**数据库事务四种隔离级别**

MYSQL标准定义了4类隔离级别，用来限定事务内外的哪些改变是可见的，哪些是不可见的。**低的隔离级一般支持更高的并发处理，并拥有更低的系统开销。**下面通过这篇文章我们来一起深入理解Mysql中的四种隔离级别。

一、首先什么是事务？

**事务是应用程序中一系列严密的操作，所有操作必须成功完成**，否则在每个操作中所作的所有更改都会被撤消。也就是事务具有原子性，一个事务中的一系列的操作要么全部成功，要么一个都不做。

事务的结束有两种，当事务中的所以步骤全部成功执行时，事务提交。如果其中一个步骤失败，将发生回滚操作，撤消撤消之前到事务开始时的所以操作。

二、事务的 ACID

事务具有四个特征：原子性（ Atomicity ）、一致性（ Consistency ）、隔离性（ Isolation ）和持续性（ Durability ）。这四个特性简称为 ACID 特性。

     1 、**原子性**。事务是数据库的逻辑工作单位，事务中包含的各操作要么都做，要么都不做

     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。因此当数据库只包含成功事务提交的结果时，就说数据库处于一致性状态。如果数据库系统 运行中发生故障，有些事务尚未完成就被迫中断，这些未完成事务对数据库所做的修改有一部分已写入物理数据库，这时数据库就处于一种不正确的状态，或者说是 不一致的状态。

     3 、**隔离性**。一个事务的执行不能其它事务干扰。即**一个事务内部的操作及使用的数据对其它并发事务是隔离的**，并发执行的各个事务之间不能互相干扰。

     4 、**持续性**。也称永久性，指一**个事务一旦提交，它对数据库中的数据的改变就应该是永久性的**。接下来的其它操作或故障不应该对其执行结果有任何影响。

三、Mysql的四种隔离级别

SQL标准定义了4类隔离级别，包括了一些具体规则，用来限定事务内外的哪些改变是可见的，哪些是不可见的。**低级别的隔离级一般支持更高的并发处理，并拥有更低的系统开销。**

**（零）没有并发控制**

存在的问题：更新遗失。

解决办法就是下面的“可读取未确认”。

**（一）可读取未确认（Read uncommitted）**

**写事务阻止其他写事务，**避免了更新遗失。但是没有阻止其他读事务。

存在的问题：脏读。即读取到不正确的数据，因为另一个事务可能还没提交最终数据，这个读事务就读取了中途的数据，这个数据可能是不正确的。

解决办法就是下面的“可读取确认”。

**（二）可读取确认（Read committed）**

**写事务会阻止其他读写事务。读事务不会阻止其他任何事务。**

存在的问题：不可重复读。即在一次事务之间，进行了两次读取，但是结果不一样，可能第一次id为1的人叫“李三”，第二次读id为1的人就叫了“李四”。因为读取操作不会阻止其他事务。

解决办法就是下面的“可重复读”。

**（三）可重复读（Repeatable read）**

**读事务会阻止其他写事务，但是不会阻止其他读事务**。

存在的问题：幻读。可重复读阻止的写事务包括update和delete（只给存在的表加上了锁），但是不包括insert（新行不存在，所以没有办法加锁），所以一个事务第一次读取可能读取到了10条记录，但是第二次可能读取到11条，这就是幻读。

解决办法就是下面的“串行化”。

**（四）可串行化（Serializable）**

**读加共享锁，写加排他锁**。这样读取事务可以并发，但是读写，写写事务之间都是互斥的，基本上就是一个个执行事务，所以叫串行化。

之所以提出事务隔离级别，是因为在一个事务执行过程中，可能会出现以下几种情况:

**1、更新丢失**

两个事务都同时更新一行数据，一个事务对数据的更新把另一个事务对数据的更新覆盖了。这是因为系统没有执行任何的锁操作，因此并发事务并没有被隔离开来。

**2、脏读**

一个事务读取到了另一个事务未提交的数据操作结果。

3、**不可重复读**（Non-repeatable Reads）：一个事务对同一行数据重复读取两次，但是却得到了不同的结果。

包括以下情况：

（1） **虚读**：事务T1读取某一数据后，事务T2对其做了修改，当事务T1再次读该数据时得到与前一次不同的值。

（2） **幻读**（Phantom Reads）：事务在操作过程中进行两次查询，第二次查询的结果包含了第一次查询中未出现的数据或者缺少了第一次查询中出现的数据（这里并不要求两次查询的SQL语句相同）。这是因为在两次查询过程中有另外一个事务插入数据造成的。

**Read Uncommitted（读取未提交内容）**

在该隔离级别，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。读取未提交的数据，也被称之为脏读（Dirty Read）。

**Read Committed（读取提交内容）**

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：一个事务只能看见已经提交事务所做的改变。这种隔离级别 也支持所谓的不可重复读（Nonrepeatable Read），因为同一事务的其他实例在该实例处理其间可能会有新的commit，所以同一select可能返回不同结果。

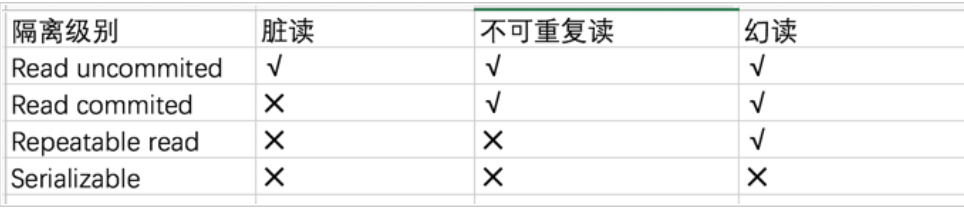
**Repeatable Read（可重读）**

这是MySQL的默认事务隔离级别，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。不过理论上，这会导致另一个棘手的问题：幻读 （Phantom Read）。简单的说，幻读指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）机制解决了该问题。

**Serializable（可串行化）**

这是最高的隔离级别，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。

在MySQL中，实现了这四种隔离级别，分别有可能产生问题如下所示：



两个命令行客户端分别为A，B；不断改变A的隔离级别，在B端修改数据。

**（一）、将A的隔离级别设置为read uncommitted(未提交读)**

事务B更新了一条记录，但是没有提交，此时事务A可以查询出未提交记录。造成脏读现象。未提交读是最低的隔离级别

**（二）、将客户端A的事务隔离级别设置为read committed(已提交读)**

已提交读隔离级别解决了脏读的问题，但是出现了不可重复读的问题，即事务A在两次查询的数据不一致，因为在两次查询之间事务B更新了一条数据。已提交读只允许读取已提交的记录，但不要求可重复读。

**(三)、将A的隔离级别设置为repeatable read(可重复读)**

可重复读隔离级别只允许读取已提交记录，而且在一个事务两次读取一个记录期间，其他事务部的更新该记录。但该事务不要求与其他事务可串行化。例如，当一个事务可以找到由一个已提交事务更新的记录，但是可能产生幻读问题(注意是可能，因为数据库对隔离级别的实现有所差别)。像以上的实验，就没有出现数据幻读的问题。

**(四)、将A的隔离级别设置为可串行化(Serializable)**

serializable完全锁定字段，若一个事务来查询同一份数据就必须等待，直到前一个事务完成并解除锁定为止。是完整的隔离级别，会锁定对应的数据表格，因而会有效率的问题。