Atitit.数据库存储引擎的原理与attilax 总结

[1. 存储引擎是什么 1](#_Toc26351)

[2. 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 2](#_Toc1270)

[3. 表的存储有三个文件：结构+数据+索引 2](#_Toc7214)

[4. 页和字段 2](#_Toc18319)

[5. 数据存取的选择：行存储还是列存储？ 3](#_Toc25470)

[6. 常见的存储引擎 3](#_Toc8686)

[6.1. 简单类型MyISAM、 3](#_Toc10899)

[6.2. 复杂类型，支持事务与外键 MySQL存储引擎【InnoDB、 3](#_Toc5307)

[6.3. InnoDB数据存储结构 3](#_Toc6703)

[6.4. Memory】(Heap) 存储引擎 5](#_Toc3452)

[6.5. NDBCluster分布式存储引擎 6](#_Toc31305)

[7. other 6](#_Toc10930)

[7.1. 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。 6](#_Toc5181)

[7.2. 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 7](#_Toc2490)

[7.3. 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 7](#_Toc27544)

[7.4. Federated存储引擎 7](#_Toc13550)

[8. 参考 7](#_Toc10823)

# 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储所有的表格数据。又或者，你也许需要一个支持[事务处理](C:/view/709594.htm" \t "C:/Users/Administrator/Desktop/db/_blank)的数据库(以确保事务处理不成功时数据的回退能力)。

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。

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](C:/view/7809.htm" \t "C:/Users/Administrator/Desktop/db/_blank)(包括大多数商业选择)仅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”的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[1]

  存储引擎说白了就是如何存储数据、如何为存储的数据建立索引和如何更新、查询数据等技术的实现方法。因为在关系数据库中数据的存储是以表的形式存储的，所以存储引擎也可以称为表类型（即存储和操作此表的类型）。

作者:: 老哇的爪子 Attilax 艾龙， EMAIL:1466519819@qq.com

转载请注明来源： <http://blog.csdn.net/attilax>

# 其它[数据库系统](C:/view/7809.htm" \t "C:/Users/Administrator/Desktop/db/_blank)(包括大多数商业选择)仅支持一种类型的数据存储

在Oracle 和SQL Server等数据库中只有一种存储引擎，所有数据存储管理机制都是一样的。而MySql数据库提供了多种存储引擎。用户可以根据不同的需求为数据表选择不同的存储引擎，用户也可以根据自己的需要编写自己的存储引擎。

另外，Mysql的存储引擎接口定义良好。有兴趣的开发者通过阅读文档编写自己的存储引擎

# 表的存储有三个文件：结构+数据+索引

.frm文件，存储表的结构

.myd文件，存储表的数据

.myi文件，存储表的索引

# 页和字段

SQL Server 中的基本数据存储单元是页

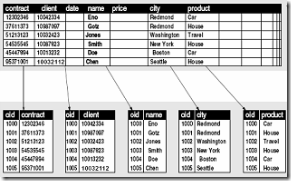
数据页包括数据行中除text、ntext和 image以外的所有数据类型，text、ntext和 image是保存在单独的页中的 。数据行是连续地放在页上，紧接在页头之后。一个行偏移的表开始于页的末尾。

行偏移表包含页上每一行的一个条目。每一个条目记录了该行的第一个字节距离页的开始的距离。行偏移表中的条目的顺序与页上行的顺序是相反的。在SQL Server 7.0中，行不能够跨越页，

# 数据存取的选择：行存储还是列存储？

其实列存储并不是什么新概念，早在1985年SIGMOD会议上就有文章” A decomposition storage model”对DSM(decomposition storage model)做了比较详细的介绍，而Sybase更在2004年左右就推出了列存储的Sybase IQ数据库系统(见200年VLDB文章” Sybase iq multiplex - designed for analytics”)，主要用于在线分析、数据挖掘等查询密集型应用。

列存储，缩写为DSM，相对于NSM(N-ary storage model)，其主要区别在于，DSM将所有记录中相同字段的数据聚合存储，而NSM将每条记录的所有字段的数据聚合存储，如下图所示：

[](http://images.cnblogs.com/cnblogs_com/happyy/WindowsLiveWriter/a4ece460b63d_11529/clip_image002_2.gif)

列存储有什么优点？

就我目前比较肤浅的理解，列存储的主要优点有两个：

1) 每个字段的数据聚集存储，在查询只需要少数几个字段的时候，能大大减少读取的数据量，据C-Store, MonetDB的作者调查和分析，查询密集型应用的特点之一就是查询一般只关心少数几个字段，而相对应的，NSM中每次必须读取整条记录；

2) 既然是一个字段的数据聚集存储，那就更容易为这种聚集存储设计更好的压缩/解压算法。

列存储适合用在什么场合？

OLAP，数据仓库，数据挖掘等查询密集型应用。当然，列存储数据库并不是说完全不能进行更新操作，其实它们的更新操作性能并不是很差，一般也够用，但是一方面不如自己的查询性能，另外一方面也不如Oracle这种专门搞OLTP的数据库，所以一般就不提这个。

列存储不适合用在什么场合？

相对来说，不适合用在OLTP，或者更新操作，尤其是插入、删除操作频繁的场合。

为啥上世纪80年代就出现的概念现在又重新炒起来了呢？

2005年VLDB有篇文章(“One Size Fits All - An Idea Whose Time Has Come and Gone”)，就是那个老牛M. Stonebraker写的，明确指出，时代变了，指望一个数据库产品就统一天下的日子已经一去不复还了。于是，这个老牛在2005年左右做了C-Store，一个列存储的数据库原型系统，在VLDB, SIGMOD等顶级国际会议上灌了几桶水后，

## 4发展史

　　有篇文章(“One Size Fits All - An Idea Whose Time Has Come and Gone”)，就是那个老牛M. Stonebraker写的，明确指出，时代变了，指望一个数据库产品就统一天下的日子已经一去不复还了。于是，这个老牛在2005年左右做了C-Store，一个列存储的数据库原型系统，在VLDB, SIGMOD等顶级国际会议上灌了几桶水后，拉了一伙人出去开了个公司叫Vertica，将其商业化，专注于数据仓库、在线分析等市场，最近貌似还挺红火的;顺便说一下，为了贯彻上面的思想，这个老牛在同一时期又做了H-Store，一个主内存数据库原型系统，没怎么灌水就又招呼了一帮人开了个公司叫VoltDB，将其商业化，专注于联机事务处理。

M. Stonebraker在上世纪70年代带头开展关系数据库管理系统的实现工作，做出来了Ingres，其中灌水无数，从这个原型系统基础上产生了很多商业数据库软件，包括 Sybase、Microsoft SQL Server、NonStop SQL、Informix 等，而所谓的最先进的开源数据库系统PostgreSQL也是Ingres的一个后继分支。

# 常见的存储引擎

## 简单类型**MyISAM、**

3.4：MYISAM存储引擎的优缺点：

优点：占用空间小，处理速度快（相对InnoDB来说）

确定：不支持事务的完整性和并发性

表级锁定：其锁定机制是表级索引，这虽然可以让锁定的实现成本很小但是也同时大大降低了其并发性能

MyISAM的Count只有在全表扫描的时候特别高效，带有其他条件的count都需要进行实际的数据访问

## 复杂类型，支持事务与外键 **MySQL存储引擎【InnoDB、**

## **InnoDB数据存储结构**

1.MySQL将所有数据都逻辑地存放在ib\_data1文件中，我们称之为表空间。当然，你也可以一个表对应一个物理文件,将innodb\_file\_per\_table设置成ON即可。  
2.表空间又划为成段，有数据段(leaf node segment)，索引段(none-leaf node segment)，回滚段(rollback segment)。表空间由这些段和页组成，比如32页碎片页。  
3.每段又划为成区，InnoDB每次最多可以申请4个区，即4M的存储空间。  
4.每个区又划为成页，一个区划分成64页，每个页的大小是16KB，大小不能够改，这也固定了一个区的大小为4M。页是MySQL操作的最小逻辑单位。  
5.InnoDB是面向行的，这就意味着数据行存放在页中，每页最多能记录7992行数据。  
6.MySQL定义了不同作用的页类型，比如B-Tree Page, Undo Log Page等，我们最关心的是B-Tree Page(数据页)。实际数据就以这样的页逻辑实体存在于表空间,总是以B+树结构索引组织的。  
7.换句话就说，实际数据一行一行地存放在B-Tree页中，这些页都放在数据段leaf node segment中。B-Tree Page是B+树的叶子节点。  
8.一个B-Tree树，由7部分构成

* 8-1.File Header，这里记录了页在表空间的一些信息，比如上一页，下一页，属于哪个表空间等等
* 8-2.Page Header, 这里记录了页本身的一些存储信息。比如第一个记录的位置，记录数，最后插入记录行的位置，该页的索引ID等等
* 8-3.Infimum & Supermum Records, MySQL虚拟的二个行记录，用来界定记录的边界。分别代表此页中任何pk值还小的值和任何pk值还大的值。
* 8-4.user records, 实际存储的行记录。
* 8-5.free space,空闲空间，同样是链表结构。当一个数据记录删除后，就会加入到空闲链表中
* 8-6.page directory, 存放了记录的相对位置。注:聚集索引本身找不到具体的一条记录。而是通过 聚集索引找到该记录所在的页，然后再通过Page Directory进行二分查找找到具体数据。
* 8-7.File Trailer, MySQL InnoDB利用它来保证页完整地写入磁盘。

上图中主要有以下几种存储文件：

1）表结构定义文件(.frm)：MySQL里每个表和每个视图都有一个对应的.frm文件用于记录表和视图的定义。注意：该文件和存储引擎无关，属于MySQL数据库本身的文件。

2）默认表空间文件(ibdata)

3）单独表空间文件(.ibd)

## Memory】(Heap) 存储引擎

4.1：Memory存储引擎的文件存储形式

Memory存储引擎也会再磁盘上形成一个 .frm的表结构文件，只是表的数据件并不以文件的形式存放在磁盘上。鉴于其数据存放在内存里，因此，访问速度更快。但需 要考虑的是：内存上数据的持久性。

4.2：Memory存储引擎的索引类型

默认的是哈希索引，也支持BTree索引

## ****NDBCluster****分布式存储引擎

1.特性

　　分布式：分布式存储引擎，可以由多个NDBCluster存储引擎组成集群分别存放整体数据的一部分

　　支持事务：和Innodb一样，支持事务

　　可与mysqld不在一台主机：可以和mysqld分开存在于独立的主机上，然后通过网络和mysqld通信交互

　　内存需求量巨大：新版本索引以及被索引的数据必须存放在内存中，老版本所有数据和索引必须存在与内存中

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

# other

## [CSV](C:/view/468993.htm" \t "C:/Users/Administrator/Desktop/db/_blank)： 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。

## 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

正如其名称所示，Archive非常适合存储归档数据，如日志信息。它只支持INSERT和SELECT操作，其设计的主要目的是提供高速的插入和压缩功能。

## Federated存储引擎

Federated存储引擎不存放数据，它至少指向一台远程MySQL数据库服务器上的表，非常类似于Oracle的透明网关

# 参考

MySQL存储引擎【InnoDB、MyISAM、Memory】 - mysql数据库栏目 - 红黑联盟

MySQ 存储引擎选择 - ggjucheng - 博客园.html

存储引擎\_百度百科.html

（ imp ）mysql 数据结构详解-singerinsky-ChinaUnix博客.html

（imp）MySQL数据库的体系架构 - dongdong\_java的专栏 - 博客频道 - CSDN.NET.html

SQLServer数据库中存储引擎深入探讨-mssql-电脑编程网.html

（imp ）大数据存取的选择：行存储还是列存储？\_存储\_比特网.html