# **########## mysql的ACID ###############**

**1.ACID是衡量事务的四个特性：**

**原子性：**语句要么全执行，要么全不执行，是事务最核心的特性，事务本身就是以原子性来定义的；实现主要基于undo log

**一致性：**事务追求的最终目标，一致性的实现既需要数据库层面的保障，也需要应用层面的保障

**隔离性：**保证事务执行尽可能不受其他事务影响；InnoDB默认的隔离级别是RR，RR的实现主要基于锁机制（包含next-key lock）、MVCC（包括数据的隐藏列、基于undo log的版本链、ReadView）

**持久性：**保证事务提交后不会因为宕机等原因导致数据丢失；实现主要基于redo log

### **2.脏读、[不可重复读](https://so.csdn.net/so/search?q=%E4%B8%8D%E5%8F%AF%E9%87%8D%E5%A4%8D%E8%AF%BB&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/zxl646801924/article/details/_blank)、幻读**

**1）脏读**：当前事务(A)中可以读到其他事务(B)未提交的数据（脏数据），这种现象是脏读。举例如下（以账户余额表为例）：



1. **不可重复读**：在事务A中先后两次读取同一个数据，两次读取的结果不一样，这种现象称为不可重复读。**脏读与不可重复读的区别在于：前者读到的是其他事务未提交的数据，后者读到的是其他事务已提交的数据**。举例如下：



1. **幻读**：在事务A中按照某个条件先后两次查询数据库，两次查询结果的条数不同，这种现象称为幻读。**不可重复读与幻读的区别可以通俗的理解为：前者是修改操作数据变了，后者是新增数据导致行数统计变了。**举例如下：



## **事务隔离级别**

**(1)读未提交:**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。

**(2)读已提交:**这是大多数数据库系统的默认隔离级别（比如SQLSever，Oracle，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，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。

**(3)可重复读:**这是MySQL的默认事务隔离级别，它确保同一事务内读取另外一个并发事务提交前后时的结果一样，而本事务提交后，重新开始事务操作时可以读到另外一个事务最新值，**mysql在可重复读事务级别中是通过MVCC机制避免不可重复读，通过间隙锁避免幻读。**

**(4)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。

## **四个事务隔离级别对应解决常见数据问题**

**（打勾表示会造成的问题，打叉表示可以解决的问题）**



## **快照读和当前读**

## **快照读：MVCC-多版本并发控制**

## **当前读：争夺锁（行锁、间隙锁、表锁）**

**常见操作场景：**

## **快照读：**执行select语句，默认开启事务，找到数据库中最新已提交的视图版本，作为本次事务周期内该sql的结果

## **当前读：**执行select for update/for lock，insert, update,delete语句，

开始事务，并确定当前执行语句锁范围（行级锁、间隙锁、表锁），判断是否存在已经获取锁但还未提交的其他事务，若存在，则当前事务就会阻塞，等待其他事务提交完释放锁，本事务就可以获取锁并读取到最新的数据。因为锁而会阻塞，索引当前读的效率会低于快照读。

**按加锁粒度分：**

**表锁**：MyIASM存储引擎支持表锁，InnoDB存储引擎支持表锁、间隙锁、行级锁，

Select for update语句where查询未匹配到索引时会触发表锁

**行锁：**使用了InnoDB引擎，select for update语句where查询匹配到索引时会触发行锁，给索引项加锁

**间隙锁：属于特殊的行锁，就是一定区间范围的多行锁，可以在可重复读事务级别下解决幻读问题**，一个事务进行索引项字段范围查询未命中记录时对条件字段索引项进行一段区间范围内的加锁，另一个并发事务涉及到该字段新增操作时，因为处于加锁范围内而进行阻塞状态，无法新增，从而保证查询事务内能避免幻读，这种范围锁就叫做间隙锁。

**按操作限制分：**

**共享锁：**并发事务时，写写互斥，读写互斥，读读不受限制

select  \*  from table ****lock in share mode****

**排他锁：**并发事务时，写写互斥，读写互斥，读读互斥

select  \*  from table where id=1 ****for update****

# **########## mysql的优化 ###############**

1. 循环写入操作改为批量一次写入
2. 查询要走索引

索引优化：

2.1将where条件和order by字段进行联合索引；

2.2 通过慢查询日志发现问题sql，添加合适的索引

2.3 遵循索引最左匹配原则

1. 连接配置优化
   1. 增加最大连接数max\_connection（默认是100）为1000，满足更改并发；
   2. 缓存池大小innodb\_buffer\_pool\_size调优，设置为机器内存大小的60%~80%；
   3. 日志缓冲区innodb\_log\_buffer\_size（默认8M）大小调优；