目录

[第3章MySQL体系结构 1](#_Toc20671)

[3.2　数据目录结构 1](#_Toc27216)

[3.3　 MySQL Server 体系结构 2](#_Toc8502)

[3.4 MySQL中的存储引擎 3](#_Toc228)

[3.5 InnoDB存储引擎体系结构 4](#_Toc26405)

[3.6　InnoDB存储引擎后台线程 5](#_Toc11573)

[3.7　MySQL前台线程 6](#_Toc5762)

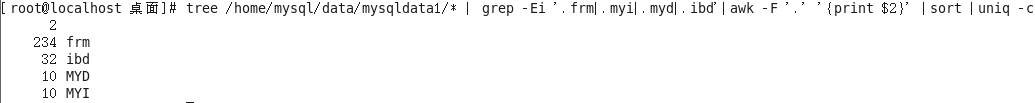
## 第3章MySQL体系结构

## 3.2　数据目录结构

# 使用如下两个命令搜索datadir下的所有文件

[root@localhost ~]# tree /home/mysql/data/mysqldata1/\* | grep -Ei '**.frm|.myi|.myd|.ibd**'\

|awk -F '.' '{print $2}' |sort |uniq -c



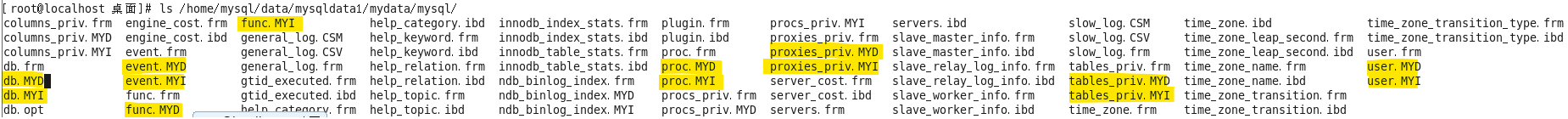
注：tree命令未必默认安装。可参考实训1笔记配置yum安装。

下面不妨进入这个位置看看：

报纸上的文字

中度可信度描述已自动生成

上面performance\_schma库将在第4章介绍，里面每个frm文件对应一张数据表。这意味着服务器里面总共有234张表。myi文件是MyISAM索引文件。myd和myi文件可以在/home/mysql/data/mysqldata1/mysql中发现，分别属于数据文件和索引文件。mysql库以及这些对应的表在12章及其后续章节介绍。



[root@localhost ~]# tree /home/mysql/data/mysqldata1/\* | grep -Eiv '.frm|.myi|.myd|.ibd'

表格

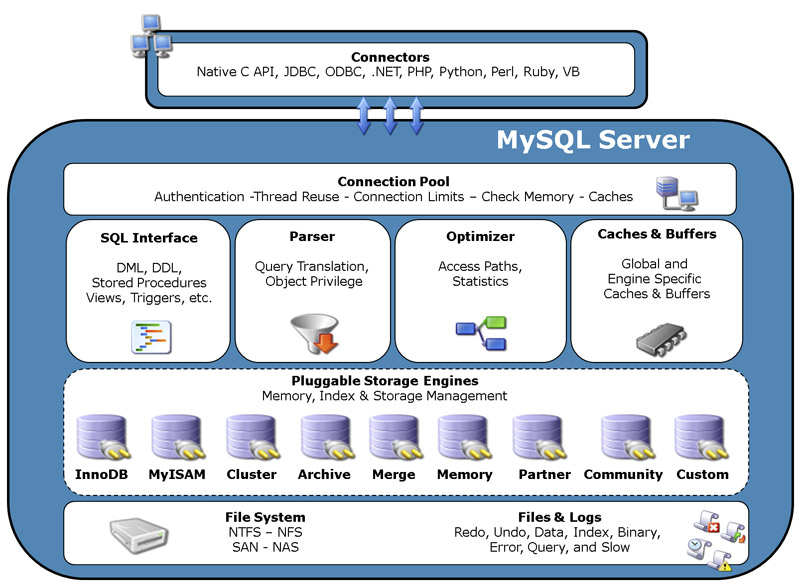
中度可信度描述已自动生成

注：以上命令列出除前一命令外的其他命令。以下是对书中文件内容的整理介绍：

图示

描述已自动生成

## 3.3　 MySQL Server 体系结构



MySQL Server体系结构图

注：以下是该图的中文翻译(翻译概念，短语。专有名词不翻译，如第3层InnoDB等存储引擎插件，一些术语要按本来意义翻译，如右下的Slow，指的是慢查询日志，其实右下都是日志，要明确它们的含义)。

该图的中文版：

注：各组件解释及书中最后关于查询会话请求流程的总结：

图片包含 文本

描述已自动生成

参考如下网页中的讲解：

<https://www.pingface.com/2020/04/structure.html>

图示

描述已自动生成

## 3.4 MySQL中的存储引擎

#关于MySQL支持的存储引擎列表，我们可以登录数据库执行如图3-2所示的语句查询。

注：

mysql>select \* from information\_schema.engines;

文本

描述已自动生成

#InnoDB是默认的存储引擎，并且是唯一一个支持事务、XA(注：分布式事务、跨数据库事务)和SAVEPOINTS(注：事务回滚到达的中间存储点)的引擎。

## 3.5 InnoDB存储引擎体系结构

该图的中文翻译(翻译原则同上)：图示

描述已自动生成

注：此图从左上角开始按箭头走向阅读。在后面结合更具体的章节内容进行理解。

**图示

中度可信度描述已自动生成**

#假设有一个UPDATE语句正在执行：UPDATE test SET idx = 2 WHERE id=10，执行流程如下：

图示

描述已自动生成

## 3.6　InnoDB存储引擎后台线程

#可以通过performance\_schema.threads表查询到所有线程，包括后台线程和前台线程

mysql> select name,type,thread\_id, processlist\_id from performance\_schema.threads;

表格

描述已自动生成

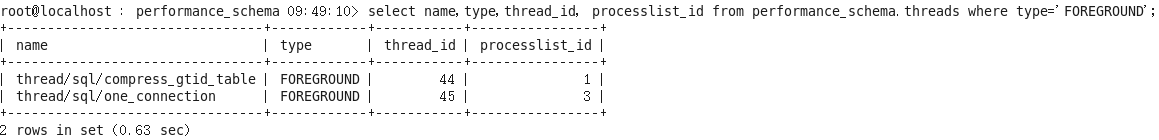
注：不妨对照前面介绍的架构和工作流程来理解这些后台线程的作用。

图片包含 文本

描述已自动生成

## 3.7　MySQL前台线程

mysql> select name,type,thread\_id, processlist\_id from performance\_schema.threads where type='FOREGROUND';



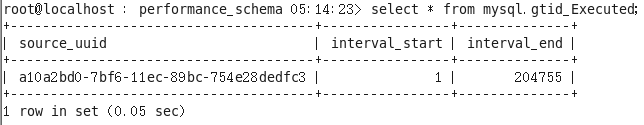
注：one\_connection就是用户连接线程，处理用户请求。

书中提到“双主复制架构”是指有多个主数据库服务器。可阅读：

<https://www.cnblogs.com/itbsl/p/13508625.html>

所以跟这里的实际结果自然不同。

GTID(Global Transaction ID)是对于一个已提交事务的编号，并且是一个全局唯一的编号(UUID)，目的是为了在主从架构的数据库服务器中确保一致性。GTID不断被记录在mysql.gtid\_Executed表需要压缩节省空间。compress\_gtid\_table线程会对GTID记录进行压缩，比如：



下图是对GTID的介绍：

图形用户界面, 图示

描述已自动生成

完整的图参见：

<https://www.hldx.cc/270>

由于书中用到主从架构，因此还会有slave\_io、slave\_sql、slave\_worker等前台线程，可参考书中相关介绍。