1. 存储引擎
2. 定义
3. MyISAM
4. InnoDB
5. MEMORY
6. MERGE
7. ARCHIVE
8. 选取存储引擎的标准
9. 存储引擎
10. 定义

MySQL中的数据用各种不同的技术存储在文件(或者内存)中。这些技术中的每一种技术都使用**不同的存储机制、索引技巧、锁定水平并且最终提供广泛的不同的功能和能力**。MySQL默认配置了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、数据库和表格的存储引擎，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性能和功能的时候为你提供最大的灵活性。

1. MyISAM（适合多读取插入，少更新删除）
2. MyISAM不支持事务，也不支持外键，尤其是访问速度快，对事务完整性没有要求或者以SELECT、INSERT为主的应用基本都可以使用这个引擎来创建表。
3. 每个MyISAM在磁盘上存储成3个文件，其中文件名和表名都相同，但是扩展名分别为：
4. .frm(存储表定义)
5. MYD(MYData，存储数据)
6. MYI(MYIndex，存储索引)
7. **数据文件和索引文件可以放置在不同的目录，平均分配IO，获取更快的速度**。要指定数据文件和索引文件的路径，需要在创建表的时候通过DATA DIRECTORY和INDEX DIRECTORY语句指定，文件路径需要使用绝对路径。
8. **每个MyISAM表都有一个标志，服务器或myisamchk程序在检查MyISAM数据表时会对这个标志进行设置。MyISAM表还有一个标志用来表明该数据表在上次使用后是不是被正常的关闭了。**如果服务器以为当机或崩溃，这个标志可以用来判断数据表是否需要检查和修复。如果想让这种检查自动进行，可以在启动服务器时使用--myisam-recover现象。这会让服务器在每次打开一个MyISAM数据表是自动检查数据表的标志并进行必要的修复处理。MyISAM类型的表可能会损坏，可以使用CHECK TABLE语句来检查MyISAM表的健康，并用REPAIR TABLE语句修复一个损坏到MyISAM表。
9. MyISAM的表还支持3种不同的存储格式：
10. 静态(固定长度)表
11. 动态表
12. 压缩表
13. 其中**静态表是默认的存储格式**。静态表中的字段都是非变长字段，这样每个记录都是固定长度的，这种存储方式的优点是存储非常迅速，容易缓存，出现故障容易恢复；缺点是占用的空间通常比动态表多。静态表在数据存储时会根据列定义的宽度定义补足空格，但是在访问的时候并不会得到这些空格，这些空格在返回给应用之前已经去掉。同时需要注意：在某些情况下可能需要返回字段后的空格，而使用这种格式时后面到空格会被自动处理掉。
14. **动态表包含变长字段，记录不是固定长度的，这样存储的优点是占用空间较少**，但是频繁到更新删除记录会产生碎片，需要定期执行OPTIMIZE TABLE语句或myisamchk -r命令来改善性能，并且出现故障的时候恢复相对比较困难。
15. 压缩表由myisamchk工具创建，占据非常小的空间，因为每条记录都是被单独压缩的，所以只有非常小的访问开支。
16. InnoDB（比较慢，并发，索引，事务）

InnoDB是一个健壮的事务型存储引擎，这种存储引擎已经被很多互联网公司使用，为用户操作非常大的数据存储提供了一个强大的解决方案。InnoDB还引入了行级锁定和外键约束，在以下场合下，使用InnoDB是最理想的选择：

(1) 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。

(2) 事务。InnoDB存储引擎是支持事务的标准MySQL存储引擎。

(3) 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。

(4) 外键约束。MySQL支持外键的存储引擎只有InnoDB。

(5) 支持自动增加列AUTO\_INCREMENT属性。

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。

1. MEMORY（内存表）

使用MySQL Memory存储引擎的出发点是速度。为得到最快的响应时间，采用的逻辑存储介质是系统内存。虽然在内存中存储表数据确实会提供很高的性能，但当mysql守护进程崩溃时，所有的Memory数据都会丢失。获得速度的同时也带来了一些缺陷。它要求存储在Memory数据表里的数据使用的是长度不变的格式，这意味着不能使用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。

1. 一般在以下几种情况下使用Memory存储引擎：
2. **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。
3. 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。
4. 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。
5. Memory同时支持散列索引(HASH)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。
6. MERGE
7. MERGE存储引擎是一组MyISAM表的组合，这些MyISAM表结构必须完全相同，尽管其使用不如其它引擎突出，但是在某些情况下非常有用。说白了，Merge表就是几个相同MyISAM表的聚合器；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存储引擎的使用场景。
8. 对于服务器日志这种信息，一般常用的存储策略是将数据分成很多表，每个名称与特定的时间端相关。例如：可以用12个相同的表来存储服务器日志数据，每个表用对应各个月份的名字来命名。当有必要基于所有12个日志表的数据来生成报表，这意味着需要编写并更新多表查询，以反映这些表中的信息。与其编写这些可能出现错误的查询，不如将这些表合并起来使用一条查询，之后再删除Merge表，而不影响原来的数据，删除Merge表只是删除Merge表的定义，对内部的表没有任何影响。
9. ARCHIVE

Archive是归档的意思，在归档之后很多的高级功能就不再支持了，仅仅支持最基本的插入和查询两种功能。在MySQL 5.5版以前，Archive是不支持索引，但是在MySQL 5.5以后的版本中就开始支持索引了。Archive拥有很好的压缩机制，它使用zlib压缩库，在记录被请求时会实时压缩，所以它经常被用来当做仓库使用。

1. 选取存储引擎的标准
2. 是否需要支持事务；
3. 是否需要使用热备；
4. 崩溃恢复：能否接受崩溃；
5. 是否需要外键支持；

