

db往年题

2021年

简答题

6道，一个3分，共18分

1. 请给出这五个调度的包含关系

1. 正确的调度
2. 串行调度
3. 可串行化调度
4. 满足两阶段封锁协议的调度
5. 满足严格两阶段封锁协议的调度

2. 请解释数据和程序在逻辑上和物理上的独立性

3. 题目给出一个sql语句，要求画出优化后的语法树具体的where条件忘记了，类似这样的sql语句

```
select sname
from s,sc,c
where s.sno = sc.sno and sc.cno = c.cno and s.age = 18;
```

4. 对于一个函数依赖集 $F=\{a \rightarrow b, y \rightarrow g\}$ ，假设 y 包含于 b ，证明 $a \rightarrow g$

5. 有一个学生表 $S(sno, sname)$ ，在 $sname$ 上建立辅助索引，请问在执行如下sql语句时，是否用到了该索引，请解释原因

```
select sname
from s
where sname like '%小%'
```

6. 利用银行转账的例子解释事务的原子性

E-R图 1题，共8分

1. 要求画出E-R图，并给出E-R图转换得到的关系模式

题目的背景是一家软件公司，里面有以下几个实体部门，员工，实习生，项目，有相应的联系。

3NF, BCNF分解

共9分，第一题5分，第二题4分

1. 3NF

题目给定属性集和函数依赖集 F

- (1) 求全部候选码
- (2) 判断这个关系是否属于3NF，说明原因
- (3) 求函数依赖集 F 的正则覆盖 F_c

2. BCNF

判断该关系模式是否属于BCNF，如果属于，说明原因，如果不属于，则给出BCNF分解的结果

SQL，关系代数，元组关系演算

7题，共35分

题目给定了四个关系模式

```
supplier(sno,sname,city,忘了)
com(cno,cname,color,weight)
project(pno,pname,city,忘了)
spc(sno,cno,pno,qurt)
```

supplier表示供应商，com表示零件，project表示项目，spc表示供应商，零件和项目之间的关联关系

供应商向项目提供零件，qurt表示零件的数目

(1) 分别用SQL，关系代数表达式，元组关系演算写出以下查询的结果：

查询颜色为“红色”的零件编号

(2) 分别用SQL，关系代数表达式，元组关系演算写出以下查询的结果：

查询提供了红色零件的供应商的名字

(3) 分别用SQL，关系代数表达式，元组关系演算写出以下查询的结果：

查询未使用“红色”零件的项目的编号

(4) 分别用SQL，关系代数表达式，元组关系演算写出以下查询的结果：

查询使用了编号为s1的供应商提供的所有零件的项目

(5) 使用SQL语句，写出以下查询的结果：

查询提供零件数最多的供应商的编号

(6) 使用SQL语句，写出以下查询的结果：

查询零件重量比编号为s1供应商提供的零件都重的零件名称

(7) 使用SQL语句更新表

将项目名为p1的项目所拥有的零件数量增加10%

以上题目的意思大体上是正确的，细节上可能有略微出入

后记

本学期的数据库考试结束了，总的来说，这次数据库考试重点明确，SQL，关系代数和元组关系演算占比为50%。而其它重点比如E-R图，3NF，BCNF分解，正则覆盖等也全部涉及。在简答题方面，没有考判断是否可串行化的题目，有点意外。还考察了索引，事务，函数依赖，语法树，数据独立性。

版权声明：本文为CSDN博主「叶卡捷琳堡」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/weixin_46841376/article/details/118307456

2019年

简答题（每题4分）

1. 三级模式和二级映像是什么东西，有什么优势？
2. $\Pi_{\text{Sname, cname, score}}(\sigma_{\text{sex}='F'}(\text{SC}))$ 的语法树、优化语法树
3. 关系模式和关系实例的区别是什么
4. 事务的概念及其特点
5. 是否存在 $A \rightarrow \rightarrow C$ 的多值传递依赖，为什么

A	B	C	D	E
a1	b1	c1	d1	e1
a1	b1	c2	d1	e1
a1	b1	c1	d2	e2
a1	b1	c2	d1	e2

6. 该并行操作是否等价于串行T4T5？为什么？

T4	T5
read(A)	
	read(B)
	B=B-50
	write(B)
read(B)	
	read(A)
Display(A+B)	
	A=A+50
	write(A)
	Display(A+B)

设计题

1. 某企业的人力资源管理需求如下：

企业有员工，员工需要关注的信息包括：员工号、员工姓名、性别、所在部门、出生日期、年龄、民族、亲属信息；企业有若干部门，部门需要关注的信息包括：部门编号、部门名称、部门地址；企业有若干岗位，岗位需要关注的信息包括：岗位名称、岗位职责；每个员工隶属于一个部门，一个部门有多个员工，一个部门有一个负责人；一个员工可以承担多个岗位的工作职责，一个岗位可以有多个员工承担；一个部门可以拥有多个岗位，一个岗位也可以在多个部门中存在。员工的亲属信息关注身份证号码、与员工的关系和姓名。

要求：

- (1) 用E-R图表示上述内容。
- (2) 将E-R图转换成相应的关系模式。

类似这个，多了一个要求，员工在食堂吃饭，有个最喜欢去的食堂

2. $R=\{ABCDE\}$, $F=\{A \rightarrow B, BC \rightarrow D, DE \rightarrow A\}$

- (1) 是否是3NF？如果是给出理由，如果不是分解成3NF并保持函数依赖、无损连接
- (2) 判断是否是BCNF,如果是，则说明理由，如果不是则分解

SQL

1. 有关系模式S(sno,saname,dno,sex, dorm, monitor), C(cno,cname,credit), SC(sno,cno,sore), D(dno,dname)。关系S、C、SC和D分别表示学生信息、课程信息、学生选课情况和院系信息。其属性分别表示如下：sno—学生编号，sname—学生姓名，dno—院系编号，sex—性别，dorm—宿舍，monitor—班长，cno—课程编号，cname—课程名称，credit—课程学分，score—成绩，dname—院系名称。
关系模式的实例如下：

S

sno	sname	dno	sex	dorm	monitor
s1	王红	d1	M	101	s5
s2	张明	d2	M	101	
s3	刘华	d1	F	201	s5
s4	李静	d3	F	201	
s5	赵军	d1	M	101	s5

C

cno	cname	credit
c1	操作系统	4
c2	离散数学	3
c3	数据库系统	4
c4	数据结构	4

SC

sno	cno	score
s1	c1	80
s1	c2	75
s1	c3	45
s2	c2	70
s3	c1	90
s3	c2	60

https://blog.csdn.net/baidu_41871794

D

dno	dname
d1	计算机学院
d2	软件学院
d3	数学院
d4	文学院

https://blog.csdn.net/baidu_41871794

(一) 针对上述实例，给出下列表达式的结果（只写出结果即可，无需计算过程，每小题3分，共24分）

- $\pi_{sname}(\sigma_{sex='M'}(S))$
- $snoGcount(cno)(SC)$
- $\pi_{dname}(D) - \pi_{dname}(S \bowtie D)$
- $\pi_{cname}((sno='s1' \text{ score} > 60(SC)) \bowtie C)$

- 5、 $\neg \exists a.sno(a.score < b.score \wedge b.sno = 's2' \wedge a.cno = 'c2' \wedge b.cno = 'c2' (a(SC)b(SC)))$
- 6、 $\neg \exists name (\neg sno, cno(SC) \div \neg cno(sno = 's3' (SC)) \bowtie S)$
- 7、 $\{ t \mid rC (t[cname] = r[cname] \wedge r[cno] = 'c1') \}$
- 8、 $\{ t \mid rD (t[dno] = r[dno]) \neg \exists vS (v[dno] = t[dno]) \}$

(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21分）

- 1、查询软件学院学生姓名，按照学生姓名降序排序。
- 2、查询没有选修“c2”课程的学生学号。
- 3、查询自己的班长住在同一个宿舍的学生学号。
- 4、查询获得“数据库系统”课程最高分的学生学号。
- 5、查询学生人数最多的院系名称。
- 6、查询被所有学生都选修的课程名称。
- 7、如果学生某门课的考试成绩低于该门课的平均成绩，将该学生的考试成绩提高5%。

JDBC (3分=)：删除表C中cname=‘数据库系统’的数据时，若有数据，则不删且给出提示，若有直接删除，已给出conn.

版权声明：本文为CSDN博主「dodoBehind」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/weixin_42925536/article/details/94399575

版权声明：本文为CSDN博主「日暮途远。」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/baidu_41871794/article/details/94654898