

12.1 事务的基本概念

事务

- **事务**：是一系列的数据库操作，是数据库应用程序的逻辑单位，即应用程序对数据库的操作都应该以事务的方式进行。
- 事务是一个操作序列，这些操作“要么都做，要么都不做”。
- 事务定义的语句如下：
 - (1) BEGIN TRANSACTION：事务开始。
 - (2) END TRANSACTION：事务结束。
 - (3) COMMIT：事务提交。该操作表示事务成功地结束，它将通知事务管理器该事务的所有更新操作现在可以被提交或永久地保留。
 - (4) ROLLBACK：事务回滚。该操作表示事务非成功地结束，它将通知事务管理器出故障了，数据库可能处于不一致状态，该事务的所有更新操作必须回滚或撤销。

事务

例：从账户A转入账户B金额x元。

```
BEGIN TRANSACTION           // 事务开始
  read(A);                   // 读账户A的余额
  A=A-x;                     // A账户金额减去x元
  IF(A<0) THEN
    print ("金额不足，不能转账");
    ROLLBACK;                // 撤销该事务，回到事务执行前的状态
  ELSE
    write(A);                 // 写入账户A的余额
    read(B);                 // 读账户B的余额
    B=B+x;
    write(B);
    COMMIT;                  // 提交事务
  ENDIF;
END TRANSACTION              // 事务结束
```


SQL中事务的开始与结束

- SQL标准规定当一条SQL语句被执行，就隐式地开始了一个事务，SQL中的Commit work和Rollback work语句之一会结束一个事务。
 - (1) Commit work: 提交当前事务。这意味着该事务所做的更新在数据库中永久保存。一旦事务被提交后，一个新的事务自动开始。
 - (2) Rollback work: 回滚当前事务。这意味着将撤销该事务对数据库的更新。
- 注意：如果事务已经执行了Commit work，就不能再用Rollback来撤销。

事务的特性

• 事务具4个特性：原子性、一致性、隔离性和持久性。

1、**原子性**：事务的所有操作在数据库中要么都做要么都不做。

2、**一致性**：一个事务独立执行的结果，将保持数据的一致性，即数据不会因为事务的执行而遭受破坏。

3、**隔离性**：一个事务的执行不能被其他事务干扰。并发事务在执行过程中可能会对同一数据进行操作，这些事务的操作应该不会相互干扰，是相互隔离的。

4、**持久性**：一个事务一旦提交，它对数据库的改变必须是永久的，即使系统出现故障也是如此。

例：事务是一个操作序列，这些操作____(1)____。“当多个事务并发执行时，任何一个事务的更新操作直到其成功提交前的整个过程，对其他事务都是不可见的。”这一性质通常被称为事务的____(2)____性质。

- (1) A. “可以做，也可以不做”，是数据库环境中可分割的逻辑工作单位
B. “可以只做其中的一部分”，是数据库环境中可分割的逻辑工作单位
C. “要么都做，要么都不做”，是数据库环境中可分割的逻辑工作单位
D. “要么都做，要么都不做”，是数据库环境中不可分割的逻辑工作单位

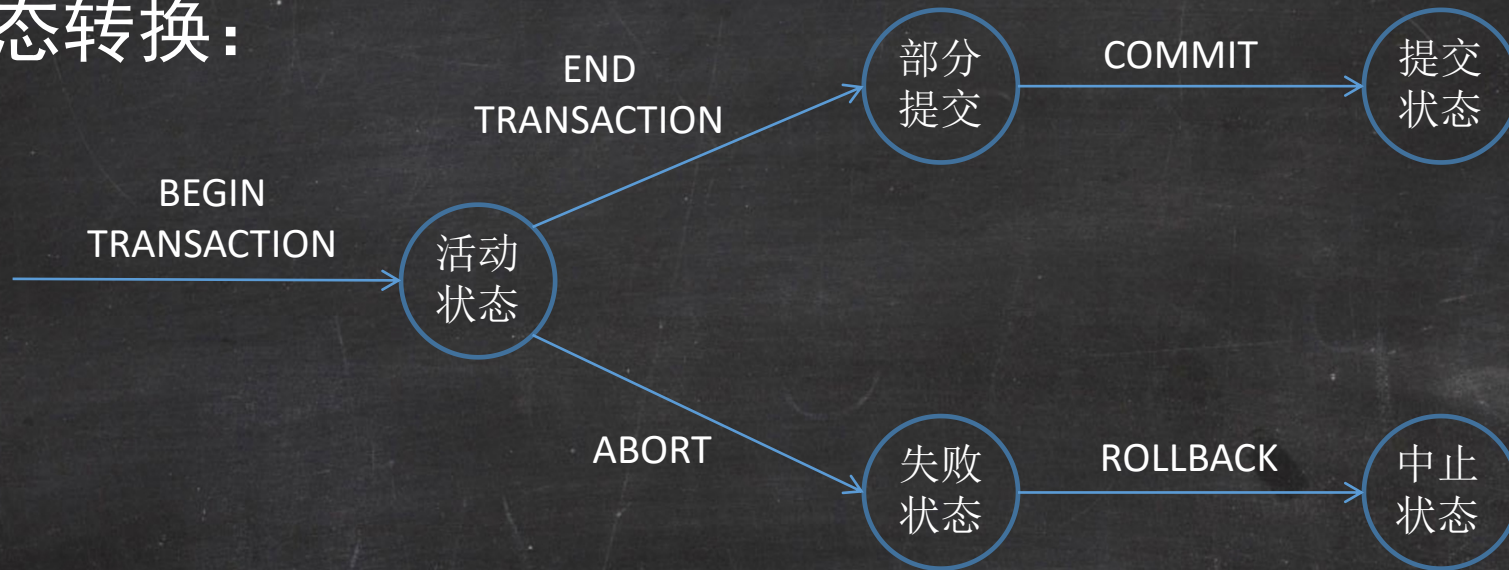
- (2) A. 原子性 B. 一致性 C. 隔离性 D. 持久性

事务的状态

- 事务是数据库的基本执行单元，如果事务成功执行，则数据库从一个一致状态进入另一个一致状态。如果因为某种原因事务没能成功执行，但其已经对数据库进行了修改，这时候可能会导致数据库处于不一致的状态，需要对事务已经造成的变更进行撤销（回滚）。
- 1、**活动状态**：事务的初始状态，事务执行时处于这个状态。
 - 2、**部分提交状态**：当操作序列的最后一条语句执行后，事务就处于部分提交状态。这时，事务虽然已经完全执行，但由于实际输出可能还临时驻留在内存中，在事务成功完成前还有可能出现硬件故障，因此，部分提交状态并不等于事务成功执行。
 - 3、**失败状态**：由于硬件或逻辑错误，使得事务不能继续正常执行，事务就进入了失败状态，处于失败状态的事务必须回滚。这样，事务就进入了中止状态。
 - 4、**中止状态**：事务回滚并且数据库恢复到事务开始执行前的状态。
 - 5、**提交状态**：当事务成功完成后，称事务处于提交状态。只有事务处于提交状态后，才能说事务已经提交。

事务的状态

- 事务的状态转换：



- BEGIN TRANSACTION：** 开始运行事务，使事务进入活动状态
- END TRANSACTION：** 说明事物中的所有读写操作都已完成，使事务进入部分提交状态，把事务的所有操作对数据库的影响存入数据库。
- COMMIT：** 标志事务已经成功地完成，事务中的所有操作对数据库的影响已经安全地存入数据库，事务进入提交状态，结束事务的运行。
- ABORT：** 标志事务进入失败状态，系统撤销事务中所有操作对数据库和其他事务的影响，结束事务的运行。