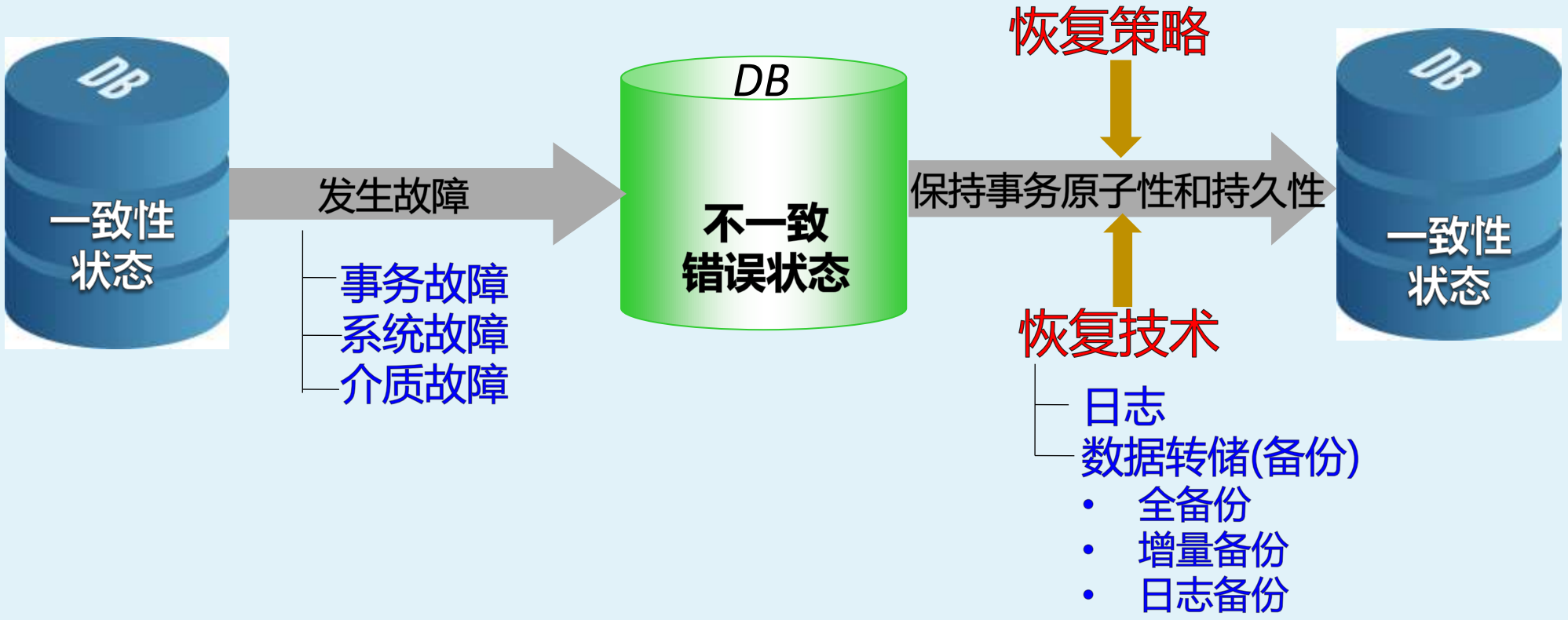
恢复策略

引言


DB
一致性
状态
发生故障
事务故障
系统故障
介质故障
DB
不一致
错误状态
恢复策略
保持事务原子性和持久性
恢复技术
日志
数据转储(备份)
全备份
增量备份
日志备份
DB
一致性
状态

引言

不一致错误

- 夭折的事务的部分执行结果已对数据库进行了更新。

  破坏事务的原子性

- 已提交的事务对数据库的更新结果有一部分甚至全部还在缓冲区中，尚未写回到磁盘上的数据库中。

  破坏事务的持久性

- 已提交的事务对数据库的更新结果不能持久地保存在磁盘上。

  破坏事务的持久性

讲授内容

1. 撤销事务(UNDO)
2. 重做事务(REDO)
3. 数据库恢复
   - 事务故障
   - 系统故障
   - 介质故障

撤销事务(UNDO)

不一致错误

- 夭折的事务的部分执行结果已对数据库进行了更新。

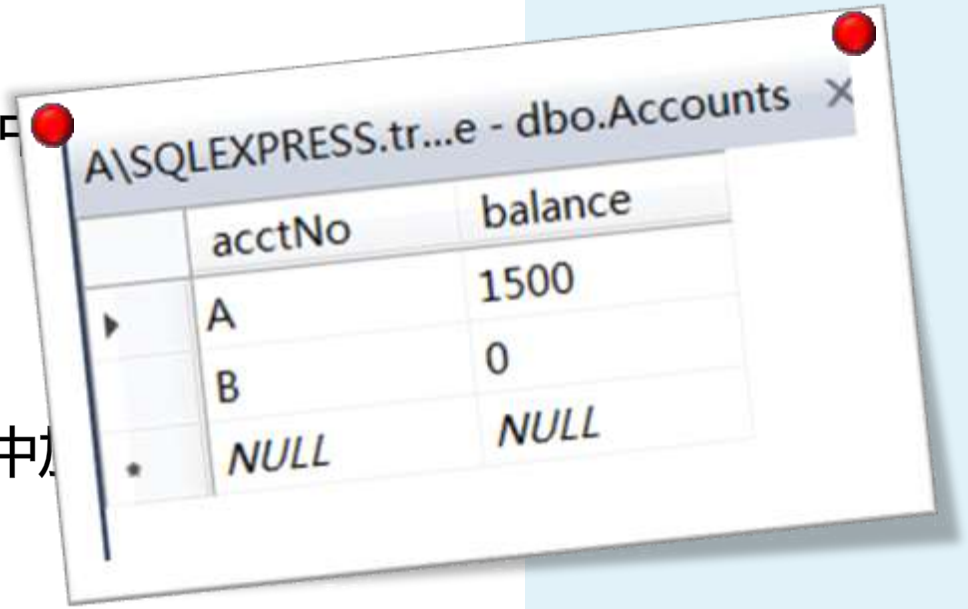
UNDO操作

撤销事务(UNDO)

事务夭折：从账户A转账1000元到账户B

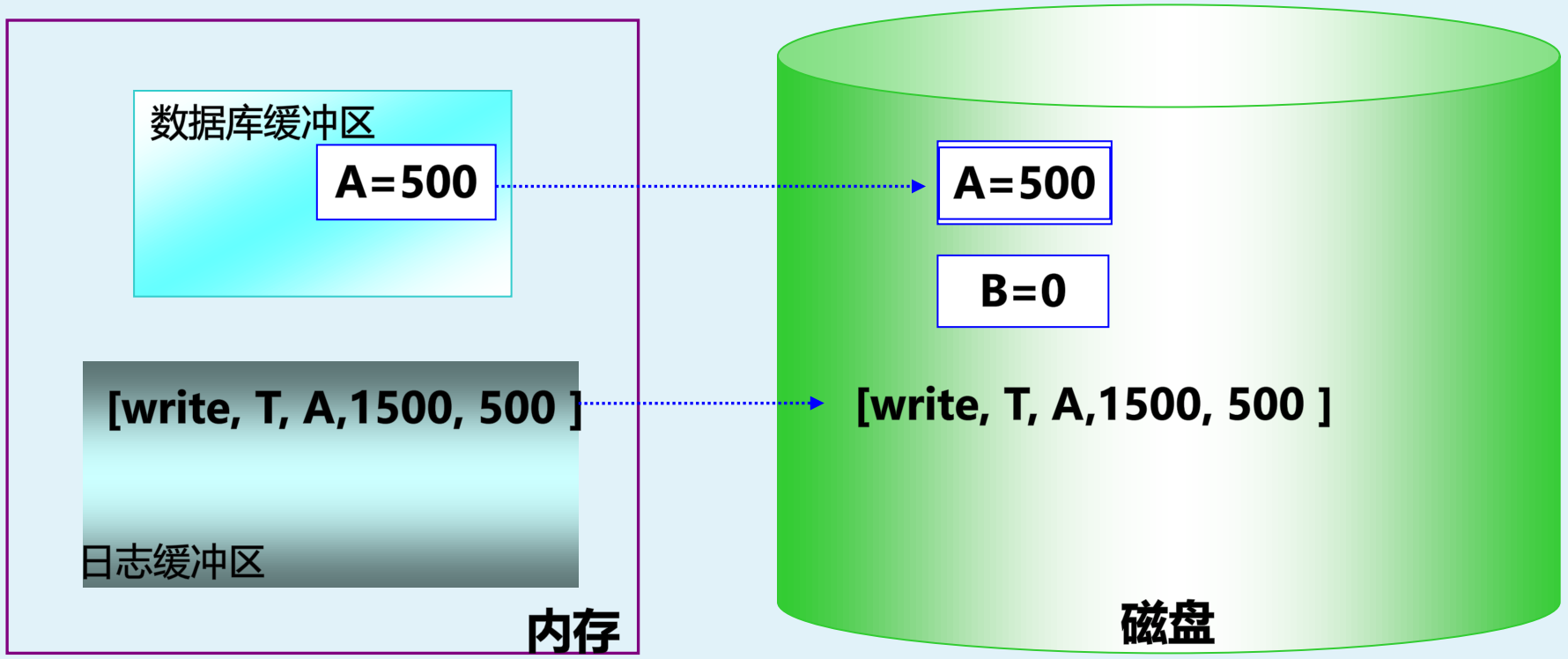
```
BEGIN TRANSACTION
  Read(A, t1)
    t1 := t1 -1000
  Write(A, t1)          /*从账户A中
  Read(B, t2)
    t2 := t2 +1000
  Write(B, t2)          /*向账户B中
COMMIT
```

故障 →



撤销事务(UNDO)

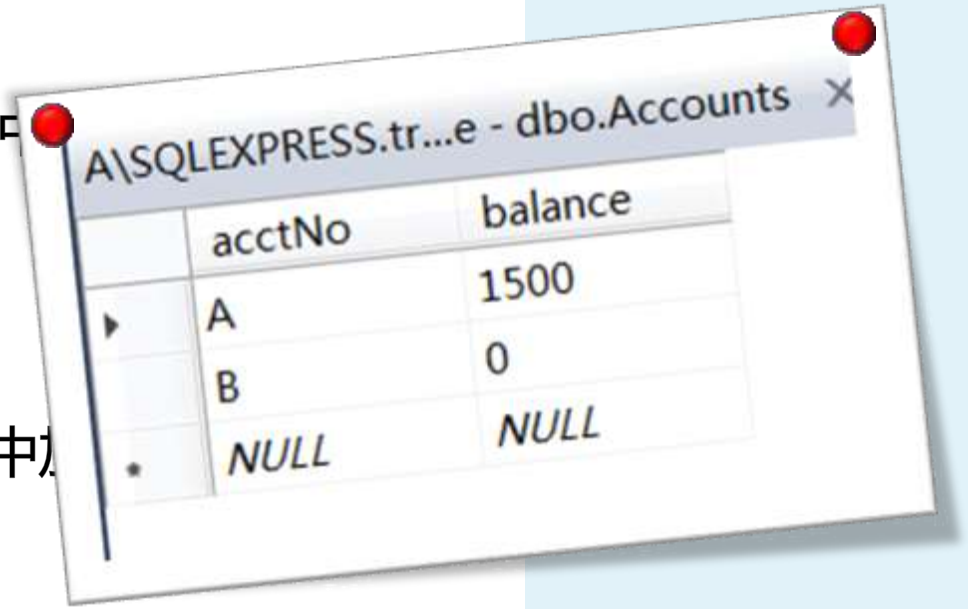
事务夭折：从账户A转账1000元到账户B



撤销事务(UNDO)

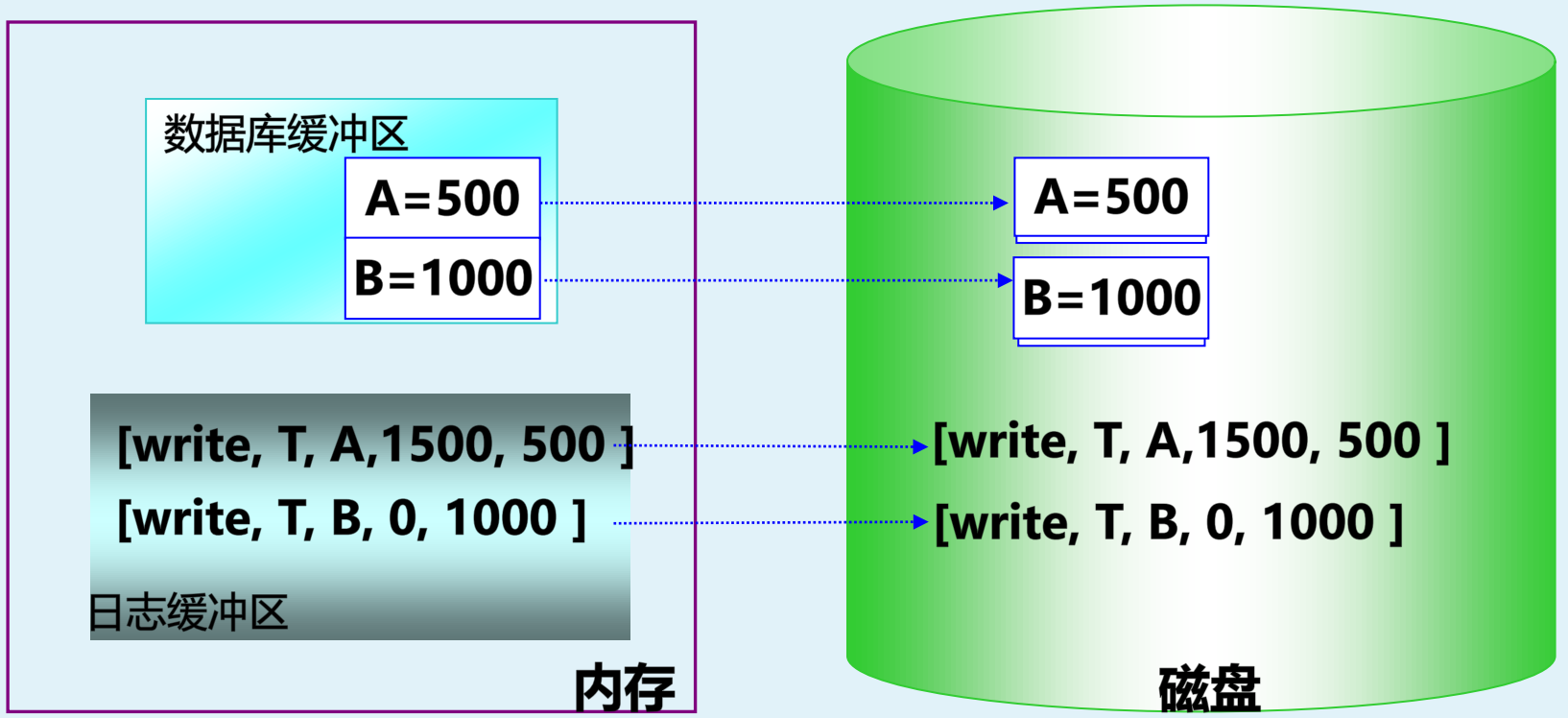
事务夭折：从账户A转账1000元到账户B

```
BEGIN TRANSACTION
  Read(A, t1)
    t1 := t1 -1000
  Write(A, t1)          /*从账户A中
  Read(B, t2)
    t2 := t2 +1000
  Write(B, t2)          /*向账户B中
故障 ──→
COMMIT
```



撤销事务(UNDO)

事务夭折：从账户A转账1000元到账户B



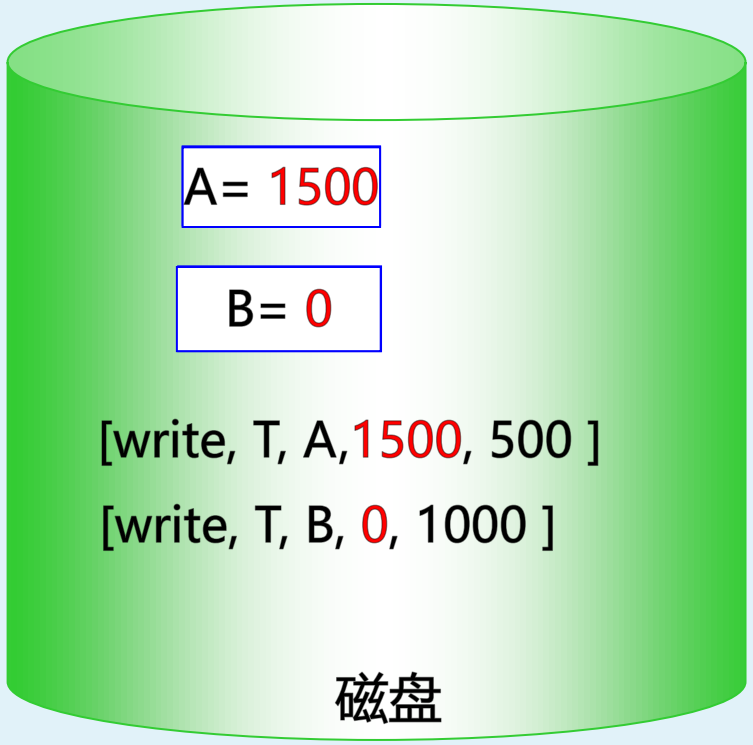
撤销事务(UNDO)

- 利用日志撤销事务对数据库的更新。
- 保持事务的原子性。
- 事务以ROLLBACK方式结束。

将更新前的旧值写回数据库

撤销事务(UNDO)

UNDO事务：从账户A转账1000元到账户B



重做事务(REDO)

不一致错误

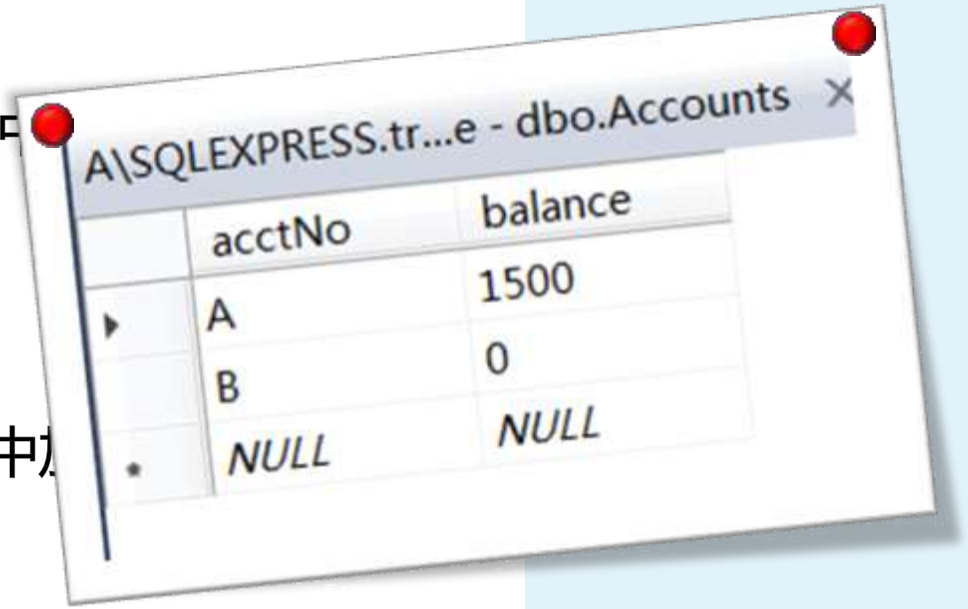
- 已提交的事务对数据库的更新结果可能有一部分甚至全部还在缓冲区中，尚未写回到磁盘上的数据库中。

REDO操作

重做事务(REDO)

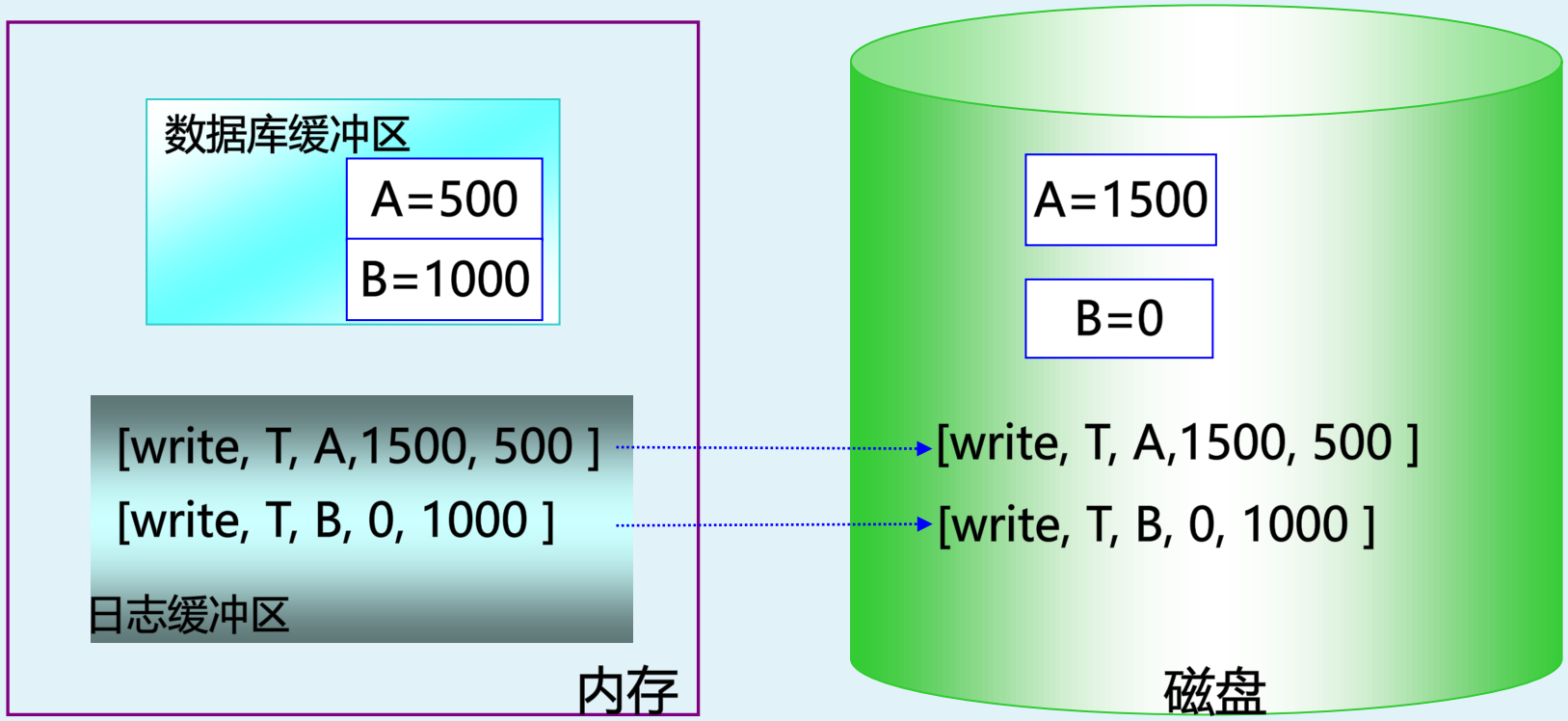
事务提交：从账户A转账1000元到账户B

```
BEGIN TRANSACTION
  Read(A, t1)
    t1 := t1 -1000
  Write(A, t1)          /*从账户A中
  Read(B, t2)
    t2 := t2 +1000
  Write(B, t2)          /*向账户B中
COMMIT
```



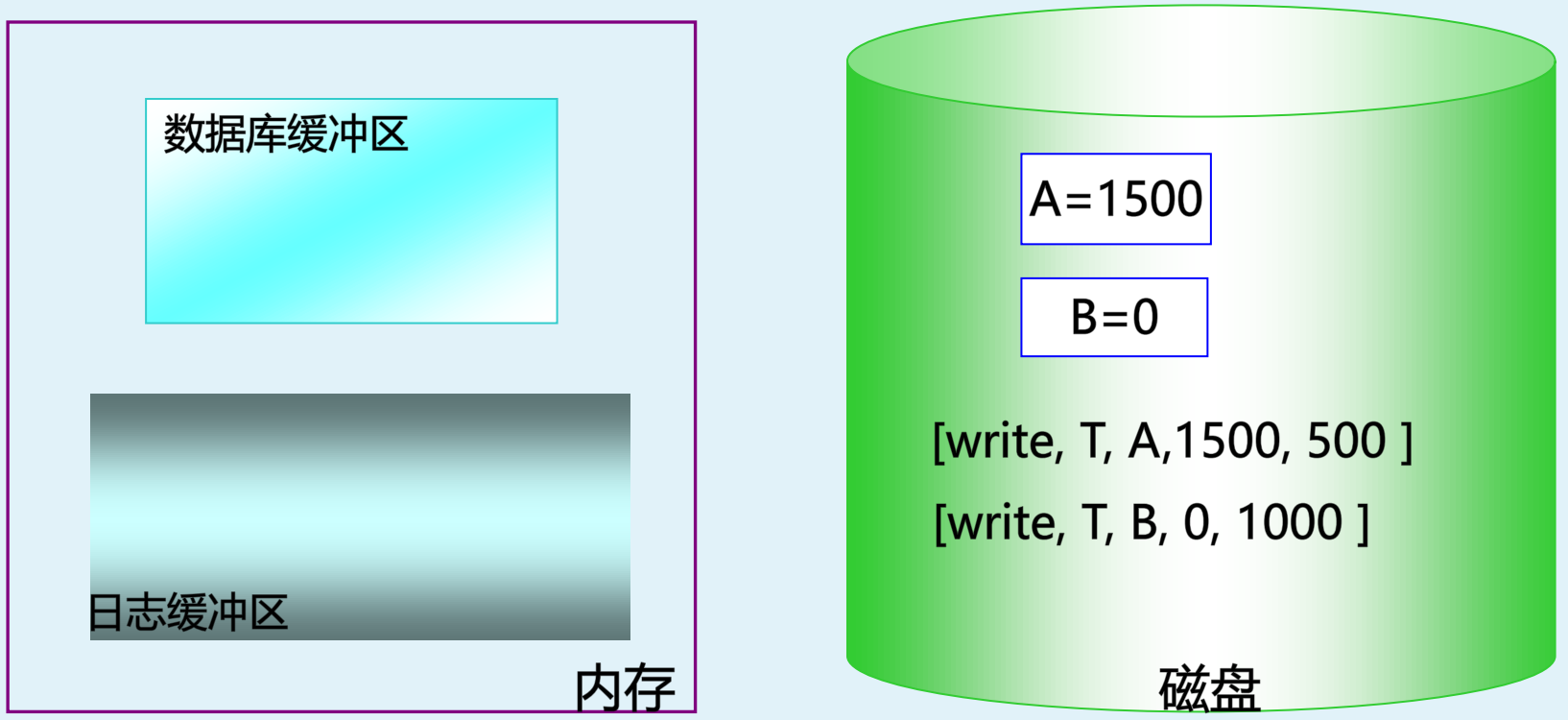
重做事务(REDO)

事务提交：从账户A转账1000元到账户B



重做事务(REDO)

事务提交：从账户A转账1000元到账户B



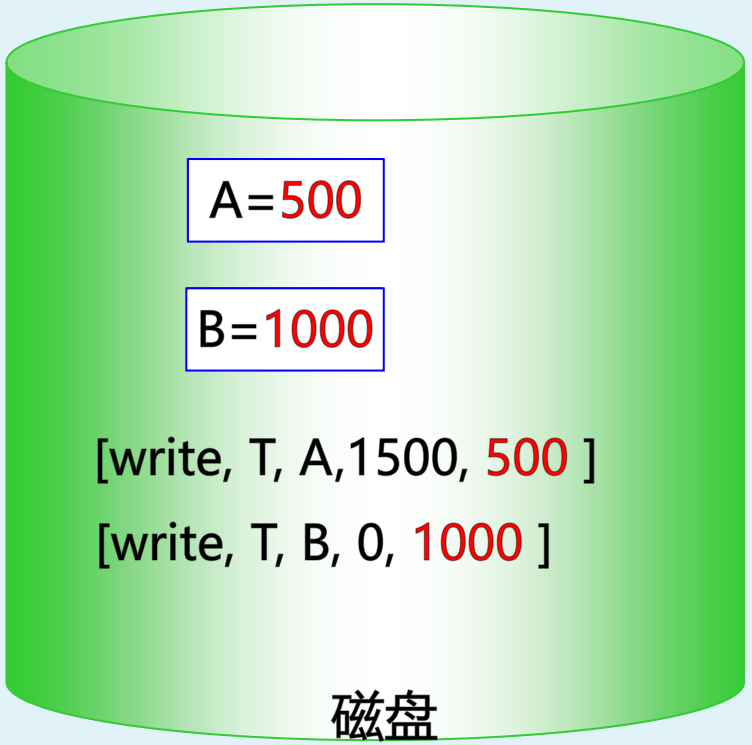
重做事务(REDO)

- 利用日志重做事务对数据库的更新。
- 保持事务的持久性。
- 事务以COMMIT方式结束。

将更新后的新值写入数据库

重做事务(REDO)

REDO事务：从账户A转账1000元到账户B



数据库恢复

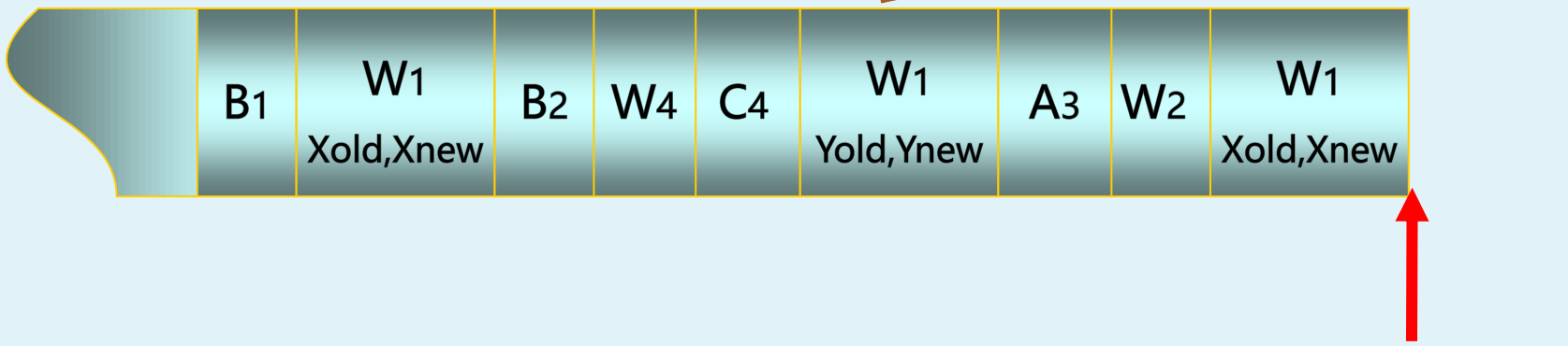
事务故障后的恢复

- 恢复机制回滚夭折事务，利用日志撤消（UNDO）事务已对数据库进行的更新。
- 恢复是由DBMS自动完成的，对用户是透明的。

数据库恢复

事务故障后的恢复

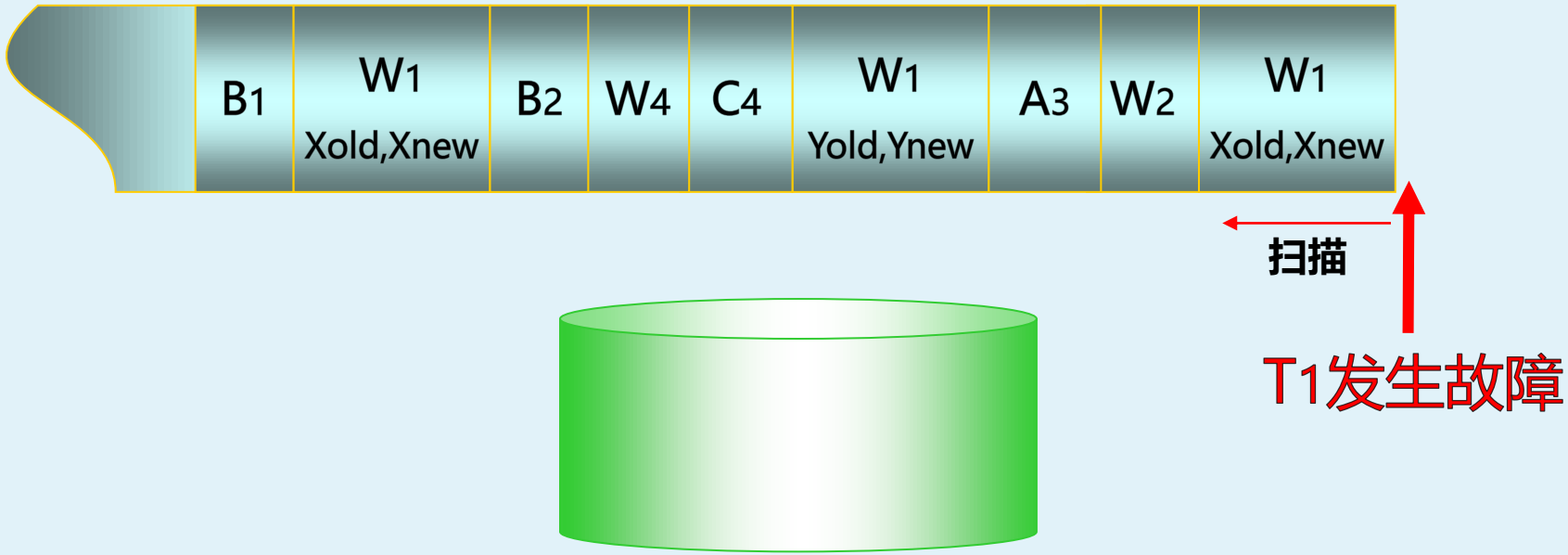
Bi：事务Ti的开始
Ci：事务Ti的提交
Wi：事务Ti的更新
Ai：事务Ti的异常中止



数据库恢复

事务故障后的恢复步骤

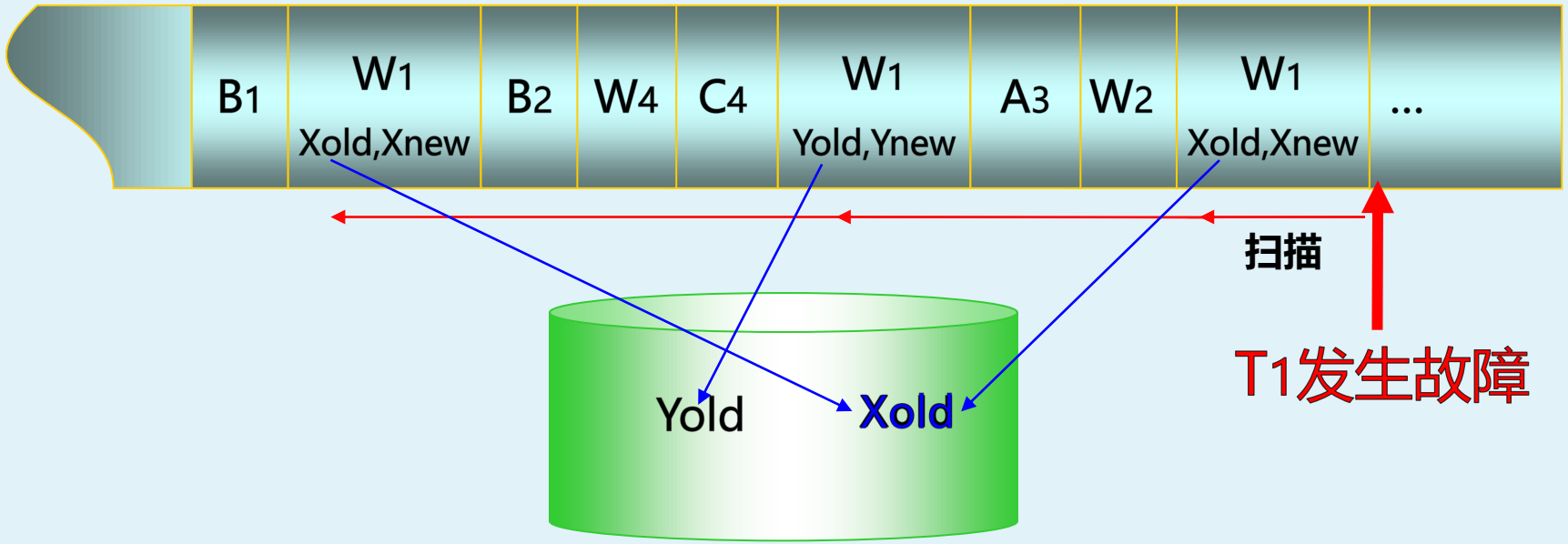
- (1)从日志尾部开始向前反向扫描日志，查找事务更新操作。



数据库恢复

事务故障后的恢复步骤

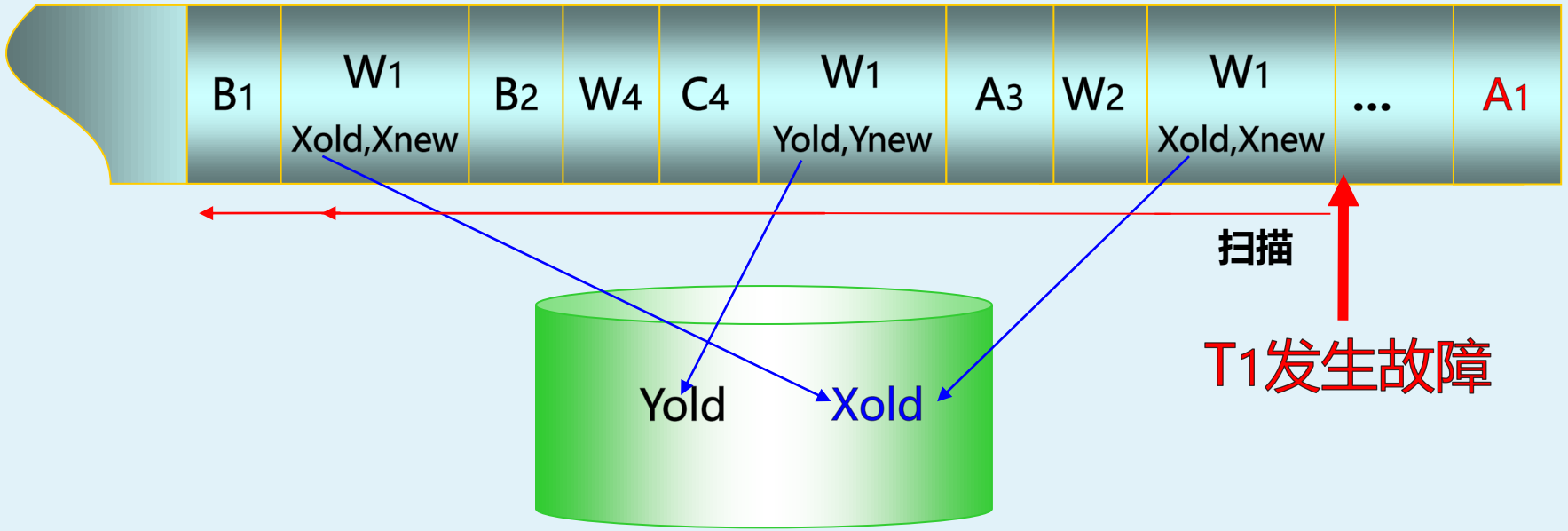
- (2)依次对事务的所有更新操作执行逆操作，将更新前的值写入数据库。



数据库恢复

事务故障后的恢复步骤

- (3)扫描到事务的开始标记，恢复过程终止，在日志中添加该事务的异常中止记录。



数据库恢复

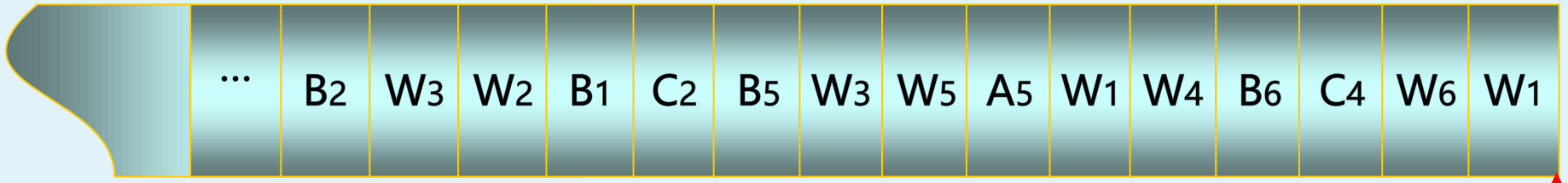
系统故障后的恢复

- 恢复机制在系统重新启动时利用日志撤消(UNDO)非正常终止的事务已对数据库进行的更新。
- 利用日志重做(REDO)所有已提交的事务对数据库的更新。
- 恢复是由DBMS在系统重新启动时自动完成的，不需要用户干预。

数据库恢复

系统故障后的恢复

Bi：事务Ti的开始
Ci：事务Ti的提交
Wi：事务Ti的更新
Ai：事务Ti的异常中止

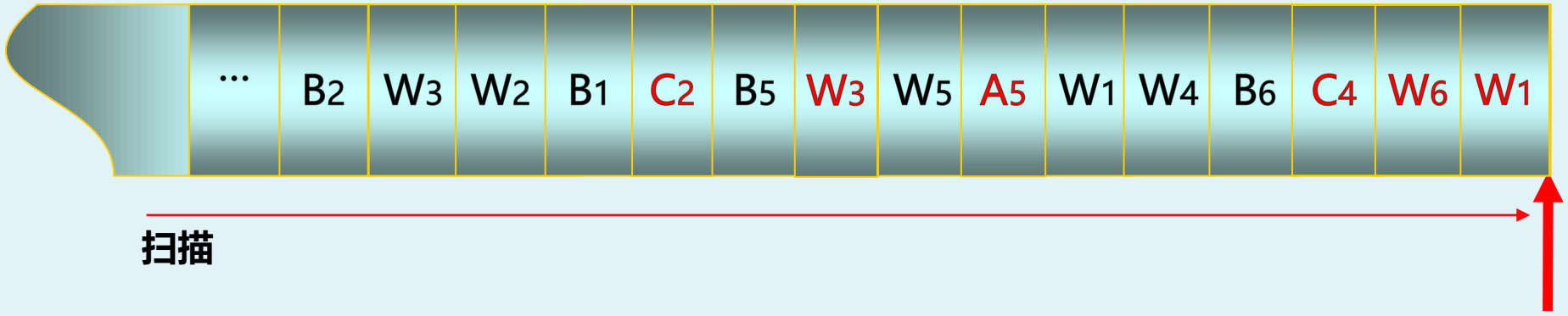


系统崩溃

数据库恢复

系统故障后的恢复步骤

- (1)从日志头部开始向后正向扫描日志，将事务划分为已提交事务和夭折事务。



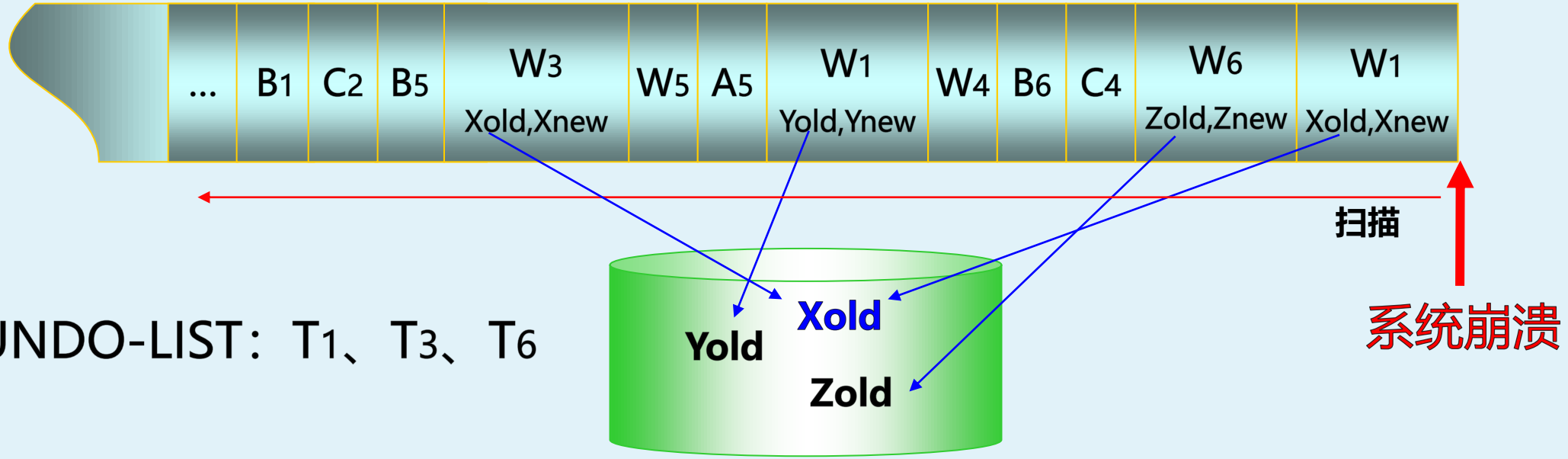
REDO-LIST：T_2 、T_4

UNDO-LIST：T_1、T_3、T_6

数据库恢复

系统故障后的恢复步骤

- (2)从日志尾部向前反向扫描日志，对撤销事务队列(UNDO-LIST)中的事务进行撤销(UNDO)操作。

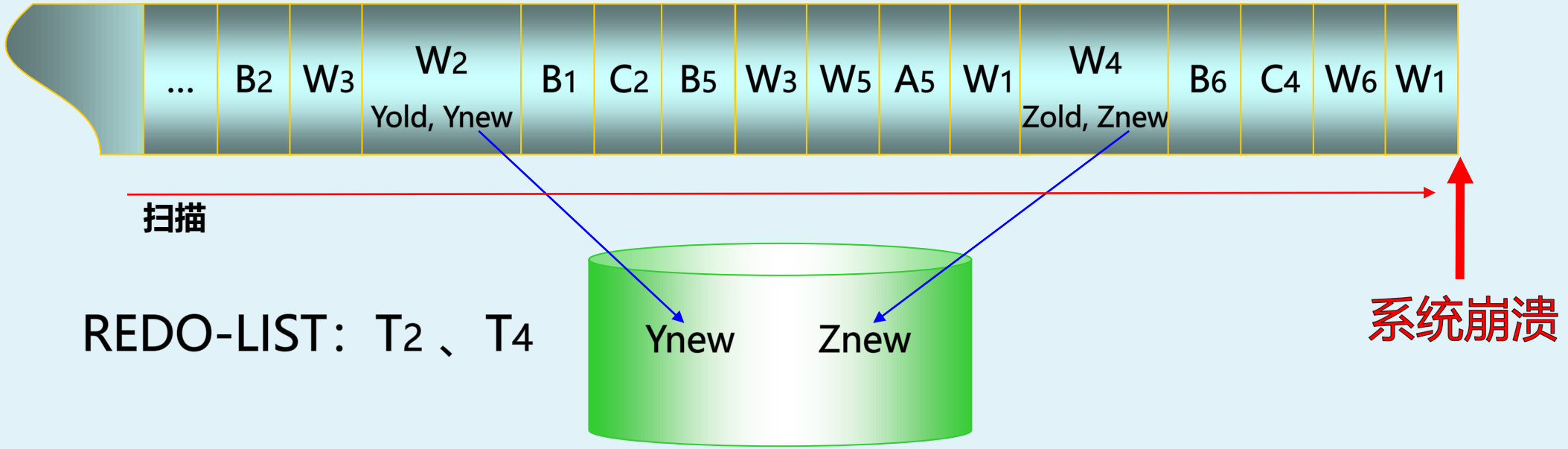


UNDO-LIST：T1、T3、T6

数据库恢复

系统故障后的恢复步骤

- (3)从日志头部向后正向扫描日志，对重做事务队列(REDO-LIST)中的各个事务进行重做(REDO)操作。



数据库恢复

系统故障后的恢复算法存在的问题

- 恢复算法要搜索整个日志，检查所有日志记录。
- 重做的许多事务的更新操作结果已写到磁盘的数据库中，恢复机制又重新执行了对数据库的更新。

检查点技术

数据库恢复

采用检查点技术实现系统故障后的恢复

- 可以限制恢复机制必须回溯的日志长度，有效减少搜索日志的时间。
- 有效减少系统重新启动后需要重做的事务，减少数据库恢复所需的时间和资源。

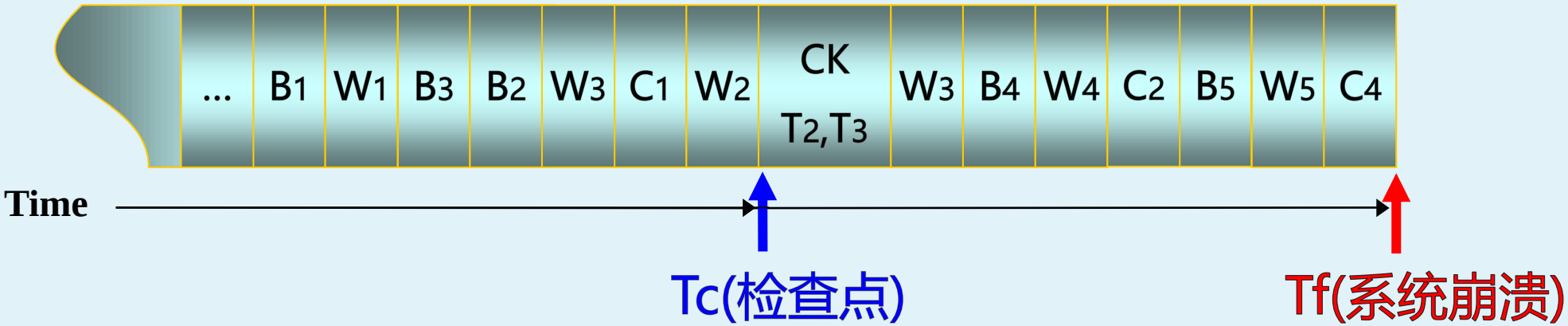
数据库恢复

采用检查点技术实现系统故障后的恢复

- **检查点的设置**
  - 定期或不定期地在生成的日志上设置检查点
  - 按照某种规则建立检查点

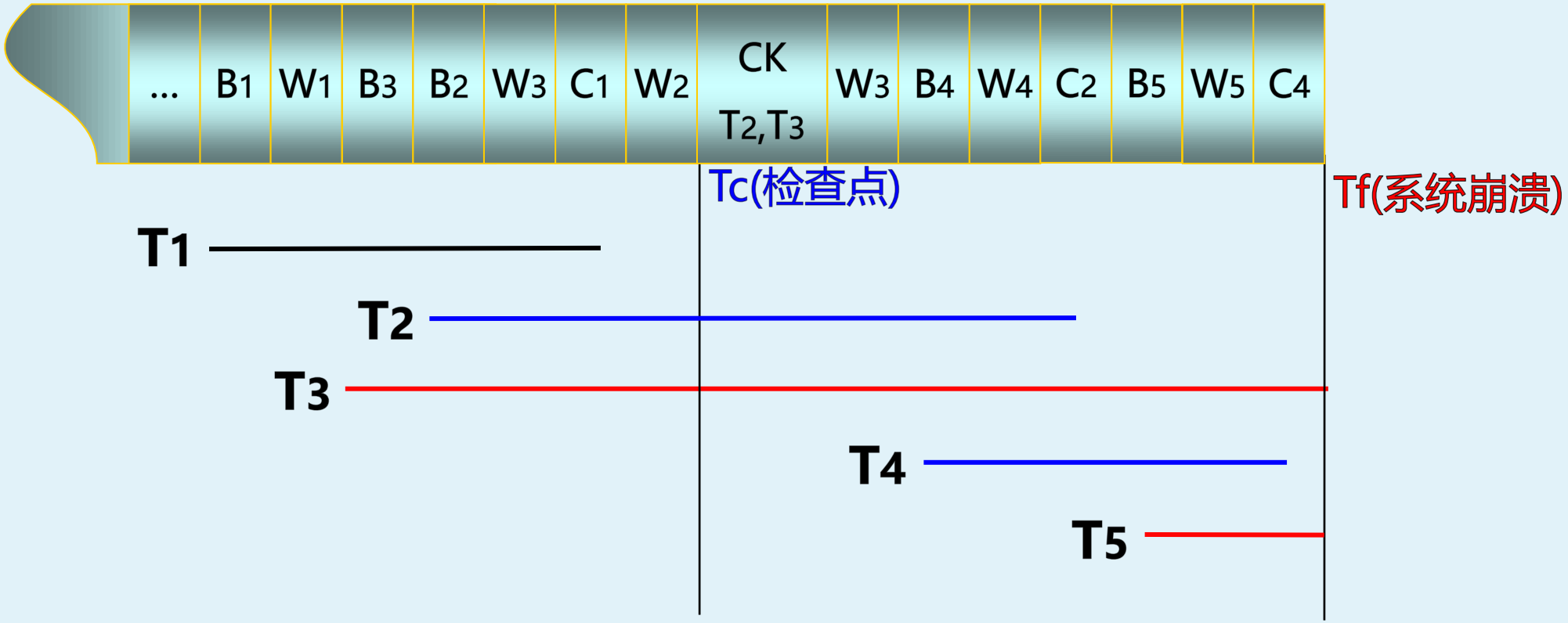
数据库恢复

采用检查点技术实现系统故障后的恢复



数据库恢复

采用检查点技术实现系统故障后的恢复



数据库恢复

采用静态检查点技术实现系统故障后的恢复

- **设置检查点时恢复机制要完成的工作**

① 暂时中止运行事务的执行；

② 将当前日志缓冲区中的所有日志记录写入磁盘的日志中；

③ 在日志中写入一个检查点记录；

④ 将当前数据缓冲区中的所有数据写入磁盘的数据库中；

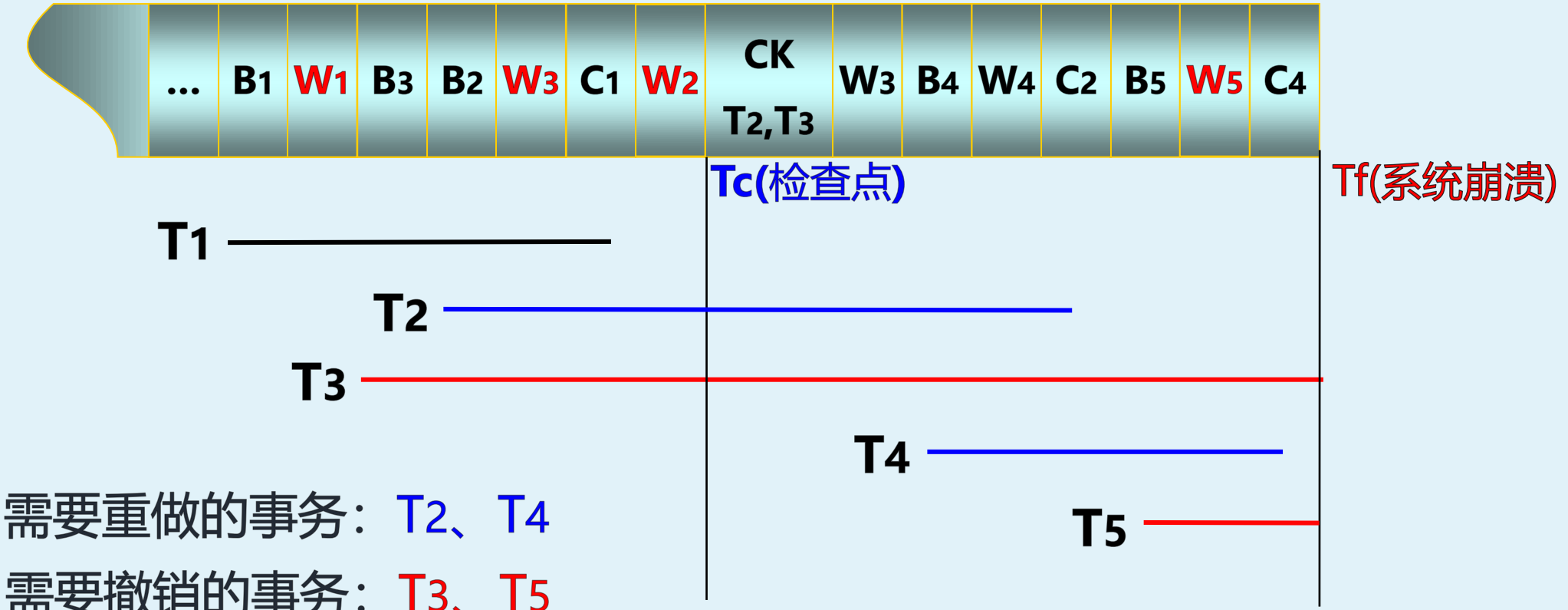
⑤ 把检查点记录在日志中的地址写入一个重新开始文件；

⑥ 重新开始执行运行的事务。

包括所有正在执行的事务

数据库恢复

采用静态检查点技术实现系统故障后的恢复



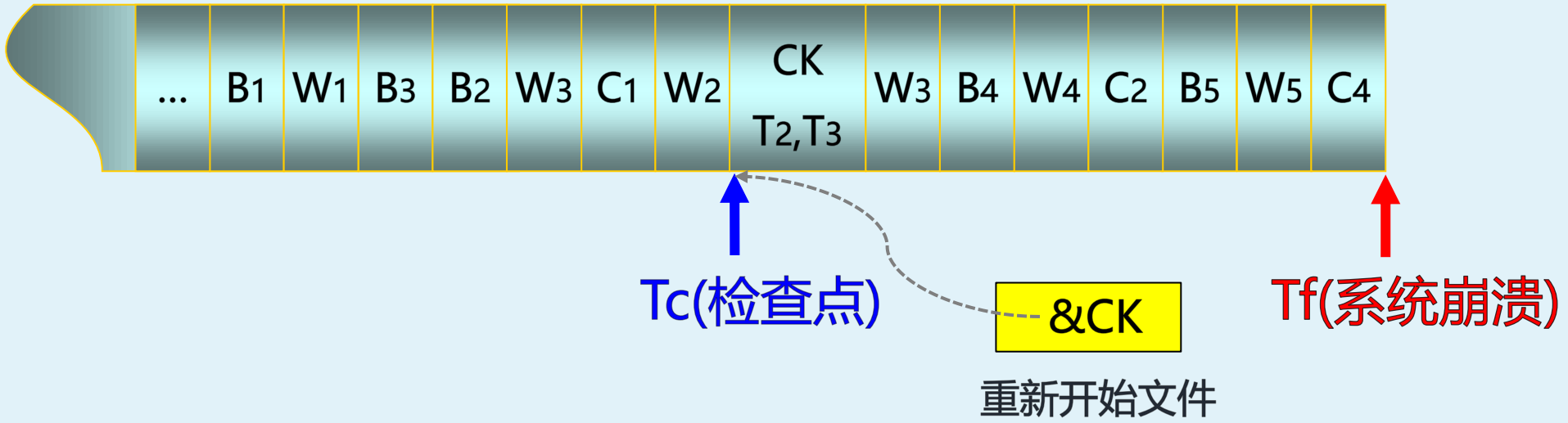
需要重做的事务：T2、T4

需要撤销的事务：T3、T5

数据库恢复

采用静态检查点技术实现系统故障后的恢复步骤

- (1)由重新开始文件中最后一个检查点记录在日志中的地址，在日志中找到最后一个检查点记录。



数据库恢复

采用静态检查点技术实现系统故障后的恢复步骤

- (2)将检查点记录中包括的所有正在执行的事务放入撤销事务UNDO-LIST队列中。

撤销事务队列 UNDO-LIST： T2 T3

数据库恢复

采用静态检查点技术实现系统故障后的恢复步骤

- (3)从**检查点记录**开始向后正向扫描日志，确定需要重做或撤销的事务。

撤销事务队列 UNDO-LIST： T2 T3 T4 T5

重做事务队列 REDO-LIST： T2 T4

数据库恢复

采用静态检查点技术实现系统故障后的恢复步骤

- (4)从日志尾部向前反向扫描日志，对撤销事务队列(UNDO-LIST)中的事务执行撤销(UNDO)操作。

撤销事务队列 UNDO-LIST： T3 T5

UDDO操作

数据库恢复

采用静态检查点技术实现系统故障后的恢复步骤

- (5)从**检查点记录**开始向后正向扫描日志，对重做事务队列(REDO-LIST)中的事务执行重做(REDO)操作。

重做事务队列 REDO-LIST： T2 T4

REDO操作

数据库恢复

采用静态检查点技术实现系统故障后的恢复步骤

- 在日志中找到最后一个检查点记录。
- 从**检查点记录**得到所有正在执行的事务，放入撤销事务UNDO-LIST队列中。
- 从**检查点记录**开始向后正向扫描日志，确定需要重做或撤销的事务。
- 对撤销事务队列UNDO-LIST中的事务执行UNDO操作。
- 从**检查点记录**开始，对重做事务队列REDO-LIST中的事务执行REDO操作。

数据库恢复

介质故障后的恢复

- 利用最新的数据库备份将数据库恢复到进行数据转储前的数据库状态。
- 利用日志备份将数据库恢复到离故障发生时更近的一致性状态。
- DBA装入最近转储的数据库备份和有关的日志文件备份，然后执行恢复命令，由DBMS完成具体的恢复操作。

数据库恢复

T-SQL对数据恢复的支持

```
RESTORE  DATABASE  <数据库名>
{FILE=logic_file_name|FILEGROUP=logical_filegroup_nam
e}
FROM {DISK|TAPE}=’ physical_backup_device_name’
[WITH
  [[,]{NORECOVERY|RECOVERY}]
  [[,]REPLACE]
]
```

恢复的数据文件

备份所在存储路径

是否撤销未提交事务

是否覆盖同名数据库

数据库恢复

T-SQL对数据恢复的支持

```
RESTORE LOG <数据库名>
FROM {DISK|TAPE}= 'physical_backup_device_name'
[WITH
[[,]{NORECOVERY|RECOVERY}]
[[,]STOPAT=date_time
|[,]STOPATMARK= 'mark_name' [AFTER datetime]
|[,]STOPBEFOREMARK= 'mark_name' [AFTER datetime]
]
]
```

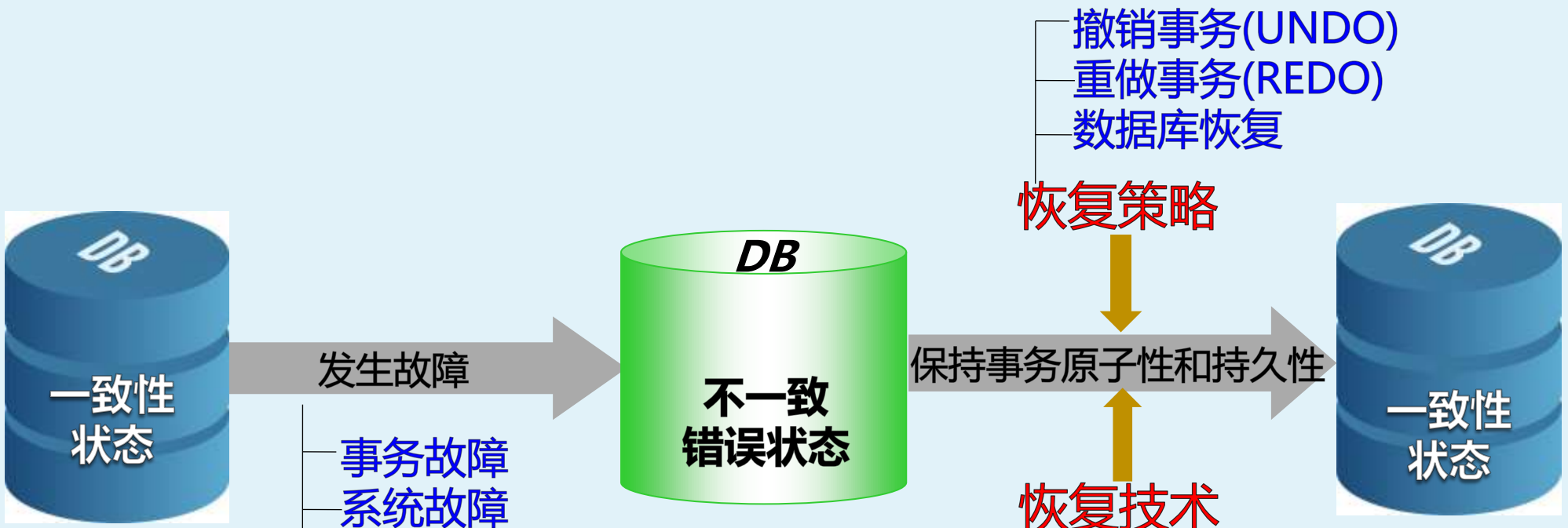
备份所在存储路径

是否撤销未提交事务

恢复到指定时间

恢复到(某时间后)指定标记(前)

小 结


撤销事务(UNDO)
重做事务(REDO)
数据库恢复
恢复策略
DB
一致性
状态
发生故障
事务故障
系统故障
介质故障
DB
不一致
错误状态
保持事务原子性和持久性
DB
一致性
状态
恢复技术
日志
数据转储
全备份
增量备份
日志备份