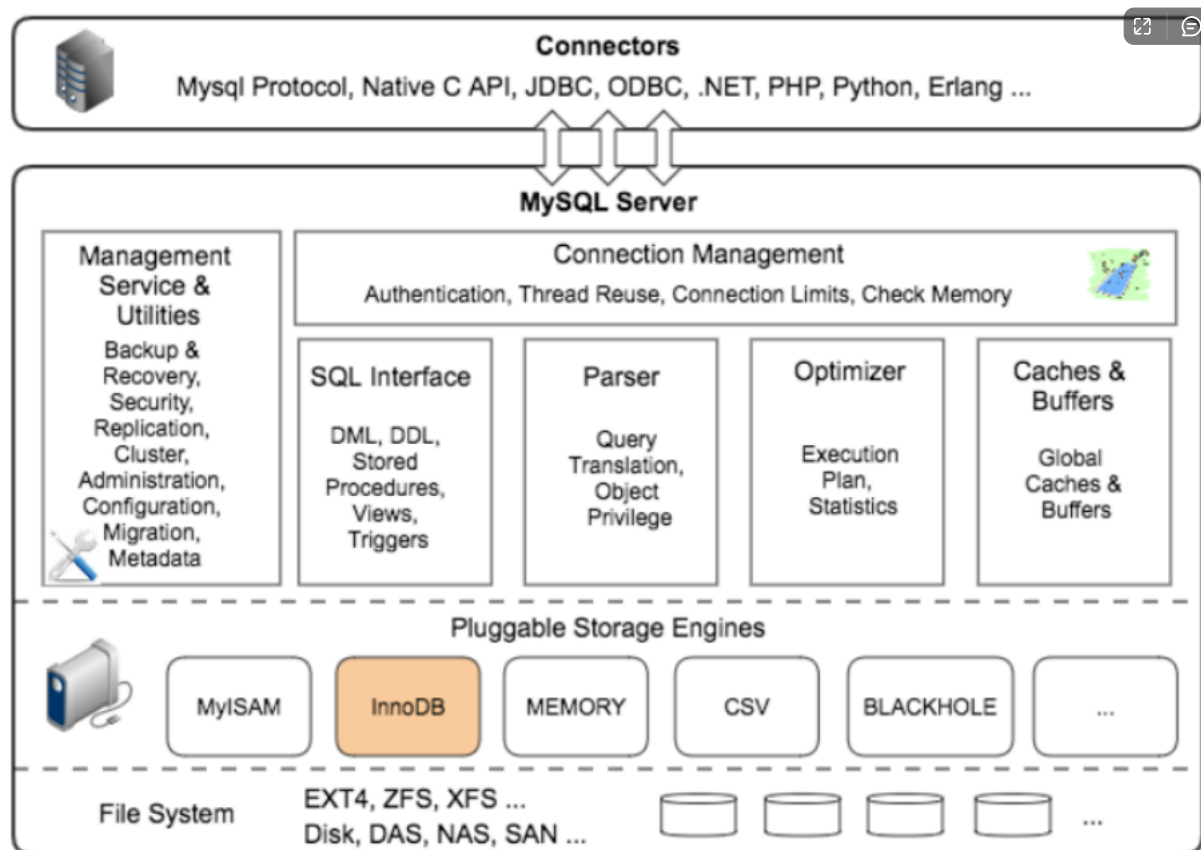
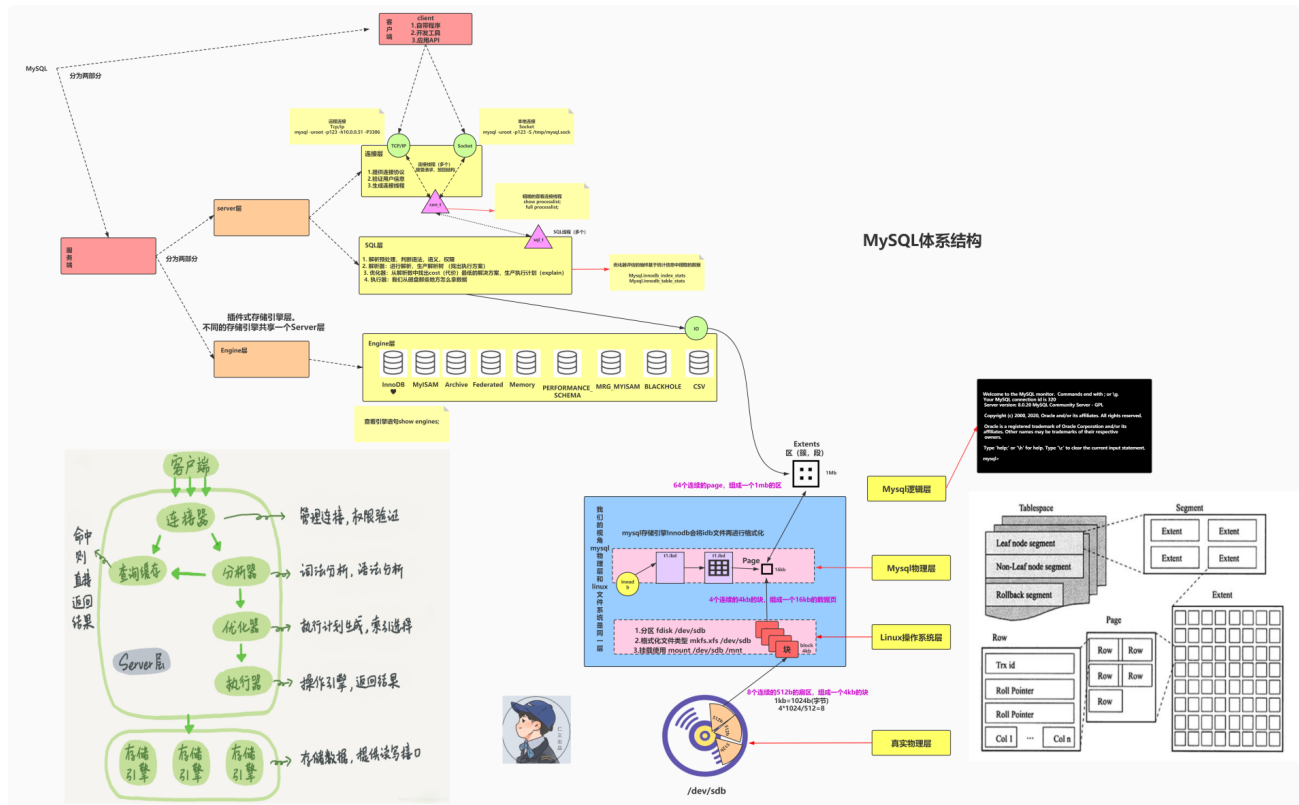


3.MySQL 体系结构√

1.mysql体系结构图（CS结构）



自己画的图



2.体系架构中各个层面功能文字概述

体系结构分为两个：客户端+服务端 (cs结构)

2.1 connectors (连接器) 客户端

1.#MySQL Protocol

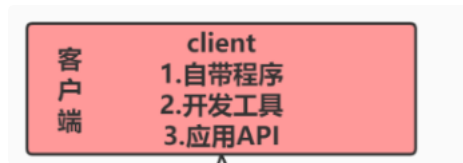
Socket (Unix Socket)

TCP/IP

2.#API

Native API

C、PHP、JDBC、ODBC、.NET、Python、Go...



服务层分为四个：管理服务 and 实用程序+连接层+sql层+存储引擎层

2.2 mysql server 服务端

2.2.1 Management Service && Utilities (管理服务 and 实用程序)功能!!!

Metadata (NEW!!) ---元数据
 Plugins(NEW!!) --- 插件
 Security (NEW!!) --- 安全
 Backup && Restore (NEW !!) ---备份和还原
 Replication (NEW!!) --- 复制
 Cluster(NEW!!) --- 集权
 Administration --- 管理
 Configuration --- 配置
 Migration --- 迁移

2.2.2 Connection Management (连接管理) 连接层 功能!!!

a.Authentication (NEW!!) --- 认证,验证用户信息

在数据库系统启动时，自动加载授权表内存

mysql.user
 mysql.db
 mysql.table_priv
 mysql.proc_priv
 mysql.column_priv
 user : 保存全库的授权信息，用户、密码、密码插件、地址来源
 db : 记录各个账号在各个数据库上的操作权限
 tables_priv : 单表级别的授权信息
 columns_priv : 记录数据列级别的操作权限
 proxies_priv : 代理授权信息
 procs_priv : 存储过程授权信息

在进行有关用户更改时，授权表会进行刷新

create user
 alter user
 drop user
 grant user
 revoke user

flush privileges (8.0版本之前更改用户权限，要及时刷新，保持内存和磁盘的一致性)

也可以通过update,insert,delete更改表的方式，手动更新授权表

b.Thread Reuse --- 专用线程（服务于每一个会话）

相关参数：

show processlist; 是瞬时我们可以查看的方式select * from information_schema.processlist; 通过这张表可以进行信息统计

c.Connection Limit --- 连接线程

连接限制参数：

select @@max_connections; 同一时间并发最大连接数(默认151)

select @@max_connect_errors; 错误连接数上限（默认100）

查询数据库连接会话线程id对应的系统id

我们以数据库中id=9为例

第一步：在数据库中 show processlist 查询线程id

```
mysql> show processlist;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | User          | Host      | db      | Command | Time | State          | Info          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 5 | event_scheduler | localhost | NULL    | Daemon  | 4903 | Waiting on empty queue | NULL         |
| 9 | root           | localhost | mysql   | Query   | 0    | starting      | show processlist |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

第二步：提取系统库performance_schema下面threads表中 processlist_id和thread_os_id两列的信息。数据库id=9对应操作系统中的1609

```
mysql> select PROCESSLIST_ID , THREAD_OS_ID  from performance_schema.threads where processlist_id=9;
```

```
mysql> select PROCESSLIST_ID , THREAD_OS_ID  from performance_schema.threads where processlist_id=9;
+-----+-----+
| PROCESSLIST_ID | THREAD_OS_ID |
+-----+-----+
| 9 | 1609 |
+-----+-----+
1 row in set (0.00 sec)
```

第三步：top命令查询数据库进程id

```

top - 23:31:43 up 2:19, 3 users, load average: 0.00, 0.01, 0.05
Tasks: 100 total, 2 running, 98 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.6 sy, 0.0 ni, 99.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2027876 total, 1310032 free, 484480 used, 233364 buff/cache
KiB Swap: 1048572 total, 1048572 free, 0 used. 1392960 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM     TIME+ COMMAND
 1487 mysql    20   0 1296844 366004 15724 S   0.6  18.0   0:34.16 mysqld
   1 root      20   0  125720   4268   2572 S   0.0   0.2   0:02.09 systemd
   2 root      20   0     0     0     0 S   0.0   0.0   0:00.00 kthreadd
   4 root       0 -20     0     0     0 S   0.0   0.0   0:00.00 kworker/0:0H
   5 root      20   0     0     0     0 S   0.0   0.0   0:00.24 kworker/u256:0
   6 root      20   0     0     0     0 S   0.0   0.0   0:00.20 ksoftirqd/0
   7 root      rt    0     0     0     0 S   0.0   0.0   0:00.00 migration/0
   8 root      20   0     0     0     0 S   0.0   0.0   0:00.00 rcu_bh
   9 root      20   0     0     0     0 S   0.0   0.0   0:00.56 rcu_sched
  10 root       0  20     0     0     0 S   0.0   0.0   0:00.00 lru-add-drain

```

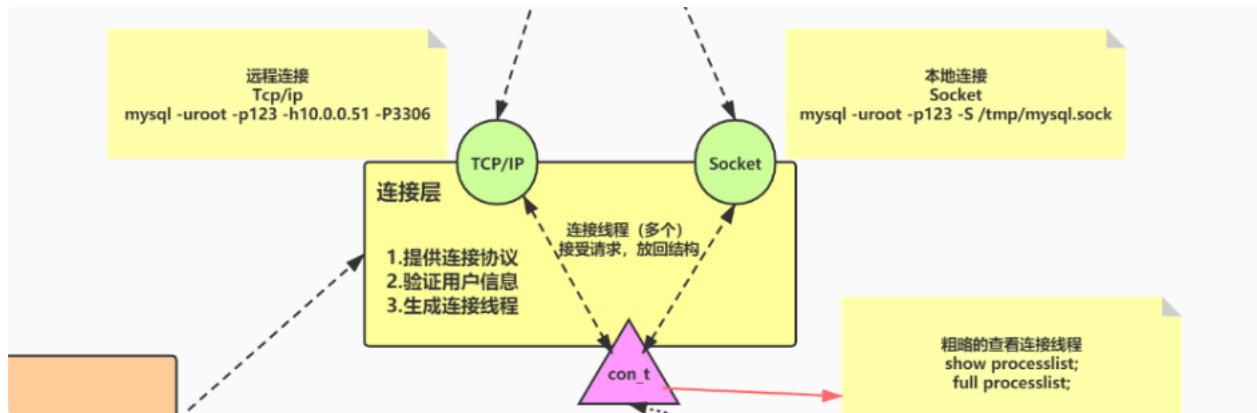
第四步：通过top -Hp 1487(数据库进程id) 找到对应连接会话线程对应的操作系统id1609

```
top - 23:33:49 up 2:21, 3 users, load average: 0.00, 0.01, 0.05
Threads: 40 total, 0 running, 40 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2027876 total, 1310160 free, 484312 used, 233404 buff/cache
KiB Swap: 1048572 total, 1048572 free, 0 used. 1393108 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1504	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.84	mysqld
1505	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.83	mysqld
1506	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.71	mysqld
1507	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.89	mysqld
1508	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.04	mysqld
1509	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.10	mysqld
1510	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.04	mysqld
1512	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.00	mysqld
1513	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.22	mysqld
1514	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.02	mysqld
1515	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.04	mysqld
1516	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.00	xpl_worker0
1517	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.00	xpl_worker1
1518	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.18	mysqld
1522	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.00	mysqld
1523	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:01.02	mysqld
1524	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.00	mysqld
1525	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.01	mysqld
1526	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.01	mysqld
1527	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.03	mysqld
1528	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.00	mysqld
1529	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.00	mysqld
1530	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.00	mysqld
1532	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.00	mysqld
1609	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.03	mysqld
1669	mysql	20	0	1296844	366004	15724	S	0.0	18.0	0:00.09	mysqld

d.Check Memory

--- 检查内存



2.2.3 SQL Interface sql层 功能!!!

DDL(NEW!!) ---sql定义语句
 DML ---sql修改数据语句
 procedures ---程序
 function (NEW!!) ---功能
 view ---视图
 triggers ---触发器
 JSON (NEW!!) --- json格式 (5.6版本支持。8.0增强)

sql处理过程:

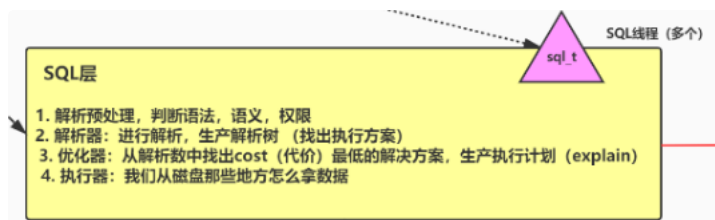
Parser 解析器

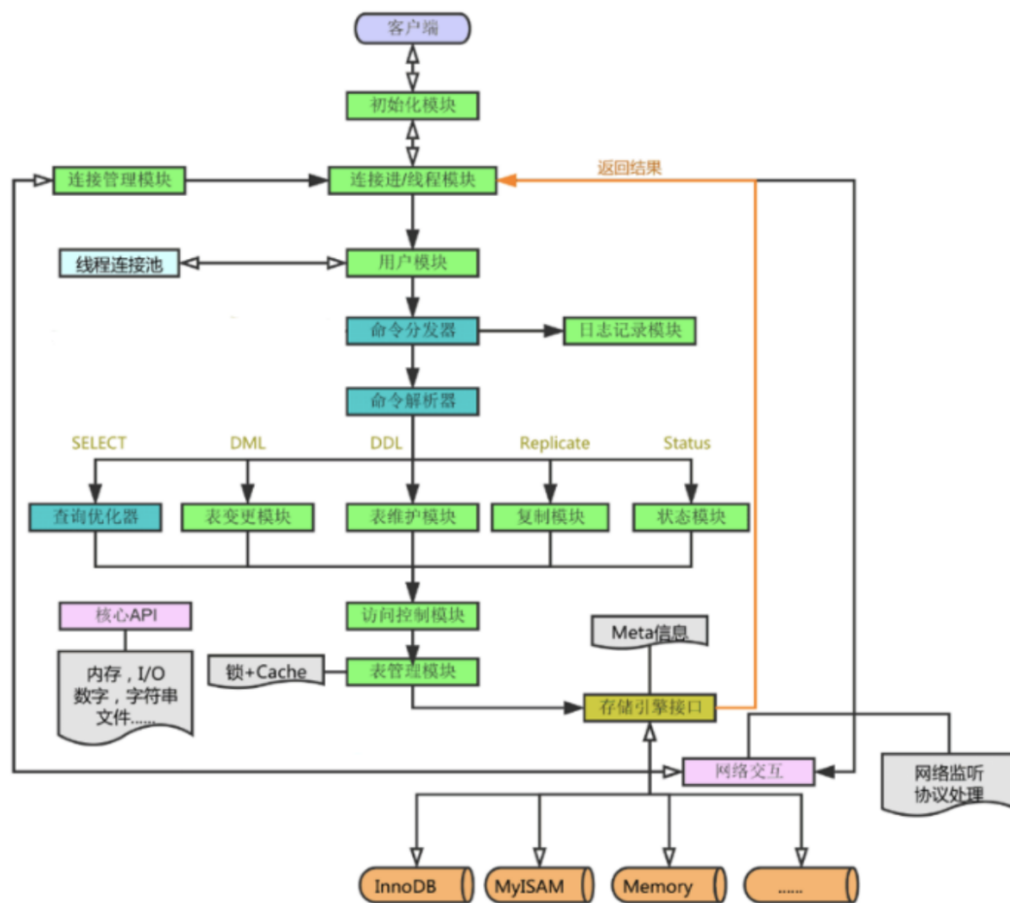
Query translation ---查询翻译
 Object privileges --- 对象权限

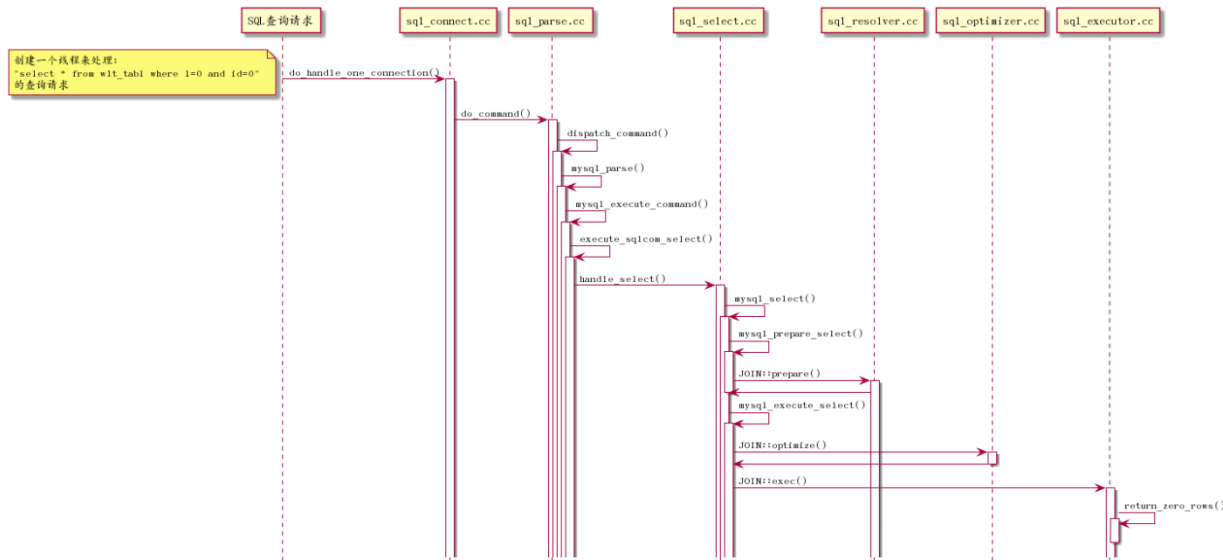
Optimizer (NEW!!) 优化器

Explain Statistics ---解释统计数据

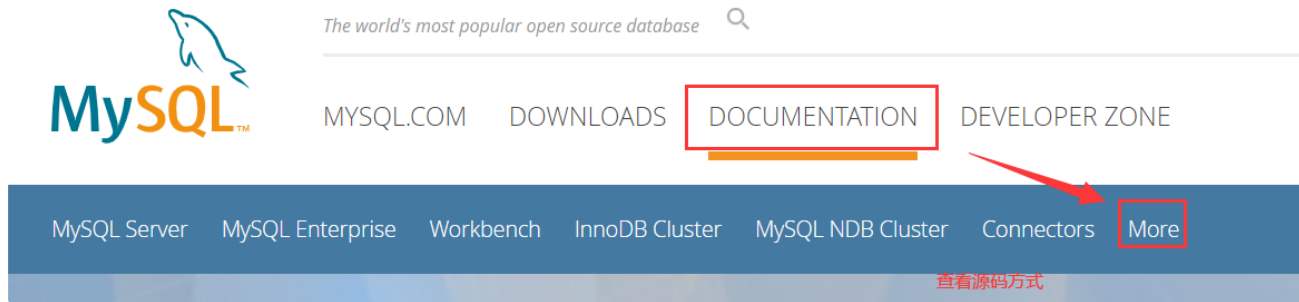
Cache && Buffers(NEW!!)

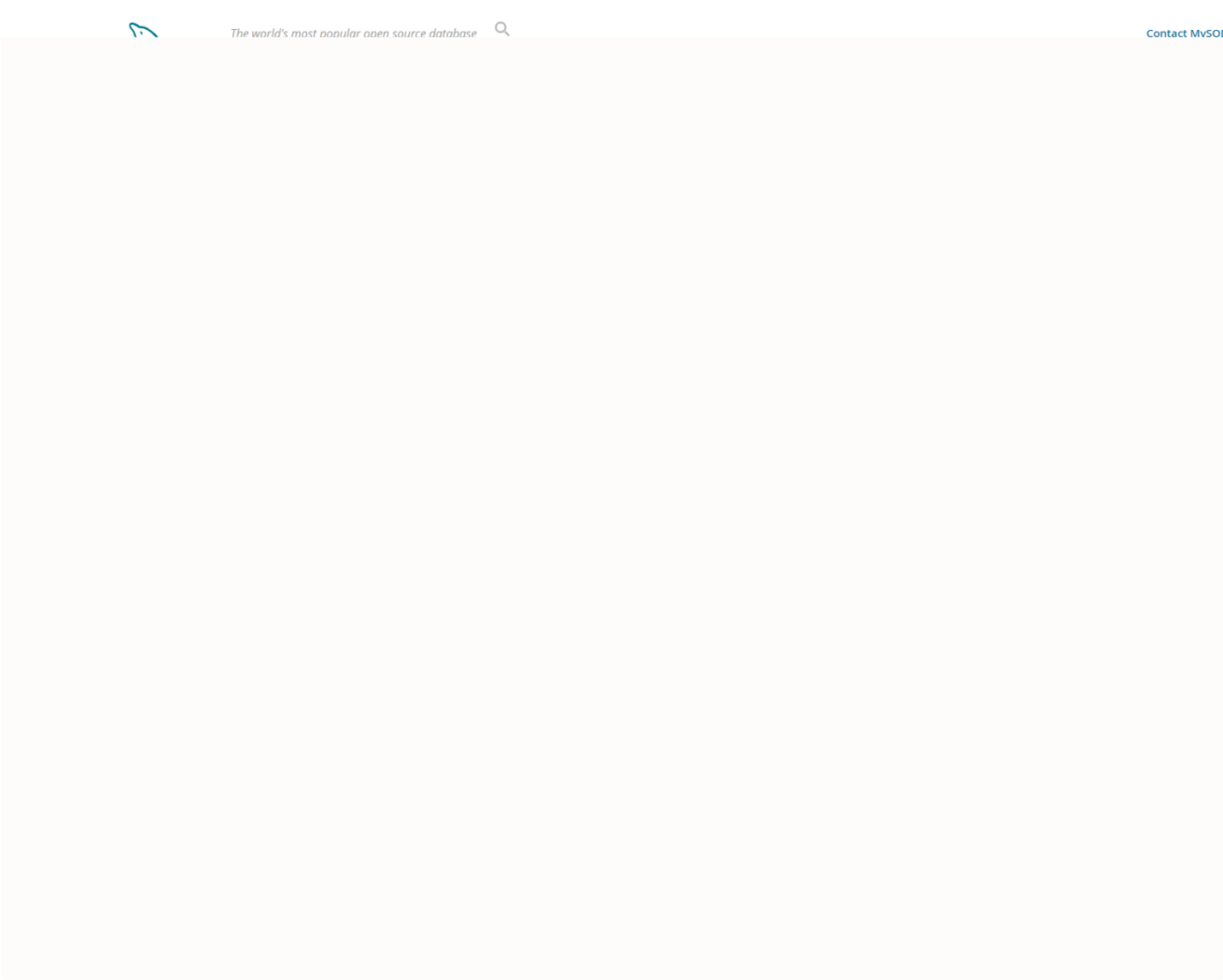


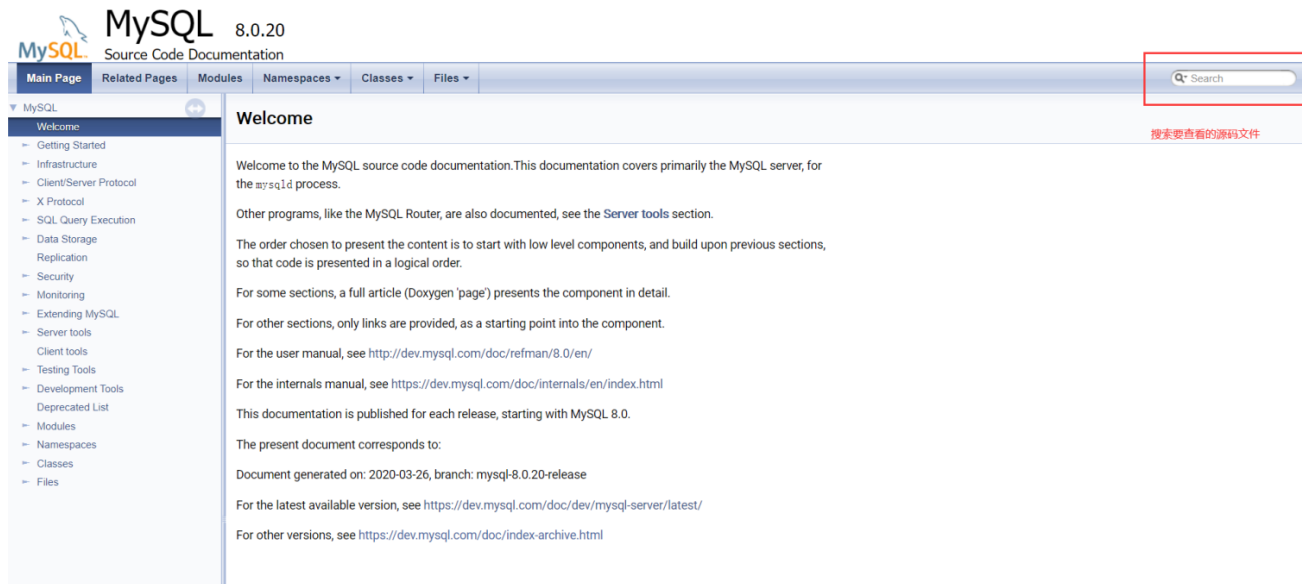




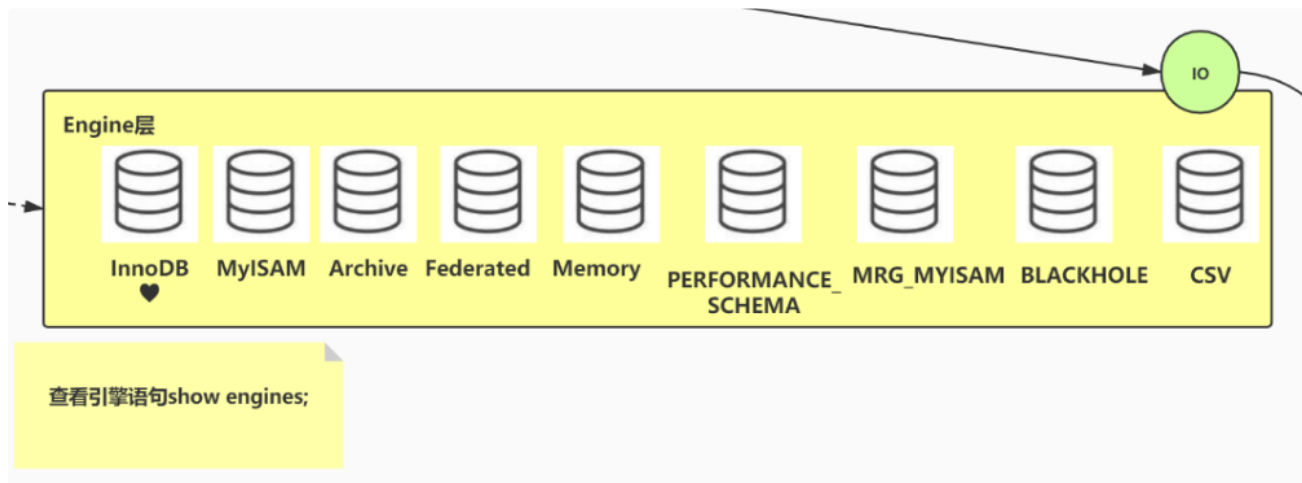
查看源码的方法







2.2.4 Pluggable storage Engines (插件式存储引擎层)



3.mysql8.0 新的特性

Metadata（元数据） 结构变化

5.7 版本问题

- 两套数据字典信息（Server层 frm，InnoDB 数据字典） 更改元数据需要在两个地方进行修改
- DDL无原子化 中间过程发生崩溃，无法保证数据的一致性
- frm和innodb层会出现不一致
- 并发处理需要小心处理， 通过锁来控制（MDL,dict_sys::mutex,dict_sys::rw_lock）
- 崩溃无法恢复

8.0的变化新特性

- 1.支持事务性DDL，崩溃可以回滚，保证一致。
- 2.保留一份数据字典信息，取消frm数据字典,数据字典存放至InnoDB表中
- 3.采用套锁机制，管理数据字典的并发访问（MDL）

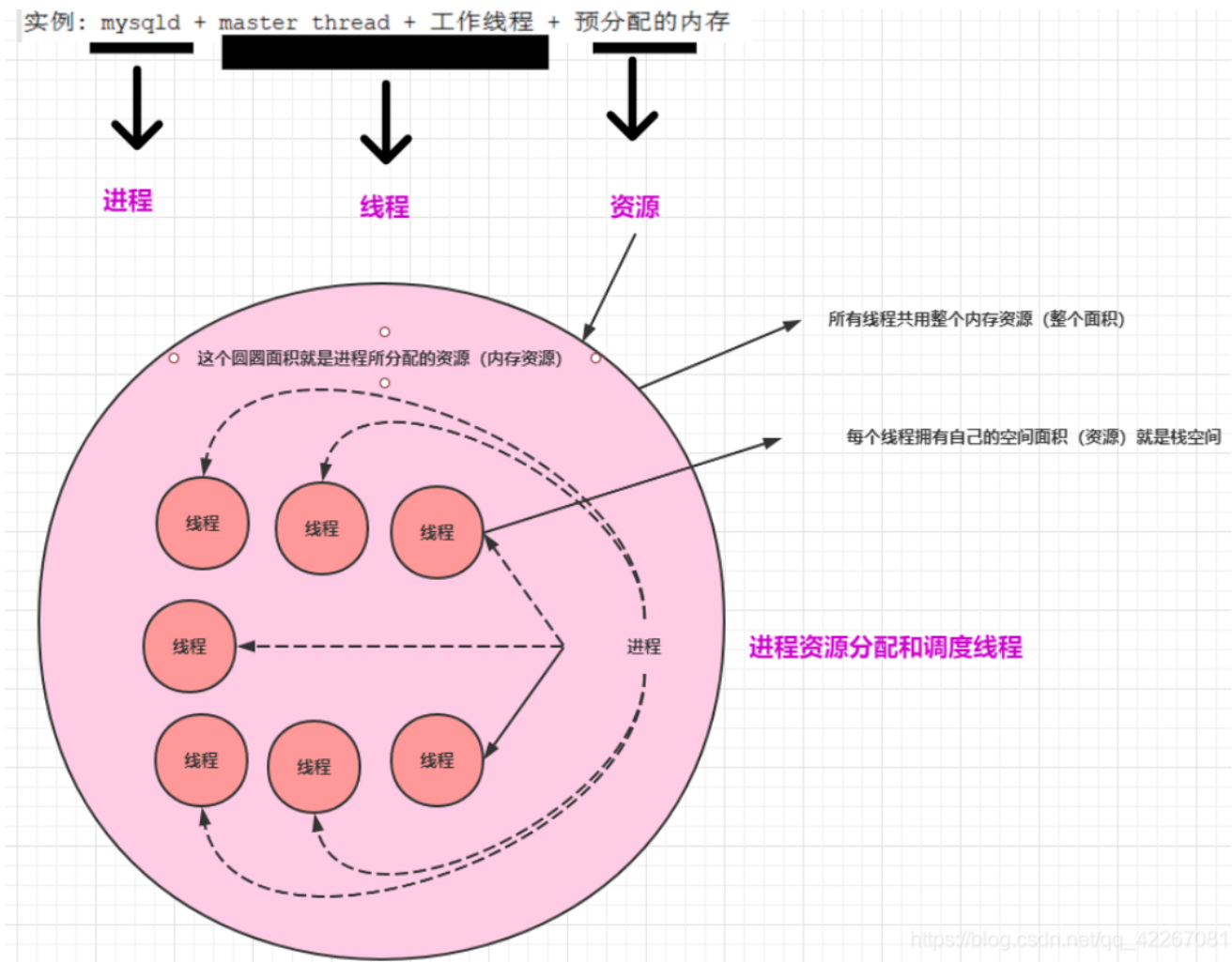
其他新特性：

- 4.全新的Plugin支持：8.0.17+ 加入Clone Plugin,更好的支持MGR架构，InnoDB Cluster的节点管理
- 5.安全加密方式的改变：改变加密方式为caching_sha2_password，SSL 将支持到 TLSv1.3 版本。
- 6.用户管理及认证方式改变：改变授权管理方式，加入role角色管理，添加更多权限
- 7.支持原子性DDL
- 8.取消Query Cache （查询缓存）

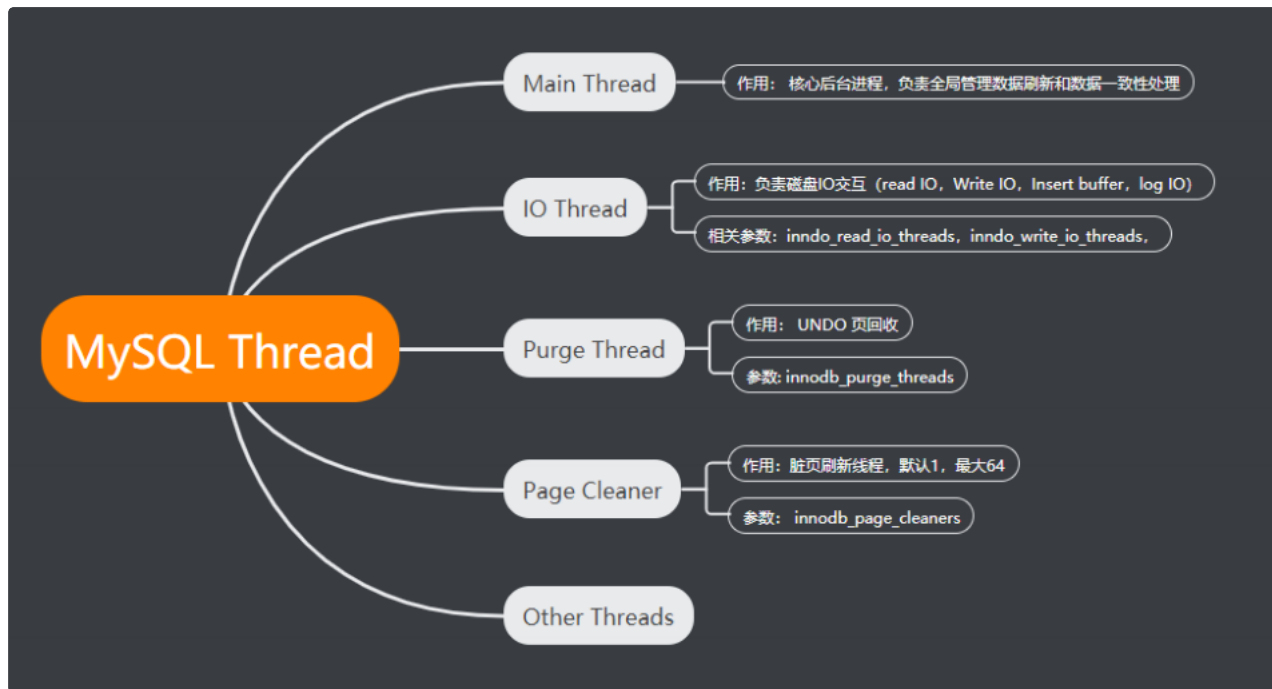
4.mysql实例结构

典型的单进程多线程工作模型。目的提高更高的并发。Oracle是多进程模式

mysqld+多线程+内存结构



4.1 mysql线程结构



a.Main Thread(Master Thread) **主线程** 作用：核心后台线程，负责全局管理数据刷新和数据一致性处理

b.io Thread **作用：负责磁盘IO交互**

- read io
- write io
- insert buffer
- log io

相关调节参数：innodb_read_io_threads, innodb_write_io_threads 默认是4

c.Purge (清楚) Thread **作用：undo页的回收**

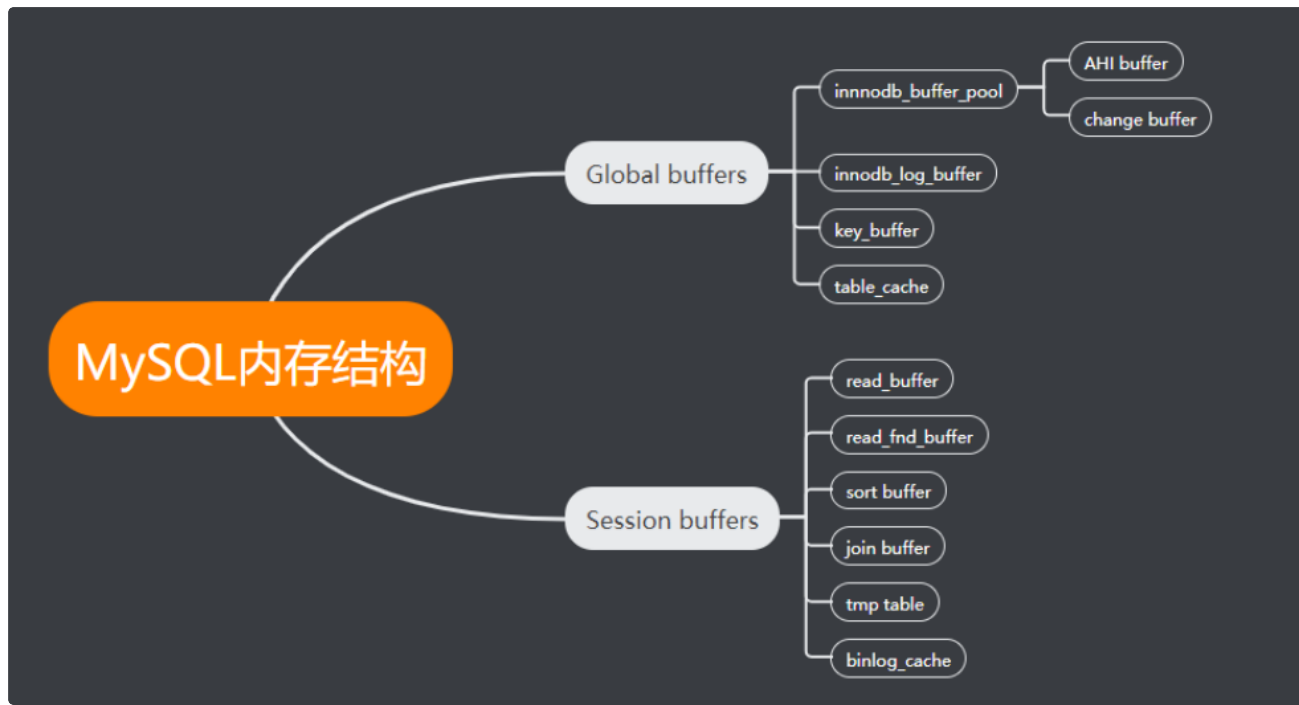
相关调节参数：innodb_read_io_threads 默认4

d.Page Cleaner **作用：刷新脏页的线程，将数据进行落盘**

相关调节参数：innodb_page_cleaners 默认1 最大64

e.Other Thread

4.2 mysql内存结构



Global buffers+Session buffers

4.2.1 Global buffers 全局（共享内存区域）组成

a. innodb_buffer_pool 数据缓冲区(占用大)

- AHI buffer
- change buffer

b.innodb_log_buffer 日志缓冲区(占用大)

c.key_buffer 针对于 myisam存储引擎存放索引

d.table_cache 表缓冲区

4.2.2 Session buffers 会话级别 每个会话连接分的内存区域

a.read_buffer 顺序读缓冲区

b.read_fnd_lbuffer 随机读缓冲区

c.sort buffer 内存分配越小越好，多用索引

d.join buffer 内存分配越小越好，多用索引

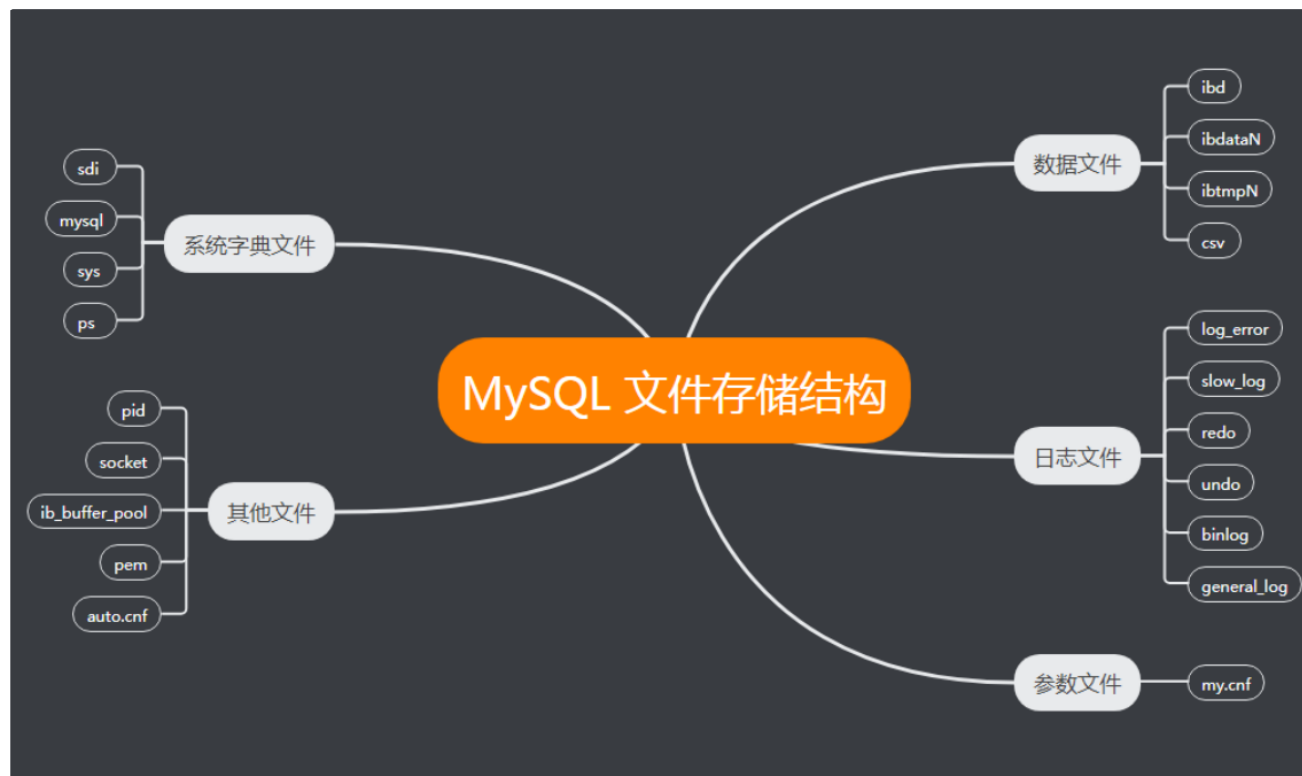
e.tmp table

f.binlog_cache

5.mysql文件存储结构

存储原则：日志与数据分开独立存储

5.1 宏观结构




```

[root@db01 data]# ll /data/3306/data/
总用量 176592
-rw-r----- 1 mysql mysql      56 3月 29 21:26 auto.cnf           ❶ 存放uuid 和主从架构有关
-rw-r----- 1 mysql mysql     200 3月 29 23:13 binlog.000001
-rw-r----- 1 mysql mysql     200 3月 29 23:14 binlog.000002
-rw-r----- 1 mysql mysql     179 3月 29 23:14 binlog.000003
-rw-r----- 1 mysql mysql     200 3月 30 22:49 binlog.000004
-rw-r----- 1 mysql mysql     156 3月 30 22:49 binlog.000005
-rw-r----- 1 mysql mysql      80 3月 30 22:49 binlog.index
-rw-r----- 1 mysql mysql    1676 3月 29 21:26 ca-key.pem
-rw-r----- 1 mysql mysql    1112 3月 29 21:26 ca.pem
-rw-r----- 1 mysql mysql    1112 3月 29 21:26 client-cert.pem
-rw-r----- 1 mysql mysql    1676 3月 29 21:26 client-key.pem
-rw-r----- 1 mysql mysql      5 3月 30 22:00 db01.pid
-rw-r----- 1 mysql mysql   196608 3月 30 22:55 #ib_16384_0.dblwr
-rw-r----- 1 mysql mysql   8585216 3月 29 21:26 #ib_16384_1.dblwr
-rw-r----- 1 mysql mysql     3569 3月 29 23:14 ib_buffer_pool
-rw-r----- 1 mysql mysql  12582912 3月 30 22:55 ibdata1
-rw-r----- 1 mysql mysql  50331648 3月 30 22:55 ib_logfile0
-rw-r----- 1 mysql mysql  50331648 3月 29 21:26 ib_logfile1
-rw-r----- 1 mysql mysql  12582912 3月 30 22:00 ibtmp1
drwxr-x--- 2 mysql mysql     187 3月 30 22:00 #innodb_temp
drwxr-x--- 2 mysql mysql     143 3月 29 21:26 mysql
-rw-r----- 1 mysql mysql  25165824 3月 30 22:00 mysql.ibd
drwxr-x--- 2 mysql mysql     8192 3月 29 21:26 performance_schema
-rw-r----- 1 mysql mysql     1680 3月 29 21:26 private_key.pem
-rw-r----- 1 mysql mysql     452 3月 29 21:26 public_key.pem
-rw-r----- 1 mysql mysql    1112 3月 29 21:26 server-cert.pem
-rw-r----- 1 mysql mysql    1680 3月 29 21:26 server-key.pem
drwxr-x--- 2 mysql mysql      28 3月 29 21:26 sys
-rw-r----- 1 mysql mysql  10485760 3月 30 22:00 undo_001
-rw-r----- 1 mysql mysql  10485760 3月 30 22:00 undo_002
[root@db01 data]#

```

❷ 存放二进制日志

❸ .pem结尾的存放ssl连接信息

❹ 存放DWB信息

❺ 存放预缓冲信息

❻ 共享表空间数据信息

❼ 存放重做日志, redolog

❽ 存放临时表空间

❾ 存放系统库表信息

❿ 存放关于优化, 统计, 监控

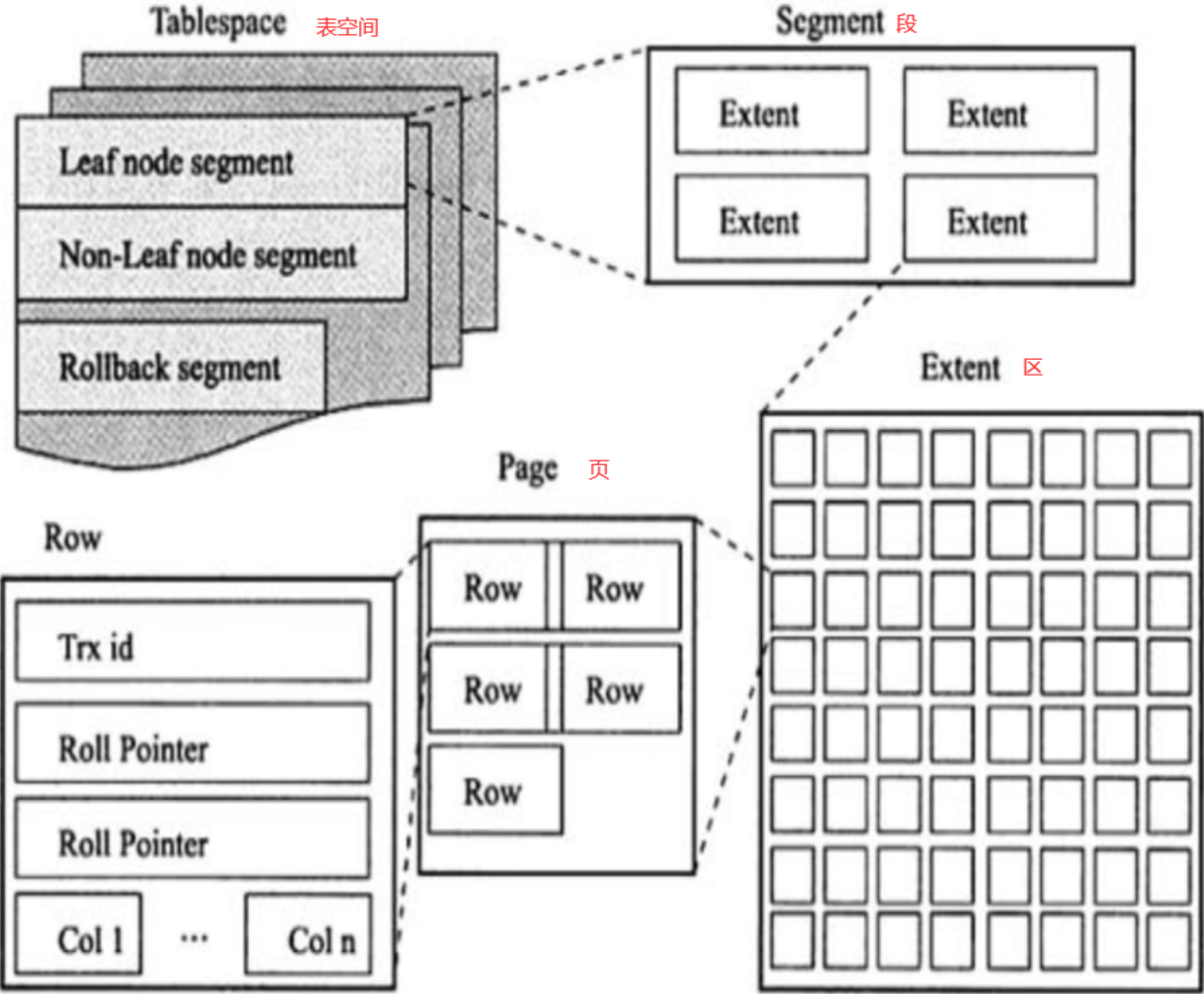
⓫ .pem结尾的存放ssl连接信息

⓬ 存储关于优化, 统计, 监控

⓭ 存放回滚日志

⓮ sys对p_s进行整合, 更容易读懂

5.2 微观结构（表空间-段-区-页-块-扇区）



a8012dc7de66.png&title=3.MySQL%E4%BD%93%E7%B3%BB%E7%BB%93%E6%9E%84%E2%88%9A%20%7C%201.mysql%E4%BD%93%E7%B3%BB%E7%BB%93%E6%9E%84%E5%9B%BE%EF%BC%