

# MyISAM 和 InnoDB

## 锁

MyISAM 支持表级锁，InnoDB 同时支持表级锁和行级锁。

表级锁的两种模式：

- **共享读锁**：MyISAM 执行查询语句前会自动给涉及到的所有表加上读锁。  
A 用户占有读锁的同时不会阻塞其他用户的读请求，但会阻塞其他用户的写请求；
- **独占写锁**：MyISAM 在执行更新、删除、新增操作时会自动给涉及到的所有表加上写锁。A 用户占有写锁的同时会阻塞其他用户的读写请求；

**注意：**

- MyISAM 也支持读写并发操作，可以通过修改变量 `concurrent_insert` 的值实现。设置为 0 时，不允许并发操作；设置为 1 时，支持一个用户读表的同时另一个用户在表的尾部插入数据；设置为 2 时，无论 MyISAM 表中有没有空洞（删除产生，执行 `OPTIMIZE TABLE` 语句来整理空间碎片），都允许在表尾插入记录，都允许在表尾并发插入记录；
- 可以通过 `LOCK TABLES` 显示的加锁，但是加锁以后不能访问未加锁的表。  
因为操作之前一次性获取所有的锁的缘故，避免了 MyISAM 表死锁的产生；
- MyISAM 的锁调度。默认情况下，**读的请求总是优于写的请求**，即使在写的请求先到的情况下，这也正是 **MyISAM 表不太适合于有大量更新操**

作和查询操作应用的原因，因为，大量的更新操作会造成查询操作很难获得读锁，从而可能永远阻塞；

InnoDB 行锁的加锁方式

- **共享锁**：允许两个事务同时读取同一行数据，但是不允许 A 事务拥有这一行共享锁时，B 事务请求这一行的排他锁；
- **排他锁**：A 事务获取排他锁时不允许其他事务请求获取相同数据行的共享和排他锁；

**注意：**

- 只有通过索引条件检索数据，InnoDB 才会使用行级锁，否则，InnoDB 将使用表锁；
- 间隙锁：InnoDB 会为范围条件而不是相等条件下的索引数据项加锁，即使是不存在的数据。例如 user 中共有 100 条数据，此时操作 id>99 的数据，即使 id=101 的数据不存在也会被加锁，这样避免了幻读的产生；
- InnoDB 中的锁是逐步获取的，因此可能会有死锁的产生，但是锁的粒度小，并发情况下性能好；

## 事务

MyISAM 不支持事务，InnoDB 支持事务。

## 外键

MyISAM 不支持，InnoDB 支持。