Atitit 稳定性与性能Mysql mssql oracle

INNODB行锁是通过给索引项加锁来实现的，即只有通过索引条件检索数据，InnoDB才使用行级锁，否则将使用表锁。

通过上面三个案例得知，InnoDB通过索引查询，使用的行锁，如果非索引查询会采用表锁。

InnoDB默认是行锁，前提条件是建立在索引之上的。如果筛选条件没有建立索引，会降级到表锁。即如果where条件中的字段都加了索引，则加的是行锁；否则加的是表锁。

InnoDB行锁是通过给索引上的索引项加锁来实现的。所以，只有通过索引条件检索数据，InnoDB才使用行级锁，否则，InnoDB将使用表锁。其他注意事项：

* 在不通过索引条件查询的时候，InnoDB使用的是表锁，而不是行锁。
* 由于MySQL的行锁是针对索引加的锁，不是针对记录加的锁，所以即使是访问不同行的记录，如果使用了相同的索引键，也是会出现锁冲突的。
* 当表有多个索引的时候，不同的事务可以使用不同的索引锁定不同的行，另外，不论是使用主键索引、唯一索引或普通索引，InnoDB都会使用行锁来对数据加锁。
* 即便在条件中使用了索引字段，但具体是否使用索引来检索数据是由MySQL通过判断不同执行计划的代价来决定的，如果MySQL认为全表扫描效率更高，比如对一些很小的表，它就不会使用索引，这种情况下InnoDB将使用表锁，而不是行锁。因此，在分析锁冲突时，别忘了检查SQL的执行计划，以确认是否真正使用了索引。

**Oracle**

在Oracle的每行数据上，都有一个标志位来表示该行数据是否被锁定。这样就大大减小了行级锁的维护开销，数据行上的锁标志一旦被置位，就表明该行数据被加X锁，Oracle在数据行上没有S锁。

### **sql server里update时,是行锁还是表锁**

udpate table a   
set column1 = 1   
where idx = 1  
---------------------  
如果你的idx是主键那么就是行级锁  
如果不是主键那么是页级索  
  
如果 where 条件那么就表级锁了