数据库原理部分习题与答案

第 3 章 关系数据库标准语言 SQL

课本例题

第 4 章 数据库安全性

1. 什么是数据库的安全性?

答:数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。

6. 对下列两个关系模式:

学生(学号,姓名,年龄,性别,家庭住址,班级号)

班级(班级号,班级名,班主任,班长)

使用 GRANT 完成下列授权功能:

① 授予用户 U1 拥有对两个表的所有权限,并可给其他用户授权;

② 授予用户 U2 对学生表具有查看权限,对家庭住址具有更新权限;

③ 将对班级表查看权限授予所有用户。

④ 将对学生表的查询、更新权限授予角色 R1。

⑤ 将角色 R1 授予用户 U1,并且 U1 可继续授权给其他角色。

答:

① GRANT ALL PRIVILEGES ON TABLE 学生,班级 TO U1 WITH GRANT OPTION;

② GRANT SELECT,UPDATE(家庭住址) ON TABLE 学生 TO U2;

③ GRANT SELECT ON TABLE 班级 TO PUBLIC;

④ GRANT SELECT,UPDATE ON TABLE 学生 TO R1;

⑤ GRANT R1 TO U1 WITH ADMIN OPTION;

第 5 章 数据库完整性

1. 什么是数据库的完整性?

答:数据库的完整性是指数据的正确性和相容性。

2. 数据库的完整性概念与数据库的安全性概念有什么区别和联系？

答：数据的完整性和安全性是两个不同的概念，但是有一定的联系。前者是为了防止数据库中存在不符合语义的数据，防止错误信息的输入和输出，即所谓垃圾进垃圾出（Garbage In Garbage Out）所造成的无效操作和错误结果；后者是保护数据库防止恶意的破坏和非法的存取。也就是说，安全性措施的防范对象是非法用户和非法操作，完整性措施的防范对象是不合语义的数据。

6. 假设有下面两个关系模式：

职工（职工号，姓名，年龄，职务，工资，部门号），其中职工号为主码；

部门（部门号，名称，经理名，电话），其中部门号为主码；

用 SQL 语言定义这两个关系模式，要求在模式中完成以下完整性约束条件的定义：（1）定义每个模式的主码；（2）定义参照完整性；（3）定义职工年龄不得超过 60 岁。

答：

```
CREATE TABLE DEPT
    (Deptno NUMBER(2)PRIMARY KEY,
     Deptname VARCHAR(10),
     Manager VARCHAR(10),

     PhoneNumber Char(12)
     );


CREATE TABLE EMP
    (Empno NUMBER(4)PRIMARY KEY,
     Ename VARCHAR(10),
     Age NUMBER(2),
     Job VARCHAR(9),
     Sal NUMBER(7,2),
     Deptno NUMBER(2),
        CONSTRAINT C1 CHECK(Age<=60),
        CONSTRAINT FK_DEPTNO FOREIGN KEY(Deptno)REFERENCES DEPT(Deptno));
```

第 11 章 数据库恢复技术

1. 试述事务的概念及事务的 4 个特性。恢复技术能保证事务的哪些特性?

答:

事务是用户定义的一个数据库操作序列,这些操作要么全做、要么全不做,是一个不可分割的工作单位。

事务具有 4 个特性:原子性(Atomicity)、一致性(Consistency)、隔离性(Isolation)和持续性(Durability)。这 4 个特性也简称为ACID 特性。

原子性:事务是数据库的逻辑工作单位,事务中包括的诸操作要么都做,要么都不做。

一致性:事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。

隔离性:一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数据对其他并发事务是隔离的,并发执行的各个事务之间不能互相干扰。

持续性:持续性也称永久性(permanence),指一个事务一旦提交,它对数据库中数据的改变就应该是永久性的。接下来的其他操作或故障不应该对其执行结果有任何影响。

故障恢复可以保证事务的原子性与持续性。

4. 考虑下图所示的日志记录:

序号	日志
1	$T1$:开始
2	$T1$:写 A,$A=10$
3	$T2$:开始
4	$T2$:写 B,$B=9$
5	$T1$:写 C,$C=11$
6	$T1$:提交
7	$T2$:写 C,$C=13$
8	$T3$:开始
9	$T3$:写 A,$A=8$
10	$T2$:回滚
11	$T3$:写 B,$B=7$
12	$T4$:开始
13	$T3$:提交
14	$T4$:写 C,$C=12$

① 如果系统故障发生在 14 之后,说明哪些事务需要重做,哪些事务需要回滚;

② 如果系统故障发生在 10 之后,说明哪些事务需要重做,哪些事务需要回滚;

③ 如果系统故障发生在 9 之后,说明哪些事务需要重做,哪些事务需要回滚;

④ 如果系统故障发生在 7 之后,说明哪些事务需要重做,哪些事务需要回滚。

答：

① 重做：$T1$、$T3$；回滚：$T2$、$T4$。

② 重做：$T1$；回滚：$T2$、$T3$。

③ 重做：$T1$；回滚：$T2$、$T3$。

④ 重做：$T1$；回滚：$T2$。

5. 考虑题 4 所示的日志记录，假设开始时 A、B、C 的值都是 0：

① 如果系统故障发生在 14 之后，写出系统恢复后 A、B、C 的值。

② 如果系统故障发生在 12 之后，写出系统恢复后 A、B、C 的值。

③ 如果系统故障发生在 10 之后，写出系统恢复后 A、B、C 的值。

④ 如果系统故障发生在 9 之后，写出系统恢复后 A、B、C 的值。

⑤ 如果系统故障发生在 7 之后，写出系统恢复后 A、B、C 的值。

⑥ 如果系统故障发生在 5 之后，写出系统恢复后 A、B、C 的值。

答：

① $A=8, B=7, C=11$。

② $A=10, B=0, C=11$。

③ $A=10, B=0, C=11$。

④ $A=10, B=0, C=11$。

⑤ $A=10, B=0, C=11$。

⑥ $A=0, B=0, C=0$。

第 12 章 并发控制

封锁的概念、类型，活锁、死锁概念和解决方法，冲突可串行化调度，封锁粒度

3. 什么是封锁？基本的封锁类型有几种？试述它们的含义。

答：

封锁就是事务 T 在对某个数据对象例如表、记录等操作之前，先向系统发出请求，对其加锁。加锁后事务 T 就对该数据对象有了一定的控制，在事务 T 释放它的锁之前，其他的事务不能更新或读取此数据对象。

基本的封锁类型有两种：排他锁（简称 X 锁）和共享锁（简称 S 锁）。

排他锁又称为写锁。若事务 T 对数据对象 A 加上 X 锁，则只允许 T 读取和修改 A，其他任何事务都不能再对 A 加任何类型的锁，直到 T 释放 A 上的锁。这就保证了其他事务在 T 释放 A 上的锁之前不能再读取和修改 A。

共享锁又称为读锁。若事务 T 对数据对象 A 加上 S 锁，则事务 T 可以读 A 但不能修改 A，其他事务只能再对 A 加 S 锁，而不能加 X 锁，直到 T 释放 A 上的 S 锁。这就保证了其他事务可以读 A，但在 T 释放 A 上的 S 锁之前不能对 A 做任何修改。

5. 什么是活锁？试述活锁的产生原因和解决方法。

答：

如果事务 T_1 封锁了数据 R，事务 T_2 又请求封锁 R，于是 T_2 等待。T_3 也请求封锁 R，当 T_1 释放了 R 上的封锁之后系统首先批准了 T_3 的请求，T_2 仍然等待。然后 T_4 又请求封锁 R，当 T_3 释放了 R 上的封锁之后系统又批准了 T_4 的请求……T_2 有可能永远等待，如下图所示。这就是活锁的情形。活锁的含义是该等待事务等待时间太长，似乎被锁住了，实际上可能被激活。

T_1	T_2	T_3	T_4
lock R			
	lock R		
	等待	Lock R	
Unlock	等待		Lock R
	等待	Lock R	等待
	等待		等待
	等待	Unlock	等待
	等待		Lock R
	等待		

活锁产生的原因：当一系列封锁不能按照其先后顺序执行时，就可能导致一些事务无限期等待某个封锁，从而导致活锁。

避免活锁的简单方法是采用先来先服务的策略。当多个事务请求封锁同一数据对象时，封锁子系统按请求封锁的先后次序对事务排队，数据对象上的锁一旦释放就批准申请队列中第一个事务获得锁。

6. 什么是死锁？请给出预防死锁的若干方法。

答：

如果事务 T_1 封锁了数据 R_1，T_2 封锁了数据 R_2，然后 T_1 又请求封锁 R_2，因 T_2 已封锁了 R_2，于是 T_1 等待 T_2 释放 R_2 上的锁。接着 T_2 又申请封锁 R_1，因 T_1 已封锁了 R_1，T_2 也只能等待 T_1 释放 R_1 上的锁。如下图所示。这样就出现了 T_1 在等待 T_2，而 T_2 又在等待 T_1 的局面，T_1 和 T_2 两个事务永远不能结束，形成死锁。

T_1	T_2
lock R_1	
	Lock R_2
Lock R_2	
等待	
等待	Lock R_1
等待	等待

防止死锁的发生其实就是要破坏产生死锁的条件。预防死锁通常有两种方法：

① 一次封锁法

要求每个事务必须一次将所有要使用的数据全部加锁，否则就不能继续执行。

② 顺序封锁法

预先对数据对象规定一个封锁顺序，所有事务都按这个顺序实行封锁。

10. 今有三个事务的一个调度 $r_3(B)\ r_1(A)\ w_3(B)\ r_2(B)\ r_2(A)\ w_2(B)\ r_1(B)\ w_1(A)$，该调度是冲突可串行化的调度吗？为什么？

答：

是冲突可串行化的调度。

$Sc1=r_3(B)\ r_1(A)\ w_3(B)\ r_2(B)\ r_2(A)\ w_2(B)\ r_1(B)\ w_1(A)$，交换 $r_1(A)$ 和 $w_3(B)$，得到

$$r_3(B)\ w_3(B)\ r_1(A)\ r_2(B)\ r_2(A)\ w_2(B)\ r_1(B)\ w_1(A)$$

再交换 $r_1(A)$ 和 $r_2(B)\ r_2(A)\ w_2(B)$，得到

$$Sc2=r_3(B)\ w_3(B)\ r_2(B)\ r_2(A)\ w_2(B)\ r_1(A)\ r_1(B)\ w_1(A)$$

由于 Sc2 是串行的，而且两次交换都是基于不冲突操作的，所以

$$Sc1=r_3(B)\ r_1(A)\ w_3(B)\ r_2(B)\ r_2(A)\ w_2(B)\ r_1(B)\ w_1(A)$$

是冲突可串行化的调度。