

- 数据库系统中的数据是由**DBMS**统一管理和控制的，为了适应数据共享的环境，**DBMS**必须提供数据保护能力，以保证数据库中数据的安全可靠和正确有效。
- 数据保护
 - 安全性
 - 完整性
 - 并发控制
 - 数据库恢复
- **事务**是恢复和并发控制的基本单位



10.1 事务的基本概念

- 一、什么是事务
- 二、事务的特性
- 三、如何定义事务



1.事务

- 事务(Transaction)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。
-



(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
 - 例如：
 - 某个转帐事务：对帐户A扣除100，对帐户B增加100。
 - 订票——查询、订位、(交钱)、出票



(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
 - 数据一致性要求，根据具体应用而定。
- 当数据库只包含成功事务提交的结果时，数据库处于一致性状态。



一致性与原子性

银行转帐：从帐号**A**中取出一万元，存入帐号**B**。在银行系统中，转帐事务的一致性要求是前后两个帐户的金额总和不变。定义一个事务，该事务包括两个操作

A	B
$A=A-1$	
	$B=B+1$

— 这两个操作要么全做，要么全不做

- 全做或者全不做，数据库都处于一致性状态。
- 如果只做一个操作，用户逻辑上就会发生错误，少了一万元，数据库就处于不一致性状态。



(3) 隔离性

一个事务的执行不能被其他事务干扰

- 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

隔离性要求一个事务，不会看到另一事务的中间结果。



(4) 持续性

- 持续性也称永久性 (Permanence)
事务一旦提交，它对数据库的影响就是永久性的。无论发生什么事情，都不能取消这种影响。

例如一个事务将50元从帐户A转到帐户B，此事务一旦提交，这种交易是无法悔改的——即便发生故障，也不能把这50元“还”回去。



2.事务的特性（ACID特性）

事务的ACID特性：

- 原子性（Atomicity）
- 一致性（Consistency）
- 隔离性（Isolation）
- 持续性（Durability）



事务的特性

- 保证事务**ACID**特性是事务管理的任务
- 破坏事务**ACID**特性的因素
 - (1) 多个事务并行运行时，不同事务的操作交叉执行
 - 数据库管理系统必须保证多个事务的交叉运行不影响这些事务的隔离性
 - (2) 事务在运行过程中被强行停止
 - 数据库管理系统必须保证被强行终止的事务对数据库和其他事务没有任何影响



如何定义事务

– 事务的显式定义方式

- 用户明确说明一个事务的开始（Begin Transaction语句）和结束（用Commit/Rollback语句）。



SQL中的事务

» 相关语句

– Begin Transaction

- 开始事务

– Commit (Transaction)

- 提交事务。此时事务正常结束，事务所做的修改操作都有效且永久地写入数据库

– Rollback (Transaction)

- 回滚事务。在事务执行过程中发生了某种故障，不能继续执行，事务所做的修改操作都无效且必须撤销(Undo，数据改回旧值)。
- 回滚后，数据库回退到事务开始前的状态，就好像事务从来没有发生过一样。



- 例如，银行转账事务，这个事务把一笔金额从一个账户甲转给另一个账户乙。

BEGIN TRANSACTION

读账户甲的余额**BALANCE**;

BALANCE=BALANCE-AMOUNT; /*AMOUNT 为转账金额*/

IF(BALANCE < 0) THEN

{打印‘金额不足，不能转账’;

ROLLBACK;

}

ELSE

{读账户乙的余额**BALANCE1**;

BALANCE1=BALANCE1+AMOUNT;

COMMIT;}

/*事务内部可能造成事务被回滚的情况*/

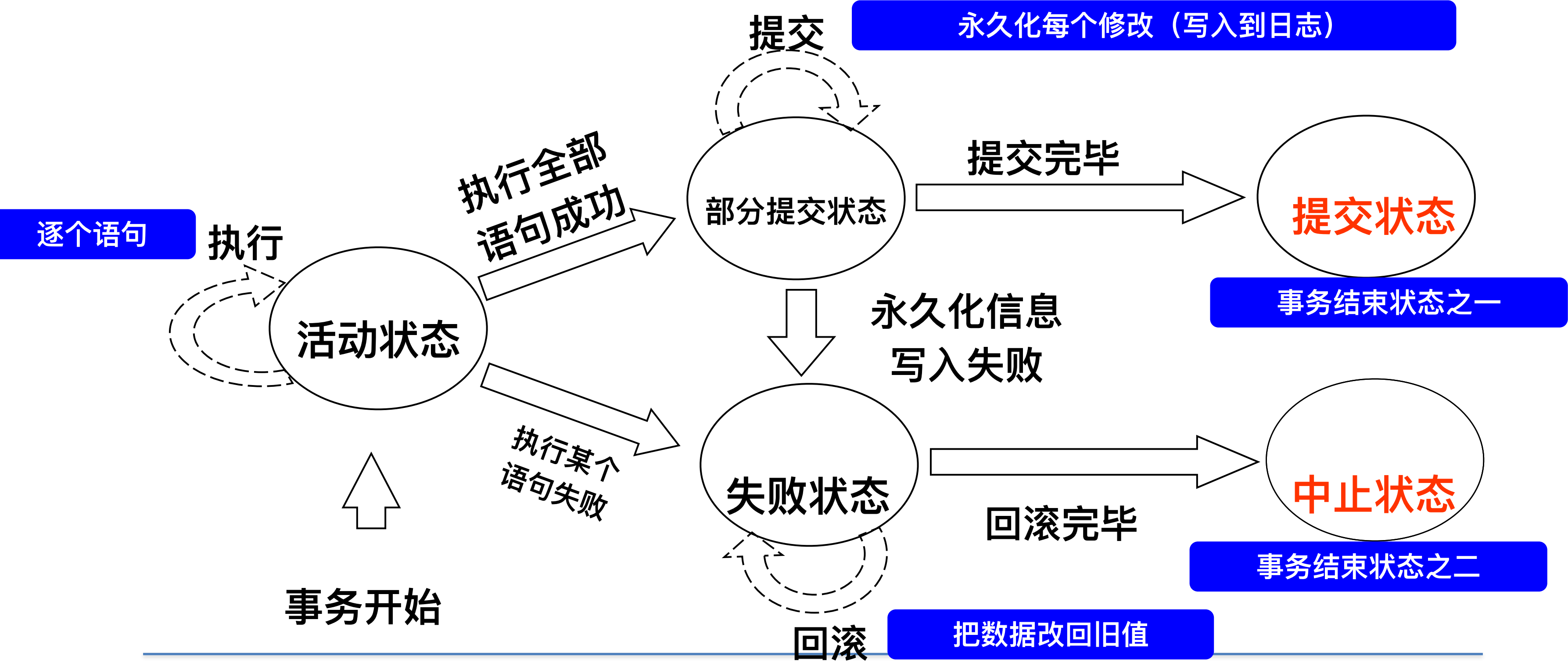
/*撤销刚才的修改，恢复事务*/



- 事务和程序是两个概念
 - 在关系数据库中，一个事务可以是一条SQL语句，一组SQL语句或整个程序
 - 一个程序通常包含多个事务



事务状态与操作



» 执行一个事务时，可能经过哪些状态序列

- 活动状态 → 部分提交状态 → 提交状态
- 活动状态 → 失败状态 → 中止状态
- 活动状态 → 部分提交状态 → 失败状态 → 中止状态



SQL中的事务

» 说明

- (1) 事务没有执行完Commit之前，因某种原因（故障）而终止了，那么系统就会自动地回滚这个事务。
- (2) 判断SQL语句属于哪个事务
 - 如果SQL语句处于某个事务的Begin transaction和Commit/Rollback之间，那么它就属于这个事务。
 - 没有明确说明一个事务的开始（有时也不说明结束）
 - 比如从第一个有效的DML语句开始事务，遇到明确的Commit/Rollback结束一个事务。
 - 遇到Create Table、Drop Table、Create View，Create Index等DDL语句，（提交上一个事务后）启动一个新的事务。
 - 如果以上不成立，那么这个SQL语句本身构成一个独立的事务。



» 下面的SQL Server程序包含几个事务?

Begin Transaction

Update account set money= money - 50 where no='A'

Update account set money= money + 50 where no='B'

Commit

» 思考，如果执行完第一个update语句之后、没有执行第二个update语句之前，系统断电。那么在重启以后，两个帐户的金额会发生什么样的变化?



» 下面的SQL Server程序包含几个事务?

```
Update account set money= money - 50 where no='A'
```

```
Update account set money= money + 50 where no='B'
```

思考，如果执行完第一个update语句之后、没有执行第二个update语句之前，系统断电。那么在重启以后，两个帐户的金额会发生什么样的变化?

