**16.3**

1. T2:R(X), T2:R(Y), T2:W(X), T1:R(X), ……T1:R(X)是脏读。
2. T2:R(X), T2:R(Y), T1:R(X), T1:R(Y), T1:W(X), ……T2:R(X)是不可重复读。
3. T2:R(X), T2:R(Y), T1:R(X), T1:R(Y), T1:W(X), T2:W(X), ……T2:W(X)把T1:W(X)给覆盖掉了。
4. 1. T1不会得到X的分享锁，因为T2已经获得了X的互斥锁，T1只能等到T2完成后才能继续。
   2. T1不会得到X的互斥锁，因为T2已经获得了X的互斥锁，T1只能等到T2完成后才能继续。
   3. 跟b一样，T1不会得到X的分享锁，因为T2已经获得了X的互斥锁，T1只能等到T2完成后才能继续。

**16.7**

1. 由于没有涉及已存在的记录，只需要READ UNCOMMITTD即可。
2. 因为需要避免脏读，必须在更新的记录上加锁，所以需要READ COMMITTED。
3. 因为我们需要读取Enrolled表的所有记录，在此期间需要避免所有对该表的更新，所以需要SERIALIZABLE。
4. 与3的情况相同，也需要SERIALIZABLE。

**17.2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 可串行化 | 冲突可串行化 | 观测可串行化 | 可恢复 | 避免级联式中止 | 严格的 |
| 1 | × | × | × | √ | √ | × |
| 2 | √ | √ | √ | \* | × | × |
| 3 | √ | √ | √ | \* | × | × |
| 4 | × | × | × | \* | × | × |
| 5 | √ | √ | √ | √ | √ | × |
| 6 | √ | × | √ | √ | √ | × |
| 7 | × | × | × | × | × | × |
| 8 | × | × | × | × | × | × |
| 9 | √ | √ | √ | × | × | × |
| 10 | √ | √ | √ | × | × | × |
| 11 | √ |  | √ | √ | √ | √ |
| 12 | √ |  | √ | √ | × | × |

\*：不能推断是否是可恢复的，因为提交和取消操作的顺序没有给出。

**17.11**

1. IS on D; IS on F2; IS on P1200; S on P1200:5.
2. IS on D;

IS on F2;

IS on P1200, S on 1201 through 1204, IS on P1205;

S on P1200:98/99/100, S on P1205:1/2.

1. IS on D; S on F1
2. IS on D; IS on F1; S on P500 through P520.
3. IS on D; S on F1
4. IS and IX on D; SIX on F1.
5. IX on D; IX on F2; X on P1200.
6. IX on D; X on F1 and F2.
7. IX on D; X on F1 and F2.

**18.3**

1. 分析阶段从日志中最近的检查点开始直到日志最后的记录，判断应该从哪里开始重做，并识别缓冲池中的脏页和崩溃时的当前操作，判断哪些应该反做；

重做阶段从日志的某个合适的点开始，重复所有操作，将数据库恢复到崩溃时的状态；

反做阶段取消没有提交的事物，使数据库只反映已提交事务的操作结果。

* 1. 检查阶段开始于LSN10，检查点结束的位置，之后进行的操作如下（TT表示事务表，DPT表示脏页表）：

LSN 20 将 (T1, 20) 插入TT， (P5, 20) 插入DPT

LSN 30 将 (T2, 30) 插入TT， (P3, 30) 插入DPT

LSN 40 将T2的状态改为“完成”

LSN 50 将T2从TT中删除

LSN 60 将(T3, 60) 加入 TT.

LSN 70 将(T1, 20) 改为 (T1, 70)

最终，TT: (T1, 70), (T3, 60); DPT: (P5, 20), (P3, 30)

* 1. 开始于LSN20，也即DPT中最小的LSN。

LSN 20 重做对P5的修改

LSN 30 根据日志中P3的修改记录判断是否要将P3写回硬盘

LSN 40 无动作

LSN 50 无动作

LSN 60 重做对P3的修改

LSN 70 无动作

* 1. 开始于LSN70，即TT中最大的LSN。

LSN 70 将LSN 20加入丢失的集合

LSN 60 反做对P3的修改并添加一个CLR进反做计划

LSN 20 反做对P5的修改并添加一个CLR。

**18.4**



|  |  |  |
| --- | --- | --- |
| **LSN** | **prevLSN** | **undonextLSN** |
| 00 |  |  |
| 10 | 00 | 00 |
| 20 |  |  |
| 30 |  |  |
| 40 | 30 |  |
| 50 | 20 | 20 |
| 60 | 50 | 50 |
| 70 | 60 |  |

**LSN** **LOG**

00 begin\_checkpoint

10 end\_checkpoint

20 更新：T1写P1

30 更新：T2写P2

40 更新：T3写P3

50 T2提交

60 更新：T3写P2

70 T2结束

80 更新：T1写P5

90 T3中止

CRASH, RESTART

1. 1. 将P3恢复至LSN 60时的状态
   2. 将P5恢复至LSN 50时的状态
   3. 将P5恢复至LSN 20时的状态

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **LSN** | **prevLSN** | **transID** | **type** | **pageId** | **undonextLSN** |
| 80 | 70 | T2 | CLR | P3 | 50 |
| 90 | 80 | T2 | CLR | P3 | 20 |
| 100 | 90 | T2 | CLR | P3 |  |
| 110 | 100 | T2 | END |  |  |