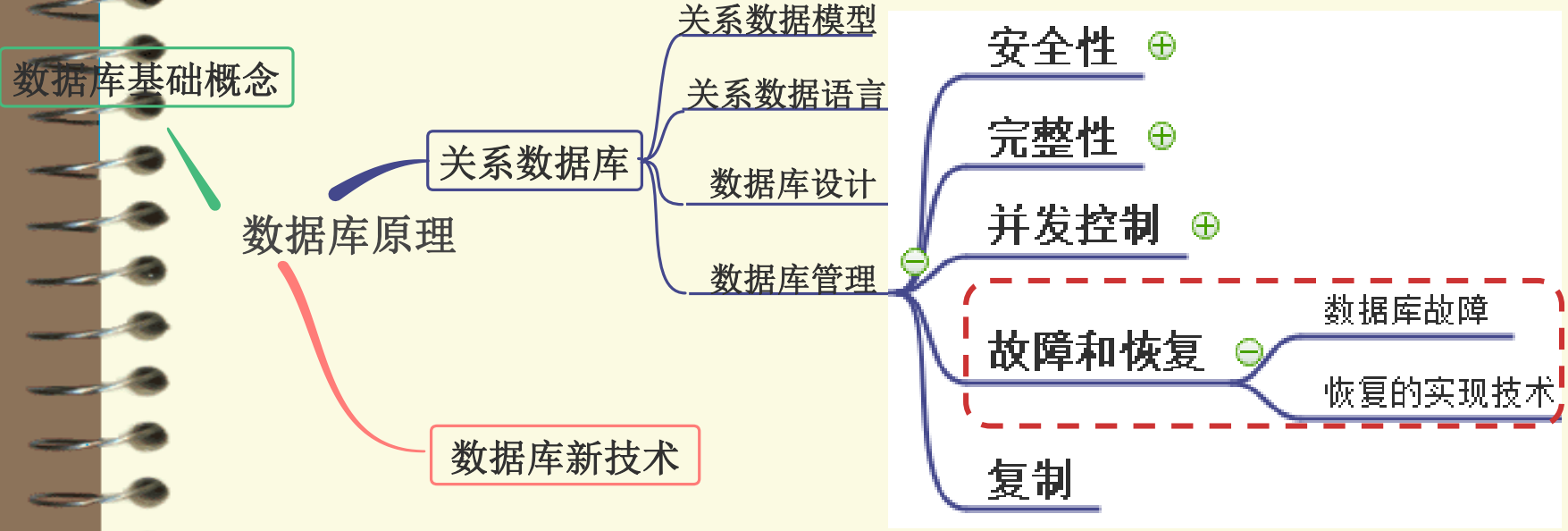
数据库原理

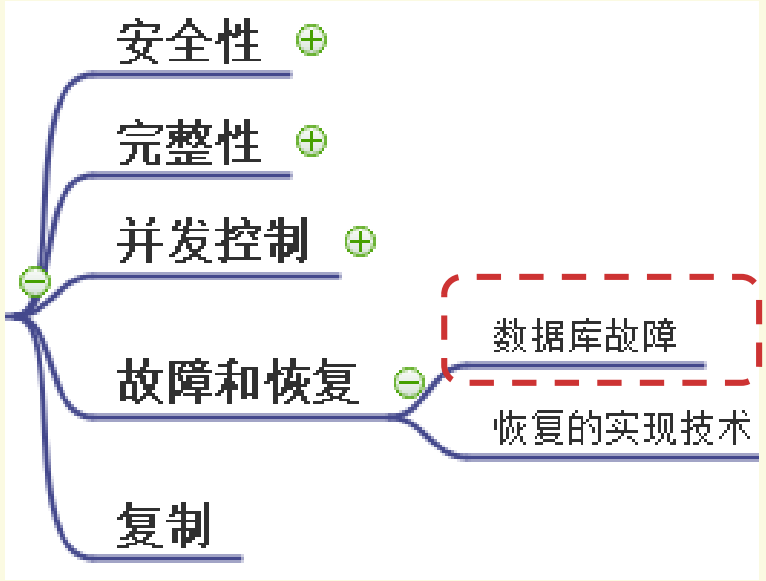
第7章 数据库管理

辽东学院 鲁 琴

本节要点


数据库基础概念
数据库原理
关系数据库
关系数据模型
关系数据语言
数据库设计
数据库管理
安全性
完整性
并发控制
故障和恢复
数据库故障
恢复的实现技术
复制
数据库新技术

4 故障和恢复



4.1 数据库故障

- 故障是不可避免的
  - 计算机硬件故障
  - 系统软件和应用软件的错误
  - 操作员的失误
  - 恶意的破坏
- 故障的影响
  - 轻则造成运行事务非正常中断，影响数据库中数据的正确性
  - 重则破坏数据库，使数据库中数据部分或全部丢失

数据库的可恢复性（Recovery）

- 数据库的可恢复性
  - 系统能把数据库从被破坏、不正确的状态恢复到最近一个正确的状态
- DBMS对故障的对策
  - DBMS恢复子系统，用来保证各种故障发生后，能把数据库中的数据从错误状态恢复到某种逻辑一致的状态
  - 即保证各个事务中的操作要么全部完成，要么全部不做
- 数据库系统所采用的恢复技术是否行之有效是衡量系统性能优劣的重要指标

数据库故障的类型

- 事务故障
- 系统故障
- 介质故障

一、事务故障

- 事务是数据库的基本工作单位
- 什么是事务故障
  - 某个事务在运行过程中由于种种原因未运行至正常终止点就夭折了
- 事务故障的常见原因
  - 输入数据有误
  - 运算溢出
  - 违反了某些完整性限制
  - 某些应用程序出错
  - 并行事务发生死锁

二、系统故障

- 什么是系统故障
  - 由于某种原因造成整个系统的正常运行突然停止，致使所有正在运行的事务都以非正常方式终止。
- 发生系统故障时，内存中数据库缓冲区的信息全部丢失
- 但存储在外部存储设备上的数据未受影响

系统故障的常见原因

- 操作系统或DBMS代码错误
- 操作员操作失误
- 特定类型的硬件错误（如CPU故障）
- 突然停电

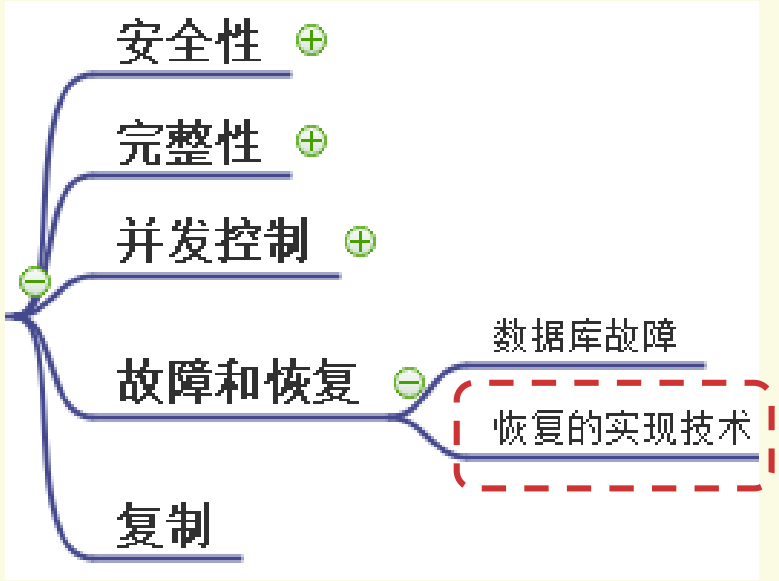
三、介质故障

- 什么是介质故障
  - 硬件故障使存储在外存中的数据部分丢失或全部丢失
  - 介质故障比前两类故障的可能性小得多，但破坏性最大
- 介质故障的常见原因
  - 硬件故障
    - 磁盘损坏
    - 磁头碰撞
  - 操作系统的某种潜在错误
  - 瞬时强磁场干扰

数据库故障小结

- 数据库系统中各类故障对数据库的影响
  - 数据库本身被破坏（介质故障）
  - 数据库处于不一致状态
    - 数据库中包含了未完成事务对数据库的修改（事务故障、系统故障）
    - 数据库中丢失了已提交事务对数据库的修改（系统故障）
- 不同类型的故障应采用不同的恢复操作

4 故障和恢复



4.2 恢复的实现技术

- 恢复技术的原理
  - 利用存储在系统其它地方的冗余数据来修复或重建数据库中被破坏的或不正确的数据。
- 恢复的实现技术：复杂

  大型数据库产品，恢复子系统的代码要占全部代码的10%以上
- 恢复机制涉及的关键问题
  - （1）如何建立冗余数据
    - 数据转储
    - 登记日志文件
  - （2）如何利用这些冗余数据实施数据库恢复

4.2 恢复的实现技术

4.2.1 数据转储

4.2.2 登记日志文件

4.2.3 恢复策略

一、什么是转储

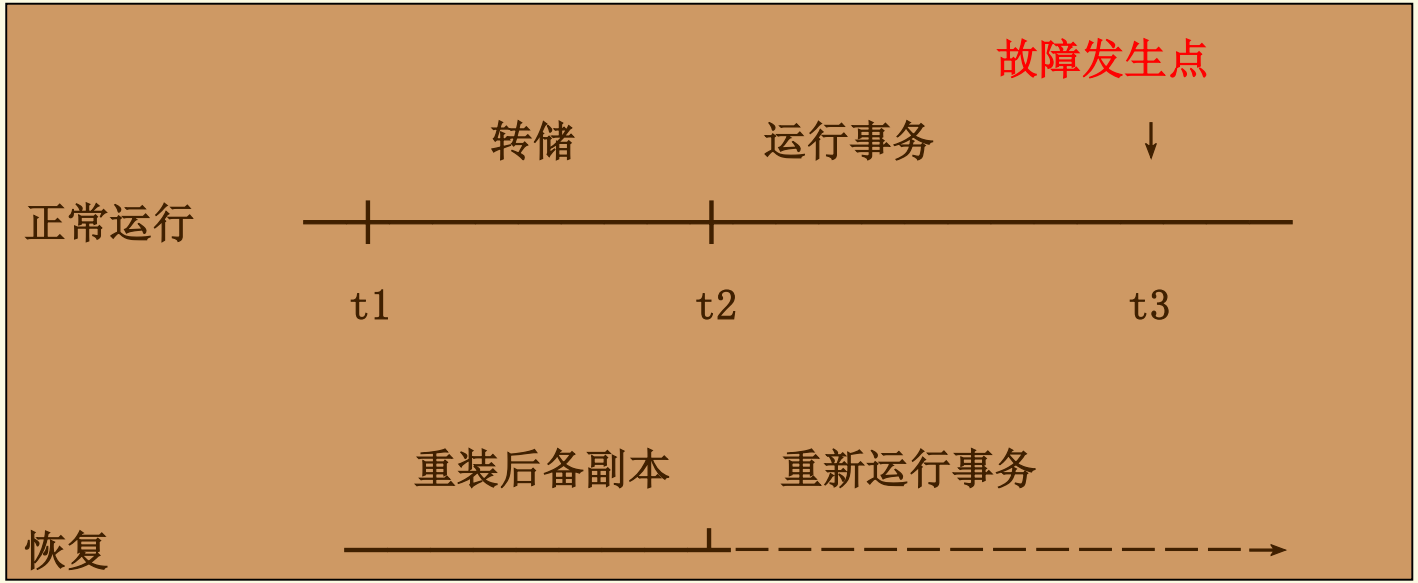
- 转储是指DBA将整个数据库复制到磁带或另一个磁盘上保存起来的过程

- 这些备用的数据文本称为后备副本或后援副本

二、转储的用途

- 供介质故障恢复时使用
  - 一旦系统发生介质故障，数据库遭到破坏，可以将后备副本重新装入，把数据库恢复起来
- 恢复的程度
  - 重装后备副本只能将DB恢复到转储时的状态
  - 要想恢复到故障发生时的状态，必须重新运行自转储以后的所有更新事务

例



三、转储方法

(1) 静态转储与动态转储

(2) 海量转储与增量转储

(3) 转储方法分类

（1）静态转储与动态转储

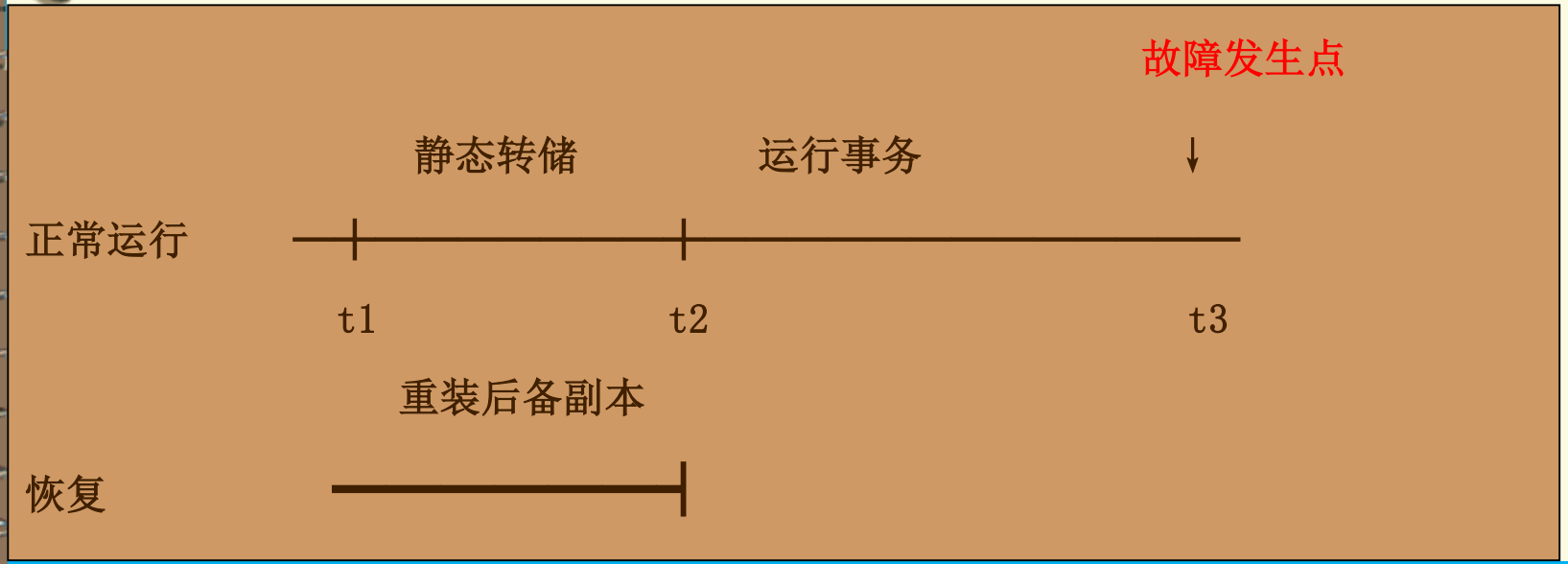
- 静态转储
  - 静态转储是在系统中无运行事务时进行的转储操作
    - 转储操作开始的时刻，数据库处于一致性状态
    - 转储期间不允许（或不存在）对数据库的任何存取、修改活动
  - 静态转储得到的一定是一个数据一致性的副本

静态转储

- 静态转储的优点
  - 实现简单
- 静态转储的缺点
  - 降低了数据库的可用性
    - 转储必须等待用户事务结束才能进行
    - 新的事务必须等待转储结束才能执行

利用静态转储副本进行恢复

- 利用静态转储得到的副本进行故障恢复
  - 只需要把静态转储得到的后备副本装入，就能把数据库恢复到转储时刻的正确状态

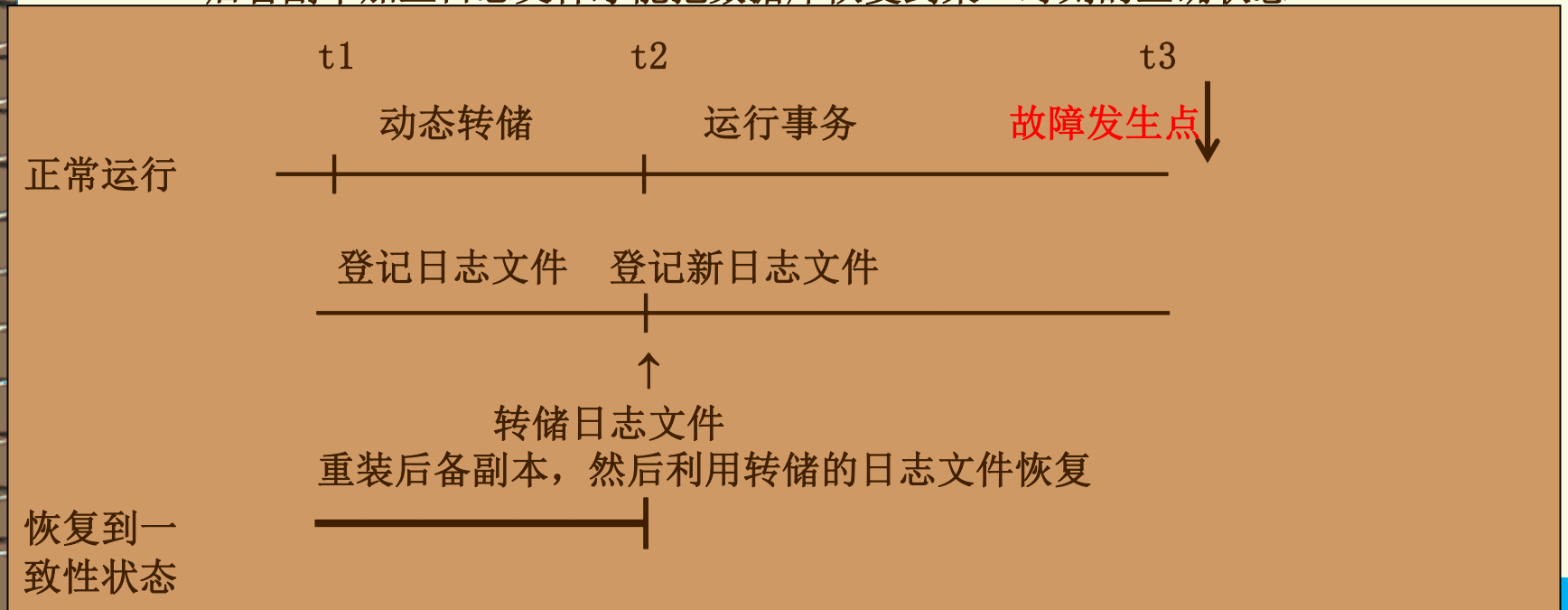


动态转储

- 动态转储
  - 动态转储是指转储操作与用户事务并发进行，转储期间允许对数据库进行存取或修改。
- 动态转储的优点
  - 不用等待正在运行的用户事务结束
  - 不会影响新事务的运行
- 动态转储的缺点
  - 不能保证副本中的数据正确有效

利用动态转储副本进行恢复

- 利用动态转储得到的副本进行故障恢复
  - 需要把动态转储期间各事务对数据库的修改活动登记下来，建立日志文件
  - 后备副本加上日志文件才能把数据库恢复到某一时刻的正确状态



(2) 海量转储与增量转储

- 海量转储
  - 每次转储全部数据库
- 增量转储
  - 只转储上次转储后更新过的数据
- 海量转储与增量转储比较
  - 从恢复角度看，使用海量转储得到的后备副本进行恢复往往更方便
  - 但如果数据库很大，事务处理又十分频繁，则增量转储方式更实用更有效

(3) 转储方法分类

- 转储方法分类

转储状态 / 转储方式	动态转储	静态转储
海量转储	动态海量转储	静态海量转储
增量转储	动态增量转储	静态增量转储

转储策略

- 从恢复方便角度看，应经常进行数据转储，制作后备副本
- 但转储又是十分耗费时间和资源的，不能频繁进行
- DBA应根据数据库使用情况确定适当的转储周期和转储方法
- 例：
  - 每天晚上进行动态增量转储
  - 每周进行一次动态海量转储
  - 每月进行一次静态海量转储

4.2 恢复的实现技术

4.2.1 数据转储

4.2.2 登记日志文件

4.2.3 恢复策略

4.2.2　登记日志文件

一、日志文件的内容

二、日志文件的用途

三、登记日志文件的原则

一、日志文件的内容

(1) 什么是日志文件

日志文件(log)是用来记录事务对数据库的更新操作的文件

(2) 日志文件的格式

以记录为单位的日志文件

以数据块为单位的日志文件

日志文件的内容（续）

(3) 日志文件内容

- **各个事务的开始标记(BEGIN TRANSACTION)**
- **各个事务的结束标记(COMMIT或ROLLBACK)**
- **各个事务的所有更新操作**

作为日志文件中的一个日志记录(log record)

日志文件的内容（续）

(4) 基于记录的日志文件

每条日志记录的内容

事务标识（标明是那个事务）

操作类型（插入、删除或修改）

操作对象

更新前数据的旧值（对插入操作而言，此项为空值）

更新后数据的新值（对删除操作而言,此项为空值）

日志文件的内容（续）

(5) 基于数据块的日志文件

- 每条日志记录的内容
  - 事务标识（标明是那个事务）
  - 更新前数据所在的整个数据块的值（对插入操作而言，此项为空值）
  - 更新后整个数据块的值（对删除操作而言，此项为空值）

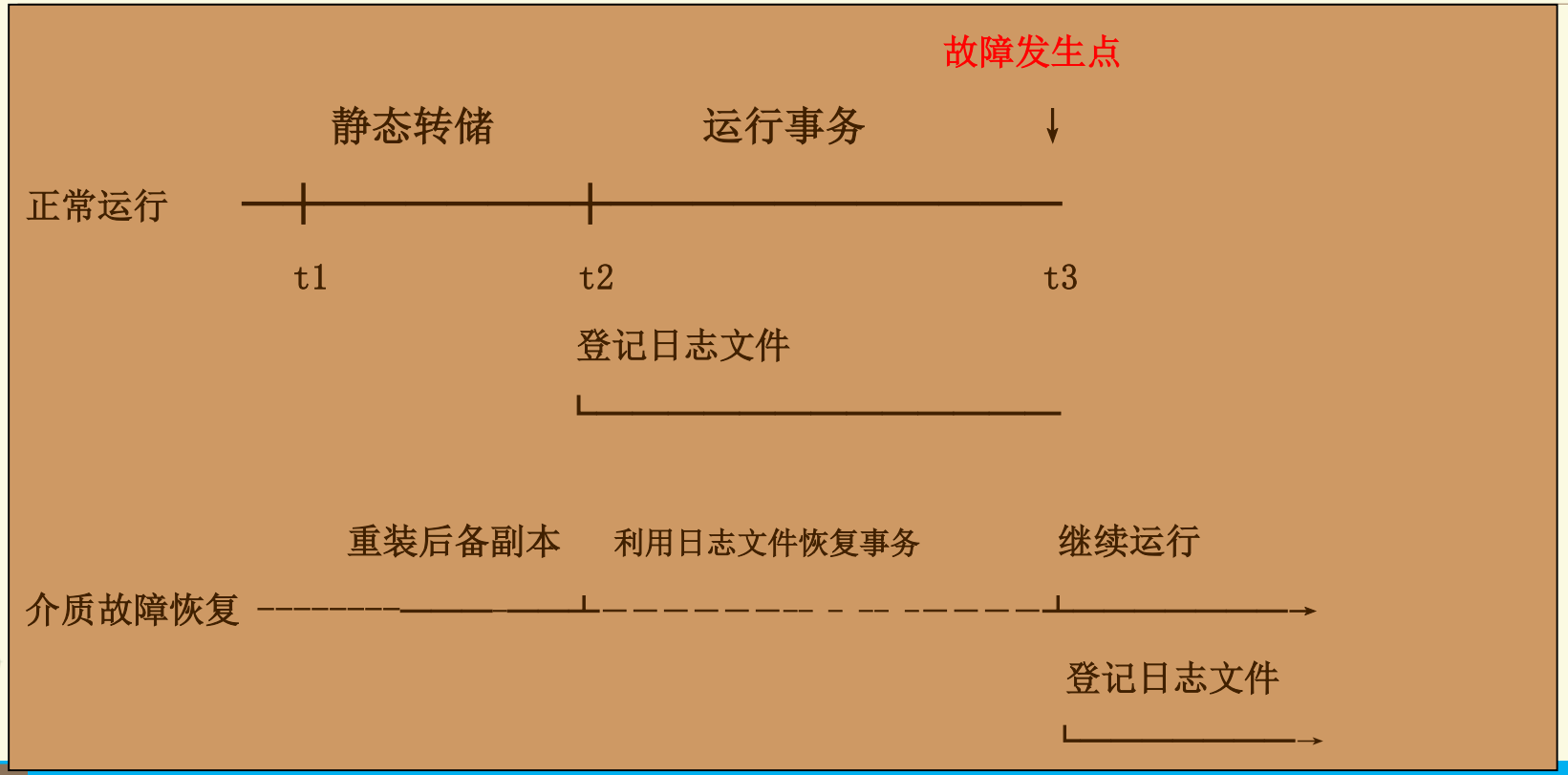
二、日志文件的用途

- 进行事务故障恢复
- 进行系统故障恢复
- 协助后备副本进行介质故障恢复

与静态转储后备副本配合进行介质故障恢复

- 静态转储的数据已是一致性的数据
- 如果静态转储完成后，仍能定期转储日志文件，则在出现介质故障重装数据副本后，可以利用这些日志文件副本对已完成的事务进行重做处理
- 这样不必重新运行那些已完成的事务程序就可把数据库恢复到故障前某一时刻的正确状态

利用日志文件恢复事务


故障发生点
静态转储
运行事务
正常运行
t1
t2
t3
登记日志文件
重装后备副本
利用日志文件恢复事务
继续运行
介质故障恢复
登记日志文件

与动态转储后备副本配合使用进行介质故障恢复

- 动态转储机制在转储数据库时，必须同时转储同一时间点的日志文件，后备副本与该日志文件结合起来才能将数据库恢复到一致性状态。
- 与静态转储一样，如果动态转储完成后，仍能定期转储日志文件，则在做介质故障恢复时，可以利用这些日志文件副本进一步恢复数据库，避免重新运行事务程序。

三、登记日志文件的原则

- 为保证数据库是可恢复的
  - 登记的次序严格按并行事务执行的时间次序
  - 必须先写日志文件，后写数据库
    - 写日志文件操作：把表示这个修改的日志记录写到日志文件
    - 写数据库操作：把对数据的修改写到数据库中

为什么要先写日志文件

- 写数据库和写日志文件是两个不同的操作
- 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
- 如果先写了数据库修改，而在日志文件中没有登记下这个修改，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，按日志文件恢复时只不过是多执行一次不必要的UNDO操作，并不会影响数据库的正确性

4.2 恢复的实现技术

4.2.1 数据转储

4.2.2 登记日志文件

4.2.3 恢复策略

4.2.3 恢复策略

(1) 事务故障的恢复

(2) 系统故障的恢复

(3) 介质故障的恢复

(1) 事务故障的恢复

- 事务故障是指事务在运行至正常终止点前被中止
- 恢复方法
  - 由恢复子系统应利用日志文件撤消（UNDO）此事务已对数据库进行的修改
- 事务故障的恢复由系统自动完成，不需要用户干预

恢复步骤

1. 反向扫描文件日志(即从最后向前扫描日志文件)，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
   - 如果记录中是插入操作，则相当于做删除操作(因为此时“更新前的值”为空)
   - 若记录中是删除操作，则相当于做插入操作(因为此时“更新后的值”为空)
   - 若是修改操作，则相当于用修改前值代替修改后值
3. 继续反向扫描日志文件，查找该事务的其他更新操作，并做同样处理
4. 如此处理下去，直至读到此事务的开始标记，事务故障恢复就完成了

(2) 系统故障的恢复

- 系统故障造成数据库不一致状态的原因
  - 一些未完成事务对数据库的更新已写入数据库
  - 一些已提交事务对数据库的更新还留在缓冲区没来得及写入数据库
- 恢复方法
  - 1. 撤消故障发生时未完成的事务
  - 2. 重做已完成的事务
- 系统故障的恢复由系统在重新启动时自动完成，不需要用户干预

恢复步骤

1. 正向扫描日志文件（即从头扫描日志文件）
   - 找出在故障发生前已经提交的事务,将事务标识记入重做队列
   - 同时找出故障发生时尚未完成的事务,将事务标识记入撤消队列
2. 对撤消队列中的各个事务进行撤消(UNDO)处理
   - 反向扫描日志文件，对每个UNDO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
3. 对重做队列中的各个事务进行重做(REDO)处理
   - 正向扫描日志文件，对每个REDO事务重新执行登记的操作。即将日志记录中“更新后的值”写入数据库

（3）介质故障的恢复

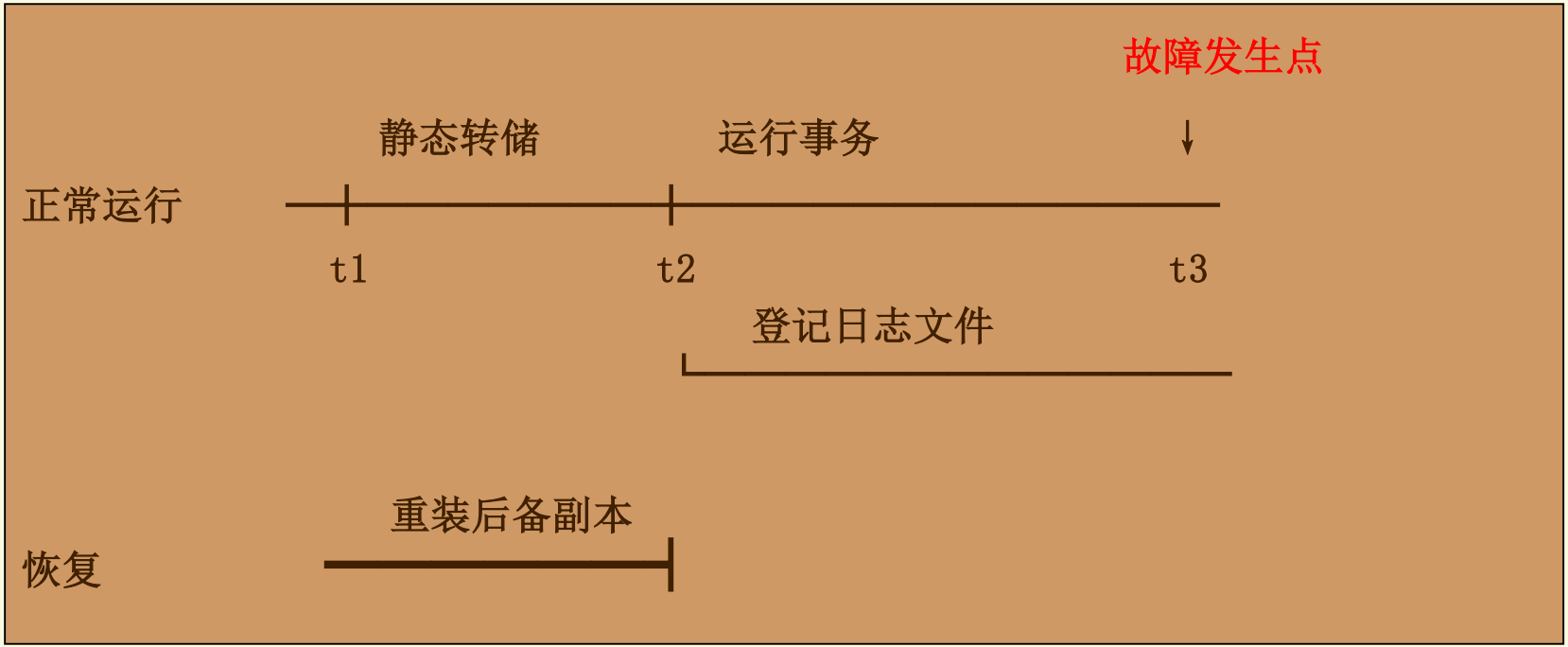
- 发生介质故障后，磁盘上的物理数据和日志文件被破坏，这是最严重的一种故障
- 恢复方法
  - 1. 重装数据库，使数据库恢复到一致性状态
  - 2. 重做已完成的事务

恢复步骤

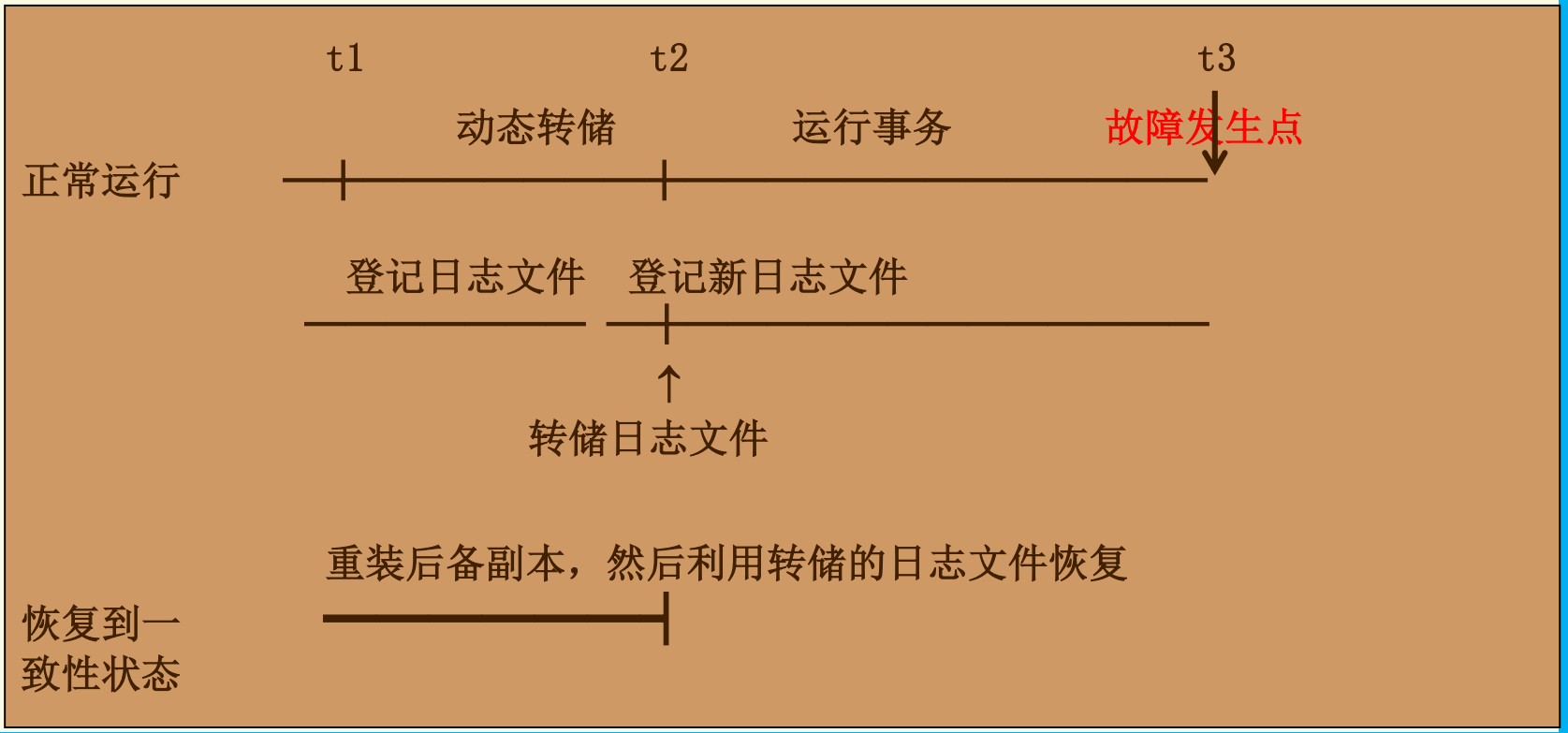
1. 装入最新的后备数据库副本，使数据库恢复到最近一次转储时的一致性状态。

- 对于静态转储的数据库副本，装入后数据库即处于一致性状态
- 对于动态转储的数据库副本，还须同时装入转储时刻的日志文件副本，利用与恢复系统故障相同的方法（即REDO+UNDO），才能将数据库恢复到一致性状态。

利用静态转储副本将数据库恢复到一致性状态



利用动态转储副本将数据库恢复到一致性状态



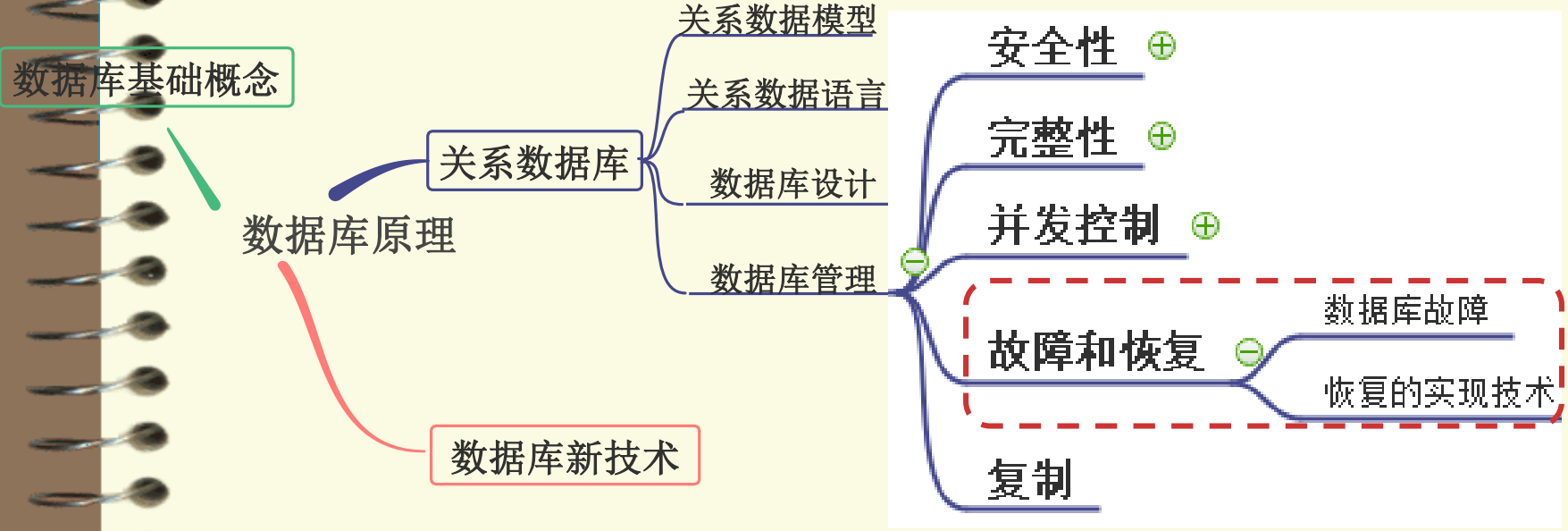
介质故障的恢复（续）

2. 装入有关的日志文件副本，重做已完成的事务。
   - 首先扫描日志文件，找出故障发生时已提交的事务的标识，将其记入重做队列。
   - 然后正向扫描日志文件，对重做队列中的所有事务进行重做处理。即将日志记录中“更新后的值”写入数据库。

介质故障的恢复（续）

- 介质故障的恢复需要DBA介入
  - DBA的工作
    - 重装最近转储的数据库副本和有关的各日志文件副本
    - 执行系统提供的恢复命令
  - 具体的恢复操作仍由DBMS完成

本节小结


数据库基础概念
数据库原理
数据库新技术
关系数据库
关系数据模型
关系数据语言
数据库设计
数据库管理
安全性
完整性
并发控制
故障和恢复
数据库故障
恢复的实现技术
复制

小结

- 如果数据库只包含成功事务提交的结果，就说数据库处于一致性状态
- 保证数据一致性是对数据库的最基本的要求。
- 事务是数据库的逻辑工作单位
  - DBMS保证系统中一切事务的原子性、一致性、隔离性和持续性

小结（续）

- **DBMS必须对事务故障、系统故障和介质故障进行恢复**
- 恢复中最经常使用的技术
  - 数据库转储
  - 登记日志文件
- 恢复的基本原理
  - 利用存储在后备副本、日志文件和数据库镜像中的冗余数据来重建数据库

小结（续）

- 常用恢复技术
  - 事务故障的恢复
    - **UNDO**
  - 系统故障的恢复
    - **UNDO + REDO**
  - 介质故障的恢复
    - 重装备份并恢复到一致性状态 + **REDO**