



链滴

Mysql 核心之存储引擎

作者: [rxz](#)

原文链接: <https://ld246.com/article/1505965111025>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

存储引擎是什么？

例如，如果你在研究大量的临时数据，你也许需要使用内存存储引擎。内存存储引擎能够在内存中存所有的表格数据。又或者，你也许需要一个支持[事务处理]的数据库(以确保事务处理不成功时数据的退能力)。

这些不同的技术以及配套的相关功能在MySQL中被称作存储引擎(也称作表类型)。MySQL默认配置许多不同的存储引擎，可以预先设置或者在MySQL服务器中启用。你可以选择适用于服务器、数据和表格的存储引擎，以便在选择如何存储你的信息、如何检索这些信息以及你需要你的数据结合什么能和功能的时候为你提供最大的灵活性。

选择如何存储和检索你的数据的这种灵活性是MySQL为什么如此受欢迎的主要原因。其它数据库系统支持一种类型的数据存储。遗憾的是，其它类型的数据库解决方案采取的“一个尺码满足一切需求”方式意味着你要么就牺牲一些性能，要么你就用几个小时甚至几天的时间详细调整你的数据库。使用MySQL，我们仅需要修改我们使用的存储引擎就可以了 -- 百度百科

总的来说，存储引擎是数据组织数据的方式，不同的存储引擎，能够满足不同的业务需求，同时存在同方面的性能差异，选择合适的存储引擎对于不同的数据的存储要求提供一种合适的存储策略，对于据的操作来说，是事半功倍的。

Mysql的常用的存储引擎及特点

通过终端命令：`SHOW ENGINES \G`我们可以查看数据库支持的存储引擎，Mysql5.6支持9种存储引擎，如下查询结果，通过表格组织。

```
mysql> SHOW ENGINES \G
***** 1. row *****
      Engine: FEDERATED          存储引擎名
      Support: NO                数据库是否支持此引擎
      Comment: Federated MySQL storage engine 简介
      Transactions: NULL        事务支持
      XA: NULL                   分布式事务
      Savepoints: NULL
***** 2. row *****
      Engine: MRG_MYISAM
      Support: YES
      Comment: Collection of identical MyISAM tables
      Transactions: NO
      XA: NO
      Savepoints: NO
***** 3. row *****
      Engine: MyISAM
      Support: YES
      Comment: MyISAM storage engine
      Transactions: NO
      XA: NO
      Savepoints: NO
***** 4. row *****
      Engine: BLACKHOLE
      Support: YES
      Comment: /dev/null storage engine (anything you write to it disappears)
      Transactions: NO
      XA: NO
```

```

Savepoints: NO
***** 5. row *****
  Engine: CSV
  Support: YES
  Comment: CSV storage engine
Transactions: NO
  XA: NO
Savepoints: NO
***** 6. row *****
  Engine: MEMORY
  Support: YES
  Comment: Hash based, stored in memory, useful for temporary tables
Transactions: NO
  XA: NO
Savepoints: NO
***** 7. row *****
  Engine: ARCHIVE
  Support: YES
  Comment: Archive storage engine
Transactions: NO
  XA: NO
Savepoints: NO
***** 8. row *****
  Engine: InnoDB
  Support: DEFAULT
  Comment: Supports transactions, row-level locking, and foreign keys
Transactions: YES
  XA: YES
Savepoints: YES
***** 9. row *****
  Engine: PERFORMANCE_SCHEMA
  Support: YES
  Comment: Performance Schema
Transactions: NO
  XA: NO
Savepoints: NO

```

看到这么多，你肯能有点迷了，这么多存储引擎，到底该如何选择，到底有什么用。其实常用的就那几种，最容易遇到（考到）的也就那么两三种。也许你除了InnoDB其他的都不知道那也没关系，因在数据库的快速发展中，很多存储引擎支撑的业务，正在逐渐被某些数据库替代，这些引擎基本不会用到，术业有专攻就是这个道理，目前的Mysql默认引擎是InnoDB，InnoDB特点是对事务的支持，种索引结构的支持，而我们大多数的业务都离不开事务。

以下是集中存储引擎的简介：

摘自[百度百科]：

- 1、**MyISAM**：拥有较高的插入，查询速度，但不支持事务
- 2、**InnoDB**：5.5版本后Mysql的默认数据库，事务型数据库的首选引擎，支持ACID事务，支持行级定
- 3、**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。但是会占用和数量成正比的内存空间。并且其内容会在Mysql重新启动时丢失
- 4、**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

5、**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有高效的插入速度，但其对查询的支持相对较差

6、**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[分布式用]

7、**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。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。

8、**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与MyISAM存储引擎的比较

名称	事务支持	存储限制	锁机制	索引支持	缓存支持	数据压缩	空间使用	内存使用	插入速度
InnoDB	支持	64T	行锁	B树/Hash/集群	数据/索引	支持	高	高	慢
MyISAM	不支持	不限	表锁	B树/全文检索	索引	不支持	低	低	快

以上两种是Mysql最常用的存储引擎，特点也很明显，MyISAM不支持事务，那么他就不支持并发的Update，但是它支持高负载的查询，他的表锁支持他的并发插入。还有一个特点是它支持全文检索。like %xxx%这种操作是十分耗时的，而且性能低下。而全文检索就可以一用来支持高效模糊匹配。

指定存储引擎 <操作>

在创建表的时候可以指定存储引擎，在test数据库中创建tetsuser和testuser2两个表如下：

```
USE test;
```

```
CREATE TABLE `testuser` (  
  `old` INT(10) NOT NULL,  
  `username` INT(11) NOT NULL,  
  `usage` INT(11) NOT NULL,  
  PRIMARY KEY (`old`)  
) ENGINE=INNODB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `testuser2` (  
  `old` VARCHAR(10) NOT NULL,  
  `username` INT(11) NOT NULL,  
  `usage` INT(11) NOT NULL,  
  PRIMARY KEY (`old`)  
) ENGINE=MYISAM DEFAULT CHARSET=utf8;
```

查看表信息：

```

mysql> SHOW TABLE STATUS FROM test WHERE NAME="testuser" \G
mysql> SHOW TABLE STATUS FROM test WHERE NAME="testuser2" \G
***** 1. row *****2.row*****
      Name: testuser                testuser2
      Engine: InnoDB                MyISAM
      Version: 10                    10
      Row_format: Compact            Dynamic
      Rows: 0                        0
      Avg_row_length: 0              0
      Data_length: 16384              0
      Max_data_length: 0              281474976710655
      Index_length: 0                1024
      Data_free: 0                   0
      Auto_increment: NULL           NULL
      Create_time: 2017-09-21 11:07:35
      Update_time: NULL
      Check_time: NULL
      Collation: utf8_general_ci
      Checksum: NULL
      Create_options:
      Comment:

```

从上面的表状态可以稍微的了解到不同存储引擎所带来的差别，具体的差异，可以通过测试，进行验证
 ..这里就略过...