

# 第十四章作业

---

## 14.12

---

### 14.12 列出 ACID 特性，解释每一特性的用途。

- 原子性：事务的所有操作在数据库中要么全部正确反映出来，要么完全不反映
- 一致性：隔离执行事务时，也即在没有其他事务并发执行的情况下，要保持数据库的一致性
- 隔离性：尽管多个事务可能并发执行，但系统必须保证对于任何一对事务，在彼此看来，另一个事务要么在自己先前完成，要么在自己之后完成，每个事务都感觉不到系统中有其他事务在并发地执行
- 持久性：一个事务成功完成后，它对数据库的改变必须是永久的，即使出现系统故障

## 14.13

---

### 14.13 事务从开始执行直到提交或终止，其间要经过几个状态。列出所有可能出现的事务状态序列，解释每一种状态变迁出现的原因。

- 活动状态：初始状态，事务执行时处于这个状态
- 部分提交状态：最后一条语句执行后
  - 活动状态→部分提交状态：事务从活动状态开始，当事务完成它的最后一条语句后就进入了部分提交状态
- 失败状态：发现正常的执行不能继续后
  - 活动状态（部分提交状态）→失败状态：系统判定事务不能继续正常执行后，例如硬件或逻辑错误，事务就进入失败状态
- 提交状态：成功完成后
  - 部分提交状态→提交状态：数据库系统往磁盘上写入足够的信息，确保即使出现故障时事务所做的更新也能在系统重启后重新创建，当最后一条这样的信息写完后，事务就进入提交状态
- 中止状态：事务回滚并且数据库已恢复到事务开始执行前的状态后
  - 失败状态→中止状态：事务进入失败状态后，这种事务必须回滚，此后它可以重启或杀死事务，注意重启的事务会被当做一个新事务

## 14.14

---

### 14.14 解释串行调度和可串行化调度的区别。

- 串行调度：每个串行调度由来自各事务的指令序列组成，其中属于同一事务的指令在调度中紧挨在一起，对于有  $n$  个事务的事务组，共有  $n!$  个不同的有效串行调度
- 可串行化调度：相较于串行调度限制更小，且等价于某些串行调度，核心概念是调度等价，调度等价又包括冲突等价和视图等价

## 14.15

---

#### 14.15 考虑以下两个事务：

```

T13: read(A);
      read(B);
      if A=0 then B:= B + 1;
      write(B);
T14: read(B);
      read(A);
      if B=0 then A:= A + 1;
      write(A);
    
```

设一致性需求为  $A=0 \vee B=0$ ，初值是  $A=B=0$ 。

- 说明包括这两个事务的每一个串行执行都保持数据库的一致性。
- 给出  $T_{13}$  和  $T_{14}$  的一次并发执行，执行产生不可串行化调度。
- 存在产生可串行化调度的  $T_{13}$  和  $T_{14}$  的并发执行吗？

a.

先执行  $T_{13}$  再执行  $T_{14}$

	A	B
初始状态	0	0
执行 $T_{13}$	0	1
执行 $T_{14}$	0	1

先执行  $T_{14}$  再执行  $T_{13}$

	A	B
初始状态	0	0
执行 $T_{14}$	1	0
执行 $T_{13}$	1	0

不会有冲突

b.

$T_{13}$	$T_{14}$
read(A)	
	read(B)
read(B)	
	read(A)
B := B + 1	
	A := A + 1
write(B)	
	write(A)

C.

不存在，因为两个事务互相依赖对方修改后的结果再读取数据再做判断，并行执行一定会违反一致性规则，即使得二者同时为1。