MySql架构与存储引擎

T A H N K Y O U F O R W A T C H I N G

主讲老师Deer：2957339855

课程咨询依娜老师：2470523467

目　录

CONTENTS

MySql逻辑架构

连接层

服务层

引擎层

存储层

存储引擎

MyISAM

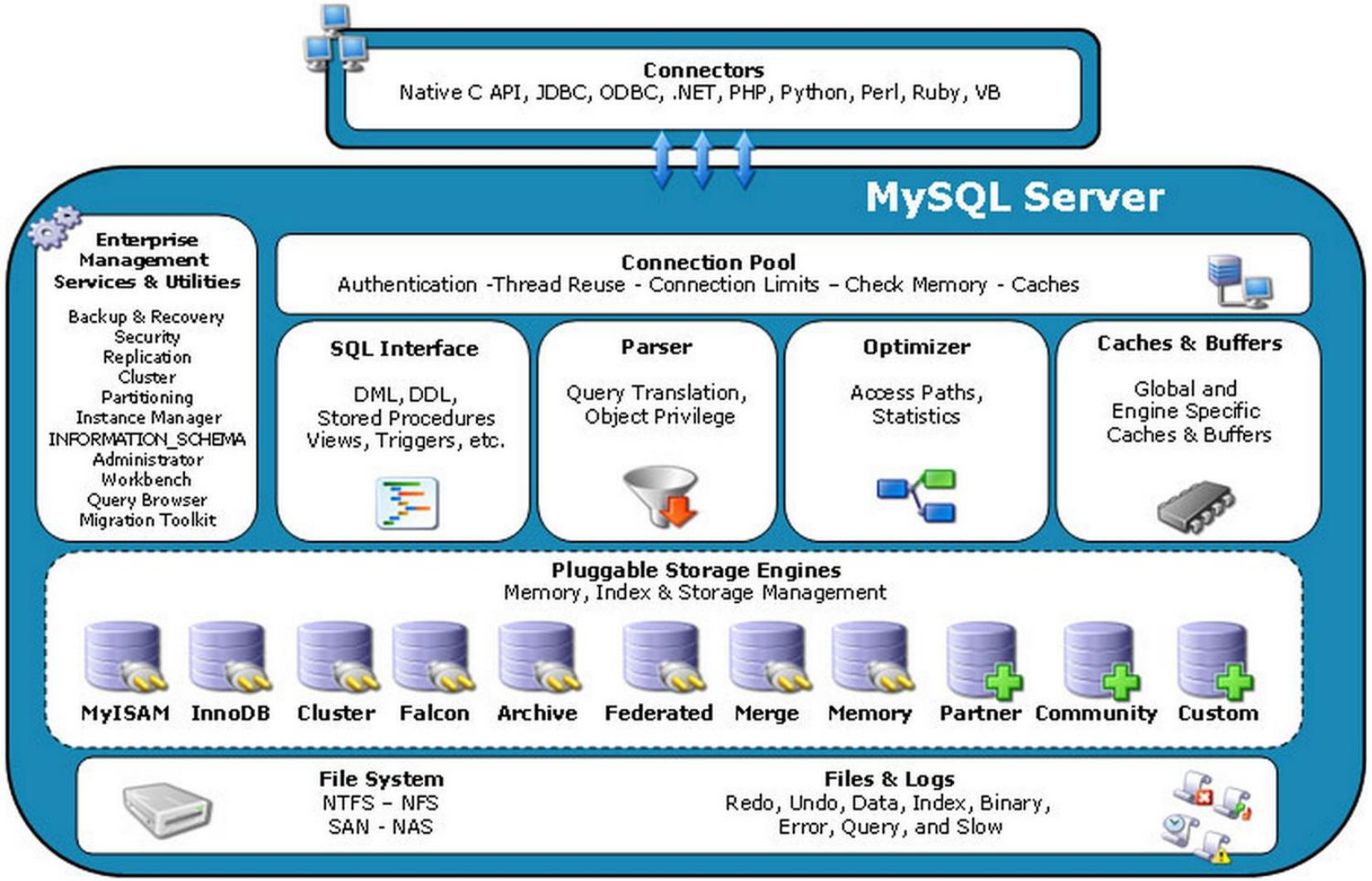
Innodb

Archive

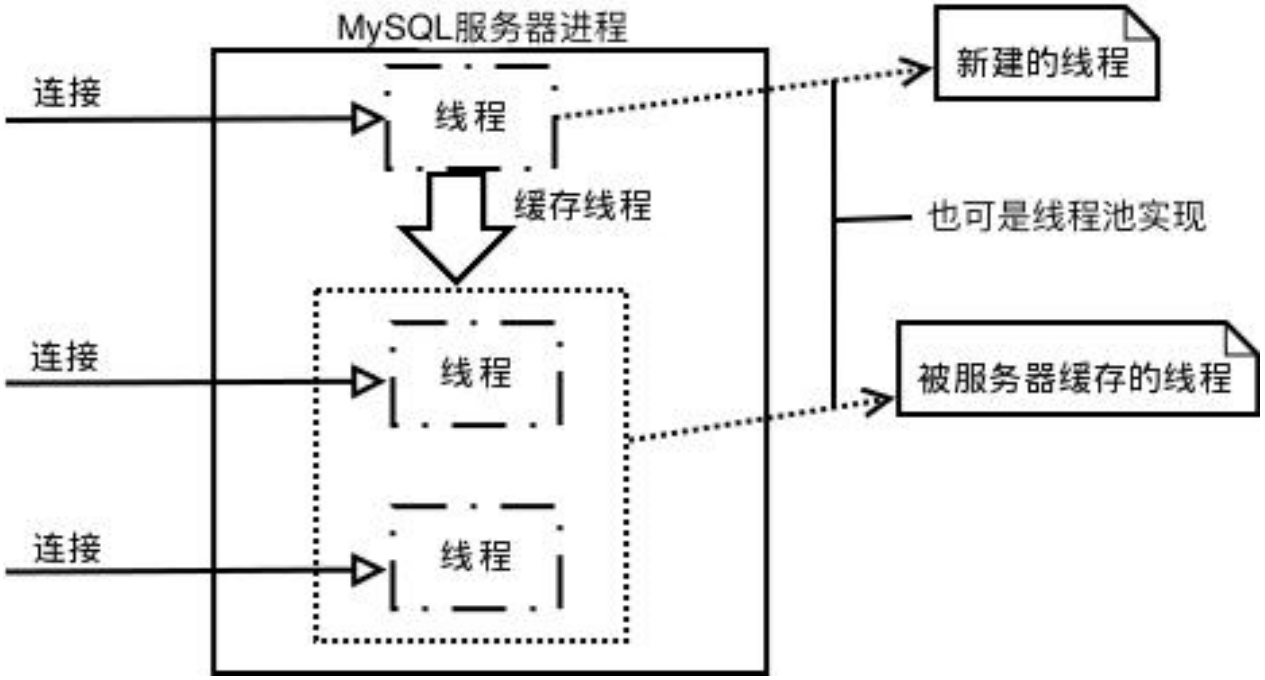
Memory

Federated

MySql逻辑架构

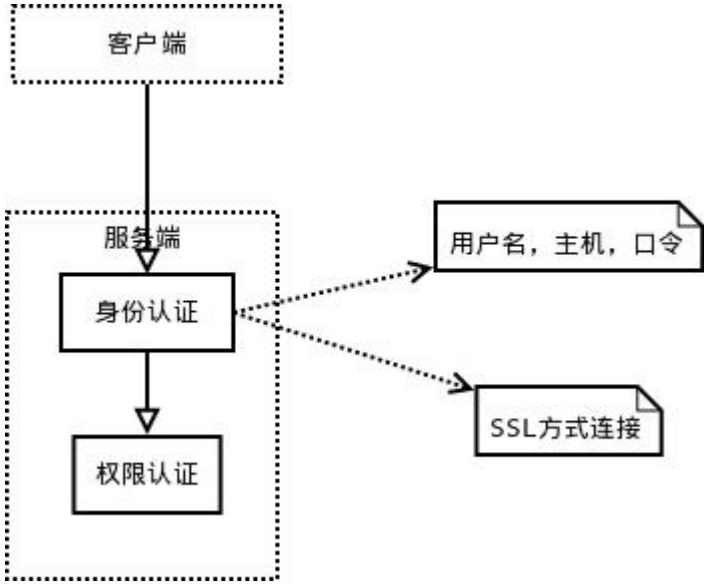

Connectors
Native C API, JDBC, ODBC, .NET, PHP, Python, Perl, Ruby, VB
MySQL Server
Enterprise Management Services & Utilities
Backup & Recovery
Security
Replication
Cluster
Partitioning
Instance Manager
INFORMATION_SCHEMA
Administrator
Workbench
Query Browser
Migration Toolkit
Connection Pool
Authentication -Thread Reuse - Connection Limits – Check Memory - Caches
SQL Interface
DML, DDL, Stored Procedures Views, Triggers, etc.
Parser
Query Translation, Object Privilege
Optimizer
Access Paths, Statistics
Caches & Buffers
Global and Engine Specific Caches & Buffers
Pluggable Storage Engines
Memory, Index & Storage Management
MyISAM
InnoDB
Cluster
Falcon
Archive
Federated
Merge
Memory
Partner
Community
Custom
File System
NTFS – NFS
SAN - NAS
Files & Logs
Redo, Undo, Data, Index, Binary, Error, Query, and Slow

MySql逻辑架构-连接层



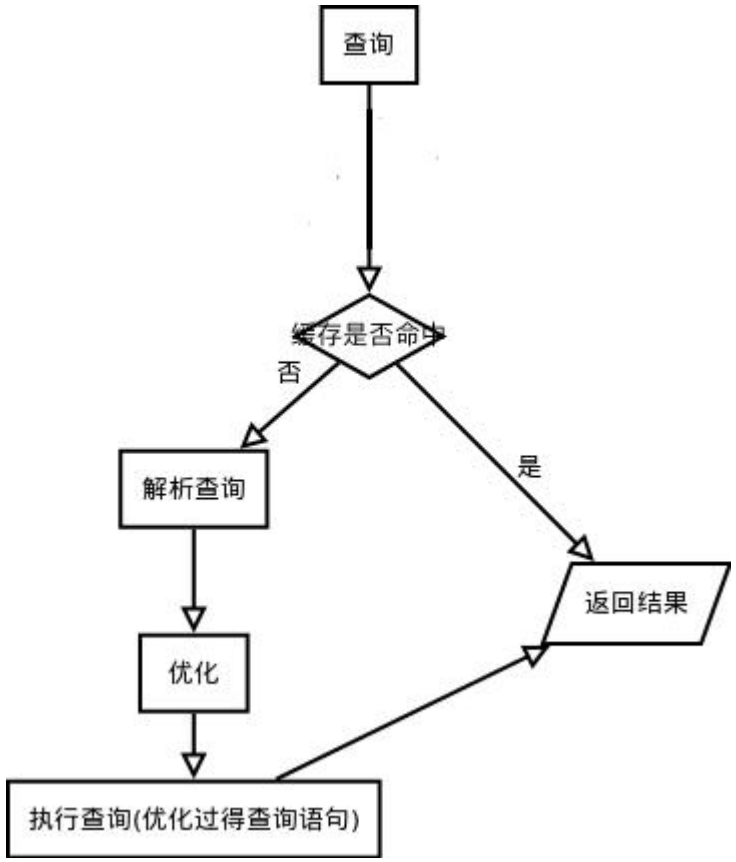
当MySQL启动（MySQL服务器就是一个进程），等待客户端连接，每一个客户端连接请求，服务器都会新建一个线程处理（如果是线程池的话，则是分配一个空的线程），每个线程独立，拥有各自的内存处理空间，但是，如果这个请求只是查询，没关系，但是若是修改数据，很显然，当两个线程修改同一块内存是会引发数据同步问题的。

MySql逻辑架构-SQL处理层



连接到服务器，服务器需要对其进行验证，也就是用户名、IP、密码验证，一旦连接成功，还要验证是否具有执行某个特定查询的权限（例如，是否允许客户端对某个数据库某个表的某个操作）

MySql逻辑架构-SQL处理层



这一层主要功能有：SQL语句的解析、优化，缓存的查询，MySQL内置函数的实现，跨存储引擎功能（所谓跨存储引擎就是说每个引擎都需提供的功能（引擎需对外提供接口）），例如：存储过程、触发器、视图等。

1.如果是查询语句（select语句），首先会查询缓存是否已有相应结果，有则返回结果，无则进行下一步（如果不是查询语句，同样调到下一步）

2.解析查询，创建一个内部数据结构（解析树），这个解析树主要用来SQL语句的语义与语法解析；

3.优化：优化SQL语句，例如重写查询，决定表的读取顺序，以及选择需要的索引等。这一阶段用户是可以查询的，查询服务器优化器是如何进行优化的，便于用户重构查询和修改相关配置，达到最优化。这一阶段还涉及到存储引擎，优化器会询问存储引擎，比如某个操作的开销信息、是否对特定索引有查询优化等。

MySql逻辑架构-缓存

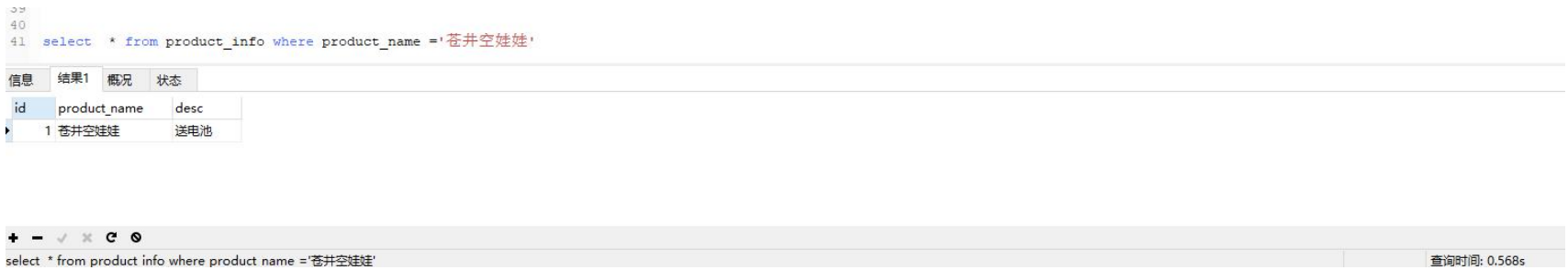
show variables like '%query_cache_type%'

C:\ProgramData\MySQL\MySQL Server 5.6

SET GLOBAL query_cache_size = 4000;

SET GLOBAL query_cache_size = 134217728;

select * from product_info where product_name ='苍井空娃娃'



MySql逻辑架构-解析查询

```
SELECT DISTINCT
    < select_list >
FROM
    < left_table > < join_type >
JOIN < right_table > ON < join_condition >
WHERE
    < where_condition >
GROUP BY
    < group_by_list >
HAVING
    < having_condition >
ORDER BY
    < order_by_condition >
LIMIT < limit_number >
```

MySql逻辑架构-解析查询

MySql逻辑架构-优化

```
1  EXPLAIN
2  select  * from product_info where product_name ='苍井空娃娃'
```

信息 | 结果1 | 概况 | 状态

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	product_i	ALL	(Null)	(Null)	(Null)	(Null)	2091355	Using where

```
1  EXPLAIN
2  select  * from product_info where 1=1
```

信息 | 结果1 | 概况 | 状态

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	product_i	ALL	(Null)	(Null)	(Null)	(Null)	2091355	(Null)

```
1  EXPLAIN
2  select  * from product_info where id = null
```

信息 | 结果1 | 概况 | 状态

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	(Null)	(Null)	(Null)	(Null)	(Null)	(Null)	(Null)	Impossible WHERE

目　录

CONTENTS

MySql逻辑架构

连接层

服务层

引擎层

存储层

存储引擎

MyISAM

Innodb

Archive

Memory

Federated

存储引擎

```
#看你的mysql现在已提供什么存储引擎:
 mysql> show engines;

#看你的mysql当前默认的存储引擎:
 mysql> show variables like '%storage_engine%';
```

存储引擎-MyISAM

MySql 5.5之前默认的存储引擎

MyISAM 存储引擎由MYD和MYI组成

```
1  show create table testmysam
```

信息 | 结果1 | 概况 | 状态

Table	testmysam
Create Table	CREATE TABLE `testmysam` (`1` varchar(255) DEFAULT NULL) ENGINE=MyISAM DEFAULT CHARSET=utf8

testmysam.frm	2018/8/21 22:41	FRM 文件	9 KB
testmysam.MYD	2018/8/21 22:41	MYD 文件	0 KB
testmysam.MYI	2018/8/21 22:41	MYI 文件	1 KB

存储引擎-MyISAM

特性：

并发性与锁级别-表级锁

支持全文检索

支持数据压缩

myisampack -b -f testmysam.MYI

存储引擎-MyISAM

适用场景：
非事务型应用（数据仓库，报表，日志数据）
只读类应用
空间类应用（空间函数，坐标）

MySQL5
空间扩展.docx

存储引擎-Innodb

MySql 5.5以及以后版本默认存储引擎

innodb_file_per_table

ON:独立的表空间：tablename.ibd

OFF:系统表空间：ibdataX

```
show create table testdemo
```

信息 | 结果1 | 概况 | 状态

Table	testdemo
Create Table	CREATE TABLE `testdemo` (`id` int(255) NOT NULL, `c1` varchar(300) DEFAULT NULL, `c2` int(50) DEFAULT NULL,

testdemo.frm	2018/8/20 20:36	FRM 文件	9 KB
testdemo.ibd	2018/8/20 20:36	IBD 文件	112 KB

```
CREATE TABLE `testdemo2` (
  `id` int(255) NOT NULL,
  `c1` varchar(300) DEFAULT NULL,
  `c2` int(50) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_c2` (`c2`) USING HASH
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

比电脑 › 本地磁盘 (C:) › ProgramData › MySQL › MySQL Server 5.6 › data

名称	修改日期	类型	大小
demo	2018/7/24 20:16	文件夹	
mysql	2018/7/17 14:15	文件夹	
mysqldemo	2018/8/22 10:32	文件夹	
performance_schema	2018/7/17 14:15	文件夹	
sakila	2018/7/17 14:16	文件夹	
test	2018/8/1 11:48	文件夹	
world	2018/7/17 14:16	文件夹	
auto.cnf	2018/7/17 14:16	CNF 文件	1 KB
DESKTOP-2EKGEE5.err	2018/8/21 22:03	ERR 文件	109 KB
DESKTOP-2EKGEE5.pid	2018/8/21 22:03	PID 文件	1 KB
DESKTOP-2EKGEE5-slow.log	2018/8/21 19:19	文本文档	6 KB
ib_logfile0	2018/8/22 10:32	文件	49,152 KB
ib_logfile1	2018/8/21 22:51	文件	49,152 KB
ibdata1	2018/8/22 10:32	文件	77,824 KB
testdemo.frm	2018/8/20 20:36	FRM 文件	9 KB
testdemo.ibd	2018/8/20 20:36	IBD 文件	112 KB
testdemo2.frm	2018/8/22 10:32	FRM 文件	9 KB
testmysam.frm	2018/8/21 22:41	FRM 文件	9 KB
testmysam.MYD	2018/8/21 22:41	MYD 文件	0 KB

存储引擎-Innodb

mysql5.6以前默认为系统表空间
系统表空间和独立表空间

- 系统表空间无法简单的收缩文件大小
- 独立表空间可以通过optimize table 收缩系统文件
- 系统表空间会产生IO瓶颈
- 独立表空间可以同时向多个文件刷新数据

- 建议：Innodb使用独立表空间

存储引擎-Innodb

特性

Innodb是一种事务性存储引擎

完全支持事务得ACID特性

Redo Log 和 Undo Log

Innodb支持行级锁（并发程度更高）

适用场景

Innodb适合于大多数OLTP应用

存储引擎

对比项	MyISAM	InnoDB
主外键	不支持	支持
事务	不支持	支持
行表锁	表锁，即使操作一条记录也会锁住整个表 不适合高并发的操作	行锁,操作时只锁某一行，不对其它行有影响 适合高并发的操作
缓存	只缓存索引，不缓存真实数据	不仅缓存索引还要缓存真实数据，对内存要求较高，而且内存大小对性能有决定性的影响
表空间	小	大
关注点	性能	事务
默认安装	Y	Y

存储引擎-CSV

组成

数据以文本方式存储在文件

.csv文件存储内容

.csm文件存储表得元数据如表状态和数据量

.frm 表结构

存储引擎-CSV

特点

以csv格式进行数据存储

所有列都不能为null的

不支持索引（不适合大表，不适合在线处理）

可以对数据文件直接编辑（保存文本文件内容）

mycsv.CSV | new 1

```
1,"aaa","bbb"
2,"cccc","dddd"
```

```
create table mycsv(id int,c1 VARCHAR(10),c2 char(10)) engine=csv;
```

信息 | 概况 | 状态

[SQL]create table mycsv(id int,c1 VARCHAR(10),c2 char(10)) engine=csv;

[Err] 1178 - The storage engine for the table doesn't support nullable columns

```
create index idx_id on mycsv(id)
```

信息 | 概况 | 状态

SQL]create index idx_id on mycsv(id)

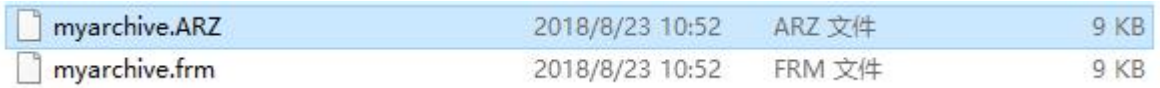
Err] 1069 - Too many keys specified; max 0 keys allowed

存储引擎-Archive

组成

以zlib对表数据进行压缩，磁盘I/O更少

数据存储在ARZ为后缀的文件中



特点：

只支持insert和select操作

只允许在自增ID列上加索引

```
10  update myarchive set c1='aaa' where id = 1
```

信息 概况 状态

[SQL]update myarchive set c1='aaa' where id = 1

[Err] 1031 - Table storage engine for 'myarchive' doesn't have this option

```
8   delete from myarchive where id = 1
```

信息 概况 状态

[SQL]delete from myarchive where id = 1

[Err] 1031 - Table storage engine for 'myarchive' doesn't have this option

```
3   create index idx_c1 on myarchive(c1)
```

信息 概况 状态

[SQL]create index idx_c1 on myarchive(c1)

[Err] 1069 - Too many keys specified; max 1 keys allowed

存储引擎-Archive

使用场景

日志和数据采集应用

存储引擎-Memory

- 文件系统存储特点
  也称HEAP存储引擎，所以数据保存在内存中
- 支持HASH索引和BTree索引
- 所有字段都是固定长度 varchar(10) = char(10)
- 不支持Blog和Text等大字段
- Memory存储引擎使用表级锁
- 最大大小由max_heap_table_size参数决定

存储引擎-Memory

```
1  create table mymemory(id int,c1 varchar(10),c2 char(10),c3 text) engine = memory;
```

信息 | 概况 | 状态

```
[SQL]create table mymemory(id int,c1 varchar(10),c2 char(10),c3 text) engine = memory;
[Err] 1163 - The used table type doesn't support BLOB/TEXT columns
```

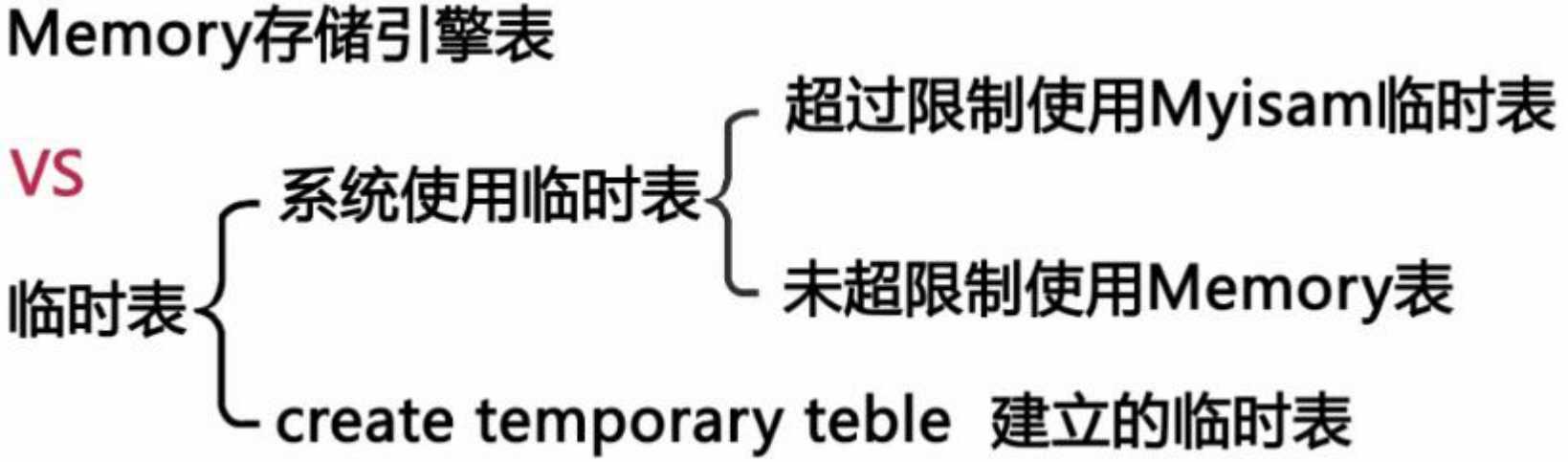
```
1  show TABLE status LIKE 'mymemory'
```

信息 | 结果1 | 概况 | 状态

Name	Engine	Version	Row_format	Rows	Avg_row_length	Da
mymemory	MEMORY	10	Fixed	0	66	0

存储引擎-Memory

容易混淆的概念



存储引擎-Memory

使用场景

- hash索引用于查找或者是映射表（邮编和地区的对应表）
- 用于保存数据分析中产生的中间表
- 用于缓存周期性聚合数据的结果表

memory数据易丢失，所以要求数据可再生

存储引擎-Ferderated

特点

- 提供了访问远程MySQL服务器上表的方法
- 本地不存储数据，数据全部放到远程服务器上
- 本地需要保存表结构和远程服务器的连接信息

使用场景

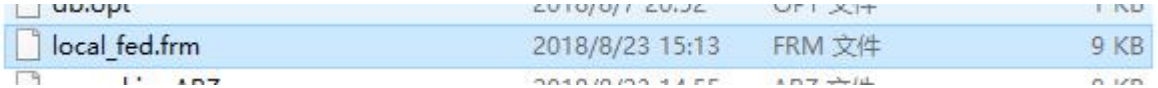
偶尔的统计分析及手工查询

存储引擎-Ferderated

如何使用

默认禁止，启用需要再启动时增加federated参数

mysql://user_name[:password]@hostname[:port_num]/db_name/table_name




```
CREATE TABLE `remote_fed` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `c1` varchar(10) NOT NULL DEFAULT '',
  `c2` char(10) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```