数据库作业十

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：

①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。
②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。
（1）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。
（2）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。
（3）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。
（4）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。
③故障恢复可以保证事务的原子性与持续性。

2. 为什么事务非正常结束时会影响数据库数据的正确性，请举例说明之。

答：

① 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。如果数据库系统运行中发生故障，有些事务尚未完成就被迫中断，这些未完成事务对数据库所做的修改有一部分已写入物理数据库，这时数据库就可能处于不正确的状态，或者说是不一致的状态。
② 例如某工厂的库存管理系统中，要把数量为 Q 的某种零件从仓库 1 移到仓库 2 存放，则可以定义一个事务 T。包括两个操作:Q1=Q1-Q，Q2 = Q2 + Q。如果 T 非正常终止时只做了第一个操作，则数据库就处于不一致性状态，库存量无缘无故少 Q。

3. 登记日志文件时为什么必须先写日志文件，后写数据库？

答：

（1）把对数据的修改写到数据库中和把表示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

成了一个。
（2）如果先写了数据库修改，而在运行记录中没有登记这个修改，则以后就无法恢复这个修改了。如果先写日志，但没有修改数据库，在恢复时只不过是多执行一次 UNDO 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，即首先把日志记录写到日志文件中，然后写数据库的修改。

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：
在系统故障发生之前已提交数据的事物重做，其他事务回滚。即开始未结束撤销，未开始，开始已回滚不需要处理，相当于没做。
①重做:T1、T3（已经提交）；回滚：T2、T4。
②重做：T1；回滚；T2、T3(T4 还没开始)。
③重做：T1；回滚：T2、73。
④重做：T1；回滚：T2（T3 还没开始）。

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 W、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：
①A = 8，B = 7，C= 11。
②A= 10，B = 0，C= 11。
③A= 10，B = 0，C= 11。
④A= 10，B = 0，C= 11。
⑤A= 10，B = 0，C= 11。
⑥A = 0，B = 0，C = 0。（都没提交，都回滚）

6.针对不同的故障，试给出恢复的策略和方法。（即如何进行事务故障的恢复、系统故障的恢复，以及如何进行介质故障恢复。）

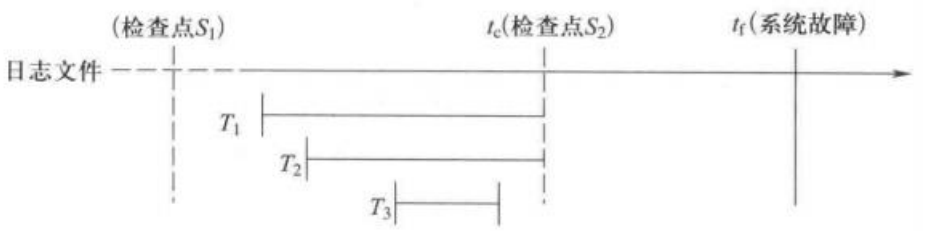
答：

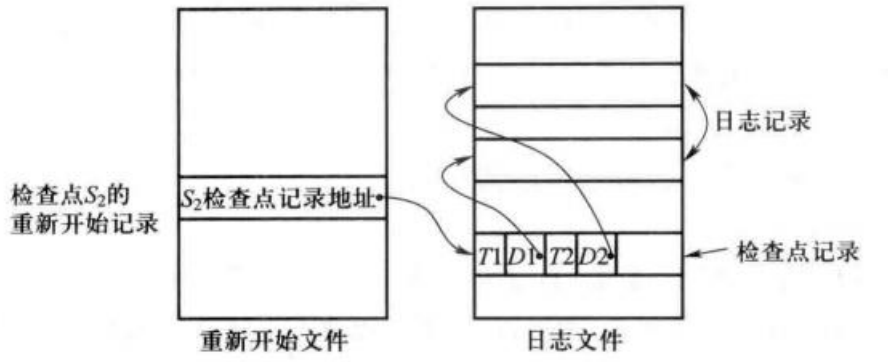
（1）事务故障的恢复步骤是：
① **反向**扫描文件日志，**查找该事务的更新操作**。
② 对该事务的更新操作执行逆操作。即将日志记录中“更新前的值”写入数据库直至读到此事务的开始标记，该事务故障的恢复就完成了。
（2）系统故障的恢复步骤是：
①**正向**扫描日志文件，找出在**故障发生前已经提交的事务队列（REDO 队列）和未完成的 事务队列（UNDO 队列）。**
②对未完成的事务队列中的各个事务进行 UNDO 处理。
③对已经提交的事务队列中的各个事务进行 REDO 处理。
（3）介质故障的恢复步骤是：
①装入最新的数据库后备副本（离故障发生时刻最近的转储副本），使数据库恢复到最近一次转储时的一致性状态。
②装入转储结束时刻的日志文件副本。
③启动系统恢复命令，由 **DBMS 完成恢复功能**，即重做已完成的事务。

7.什么是检查点记录，检查点记录包括哪些内容？

答：

（1）检查点记录是一类新的日志纪录。它的内容包括：
①建立检查点时刻所有正在执行的事务清单，如下图中的 T1、T2
②这些事务的最近一个日志记录的地址，如下图中的 D1、D2

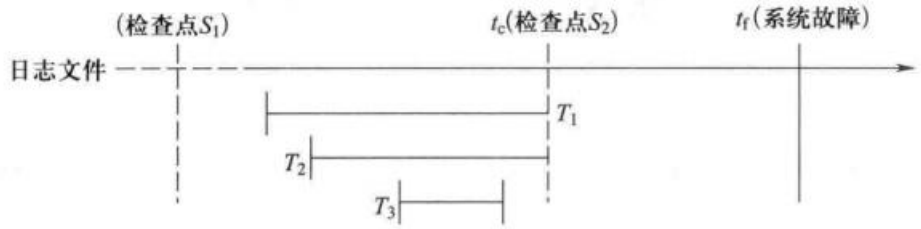




8. 具有检查点的恢复技术有什么优点？试举一个具体的例子加以说明。

答：

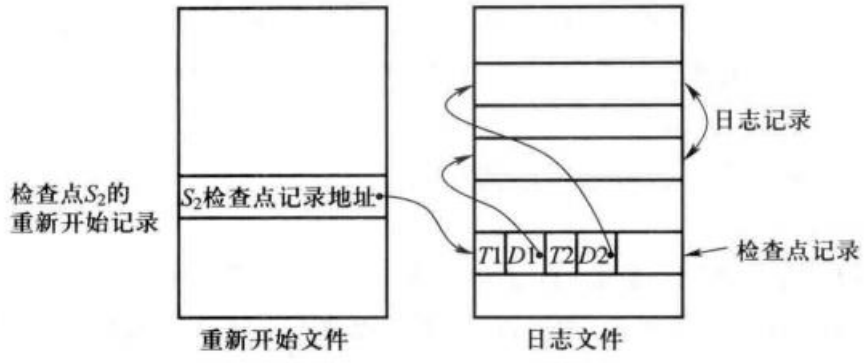
利用日志技术进行数据库恢复时，恢复子系统必须搜索整个日志，这将耗费大量的时间。 此外，需要 REDO 处理的事务实际上已经将它们的更新操作结果写到数据库中了，恢复子系统又重新执行了这些操作，浪费了大量时间。检查点技术就是为了解决这些问题。
例如：



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 tc 开始扫描日志，这就缩短了扫描日志的时间。
事务 T3 的更新操作实际上已经写到数据库中了，进行恢复时没有必要再 REDO 处理，采用检查点技术做到了这一点。

9.试述使用检查点方法进行恢复的步骤

答：



①在重新开始文件中，找到最后一个检查点记录在日志文件中的地址， 由该地址在日志文件中找到最后一个检查点记录。
②由该检查点记录得到检查点建立时刻所有正在执行的事务清单 ACTIVE-LIST。这里建立两个事务队列：
（1）UNDO-LIST：需要执行 undo 操作的事务集合；
（2）REDO-LIST：需要执行 redo 操作的事务集合。
把 ACTIVE-LIST 暂时放入 UNDO-LIST 队列。REDO 队列暂为空。
③从检查点开始正向扫描日志文件
（1）如有新开始的事务 Ti，把 Ti 暂时放入 UNDO-LIST 队列；
（2）如有提交的事务 Tj；，把 Tj 从 UNDO-LIST 队列移到 REDO-LIST 队列，直到日志文件结束。
④对 UNDO-LIST 中的每个事务执行 UNDO 操作，对 REDO-LIST 中的每个事务执行 REDO 操作。

10.什么是数据库镜像？它有什么用途？

答：

（1）数据库镜像即**根据 DBA 的要求，自动把整个数据库或者其中的部分关键数据复制到另一个磁盘上**。每当主数据库更新时，DBMS 自动把更新后的数据复制过去，即 DBMS 自动保证镜像数据与主数据的一致性。
（2）数据库镜像的用途：
①用于数据库恢复。当出现介质故障时，镜像磁盘可继续使用，同时 DBMS 自动利用镜像磁盘数据进行数据库的恢复，不需要关闭系统和重装数据库副本。
②提高数据库的可用性。在没有出现故障时，当一个用户对某个数据加排他锁进行修改时，其他用户可以读镜像数据库上的数据，而不必等待该用户释放锁。