

1. 传统数据管理经历了人工管理、文件系统、数据库系统等阶段,在这一过程中,数据独立性越来越高,为什么数据管理系统的数据独立性越高越好? (10 分)

(阐明数据独立性的意义即可)

数据独立性是指建立在数据的逻辑结构和物理结构分离的基础上,用户以简单的逻辑结构操作数据而无需考虑数据的物理结构,转换工作由数据库管理系统实现。数据独立性分为数据的物理独立和数据的逻辑独立。

(1) 数据的物理独立 意义:数据的存取与程序分离。数据存储结构与存取方法的改变不一定要求修改程序。使初步数据共享成为可能,只要知道数据存取结构,不同程序可共用同一数据文件。

(2) 数据的逻辑独立 意义:数据的使用与数据的逻辑结构相分离。

2. 关系 R, S 如下图所示。请给出下列各表达式的运算结果(每小题 5 分,共 10 分)

R				S	
A	B	C	D	C	D
a ₁	b ₁	c ₁	d ₁	c ₁	d ₁
a ₁	b ₁	c ₂	d ₂	c ₂	d ₂
a ₁	b ₁	c ₁	d ₃		
a ₂	b ₂	c ₁	d ₁		
a ₂	b ₂	c ₂	d ₂		
a ₃	b ₃	c ₁	d ₁		

(1) 计算关系代数运 $\Pi_{A,C,D}(A \neq a_1(R)) \div S$ 算:

(2) 用域关系演算表达除法操作。

(1)

A
a ₂

(2) 设有关系 R (X, Y) 和 S (Y), X 和 Y 为属性组。则域关系演算表达式如下:

$$\{X | \exists Y \forall Y (R(X, Y) \text{ AND } S(Y))\}$$

3. 某医院设计了一个数据库以方便对住院病人、医生和手术信息的管理。主要关系模式如下:

医生 (医生编号, 姓名, 性别, 年龄, 职称, 所属科室)

病人 (病人编号, 姓名, 性别, 年龄, 地址)

手术安排 (病人编号, 手术室编号, 手术日期, 手术名称, 手术类别)

手术医生安排 (医生编号, 病人编号, 手术室编号, 手术日期, 医生责任)

注:病人一次住院期间由一名医生对其进行诊断,并填写一份诊断书,每次住院可能进行一次或多次手术;手术室编号可以唯一确定每个手术室,每个手术室每天只能安排一台手术。日期属性为 Date 型,描述某年某月某日,GetYear()和 GetMon()分别为获取年份、月份的函数。

试写出表达下列查询要求的 SQL 语句(38 分,必须用单条 SQL 语句表达):

(1) 查找参与了 2019 年 10 月 10 日三号手术室手术的医生编号和姓名。(6 分)

SELECT 医生编号, 姓名

```
FROM 医生, 手术医生安排
WHERE 手术日期=2019 年 10 月 10 日 and 手术室编号=3 and 医生.医生编号=手术
医生安排.医生编号;
```

(2) 查询 2019 年 10 月份, 仅参加过一台手术的医生编号和姓名。 (8 分)

```
SELECT 医生编号, 姓名
FROM 医生 WHERE 医生编号 IN (
    SELECT 医生编号 FROM 手术医生安排
    WHERE 手术日期.GetYear()=2019 and 手术日期.GetMon()=10
    GROUP BY 医生编号 HAVING COUNT(*)=1 );
```

(3) 查询 2 次以上进行同类别手术的病人编号和姓名。 (8 分)

```
SELECT 病人编号, 姓名
FROM 病人, 手术安排
WHERE 病人.病人编号=手术安排.病人编号
GROUP BY 病人编号, 手术类别
HAVING COUNT(*)>2;
```

(4) 查询以主刀医生身份为同一名病人进行 2 次以上同类别手术的医生姓名。 (8 分)

```
SELECT 姓名
FROM 医生, 手术安排, 手术医生安排
WHERE 医生.医生编号=手术安排.医生编号 and 医生责任="主刀医生"
and 手术医生安排.病人编号=手术安排.病人编号
and 手术医生安排.手术日期=手术安排.手术日期
and 手术医生安排.手术室编号=手术安排.手术室编号
GROUP BY 病人编号, 手术类别
HAVING COUNT(*)>2;
```

(5) 按医生职称查询 2019 年各职称级别完成手术次数最多的医生姓名及其完成的手术次数。 (8 分)

```
SELECT 职称, 姓名, 手术次数
FROM 医生, (
    SELECT 医生编号, COUNT (*) AS 手术次数
    FROM 医生, 手术医生安排 AS A
    WHERE 手术日期.GetYear () =2019 and 医生.医生编号=手术医生安排.医生编号
    and NOT EXISTS
    (SELECT * FROM 医生, 手术医生安排 AS B
    WHERE 医生.医生编号=手术医生安排.医生编号 and 手术日期.GetYear ()
    =2019 and COUNT(B.医生编号) > COUNT(A.医生编号) and A.职称=B.职称
    GROUP BY 医生编号)
    GROUP BY 医生编号
} AS S
WHERE 医生.医生编号=S.医生编号;
```

4. 查询优化是关系型数据库的重要功能，谈谈你对下述论述的看法：

(1) “B-树为代表的树形索引成为当前数据库系统主流索引具有必然性”？(5 分)

(2) 组织 SQL 查询语句时，一般不建议在 Where 条件中使用“or”谓词？(5 分)

(1)

索引文件是一种适应面比较广的文件结构，因此在数据库系统中得到了广泛的应用。对于经常变动的文件，静态索引的性能会随时间变化而变坏，所以目前在数据库系统中应用更多的是动态索引。而 B-数就是为了磁盘或其它存储设备而设计的一种平衡多分树，能很好地进行动态索引。B-树提供了三种存取路径：

1.通过索引集进行树形搜索；

2.通过顺序集进行顺序搜索；

3.先通过索引找到入口，再沿顺序集顺序搜索。

B-树不仅提供了灵活的存取路径，而且能够自动保持平衡，不须定期重组，因此 B-树为代表的树索引系列在数据库系统中应用甚广，成为数据库主流具有必然性。

(2)

使用 Or 谓词的析取选择条件，并没有好的优化条件，只能按其中各个条件分别选出一个元组集，再求这些元组集的并。并是开销大的操作，而且在 OR 连接的多个条件中，只要有一个条件没有合适的存取路径，就不得不采用顺序扫描来处理这种查询，导致效率大大降低。

5. 加锁是现代数据库系统中事务并发管理的重要技术，试回答下述问题：

(1) 已有的 (S,X)、(S,U,X) 锁能解决事务并发中的死锁问题么？为什么？(5 分)

(2) (S,U,X) 锁的相容矩阵如下图，为什么已经加了 U 锁，不允许其它事务申请加 U 锁？

如果允许会出现什么情况？(5 分)

其它事务已拥有的锁				
锁 请 求		S	U	X
	S	Y	Y	N
	U	Y	N	N
	X	N	N	N

(1)

不能解决并发事务中的死锁问题。当一个事物 A 占用数据对象 a 的 X 锁，事务 B 占用数据对象 b 的 X 锁，事务 A 和事务 B 又分别申请数据对象 b 和数据对象 a 的锁，在 (S,X) 和 (S,U,X) 锁中，均无法获准，需要等待对方事务释放锁，而进入等待状态则无法释放自己所占用的锁，从而陷入循环等待，即死锁。

(2)

U 锁表示事务对数据对象进行更新的操作，在最后写入阶段事务再将其升级为 X 锁，导致最终写操作时若在 U 锁阶段允许其他事务申请 U 锁，则在事务 A 想将 U 锁升级为 X 锁进行数据写操作时，由于存在其他事物对数据对象的 U 锁，而无法升级为 X 锁，从而导致死锁。

6. 假设运行记录与数据库的存储磁盘有独立失效模式，介质失效恢复时，对运行记录中上一检查点以前的已提交事务应该 redo 否？为什么？(10 分)

介质失效指磁盘发生故障，数据库受损，如磁盘、刺头破损。

介质失效后应该在新介质中加载最近后备副本，并用档案存储器内运行记录中的后像，redo 后备副本以后提交的所有更新事物。如果检查点比后备副本要新，则对后备副本以后，检查点以前的事物也应该 redo。

7. 结合第 3 题中业务场景，回答以下问题 (12 分)：

(1) 分析各关系表的主键和可能存在的所有候选键、外键。(6 分)

(2) “手术安排”和“手术医生安排”关系表各满足第几范式？现有模式设计是否有问题？数据库模式设计时，达到的范式是越高越好么，为什么？(6 分)

(1) 医生：主键：{医生编号} 外键：无

病人：主键：{病人编号} 外键：无

手术安排：主键：{手术室编号，手术日期} 外键：{病人编号}

手术医生安排：主键：{手术室编号，手术日期，医生编号}

外键：{医生编号}，{病人编号}

(2) 手术安排：第一范式（手术类别依赖手术名称，不满足二范式）

手术医生安排：第二范式（病人编号依赖{手术室编号，手术日期}，不满足三范式）

数据库设计中，数据模式遵循的范式不是越高越好，应取决于应用。数据库的范式主要目的是防止数据冗余，更新异常、插入异常和删除异常，而范式高会存在处理速度缓慢和处理逻辑复杂的问题，从而降低数据库性能，因此需要权衡考虑。