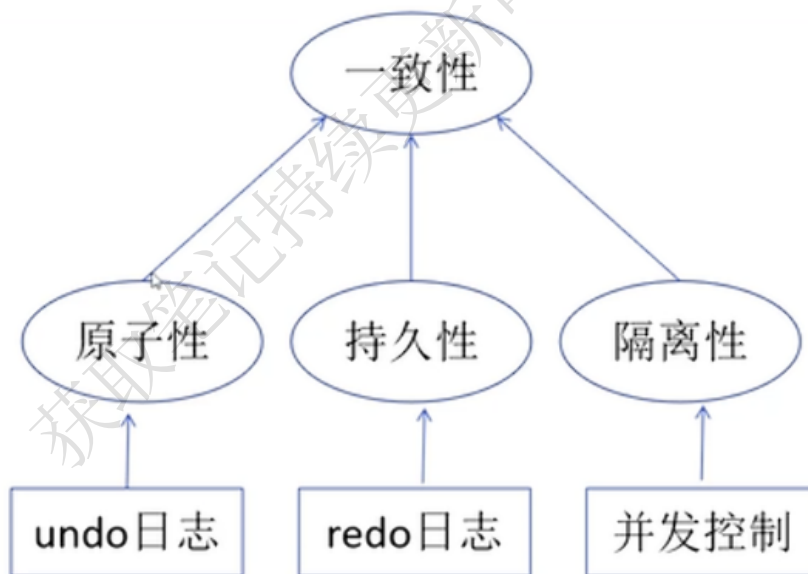


Lecture 7: 数据库恢复技术

事务

概念: 什么是事务

- 概念:
 - 事务 (transaction control language, TCL) 即事务控制语言, 由一组 SQL 语句组成一个执行单元, 这个执行单元要么全部执行, 要么全部不执行;
- 概念: ACID
 - 原子性 (Atomicity):
 - 事务是数据库的逻辑工作单位;
 - 一致性 (Consistency):
 - 事务的执行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;
 - 一致性状态是指数据库中只包含成功事务的提交结果;
 - 隔离性 (Isolation):
 - 一个事务的执行不能被其他事务干扰;
 - 持久性 (Durability):
 - 也称永久性 (Permanence) 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;
- 图示: 事务的机制实现



数据转存

概念: 数据转存

- 概念:
 - 数据转储 (Backup) 是指 DBA 将整个数据库复制到另一个磁盘上保存起来的过程;
 - 备用的数据称为后备副本或后援副本;
 - 数据库遭到破坏后, 可以将后备副本重新装入重装后备副本;

- 只能将数据库恢复到转储时的状态；

- 图示：

		转储状态	
		动态(热备份)	静态(冷备份)
转储方式	海量(完全)	动态海量转储	静态海量转储
	增量	动态增量转储	静态增量转储

日志

基础概念

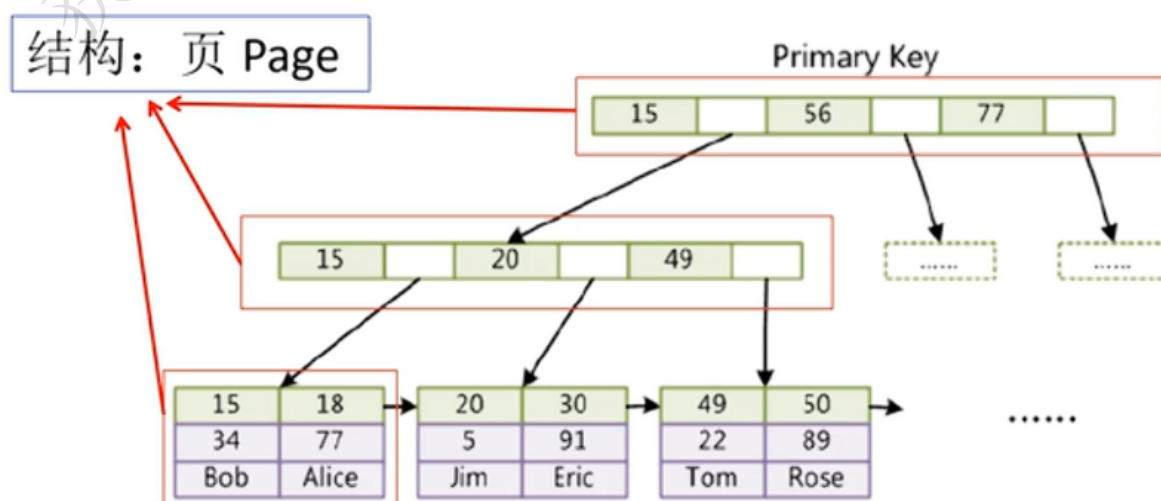
概念： 日志

- 概念：
 - 指用来记录事务对数据库的更新操作的文件；
 - 登记日志文件必须遵循的原则，包括登记的次序严格按并发事务执行的时间次序，且必须先写日志文件、后写数据库；
- 内容：
 - 逻辑日志内容包括：各事务的标记、各事务的开始和结束标记、各事务的所有更新操作；
- 概念：二进制日志
 - 具有归档性质的逻辑日志，记录提交的事务对数据库的更新操作，主要作用是数据库恢复和主从数据库复制；

redo 日志

概念： redo 日志

- 引入：
 - MySQL 的 B+树 结构；



- 为何使用 redo：数据写入磁盘时可能会有脏页，脏页大小为 k，即使修个 1 B 也要写入 16K；
- 另外，数据页为随机访问，保存多个页时效率低；
- 概念：
 - 使用 redo 日志来保证事务的持久性；
 - 因此 redo 日志保存事务提交但尚未写入硬盘的数据；
 - 分 block 保存，每个 block 包含表号、页号、偏移文件量、修改的值，所以是物理日志；

undo 日志

概念：undo 日志

- 概念：
 - 是一种用于撤销回退的逻辑日志；
 - 简单理解为当 delete 一条记录时，undo log 中会记录条对应的 insert 记录，反之亦然；
 - 当 update 一条记录时，它记录一条对应相反的 update 记录；
 - 当执行 rollback 时，就可以从 undo log 中的逻辑记录读取到相应内容并进行回滚，保证事务的原子性；
- 概念：Undo 日志的产生与销毁
 - 在事务 DML 增删改操作开始前产生，DQL 查询操作不产生 undo 日志；
 - 插入事务提交后，这条 insert undo log 直接删除，更新和删除事务在提交时，并不会立刻删除；
 - 将该事务对应的 undolog 放入到删除列表中，后面会通过后台线程 purge thread 进行回收处理；

事务故障

基础概念

概念：事务故障

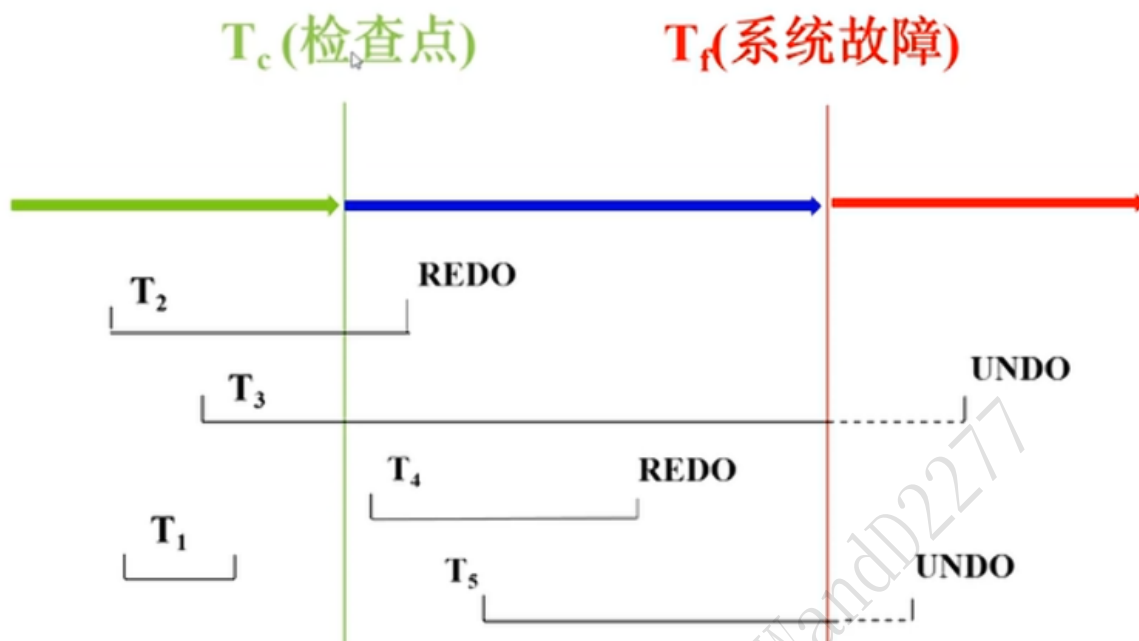
- 概念：
 - 某个事务在运行过程中由于种种原因运行至正常终止点就夭折了；
 - 运算溢出、并发事务发生死锁而被选中撤销该事务、违反了某些完整性限制等；
- 策略：
 - 使用 Undo 日志回滚；

软故障

概念：软故障

- 概念：
 - 又称为系统故障，是指造成系统停止运转的任何事件，使得系统要重新启动；
- 策略：

- 通过 **Undo** 日志，回滚未提交的事务；
- 已修改数据则通过 **Redo** 日志，重做已提交的事务，但未写入硬盘；
- 举例：

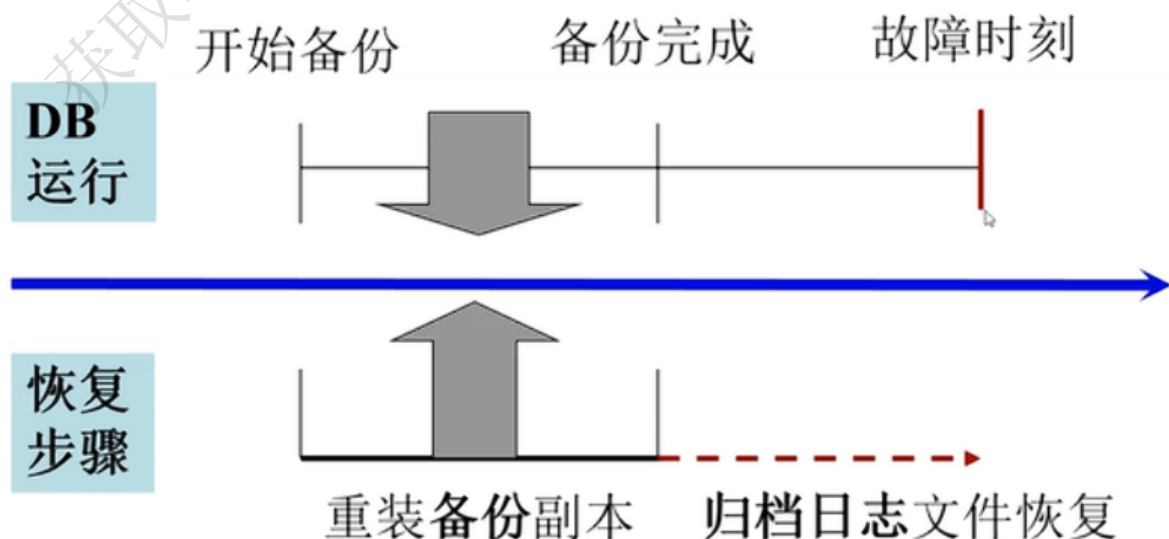


-
- 检查点和系统故障之间的，可以回滚、重做；
- 而如果是系统故障之后的，则直接撤销；

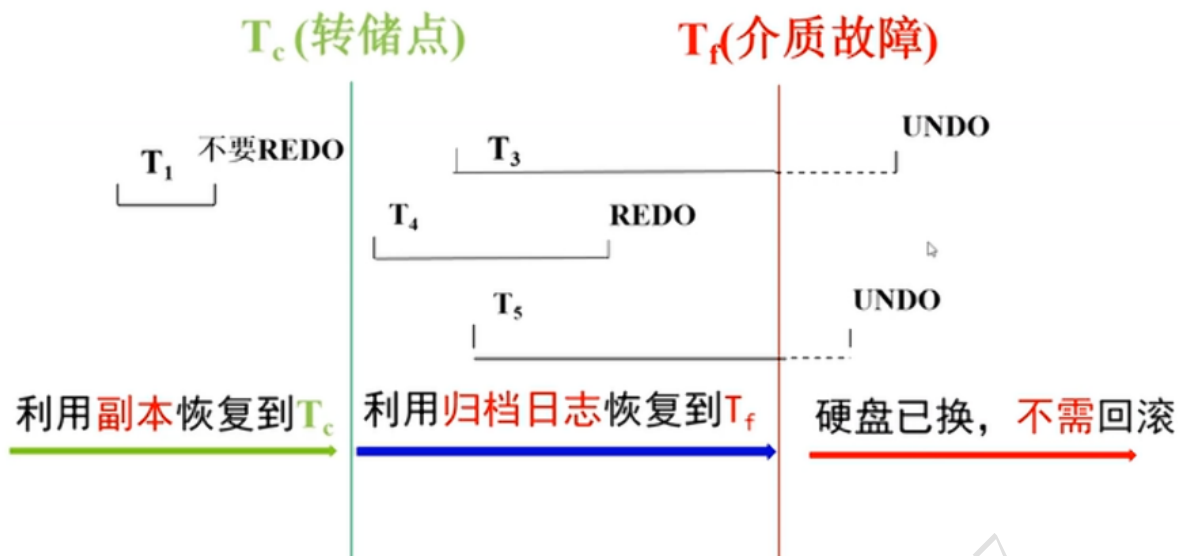
硬故障

概念：硬故障

- 概念：
 - 又称为介质故障，指外存故障；
- 策略：
 - 通过备份恢复到转储点；
 - 通过归档日志恢复到故障点；
- 图示：



- 分析：



获取笔记持续更新可加微信: WandD2277