、综合题

### (一)



关系模式：

科室 (科室编号, 科室名称)

病房 (病房编号, 病房名称)

医生 (医生编号, 医生名称, 科室编号)

护士 (护士编号, 护士名称, 科室编号)

病人 (病人编号, 病人姓名, 病房编号, 医生编号, 护士编号)

病人联系人 (病人编号, 联系人姓名, 联系人电话)

## (二) 规范化

1 .

$F_c = \{sno \rightarrow sname, cno \rightarrow cname, tno \rightarrow (tname, cno), (sno, cno) \rightarrow tno\}$

3NF :

$\{(sno, sname), (cno, cname), (tno, tname, cno), (sno, cno, tno)\}$

2 .

$\{(sno, sname), (cno, cname), (tno, tname, cno), (sno, tno)\}$

或者  $\{(tno, tname, cno), (sno, sname), (sno, cname, tno)\}$

或者  $\{(sno, tno), (tno, cno), (sno, sname), (cno, cname), (tno, tname)\}$

## (三) 证明题

1. 假设  $\forall r \in R$ , 分解得到  $r_1, r_2$ , 只需证明  $r = r_1 \bowtie r_2$ 。

容易知道,  $\forall t \in r, t \in r_1 \bowtie r_2$ , 所以  $r \subseteq r_1 \bowtie r_2$ 。

令  $\alpha = R_1 \cap R_2$ ,  $\beta = R_1 - \alpha$ ,  $\gamma = R_2 - \alpha$

$\forall t \in r_1 \bowtie r_2$ , 得  $t[\alpha\beta] \in r_1, t[\alpha\gamma] \in r_2$  ;

设  $t[\alpha\beta]$  在  $r$  里面对应元组为  $t_1$ , 则  $t_1[\alpha] = t[\alpha], t_1[\beta] = t[\beta]$  ;

设  $t[\alpha\gamma]$  在  $r$  里面对应元组为  $t_2$ , 则  $t_2[\alpha] = t[\alpha], t_2[\gamma] = t[\gamma]$  ;

由  $\alpha \rightarrow_{R_1} \beta$  得  $\alpha \rightarrow \beta$ , 又因为  $t_1[\alpha] = t_2[\alpha] = t[\alpha]$ , 得  $t_1[\beta] = t_2[\beta]$ , 所以  $t_2[\beta] = t[\beta]$ , 所以  $t_2 = t$ , 所以  $t \in r$  ;

得  $r_1 \bowtie r_2 \subseteq r$ 。

因此  $r = r_1 \bowtie r_2$  得证。

2.

$\alpha$	$\gamma$	$\beta$
1	1	3
1	2	3

## 三、运算题

(一)

1.

$\Pi_{sname} (\sigma_{sex='M'}(S))$

2.

$snoGcount(cno)(SC)$

3.

$$\Pi_{\text{dname}}(D) - \Pi_{\text{dname}}(S \bowtie D)$$

4.

$$\Pi_{\text{ctitle}}((\sigma_{\text{sno}='s1' \wedge \text{score} > 60}(SC)) \bowtie C)$$

5.

a.sno
S <sub>3</sub>

6.

Sname
赵晨
肖莹

7.

$$\{ t \mid \exists r \in C (t[\text{ctitle}] = r[\text{ctitle}] \wedge r[\text{cno}] = 'c_1' ) \}$$

8.

$$\{ t \mid \exists r \in D (t[\text{dno}] = r[\text{dno}]) \wedge \neg \exists v \in S (v[\text{dno}] = t[\text{dno}]) \}$$

(二)

1.

```
Select sname
From s,d
Where d.dno = s.dno and dname = '物理学院'
Order by sname desc;
```

2.

```
Select sno, sname
From s
Where sno not in (select sno
                  from sc
```



where cno = 'c2');

3.

Select a.sno, a.sname

From s a, s b

Where a.monitor=b.sno and a.dorm=b.dorm;

评分细则：表连接正确得 2 分，选择条件正确得 1 分。

4.

Select s.sno, s.sname

From s, sc, c

Where s.sno= sc.sno and sc.cno=c.cno and c.title='离散数学'and

sc.score = (select max(score) from sc, c

where sc.cno=c.cno and c.title='离散数学')

5.

Select dno, dname

From d

Where dno in (select dno from s

group by dno

having count(\*) >= all (select count(\*) from s group by  
dno))

评分细则：group by 和 having 各 1 分，>=all 得 1 分。

6.

Select cno, ctitle

From c

Where not exists ( select \*

from s

where not exists ( select \*

from sc

where s.sno=sc.sno and

sc.cno=c.cno))

或

Select cno, ctitle

From c

```
Where not exists ( (select sno
                    From s)
                Except
                (select sno
                From sc
                Where sc.cno=c.cno))
```

或

```
Select cno, ctitle
From c
Where cno in (select cno
              from sc
              group by cno
              having count(*) = (select count(*) from s))
```

7.

```
Update sc
Set score=score*1.05
Where score < (select avg(score) from sc sc1
               where sc1.cno=sc.cno)
```

8.

```
select c.cno, c.title, avg(score)
from sc, c
where sc.cno=c.cno
group by c.cno
having avg(score)>=85
order by c.cno
```