数据库原理试题---A 卷答案

一. 名词解释（12 分）

1. 数据独立性：数据独立性是指应用程序与 DB 的数据结构之间相互独立。

评分细则：描述正确给全分

2. 正则覆盖：满足下列条件的函数依赖集 F 称为正则覆盖，记作 Fc：1）Fc 与 F 等价 2）Fc 中任何函数依赖都不含无关属性 3）Fc 中函数依赖的左半部都是唯一的

评分细则：每条 1 分

3. 两段锁协议：可以保证可串行性。两段锁协议要求每个事物分成两个阶段提出加锁和解锁申请：增长阶段：事物可以获得封锁，不能释放锁；缩减阶段：事物可以释放锁，但不能获得新锁。

评分细则：保证可串行 1 分，每个阶段 1 分

4. 实体完整性约束：关系的主码中的属性值不能为空值

评分细则：描述正确给全分

二. 简答（20 分）

1. 事务的 ACID 特性分别是什么？每个特性的用途是什么？

原子性(Atomicity) 事务中包含的所有操作要么全做，要么全不做

一致性(Consistency) 事务的隔离执行必须保证数据库的一致性

隔离性(Isolation) 系统必须保证事务不受其它并发执行事务的影响

持久性(Durability) 一个事务一旦提交之后，它对数据库的影响必须是永久的

评分细则： ACID 特性 1 分，用途每条 1 分

2. 死锁的发生是坏事还是好事？试说明理由。如何解除死锁状态？

答：在 DBS 运行时，死锁状态是我们不希望发生的，因此死锁的发生本身是一件坏事。但是坏事可以转换为好事。如果我们不让死锁发生，让事务任意并发做下去，那么有可能破坏 DB 中数据，或用户读了错误的数据。从这个意义上讲，死锁的发生是一件好事，能防止错误的发生。

在发生死锁后，系统的死锁处理机制和恢复程序就能起作用，抽取某个事务作为牺牲品，把它撤消，做 ROLLBACK 操作，使系统有可能摆脱死锁状态，继续运行下去。

评分细则： 好坏 1 分，理由 1 分，解除死锁 3 分

3. 在嵌入式 SQL 中，什么情况下的 DML 语句不必涉及到游标操作？

INSERT、DELETE 和 UPDATE 语句；

对于 SELECT 语句，如果已知查询结果肯定是单值时。

评分细则： INSERT、DELETE 和 UPDATE 语句各 1 分，select 语句 2 分

4. 试述 ER 模型、层次模型、网状模型、关系模型和面向对象模型的主要特点。

答：ER 模型直接表示实体类型及实体间联系，与计算机系统无关，充分反映用户的需求，用户容易理解。

层次模型的数据结构为树结构，记录之间联系通过指针实现，查询较快，但 DML 属于过程化的，操作复杂。

网状模型的数据结构为有向图，记录之间联系通过指针实现，查询较快，并且容易实现 M:N 联系，但 DML 属于过程化的语言，编程较复杂。

关系模型的数据结构为二维表格，容易为初学者理解。记录之间联系通过关键码实现。DML 属于非过程化语言，编程较简单。

面向对象模型能完整描述现实世界的数据结构，具有丰富的表达能力，能表达嵌套、递归的数据结构。但涉及的知识面较广，用户较难理解

评分细则：各 1 分

三. 设 R 和 S 是下图表示的关系, 计算下列关系代数表达式和元组表达式的值。（8 分）

A	B	C
2	4	6
3	2	1
5	4	4

R

A	D	E
3	6	9
3	4	5
2	4	7

S

1. $R \bowtie S$

A	B	C	D	E
2	4	6	4	7
3	2	1	6	9
3	2	1	4	5

2. $\sigma_{A<E}(R \times S)$

A	B	C	S_A	D	E
2	4	6	3	6	9
2	4	6	3	4	5
2	4	6	2	4	7
3	2	1	3	6	9
3	2	1	3	4	5
3	2	1	2	4	7
5	4	4	3	6	9
5	4	4	2	4	7

3. $\{ t \mid \exists v \in S(\exists u \in R(u[C]>v[A] \wedge t[A]=u[B] \wedge t[B]=v[E] \wedge t[C]=u[A]))\}$

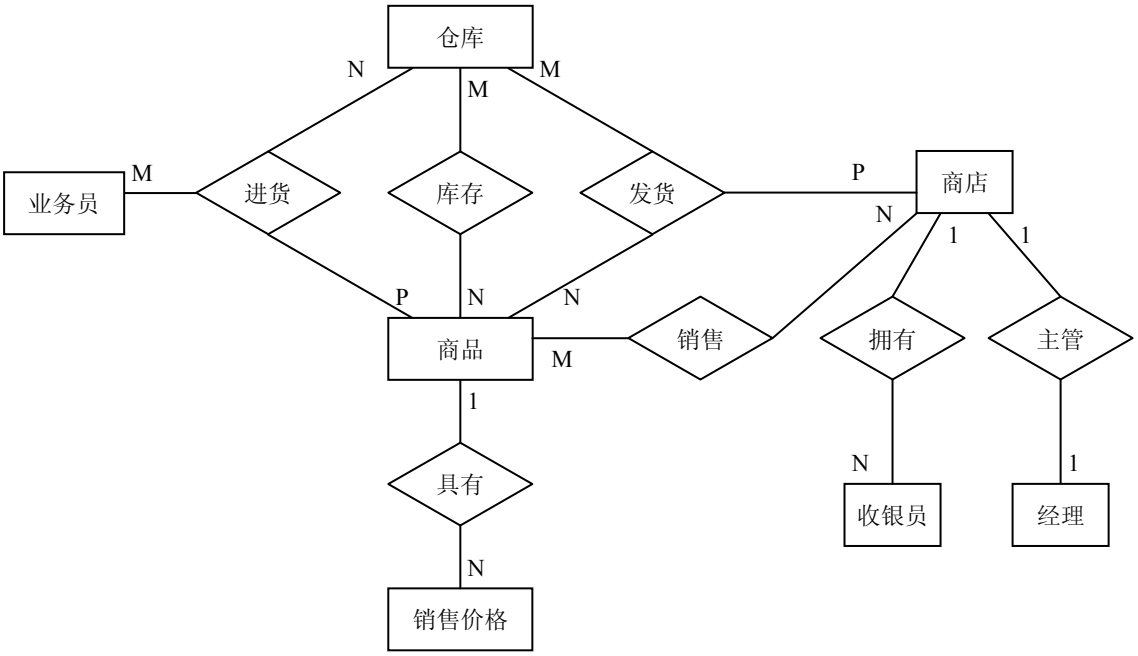
A	B	C
4	9	2
4	5	2
4	7	2
4	9	5
4	5	5
4	7	5

4. $\{ t \mid t \in R \wedge \forall u \in S(t[A] < u[E])\}$

A	B	C
2	4	6
3	2	1

评分细则：结果正确得全分，结果有错误得 0 分。

四. （10 分）



评分细则：实体集 5 分，联系集 5 分需要表明联系的映射基数

五. 试解决下列问题（10 分）

1. 解：①从已知的 F，可推出 BD→BCD，所以$(BD)^+$=BCD。

②由于 B^+=BC，因此左部是 B 的 FD 有四个：

B→ ϕ，B→B，B→C，B→BC。

评分细则：①2 分②2 分

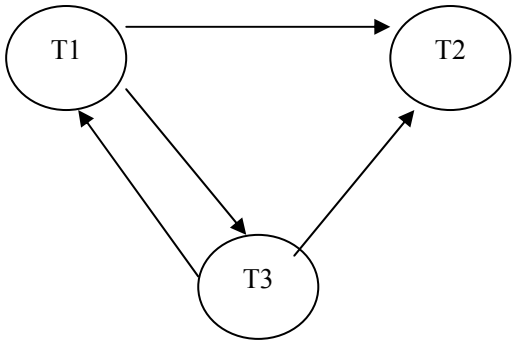
2. 解：①从已知 FD 集 F，可知 R 的候选键是 C。

从 C→B 和 B→A，可知 C→A 是一个传递依赖，因此 R 不是 3NF 模式。

②此时 R 应分解成 ρ ={ CB，BA }， ρ 是 3NF 模式集。

评分细则：①3 分②3 分

六.



有环，不可串行

评分细则：图不对写明是不可串行化的，得 3 分，图对写明是不可串行化的，得 5 分

七. 设数据库中有三个关系：

职工表 EMP（E#，ENAME，AGE，SEX，ECITY），

其属性分别表示职工工号、姓名、年龄、性别和籍贯。

工作表 WORKS（E#，C#，SALARY），

其属性分别表示职工工号、工作的公司编号和工资。

公司表 COMP（C#，CNAME，CITY），

其属性分别表示公司编号、公司名称和公司所在城市。

试写出下列操作：

1 分别使用 SQL 语句、关系代数和元组关系演算，检索超过 50 岁的男职工的工号和姓名。

2 假设每个职工可在多个公司工作，分别使用 SQL 语句、关系代数检索在编号为 C4 和 C8 公司兼职的职工工号和姓名。

3 假设每个职工可在多个公司工作，使用一 SQL 语句，检索每个职工的兼职公司数目和工资总数.显示（E#，NUM，SUM_SALARY），分别表示工号、公司数目和工资总数。

4 分别使用关系代数和 SQL 语句，求不在 C3 公司工作的职工姓名。

5 工号为 E6 的职工在多个公司工作，分别使用 SQL 语句、关系代数和元组关系演算，检索至少在 E6 职工兼职的所有公司工作的职工工号。

6 使用一 SQL 语句，检索联华公司中低于本公司平均工资的职工工号和姓名。

7 使用一 SQL 语句，在每一公司中为 50 岁以上职工加薪 100 元（若职工为多个公司工作，可重复加）。

8 使用一 SQL 语句，在 EMP 表和 WORKS 表中删除年龄大于 60 岁的职工有关元组。

1
```
SELECT E#, ENAME
FROM EMP
WHERE AGE>50 AND SEX='M';
```

$$\pi_{E\#,ENAME}\left(\Sigma_{age>50\wedge sex='m'}(EMP)\right)$$

$$\{t \mid \exists s \in EMP(t[E\#]=s[E\#] \wedge t[ENAME]=s[ENAME] \wedge s[AGE]>50 \wedge s[SEX]='M')\}$$

2
```
SELECT A.E#, A.ENAME
FROM EMP A, WORKS B, WORKS C
WHERE A.E#=B.E# AND B.E#=C.E#
  AND B.C#='C4' AND C.C#='C8';
```

$$\pi_{E\#,ENAME}\left(\Sigma_{works.c\#='c4'\wedge emp.e\#=works.e\#}(EMP\times WORKS)\right)\cap \pi_{E\#,ENAME}\left(\Sigma_{works.c\#='c8'\wedge emp.e\#=works.e\#}(EMP\times WORKS)\right)$$

3
```
SELECT E#, COUNT(C#) AS NUM, SUM(SALARY) AS SUM_SALARY
FROM WORKS
GROUP BY E#;
```

4
```
select ENAME
From  emp
Where e# not in (select e# from works where c#='C3')
```

$$\pi_{ENAME}(EMP)-\pi_{ENAME}\left(\sigma_{C\#='C3'\wedge emp.e\#=works.e\#}(EMP\times WORKS)\right)$$

5
```
SELECT X.E#
FROM WORKS X
WHERE NOT EXISTS
        (SELECT *
          FROM WORKS Y
```

```
                WHERE E#='E6'
                  AND NOT EXISTS
                      (SELECT *
                        FROM WORKS Z
                        WHERE Z.E#=X.E#
                          AND Z.C#=Y.C#));
```

$\Pi_{E\#,\ C\#}(WORKS) \div \Pi_{C\#}(\sigma_{E\#='E6'}(WORKS))$

$\{t \mid \exists w \in EMP\ (t[E\#]=w[E\#]) \wedge (\forall u \in WORKS(u[E\#]='E6' \Rightarrow \exists v \in WORKS(u[C\#]=v[C\#] \wedge t[E\#]=v[E\#]))\}$

6
```
SELECT A.E#, A.ENAME
FROM EMP A, WORKS B, COMP C
WHERE A.E#=B.E# AND B.C#=C.C#
  AND CNAME='联华公司'
  AND SALARY<(SELECT AVG(SALARY)
              FROM WORKS, COMP
              WHERE WORKS.C#=COMP.C#
                AND CNAME='联华公司');
```

7
```
UPDATE WORKS
SET SALARY=SALARY+100
WHERE E# IN (SELECT E# FROM EMP WHERE AGE>50);
```

8
```
DELETE FROM WORKS
WHERE E# IN (SELECT E# FROM EMP WHERE AGE>60);
DELETE FROM EMP
    WHERE AGE>60;
```

评分细则：结果正确得全分，结果有错误得 0 分。