



# 第9节 事务

讲师：李玉婷

# 目 标

通过本章学习，您将可以：

- 事务的概念和特性
- 事务的隔离级别
- 事务的案例演示

- **事务：**事务由单独单元的一个或多个SQL语句组成，在这个单元中，每个MySQL语句是相互依赖的。而整个单独单元作为一个不可分割的整体，如果单元中某条SQL语句一旦执行失败或产生错误，整个单元将会回滚。所有受到影响的数据将返回到事物开始以前的状态；如果单元中的所有SQL语句均执行成功，则事物被顺利执行。

- 1、概念：在mysql中的数据用各种不同的技术存储在文件（或内存）中。
- 2、通过show engines；来查看mysql支持的存储引擎。
- 3、在mysql中用的最多的存储引擎有：innodb, myisam ,memory 等。其中innodb支持事务，而myisam、memory等不支持事务

## ● 事务的ACID(acid)属性

### ➤ 1. 原子性 (Atomicity)

原子性是指事务是一个不可分割的工作单位，事务中的操作要么都发生，要么都不发生。

### ➤ 2. 一致性 (Consistency)

事务必须使数据库从一个一致性状态变换到另外一个一致性状态。

### ➤ 3. 隔离性 (Isolation)

事务的隔离性是指一个事务的执行不能被其他事务干扰，即一个事务内部的操作及使用的数据对并发的其他事务是隔离的，并发执行的各个事务之间不能互相干扰。

### ➤ 4. 持久性 (Durability)

持久性是指一个事务一旦被提交，它对数据库中数据的改变就是永久性的，接下来的其他操作和数据库故障不应该对其有任何影响。

- 以第一个 DML 语句的执行作为开始
- 以下面的其中之一作为结束:
  - **COMMIT 或 ROLLBACK 语句**
  - DDL 或 DCL 语句（自动提交）
  - 用户会话正常结束
  - 系统异常終了

- 对于同时运行的多个事务, 当这些事务访问**数据库中相同的数据**时, 如果没有采取必要的隔离机制, 就会导致各种并发问题:
  - **脏读**: 对于两个事务 T1, T2, T1 读取了已经被 T2 更新但还**没有被提交**的字段. 之后, 若 T2 回滚, T1 读取的内容就是临时且无效的.
  - **不可重复读**: 对于两个事务 T1, T2, T1 读取了一个字段, 然后 T2 **更新**了该字段. 之后, T1 再次读取同一个字段, 值就不同了.
  - **幻读**: 对于两个事务 T1, T2, T1 从一个表中读取了一个字段, 然后 T2 在该表中**插入**了一些新的行. 之后, 如果 T1 再次读取同一个表, 就会多出几行.
- **数据库事务的隔离性**: 数据库系统必须具有隔离并发运行各个事务的能力, 使它们不会相互影响, 避免各种并发问题.
- **一个事务与其他事务隔离的程度称为隔离级别**. 数据库规定了多种事务隔离级别, 不同隔离级别对应不同的干扰程度, 隔离级别越高, 数据一致性就越好, 但并发性越弱.

- 数据库提供的 4 种事务隔离级别:

隔离级别	描述
READ UNCOMMITTED (读未提交数据)	允许事务读取未被其他事物提交的变更,脏读,不可重复读和幻读的问题都会出现
READ COMMITED (读已提交数据)	只允许事务读取已经被其它事务提交的变更,可以避免脏读,但不可重复读和幻读问题仍然可能出现
REPEATABLE READ (可重复读)	确保事务可以多次从一个字段中读取相同的值,在这个事务持续期间,禁止其他事物对这个字段进行更新,可以避免脏读和不可重复读,但幻读的问题仍然存在.
SERIALIZABLE(串行化)	确保事务可以从一个表中读取相同的行,在这个事务持续期间,禁止其他事务对该表执行插入,更新和删除操作,所有并发问题都可以避免,但性能十分低下.

- Oracle 支持的 2 种事务隔离级别: **READ COMMITED**,  
SERIALIZABLE。Oracle 默认的事务隔离级别为: READ  
COMMITTED
- Mysql 支持 4 种事务隔离级别. Mysql 默认的事务隔离级别  
为: **REPEATABLE READ**

- 每启动一个 mysql 程序, 就会获得一个单独的数据库连接. 每个数据库连接都有一个全局变量 @@tx\_isolation, 表示当前的事务隔离级别.
- 查看当前的隔离级别: `SELECT @@tx_isolation;`
- 设置当前 mySQL 连接的隔离级别:
  - `set transaction isolation level read committed;`
- 设置数据库系统的全局的隔离级别:
  - `set global transaction isolation level read committed;`

