数据库系统概论

第11章　故障恢复

第3题 登记日志文件时为什么必须先写日志文件，后写数据库?

把对数据的修改写到数据库中，和把记录这个修改的日志记录写到日志文件中是两个不同的操作，有可能在这两个操作之间发生故障，即这两个写操作只完成了其中一个。

1）如果先写了数据库修改而在运行日志中没有登记这个修改，则以后就无法恢复这个修改了。

2）如果先写日志但没有修改数据库，在恢复时只不过多执行一次undo或者redo操作，并不会影响数据库的正确性，所以一定要先写日志文件，然后再写数据库文件。

故障发生时：

1）未提交事务：可能磁盘数据已有发生了部分变化，保险起见回滚(写旧值)一遍（即使没有变化，把旧值写一次也不会有错，但万一数据已经变成新值了，不写回旧值就有问题了）

2）已提交事务：可能数据还未从缓冲区写回磁盘，重写（写新值）一次（即使已经写回去了，覆盖一次也不会有错，但万一数据已经还没有写回磁盘，不写新值就有问题了）

第4题

考虑如图所示的日志记录

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，说明哪些事物需要重做，哪些事物需要回滚；
2. 如果系统故障发生在10之后，说明哪些事物需要重做，哪些事物需要回滚；
3. 如果系统故障发生在9之后，说明哪些事物需要重做，哪些事物需要回滚；
4. 如果系统故障发生在7之后，说明哪些事物需要重做，哪些事物需要回滚。

第4题

1. rollback和commit都算事务结束
2. redo队列：正向扫描日志，重新执行每条操作（通常操作是写入更新后的新值，但回滚是写旧值）
3. undo队列：反向扫描日志，撤销原有的每条操作（通常是写入更新前的旧值）

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）故障在14之后：重做(redo)：T1, T3; 回滚（Undo）：T2, T4
2）故障在10之后：重做(redo)：T1　　回滚（Undo）：T2, T3
3）故障在9之后：重做(redo)：T1　　回滚（Undo）：T2, T3
4）故障在7之后：重做(redo)：T1　　回滚（Undo）：T2

上述的1）和2）：T2已经回滚结束，所以放入了redo的队列中，但是执行的操作本身是回滚

或者(如果按照《数据库系统概念》第六版 中的具体例子)：

1）故障在 14 之后：重做(redo)：T1, T2, T3, T4 ；　回滚（Undo）：T4

2）故障在 10 之后：重做(redo)：T1, T2, T3;　　回滚（Undo）：T3

3）故障在 9 之后：　重做(redo)：T1, T2, T3;　　回滚（Undo）：T2, T3

4）故障在 7 之后：　重做(redo)：T1, T2;　　回滚（Undo）：T2

第5题

考虑如图所示的日志记录

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

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。

第5题

考虑如图所示的日志记录

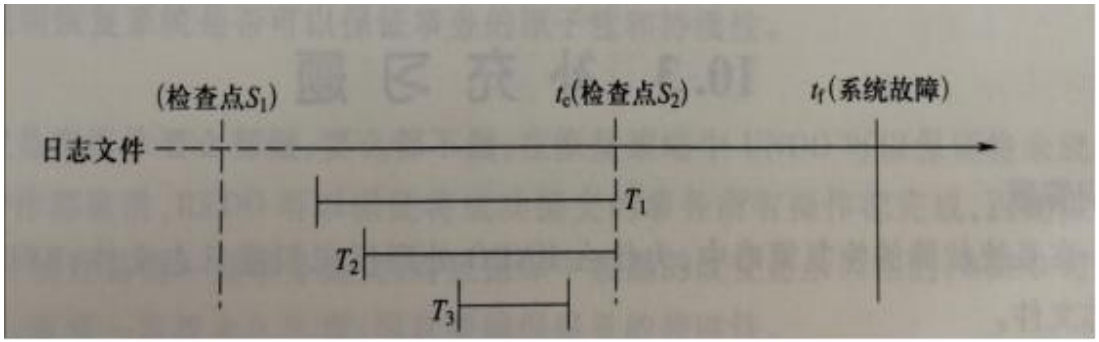
序号	日志
1	T1：开始
2	T1：写 A, A=10
3	T2：开始
4	T2：写 B，B=9
5	T1：写 C，C=11
6	T1：提交
7	T2：写 C，C=13
8	T3：开始
9	T3：写 A，A=8
10	T2：回滚
11	T3：写 B，B=7
12	T4：开始
13	T3：提交
14	T4：写 C，C=12

假设开始时A，B，C的值都是0，故障发生在某点，写出系统恢复后A，B，C的值。

	故障点	A	B	C
1)	14 之后	8	7	11
2)	12 之后	10	0	11
3)	10 之后	10	0	11
4)	9 之后	10	0	11
5)	7 之后	10	0	11
6)	5 之后	0	0	0

第8题 具有检查点的恢复技术有什么优点？试举一个具体的例子加以说明

利用日志技术进行数据库恢复时，恢复子系统必须搜索整个日志，这将耗费大量的时间。此外，需要redo处理的事务实际上已经将他们的更新操作结果写到了数据库中，恢复子系统又重新执行了这些操作，浪费了大量时间。检查点技术就是为了解决这些问题。在采用检查点技术之前恢复时需要从头扫描日志文件，而利用检查点技术，只需要从t0时刻开始扫描日志，这就缩短了扫描日志的时间。



如上图中事物T3的更新操作，实际上已经写到了数据库中，进行恢复时没有必要再进行redo处理，采用检查点技术就可以做到这点。