# 第15章 事 务

1. 事务（transaction）：

事务是构成单一逻辑工作单元的操作集合；

事务是访问并可能更新各种数据项的一个程序执行单元（unit）。

（即使有故障，数据库系统也必须保证事务的正确执行，即执行整个事务或属于该事务的操作一个也不执行）

1. ACID特性：

为了保证数据完整性，要求数据库系统维护事务的以下性质：

**原子性（atomicity）：**事务的所有操作在数据库中要么全部正确反映，要么全部不反映。

**一致性（consistency）：**保证事务执行后数据性质的一致性（如一增一减后总

和不变）。

**隔离性（isolation）：**保证事务并发执行的正确性。

**持久性（durability）：**一个事务成功完成后，它对数据库的改变是永久的，即使系统可能出现故障。

1. ACID零碎知识点：
2. 保证原子性的工作是由**事务管理部件（transaction-management component）**处理；
3. 确保持久性是数据库系统中称为**恢复管理部件（recovery-management component）**的软件部件负责；
4. 确保隔离性是数据库系统中**并发控制部件（concurrency-control component）**负责的；
5. 由于系统故障，系统的状态不能再反映数据库本应描述的现实世界的真实状态。我们把这种状态称为**不一致状态（inconsistent state）**；
6. 对于原子性和持久性的支持，一个简单的方案是**影子拷贝（shadow copy）**。
7. 事务分类
8. **中止事务（aborted）**：不能顺利执行完成的事务。
9. **已提交事务（committed）**：成功执行完成的事务。
10. **补偿事务（compensating translation）**：撤销已提交事务所造成影响的事

务。

1. **已结束事务（terminated）**：提交的或中止的事务。

（中止事务对数据库所做过的任何改变必须撤销，一旦中止事务造成的更改撤销，我们就说事务已**回滚rolled back**）

1. 事务状态：
2. 活动状态（active）：初始状态。
3. 部分提交状态（partially committed）：最后一条语句执行后。
4. 失败状态（failed）：发现正常的执行不能继续之后。
5. 中止状态（aborted）：事务回滚并且数据库已恢复到事务开始执行前的状态之后。
6. 提交状态（committed）：成功完成后。
7. 事务状态图：



1. 系统判定事务不能继续正常执行后，事务进入失败状态，这种事务必须**回滚**，这样事务就进入中止状态；
2. 在中止状态，系统有两种选择：
3. **重启事务（restart）**：仅当引起事务中止的是硬件错误；
4. **杀死事务（kill）**：由于事务的内部逻辑错误引起事务中止。
5. 零散的知识点
6. *冲突等价（conflict eqivalent）*：如果调度S可以经过一系列非冲突指令交换转换成S’，我们称S与S’是冲突等价的。
7. *冲突可串行化（conflict serializable）*：若一个调度S与一个串行调度冲突等价，称调度S是冲突可串行化的。
8. *可恢复调度（recoverable schedule）*：对于每对事务Ti和Tj，如果Tj读取了由Ti所写的数据项，则Ti先于Tj提交。
9. *联级回滚（cascading rollback）*：因一个事务故障导致一系列事务回滚的现象。
10. *无级联调度*：对于每对事务Ti和Tj，如果Tj读取了由Ti所写的数据项，则Ti必须在Tj这一读取前提交。
11. 如果调度S的优先图中有环，则调度S是非冲突可串行化的；如果图中无环，则调度S是冲突可串行化的。