**[14的路](http://www.cnblogs.com/vicenteforever/)**

[InnoDB与Myisam的六大区别](http://www.cnblogs.com/vicenteforever/articles/1613119.html)

|  |  |
| --- | --- |
| **MyISAM** | **InnoDB** |
| **构成上的区别：** | 每个MyISAM在磁盘上存储成三个文件。第一个文件的名字以表的名字开始，扩展名指出文件类型。    .frm文件存储表定义。    数据文件的扩展名为.MYD (MYData)。    索引文件的扩展名是.MYI (MYIndex)。 | 基于磁盘的资源是InnoDB表空间数据文件和它的日志文件，InnoDB 表的大小只受限于操作系统文件的大小，一般为 2GB |
| **事务处理上方面:** | MyISAM类型的表强调的是性能，其执行数度比InnoDB类型更快，但是不提供事务支持 | InnoDB提供事务支持事务，外部键等高级数据库功能 |
| **SELECT   UPDATE,INSERT，Delete操作** | 如果执行大量的SELECT，MyISAM是更好的选择 | **1.**如果你的数据执行大量的**INSERT或UPDATE**，出于性能方面的考虑，应该使用InnoDB表    **2.DELETE   FROM table**时，InnoDB不会重新建立表，而是一行一行的删除。    **3.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 |
| **对AUTO\_INCREMENT的操作** | 每表一个AUTO\_INCREMEN列的内部处理。    **MyISAM为INSERT和UPDATE操作自动更新这一列**。这使得AUTO\_INCREMENT列更快（至少10%）。在序列顶的值被删除之后就不能再利用。(当AUTO\_INCREMENT列被定义为多列索引的最后一列，可以出现重使用从序列顶部删除的值的情况）。    AUTO\_INCREMENT值可用ALTER TABLE或myisamch来重置    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    更好和更快的auto\_increment处理 | 如果你为一个表指定AUTO\_INCREMENT列，在数据词典里的InnoDB表句柄包含一个名为自动增长计数器的计数器，它被用在为该列赋新值。    自动增长计数器仅被存储在主内存中，而不是存在磁盘上    关于该计算器的算法实现，请参考    **AUTO\_INCREMENT列在InnoDB里如何工作** |
| **表的具体行数** | select count(\*) from table,MyISAM只要简单的读出保存好的行数，注意的是，当count(\*)语句包含   where条件时，两种表的操作是一样的 | 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 |
| **锁** | 表锁 | 提供行锁(locking on row level)，提供与 Oracle 类型一致的不加锁读取(non-locking read in    SELECTs)，另外，InnoDB表的行锁也不是绝对的，如果在执行一个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%” |