MySQL区别于其他数据库的最重要的一个特点就是插件式的表存储引擎，也就是说存储引擎是基于表的。

存储引擎的概念是MySQL里面才有的，不是所有的关系型数据库都有存储引擎这个概念 。其它数据库系统（包括大多数商业选择)仅支持一种类型的数据存储， 也就是说采用“ 一个尺码满足一切需求 ”的存储方式，也意味着“功能强大，性能平庸”。而MySQL默认配置了许多不同的存储引擎，你可以根据业务需求选取一种最适配最高效的存储引擎。这也是为什么MySQL为何如此受欢迎的主要原因之一。

官网5.7版本支持的10种存储引擎：

MyISAM： 拥有较高的插入，查询速度，但不支持事务
InnoDB ：5.5.8版本后Mysql的默认数据库引擎，支持ACID事务，支持行级锁定
BDB： 源自Berkeley DB，事务型数据库的另一种选择，支持COMMIT和ROLLBACK等其他事务特性
Memory ：所有数据置于内存的存储引擎，拥有极高的插入，更新和查询效率。但是会占用和数据量成正比的内存空间。并且其内容会在Mysql重新启动时丢失
Merge ：将一定数量的MyISAM表联合而成一个整体，在超大规模数据存储时很有用
Archive ：非常适合存储大量的独立的，作为历史记录的数据。因为它们不经常被读取。Archive拥有高效的插入速度，但其对查询的支持相对较差
Federated： 将不同的Mysql服务器联合起来，逻辑上组成一个完整的数据库。非常适合分布式应用
Cluster/NDB ：高冗余的存储引擎，用多台数据机器联合提供服务以提高整体性能和安全性。适合数据量大，安全和性能要求高的应用
CSV： 逻辑上由逗号分割数据的存储引擎。它会在数据库子目录里为每个数据表创建一个.CSV文件。这是一种普通文本文件，每个数据行占用一个文本行。CSV存储引擎不支持索引。
BlackHole ：黑洞引擎，写入的任何数据都会消失，一般用于记录binlog做复制的中继

InnoDB: MySQL 默认的存储引擎，支持事务、MVCC、外键、行级锁和自增列。

MyISAM: 支持全文索引、压缩、空间函数、表级锁，不支持事务，插入速度快。

Memory: 数据都在内存中，数据的处理速度快，但是安全性不高。

Notes:
在服务器中实现，而不是在存储引擎中。

只有使用压缩行格式时，才支持压缩的MyISAM表。使用MyISAM压缩行格式的表是只读的。

在服务器端通过加密功能实现。

在服务器端通过加密功能实现;在MySQL 5.7和更高版本中，支持数据静止表空间加密。

在MySQL Cluster NDB 7.3和更高版本中支持外键。

InnoDB在MySQL 5.6和更高版本中提供对 全文索引（FULLTEXT ） 的支持。

InnoDB在MySQL 5.7和更高版本中提供对地理空间索引的支持。

InnoDB内部利用哈希索引来实现自适应哈希索引特性。

下文主要介绍InnoDB MyISAM Memory三种存储引擎，以下是三者简要特性对比

功能	MyISAM	MEMORY	InnoDB
存储限制	256TB	RAM	64TB
支持事务	No	No	Yes
支持全文索引	Yes	No	No
支持B树索引	Yes	Yes	Yes
支持哈希索引	No	Yes	No
支持集群索引	No	No	Yes
支持数据索引	No	Yes	Yes
支持数据压缩	Yes	No	No
空间使用率	低	N/A	高
支持外键	No	No	Yes

InnoDB引擎

InnoDB 是一个事务安全的存储引擎，它具备提交、回滚以及崩溃恢复的功能以保护用户数据。InnoDB 的行级别锁定保证数据一致性提升了它的多用户并发数以及性能。InnoDB 将用

户数据存储在聚集索引中以减少基于主键的普通查询所带来的 I/O 开销。为了保证数据的完整性，InnoDB 还支持外键约束。默认使用B+TREE数据结构存储索引。

InnoDB引擎特点：

- 支持事务，支持4个事务隔离（ACID）级别
- 行级锁定（更新时锁定当前行）
- 读写阻塞与事务隔离级别相关
- 既能缓存索引又能缓存数据
- 支持外键
- InnoDB更消耗资源，读取速度没有MyISAM快
- 在InnoDB中存在着缓冲管理，通过缓冲池，将索引和数据全部缓存起来，加快查询的速度；
- 对于InnoDB类型的表，其数据的物理组织形式是聚簇表。所有的数据按照主键来组织。数据和索引放在一块，都位于B+数的叶子节点上；

InnoDB引擎调优

- 主键尽可能小，否则会给Secondary index带来负担
- 避免全表扫描，这会造成锁表
- 尽可能缓存所有的索引和数据，减少IO操作
- 避免主键更新，这会造成大量的数据移动

MyISAM引擎

MyISAM既不支持事务、也不支持外键、其优势是访问速度快，但是表级别的锁定限制了它在读写负载方面的性能，因此它经常应用于只读或者以读为主的数据场景。默认使用B+TREE数据结构存储索引。

MyISAM引擎特点：

- 不支持事务
- 表级锁定（更新时锁定整个表）
- 读写互相阻塞（写入时阻塞读入、读时阻塞写入；但是读不会互相阻塞）
- 只会缓存索引（通过key_buffer_size缓存索引，但是不会缓存数据）
- 不支持外键
- 读取速度快

MyISAM引擎调优

- 设置合适索引

- 启用延迟写入，尽量一次大批量写入，而非频繁写入
- 尽量顺序insert数据，让数据写入到尾部，减少阻塞
- 降低并发数，高并发使用排队机制
- MyISAM的count只有全表扫描比较高效，带有其它条件都需要进行实际数据访问

Memory引擎

在内存中创建表。每个MEMORY表只实际对应一个磁盘文件(frm 表结构文件)。MEMORY类型的表访问非常得快，因为它的数据是放在内存中的，并且默认使用HASH索引。要记住，在用完表格之后就删除表格，不然一直占据内存空间。

Memory引擎特点：

- 支持的数据类型有限制，比如：不支持TEXT和BLOB类型（长度不固定），对于字符串类型的数据，只支持固定长度的行，VARCHAR会被自动存储为CHAR类型；
- 支持的锁粒度为表级锁。所以，在访问量比较大时，表级锁会成为MEMORY存储引擎的瓶颈；
- 由于数据是存放在内存中，一旦服务器出现故障，数据都会丢失；
- 查询的时候，如果有用到临时表，而且临时表中有BLOB，TEXT类型的字段，那么这个临时表就会转化为MyISAM类型的表，性能会急剧降低；
- 默认使用hash索引。
- 如果一个内部表很大，会转化为磁盘表。