

## 存储引擎

查看MySQL提供的所有存储引擎

```
mysql> show engines;
```

查看MySQL当前默认的存储引擎

```
mysql> show variables like '%storage_engine%';
```

查看表的存储引擎

```
show table status like "table_name" ;
```

### • MyISAM和InnoDB区别

5.5版本之前，MyISAM是MySQL的默认数据库引擎。MyISAM不支持事务和行级锁，而且最大的缺陷就是崩溃后无法安全恢复。

5.5版本之后，默认的存储引擎为InnoDB，事务性数据库引擎。在某些情况下使用 MyISAM 也是合适的比如读密集的情况下。

对比：

1. 是否支持行级锁：MyISAM 只有表级锁(table-level locking)，而InnoDB 支持行级锁(row-level locking)和表级锁,默认为行级锁。
2. 是否支持事务和崩溃后的安全恢复：

MyISAM 强调的是性能，每次查询具有原子性，其执行速度比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。 具有事务(commit)、回滚(rollback)和崩溃修复能力(crash recovery capabilities)的事务安全(transaction-safe (ACID compliant))型表。

3. 是否支持外键： MyISAM不支持，而InnoDB支持。
4. 是否支持MVCC：仅 InnoDB 支持。应对高并发事务, MVCC比单纯的加锁更高效;MVCC只在 READ COMMITTED 和 REPEATABLE

READ 两个隔离级别下工作;MVCC可以使用 乐观(optimistic)锁 和 悲观(pessimistic)锁来实现;各数据库中MVCC实现并不统一。

## 索引

MySQL索引使用的数据结构主要有BTree索引 和 哈希索引 。对于哈希索引，底层的数据结构就是哈希表，因此在绝大多数需求为单条记录查询的时候，可以选择哈希索引，查询性能最快；其余大部分场景，建议选择BTree索引。

MySQL的BTree索引使用的是B树中的B+Tree，但对于主要的两种存储引擎的实现方式是不同的。

- MyISAM: B+Tree叶节点的data域存放的是数据记录的地址。在索引检索的时候，首先按照B+Tree搜索算法搜索索引，如果指定的Key存在，则取出其 data 域的值，然后以 data 域的值作为地址读取相应的数据记录。这被称为“非聚簇索引”。
- InnoDB: 其数据文件本身就是索引文件。相比MyISAM，索引文件和数据文件是分离的，其表数据文件本身就是按B+Tree组织的一个索引结构，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。这被称为“聚簇索引（或聚集索引）”。而其余的索引都作为辅助索引，辅助索引的data域存储相应记录主键的值而不是地址，这也是和MyISAM不同的地方。在根据主索引搜索时，直接找到key所在的节点即可取出数据；在根据辅助索引查找时，则需要先取出主键的值，再走一遍主索引。因此，在设计表的时候，不建议使用过长的字段作为主键，也不建议使用非单调的字段作为主键，这样会造成主索引频繁分裂。 PS：整理自《Java工程师修炼之道》

