MySQL事务特性及隔离级别

什么是事务?

**事务是逻辑上的一组操作，要么都执行，要么都不执行。**

事务最经典也经常被拿出来说例子就是转账了。

假如小明要给小红转账1000元，这个转账会涉及到两个关键操作就是：将小明的余额减少1000元，将小红的余额增加1000元。万一在这两个操作之间突然出现错误比如银行系统崩溃，导致小明余额减少而小红的余额没有增加，这样就不对了。事务就是保证这两个关键操作要么都成功，要么都要失败。

**原子性（Atomicity）： 事务是最小的执行单位，不允许分割**。事务的原子性确保动作要么全部完成，要么完全不起作用；**因此事务的操作如果成功就必须要完全应用到数据库，如果操作失败则不能对数据库有任何影响**。

**一致性（Consistency）： 执行事务前后，数据保持一致**；一致性是指事务必须使数据库从一个一致性状态变换到另一个一致性状态，也就是说一个事务执行之前和执行之后都必须处于一致性状态。拿转账来说，小明和小红两者的钱加起来一共是5000，那么不管A和B之间如何转账，转几次账，事务结束后两个用户的钱相加起来应该还得是5000，这就是事务的一致性。

**隔离性（Isolation）： 并发访问数据库时，一个用户的事物不被其他事物所干扰，各并发事务之间数据库是独立的**；当多个用户并发访问数据库时，比如操作同一张表时，数据库为每一个用户开启的事务，不能被其他事务的操作所干扰，多个并发事务之间要相互隔离。即要达到这么一种效果：对于任意两个并发的事务T1和T2，在事务T1看来，T2要么在T1开始之前就已经结束，要么在T1结束之后才开始，这样每个事务都感觉不到有其他事务在并发地执行。

**持久性（Durability）: 一个事务被提交之后**。它对数据库中数据的改变是持久的，即使数据库发生故障也不应该对其有任何影响。例如我们在使用JDBC操作数据库时，在提交事务方法后，提示用户事务操作完成，当我们程序执行完成直到看到提示后，就可以认定事务以及正确提交，即使这时候数据库出现了问题，也必须要将我们的事务完全执行完成，否则就会造成我们看到提示事务处理完毕，但是数据库因为故障而没有执行事务的重大错误。

**并发一致性问题**

在并发环境下，事务的隔离性很难保证，因此会出现很多并发一致性问题。

**1. 更新丢失（Lost Update）**：定义：当两个或多个事务选择同一行，然后基于最初选定的值更新该行时，由于每个事务都不知道其他事务的存在，就会发生丢失更新问题，最后的更新覆盖了由其他事务所做的更新

**2. 脏读（Dirty Reads）**：定义：一个事务正在对一条记录做修改，在这个事务完成并提交前， 这条记录的数据就处于不一致状态； 这时， 另一个事务也来读取同一条记录，如果不加控制，第二个事务读取了这些“脏”数据，并据此做进一步的处理，就会产生未提交的数据依赖关系。这种现象被形象地叫做"脏读"。T1 修改一个数据，T2 随后读取这个数据。如果 T1 撤销了这次修改，那么 T2 读取的数据是脏数据。

**3. 不可重复读（Non-Repeatable Reads）：**

定义：一个事务在读取某些数据后的某个时间，再次读取以前读过的数据，却发现其读出的数据已经发生了改变、或某些记录已经被删除了！这种现象就叫做“不可重复读” 。

**4. 幻读（Phantom Reads）：**

定义：一个事务按相同的查询条件重新读取以前检索过的数据，却发现其他事务插入了满足其查询条件的新数据，这种现象就称为“幻读” 。

**幻读和不可重复读的区别：**

**不可重复读的重点是修改**：在同一事务中，同样的条件，第一次读的数据和第二次读的数据不一样。（因为中间有其他事务提交了修改）

**幻读的重点在于新增或者删除**：在同一事务中，同样的条件,，第一次和第二次读出来的记录数不一样。（因为中间有其他事务提交了插入/删除）

**不可重复读和脏读的区别**：脏读是读到未提交的数据，而不可重复读读到的却是已经提交的数据。

**隔离级别**

**SQL标准定义了4类隔离级别，每一种级别都规定了一个事务中所做的修改，哪些在事务内和事务间是可见的，哪些是不可见的**。低级别的隔离级别一般支持更高的并发处理，并拥有更低的系统开销。

**第1级别：Read Uncommitted(未提交读)**

所有事务都可以看到其他未提交事务的执行结果

本隔离级别很少用于实际应用，因为它的性能也不比其他级别好多少

**该级别引发的问题是——脏读(Dirty Read)：读取到了未提交的数据**

**第2级别：Read Committed(提交读)**

这是大多数数据库系统的默认隔离级别（但不是MySQL默认的）

**它满足了隔离的简单定义：一个事务只能读取已经提交的事务所做的修改**。换句话说，一个事务所做的修改在提交之前对其它事务是不可见的。

**这种隔离级别出现的问题是——不可重复读(Nonrepeatable Read**)：不可重复读意味着我们在同一个事务中执行完全相同的select语句时可能看到不一样的结果。

导致这种情况的原因可能有：

有一个交叉的事务有新的commit，导致了数据的改变;

一个数据库被多个实例操作时，同一事务的其他实例在该实例处理期间可能会有新commit

**第3级别：Repeatable Read(可重复读)**

这是MySQL的默认事务隔离级别

保证在同一个事务中多次读取同样数据的结果是一样的。

它确保同一事务的多个实例在并发读取数据时，会看到同样的数据行

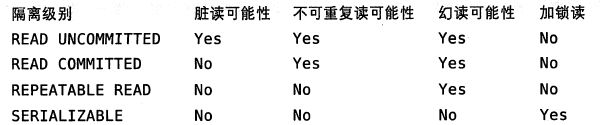
**此级别可能出现的问题——幻读(Phantom Read)**：当用户读取某一范围的数据行时，另一个事务又在该范围内插入了新行，当用户再读取该范围的数据行时，会发现有新的“幻影” 行InnoDB和Falcon存储引擎通过多版本并发控制(MVCC，Multiversion Concurrency Control)机制解决幻读问题；InnoDB还通过间隙锁解决幻读问题

**第4级别：Serializable(可串行化)**

这是最高的隔离级别，强制事务串行执行。

它通过强制事务排序，使之不可能相互冲突，从而解决幻读问题。简言之,它是在每个读的数据行上加上共享锁。

在这个级别，可能导致大量的超时现象和锁竞争



[https://juejin.im/post/5cd69ed7e51d453af9052143#heading-9](https://juejin.im/post/5cd69ed7e51d453af9052143" \l "heading-9)

<https://zhuanlan.zhihu.com/p/65332242>