

InnoDB和MyISAM是许多人在使用MySQL时最常用的两个表类型，这两个表类型各有优劣，视具体应用而定。基本的差别为：MyISAM类型不支持事务处理等高级处理，而InnoDB类型支持。MyISAM类型的表强调的是性能，其执行速度比InnoDB类型更快，但是不提供事务支持，而InnoDB提供事务支持以及外部键等高级数据库功能。

以下是一些细节和具体实现的差别：

- 1.InnoDB不支持FULLTEXT类型的索引。
- 2.InnoDB 中不保存表的具体行数，也就是说，执行select count() from table时，InnoDB要扫描一遍整个表来计算有多少行，但是MyISAM只要简单的读出保存好的行数即可。注意的是，当count()语句包含 where条件时，两种表的操作是一样的。
- 3.对于AUTO\_INCREMENT类型的字段，InnoDB中必须包含只有该字段的索引，但是在MyISAM表中，可以和其他字段一起建立联合索引。
- 4.DELETE FROM table时，InnoDB不会重新建立表，而是一行一行的删除。
- 5.LOAD TABLE FROM MASTER操作对InnoDB是不起作用的，解决方法是首先把InnoDB表改成MyISAM表，导入数据后再改成InnoDB表，但是对于使用的额外的InnoDB特性(例如外键)的表不适用。
- 另外，InnoDB表的行锁也不是绝对的，假如在执行一个SQL语句时MySQL不能确定要扫描的范围，InnoDB表同样会锁全表，例如update table set num=1 where name like "%aaa%"

两种类型最主要的差别就是InnoDB 支持事务处理与外键和行级锁。而MyISAM不支持.所以MyISAM往往就容易被人认为只适合在小项目中使用。

作为使用MySQL的用户角度出发，InnoDB和MyISAM都是比较喜欢的，如果数据库平台要达到需求：99.9%的稳定性，方便的扩展性和高可用性来说的话，MyISAM绝对是首选。原因如下：

- 1、平台上承载的大部分项目是读多写少的项目，而MyISAM的读性能是比InnoDB强不少的。
- 2、MyISAM的索引和数据是分开的，并且索引是有压缩的，内存使用率就对应提高了不少。能加载更多索引，而InnoDB是索引和数据是紧密捆绑的，没有使用压缩从而会造成InnoDB比MyISAM体积庞大不小。
- 3、经常隔1，2个月就会发生应用开发人员不小心update一个表where写的范围不对，导致这个表没法正常用了，这个时候MyISAM的优越性就体现出来了，随便从当天拷贝的压缩包取出对应表的文件，随便放到一个数据库目录下，然后dump成sql再导回到主库，并把对应的binlog补上。如果是InnoDB，恐怕不可能有这么快速度，别和我说让InnoDB定期用导出xxx.sql机制备份，因为最小的一个数据库实例的数据量基本都是几十G大小。
- 4、从接触的应用逻辑来说，select count(\*) 和order by 是最频繁的，大概能占了整个sql总语句的60%以上的操作，而这种操作InnoDB其实也是会锁表的，很多人以为InnoDB是行级锁，那个只是where对它主键是有效，非主键的都会锁全表的。
- 5、还有就是经常有很多应用部门需要我给他们定期某些表的数据，MyISAM的话很方便，只要发给他们对那表的frm.MYD,MYI的文件，让他们自己在对应版本的数据库启动就行，而InnoDB就需要导出xxx.sql了，因为光给别人文件，受字典数据文件的影响，对方是无法使用的。
- 6、如果和MyISAM比insert写操作的话，InnoDB还达不到MyISAM的写性能，如果是针对基于索引的update操作，虽然MyISAM可能会逊色InnoDB,但是那么高并发的写，从库能否追的上也是一个问题，还不如通过多实例分库分表架构来解决。
- 7、如果是用MyISAM的话，merge引擎可以大大加快应用部门的开发速度，他们只要对这个merge表做一些select count(\*)操作，非常适合大项目总量约几亿的rows某一类型(如日志，调

查统计)的业务表。

当然Innodb也不是绝对不用，用事务的项目就用Innodb的。另外，可能有人会说你MyISAM无法抗太多写操作，但是可以通过架构来弥补。

天健JAVA教学部