数据库的备份与恢复

主要考点：

1. 数据库系统故障种类
2. 数据库备份
3. 数据库恢复

1、数据库系统故障的种类

> 数据库系统故障有3类：

1. **事务故障：**是由于程序执行错误而引起事务非预期的、异常终止的故障。通常有如下两类错误引起事务执行失败：

   > （1）逻辑错误：如非法输入、找不到数据、溢出、超出资源限制等原因引起的事务执行失败。
   > （2）系统错误：系统进入一种不良状态（如死锁），导致事务无法继续执行。

2. **系统故障：**是指硬件故障、软件（如DBMS、OS或应用程序）漏洞的影响，导致丢失内存中的信息，影响正在执行的事务，但未破坏存储在外存上的信息。
3. **介质故障：**是指数据库的存储介质发生故障，如磁盘损坏、瞬间强磁场干扰等。这种故障直接破坏了数据库，会影响到所有正在读取这部分数据的事务。

2、数据库备份

- 数据库转储是将数据库自制到另一个磁盘或磁带上保存起来的过程，又称为**数据备份**。
  1. **静态转储和动态转储：**静态转储是指在转储期间不允许对数据库进行任何存取、修改操作；动态转储是在转储期间允许对数据库进行存取、修改操作；因此，转储和用户事务是可并发执行。
  2. **海量转储和增量转储：**海量转储是指每次转储全部数据；增量转储是指每次只转储上次转储后更新过的数据。
  3. **日志文件：**在事务处理的过程中，DBMS把事务开始、事务结束以及对数据库的插入、删除和修改的每一次操作写入日志文件。
  4. **数据库镜像：**为了避免磁盘介质出现故障影响数据库的可用性，许多DBMS提供数据库镜像功能用于数据库恢复。

3、数据库恢复

要使数据库在发生故障后能够恢复，必须建立冗余数据，在故障发生后利用这些冗余数据实施数据库恢复，常用的是**数据转储**和**日志文件**。

1. **故障恢复的两个操作：**

（1）**撤销事务（UNDO）**：将未完成的事务撤销，使数据库恢复到事务执行前的正确状态。

> 撤销事务的过程：反向扫描日志文件（由后往前扫描），查找事务的更新操作；对该事务的更新操作执行逆操作，用日志文件记录中更新前的值写入数据库，插入的记录从数据库中删除，删除的记录重新插入数据库中；继续反向扫描日志文件，查找该事务的其他更新操作并执行逆操作直至事务开始标志。

（2）重做事务（PEDO）：将已提交的事务重新执行。

> 重做事务的过程：从事务的开始标志起，正向扫描日志文件，重新执行日志文件登记的该事务对数据库的所有操作，直至事务结束标志。

2. **故障恢复策略：**

（1）事务故障的恢复：事务故障是事务在运行至正常终止点（SUMMIT或ROLLBACK）前终止，日志文件只有该事务的开始标识而没有结束的标识。对这类故障的恢复通常是通过撤销（UNDO）产生故障的事务，是数据库恢复到该事务执行前的正确状态来完成的。

具体做法：

1. 反向扫描日志文件，查找该事务的更新操作。
2. 对事务的更新操作执行逆操作。
3. 继续反向扫描日志文件，查找该事务的其他更新操作，并做同样的处理，直到事务的开始标志。

> 注：事务故障的恢复是由系统自动完成的，对用户是透明的。

（2）系统故障的恢复：系统故障会使数据库的数据不一致，如下：

> 一是未完成的事务对数据库的更新可能已经写入数据库；
> 二是已提交的事务对数据库的更新可能还在缓冲区没来得及写入数据库

因此对于系统故障，恢复操作是UNDO+REDO：

1. 撤销故障发生时未完成的事务（UNDO）
2. 重做已经提交的事务（REDO）

（3） 介质故障的恢复：介质故障时数据库遭到破坏，需要重装数据库，一般需要DBA的参与，装载故障前最近一次的备份和故障前的日志文件副本，再按照系统故障的过程执行撤销（UNDO）和重做（REDO）来恢复。

4、具有检查点的恢复技术

1、检查点（CHECKPOINT）出现的原因：

- 搜索整个日志将耗费大量的时间
- 很多需要重做处理的事务实际上已经将其更新操作结果写到了数据库中，而恢复子系统又重新执行了这些操作，浪费了大量时间。

2、检查点记录的内容：

- 建立检查点时刻所有正在执行的事务清单。
- 这些事务最近一个日志记录的地址。

3、系统可以定期或不定期地建立检查点、保存数据库状态：

- 将当前日志缓冲区中的所有日志记录写入磁盘的日志文件中。
- 在日志文件中写入一个检查点记录
- 将当前数据缓冲区的所有数据记录写入磁盘的数据库中
- 把检查点记录在日志文件中的地址写入一个重新开始文件

4、系统使用检查点方法进行恢复的步骤是：（背）

- 找到最后一个检查点时所有的活动事务清单ACTIVE-LIST .
- 建立两个事务队列：UNDO - LIST和REDO-LIST，把ACTIVE-LIST中的所有活动事务暂时放入UNDO - LIST，REDO - LIST暂为空。
- 从检查点开始正向扫描日志文件：

  （a）如有新开始的事务T_i，把T_i暂时放入UNDO队列。

  （b）如有已提交的事务T_j，把T_j从UNDO队列移到REDO队列，直到日志文件结束。
- 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。

> 简单理解：
> 在检查点之前提交的，不需要做REDO和UNDO
> 在检查点时，已经开始但还未结束的，且在故障发生前提交的作REDO，没提交即做UNDO
> 在检查点后开始，在故障发生前提交的作REDO，没提交作UNDO
> （在REDO-LIST是事务生效，UNDO-LIST是不生效）