**数据库锁机制**

**锁：资源的并发访问控制**

**(1)锁的分类：**

**①操作类型分类：读锁/写锁**

**②锁定粒度分类：表锁/行锁**

**读锁(共享S锁)：针对同一份数据，多个读操作可以同时进行而不会相互阻塞。**

**写锁(排他X锁)：当前写操作没有完成前，它会阻塞其他写锁和读锁。**

**表锁：偏向MyISAM存储引擎，开销小，加锁快，无死锁，锁定粒度大，发生锁冲突 的概率最高，并发度最低。**

**行锁：InnoDB存储引擎，开销大，加锁慢，会出现死锁;锁定粒度小,发生锁冲突的概率 最低，并发度最高。**

**(2)SQL命令：**

**①查看表的锁定状态：show open tables;#In use = 0代表该表上没有锁**

**②手动增加表锁(不会自动释放，必须手动释放)：lock table 表名 read/write;**

**③手动释放表锁(不会自动释放，必须手动释放)：unlock tables;**

**注意：只会释放当前事务锁住的表**

**④查看系统上的表锁定情况：**

**show status like ‘table%’;**

**Table\_locks\_immediate：产生表级锁定的次数**

**Table\_locks\_waited：出现表级锁定导致的等待的次数，此值较高说明存在较为严 重的表级锁争抢情况**

**(3)存储引擎：**

**①MyISAM存储引擎：**

**1.MyISAM在执行查询语句前，会自动给涉及到的所有表加读锁(读完自动释放)，其 他线程可以读，写操作阻塞。在执行更新操作前，会自动给涉及到的表加写锁(写 完自动释放)，其他线程的读写操作均阻塞。阻塞将会一直持续到锁释放。**

**2.MyISAM适合偏读，并且对事务要求不高的场景(不支持事务)。锁调度时写优先， 即当表锁释放时，写操作优先获得锁。如果有大量的写操作将导致读操作很难获 得锁，从而造成永远阻塞。**

**②InnoDB存储引擎：**

**1.支持事务、行锁**