# 一、什么是存储引擎

关系数据库表是用于存储和组织信息的数据结构，可以将表理解为由行和列组成的表格，类似于Excel的电子表格的形式。有的表简单，有的表复杂，有的表根本不用来存储任何长期的数据，有的表读取时非常快，但是插入数据时却很差；而我们在实际开发过程中，就可能需要各种各样的表，不同的表，就意味着存储不同类型的数据，数据的处理上也会存在着差异，对于MySQL来说，它提供了很多种类型的存储引擎，我们可以根据对数据处理的需求，选择不同的存储引擎，从而最大限度的利用MySQL强大的功能。

# 二、各种存储引擎的特点

* 1. **MyISAM**

默认的MySQL插件式存储引擎，它是在Web、和其他应用环境下最常使用的存储引擎之一.支持全文索引、压缩、空间函数(GIS)等，但MyISAM不支持事务和行级锁，有一个毫无疑问的缺陷就是崩溃后无法安全恢复。

MyISAM表是独立于操作系统的，这说明可以轻松地将其从Windows服务器移植到Linux服务器；每当我们建立一个MyISAM引擎的表时，就会在本地磁盘上建立三个文件，文件名就是表名。例如，建立了一个MyISAM引擎的tb\_Demo表，那么就会生成以下三个文件：

* tb\_demo.frm，存储表定义；
* tb\_demo.MYD，存储数据；
* tb\_demo.MYI，存储索引。

MyISAM表支持三种不同的存储格式：

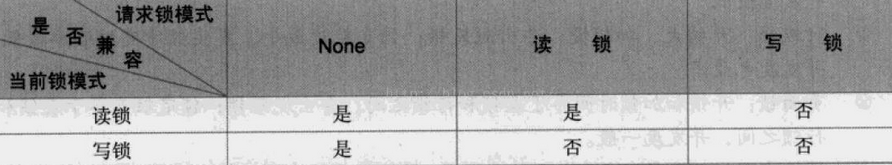
* 静态表：默认的存储格式，静态表的字段都是非变长字段。这样每个记录都是固定长度的。从而使存储迅速、容易缓存、出现故障易恢复，但由于静态表存储数据时会补足空格，所以他会比动态表占用更多空间。
* 动态表：包含变长字段，记录长度不固定，这样存储空间占用小，但频繁的更新和删除数据会出现碎片，需定期OPTIMIZE TABLE或myisamchk -r命令来改善性能，并且出现故障恢复相对比较难。
* 压缩表：由myisampack工具创建，磁盘占用很小，每个记录单独压缩，所以只有非常小的访问开支。

MyISAM表无法处理事务，这就意味着有事务处理需求的表，不能使用MyISAM存储引擎。MyISAM存储引擎特别适合在以下几种情况下使用：

* 选择密集型的表。MyISAM存储引擎在筛选大量数据时非常迅速，这是它最突出的优点。
* 插入密集型、并发相对较低的表。MyISAM的并发插入特性允许同时选择和插入数据。例如：MyISAM存储引擎很适合管理邮件或Web服务器日志数据、数据仓储等。

MyISAM支持两种表锁机制：表共享读和表独占写

* 共享读锁（S）之间是兼容的，但共享读锁（S）与排他写锁（X）之间，以及排他写锁（X）之间是互斥的，也就是说读和写是串行的。
* 在一定条件下，MyISAM 允许查询和插入并发执行，我们可以利用这一点来解决应用中对同一表查询和插入的锁争用问题。concurrent\_insert，专门用以控制其并发插入的行为，其值分别可以为0、1或2。
* MyISAM 默认的锁调度机制是写优先，这并不一定适合所有应用，用户可以通过设置LOW\_PRIORITY\_UPDATES 参数，或在INSERT、UPDATE、DELETE 语句中指定LOW\_PRIORITY 选项来调节读写锁的争用。
* 由于表锁的锁定粒度大，读写之间又是串行的，因此，如果更新操作较多，MyISAM表可能会出现严重的锁等待。



表级锁更适合于以查询为主，只有少量按索引条件更新数据的应用；

* 1. **InnoDB**

MySQL的默认事务型引擎，用于处理大量的短期(short-lived)事务；支持高并发，并且实现了四个标准的隔离级别；基于聚簇索引建立，聚簇索引对主键查询有很高的性能；存储格式是平台独立；支持真正的热备份，MySQL的其他存储引擎不支持热备份；支持行级锁定和外键约束。在InnoDB中，表数据文件本身就是按B+Tree组织的一个索引结构，这棵树的叶节点data域保存了完整的数据记录。这个索引的key是数据表的主键，因此InnoDB表数据文件本身就是主索引。

在 Innodb 存储引擎中，存在两种不同形式的索引，一种是 Cluster 形式的主键索引（ Primary Key ），另外一种则是和其他存储引擎（如 MyISAM 存储引擎）存放形式基本相同的普通 B-Tree 索引，这种索引在 Innodb 存储引擎中被称为 Secondary Index 。

InnoDB数据库表的本地存储结构如下：

* frm文件：存储数据表的框架结构，文件名与表名相同，每个表对应一个同名frm文件，与操作系统和存储引擎无关
* Ibd/ibdata文件：单表表空间文件，每个表使用一个表空间文件（file per table），存放用户数据库表数据和索引。独享表空间存储方式使用.ibd文件，并且每个表一个ibd文件;共享表空间存储方式使用.ibdata文件，所有表共同使用一个ibdata文件

除非有特别的原因，否则应该优先考虑InnoDB引擎。

在以下场合下，使用InnoDB是最理想的选择：

* 更新密集的表。InnoDB存储引擎特别适合处理多重并发的更新请求。
* 事务。InnoDB存储引擎是支持事务的标准MySQL存储引擎。
* 自动灾难恢复。与其它存储引擎不同，InnoDB表能够自动从灾难中恢复。
* 外键约束。MySQL支持外键的存储引擎只有InnoDB。

一般来说，如果需要事务支持，并且有较高的并发读取频率，InnoDB是不错的选择。

InnoDB实现了以下两种类型的行锁，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！：

* 共享锁（S）：允许一个事务去读一行，阻止其他事务获得相同数据集的排他锁。
* 排他锁（X)：允许获得排他锁的事务更新数据，阻止其他事务取得相同数据集的共享读锁和排他写锁。

为了允许行锁和表锁共存，实现多粒度锁机制，InnoDB还有两种内部使用的意向锁（Intention Locks），这两种意向锁都是表锁。

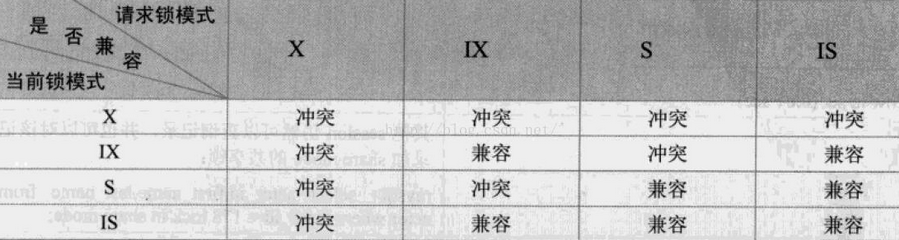
* 意向共享锁（IS）：事务打算给数据行加行共享锁，事务在给一个数据行加共享锁前必须先取得该表的IS锁。
* 意向排他锁（IX）：事务打算给数据行加行排他锁，事务在给一个数据行加排他锁前必须先取得该表的IX锁。

如果一个事务请求的锁模式与当前的锁兼容，InnoDB就将请求的锁授予该事务；反之，如果两者不兼容，该事务就要等待锁释放。

对于UPDATE、DELETE和INSERT语句，InnoDB会自动给涉及数据集加排他锁（X)；对于普通SELECT语句，InnoDB不会加任何锁；

事务可以通过以下语句显示给记录集加共享锁或排他锁。

* 共享锁（S）：SELECT \* FROM table\_name WHERE ... LOCK IN SHARE MODE。
* 排他锁（X)：SELECT \* FROM table\_name WHERE ... FOR UPDATE。



* 1. Memory

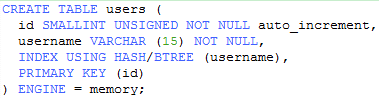
将所有数据保存在RAM中，在需要快速查找引用和其他类似数据的环境下，可提供极快的访问。虽然在内存中存储表数据确实会提供很高的性能，但当mysqld守护进程崩溃时，所有的Memory数据都会丢失。Memory引擎要求存储在Memory数据表里的数据使用的是长度不变的格式，即不能使用BLOB和TEXT这样的长度可变的数据类型，VARCHAR是一种长度可变的类型，但因为它在MySQL内部当做长度固定不变的CHAR类型，所以可以使用。

一般在以下几种情况下使用Memory存储引擎：

* 目标数据较小，而且被非常频繁地访问。在内存中存放数据，所以会造成内存的使用，可以通过参数max\_heap\_table\_size控制Memory表的大小，设置此参数，就可以限制Memory表的最大大小。
* 如果数据是临时的，而且要求必须立即可用，那么就可以存放在内存表中。
* 存储在Memory表中的数据如果突然丢失，不会对应用服务产生实质的负面影响。

Memory同时支持散列索引和B树索引。B树索引的优于散列索引的是，可以使用部分查询和通配查询，也可以使用<、>和>=等操作符方便数据挖掘。散列索引进行“相等比较”非常快，但是对“范围比较”的速度就慢多了，因此散列索引值适合使用在=和<>的操作符中，不适合在<或>操作符中，也同样不适合用在order by子句中。

可以在表创建时利用USING子句指定要使用的索引的版本，如：



* 1. Merge

MERGE存储引擎把一组MyISAM数据表当做一个逻辑单元来对待，让我们可以同时对他们进行查询。构成一个MERGE数据表结构的各成员MyISAM数据表必须具有完全一样的结构。每一个成员数据表的数据列必须按照同样的顺序定义同样的名字和类型，索引也必须按照同样的顺序和同样的方式定义。Merge表中并没有数据，对Merge类型的表可以进行查询、更新、删除操作，这些操作实际上是对内部的MyISAM表进行操作。

Merge存储引擎在本地会创建来弄个个文件：.frm文件存储表的定义；.MRG文件包含组合表的信息，包括MERGE表由哪些表组成，插入数据时的依据。

Merge存储引擎对于诸如数据仓储等VLDB环境十分适合。

* 1. Archive

归档的意思，在归档之后很多的高级功能就不再支持了，仅仅支持最基本的INSERT和SELECT操作，在MySQL5.1之前不支持索引；适合日志和数据采集类应用；支持行级锁和专用的缓存区，所以可以实现高并发的插入，但它不是一个事务型的引擎，而是一个针对高速插入和压缩做了优化的简单引擎。

1. 最常使用的2种存储引擎

## 3.1 Myisam特点

Myisam是Mysql的默认存储引擎。当create创建新表时，未指定新表的存储引擎时，默认使用Myisam。每个MyISAM在磁盘上存储成三个文件。文件名都和表名相同，扩展名分别是.frm（存储表定义）、.MYD (MYData，存储数据)、.MYI (MYIndex，存储索引)。数据文件和索引文件可以放置在不同的目录，平均分布io，获得更快的速度。

## 3.2 InnoDB特点

InnoDB存储引擎提供了具有提交、回滚和崩溃恢复能力的事务安全。但是对比Myisam的存储引擎，InnoDB写的处理效率差一些并且会占用更多的磁盘空间以保留数据和索引

1. 如何选择合适的存储引擎

**选择标准：**根据应用特点选择合适的存储引擎，对于复杂的应用系统可以根据实际情况选择多种存储引擎进行组合。选择标准可以分为：

* 是否需要支持事务；
* 是否需要使用热备；
* 崩溃恢复：能否接受崩溃；
* 是否需要外键支持；

然后按照标准，选择对应的存储引擎即可。

