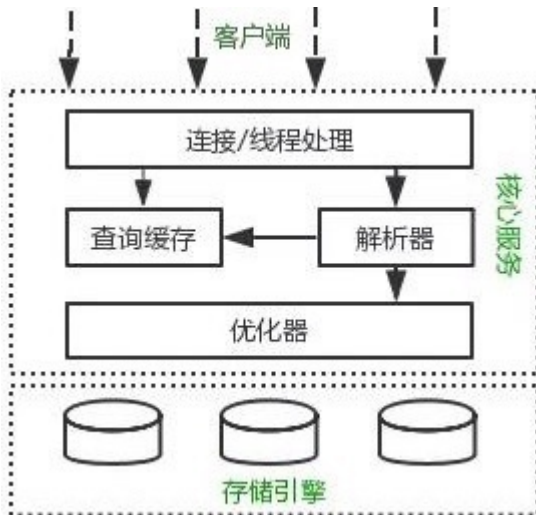


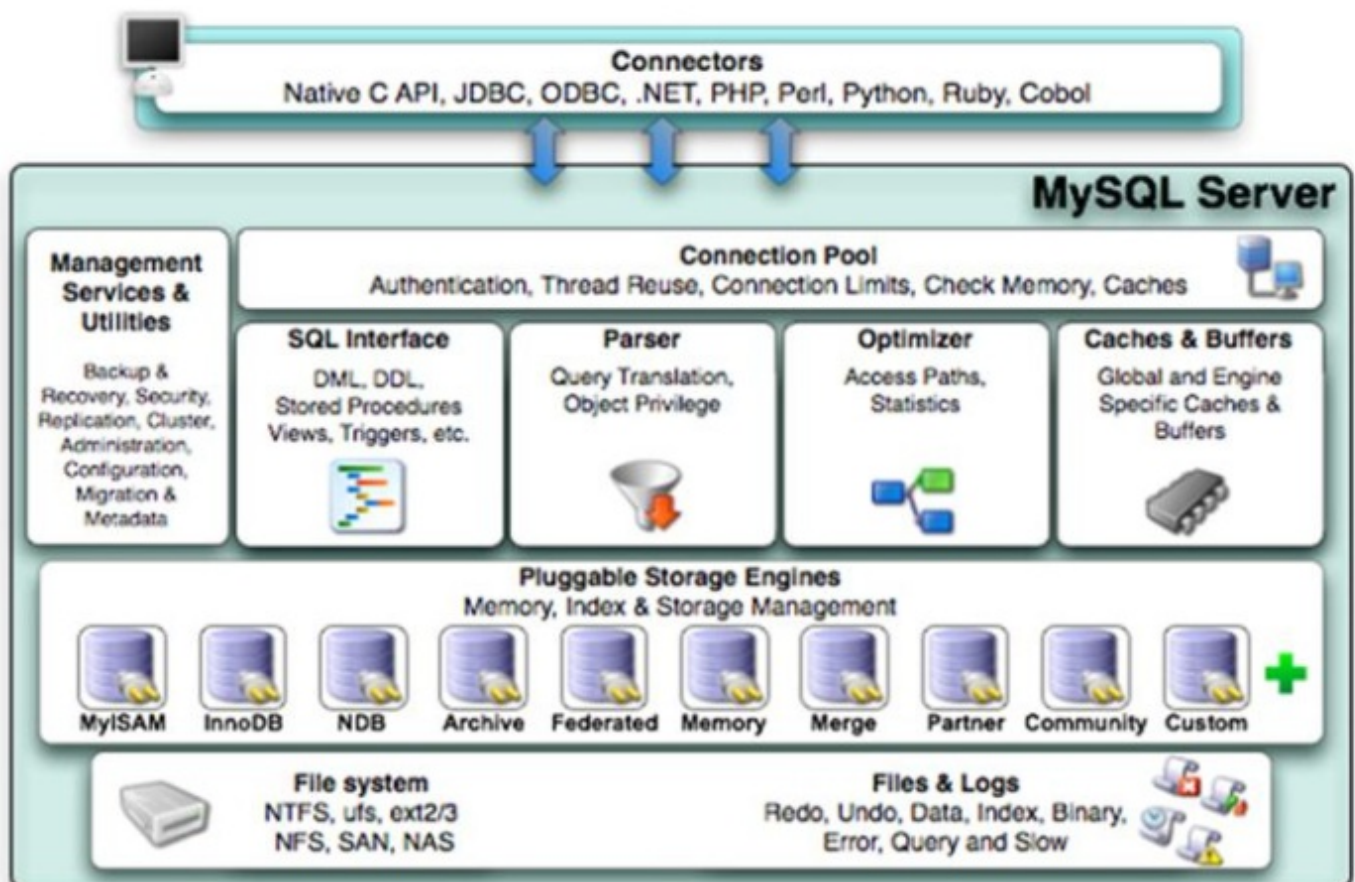
# mysql结构与sql执行

## 1. mysql架构详细讲解

简化版mysql架构:



详细版mysql架构:



MySQL Server 逻辑架构图

#### 1).最上层:

最上层是一些客户端和连接服务, 包含本地的sock通信和大多数基于客户端/服务端工具实现的类似于tcp/ip的通信, 主要完成一些类似于连接处理、授权认证及相关的安全方案, 在该层上引用了线程池的概念, 为通过认证安全接入的客户端提供线程。同样在该层上可以实现基于ssl的安全链接。服务器也会为安全接入的每个客户端验证它所具有的操作权限。

#### 2).第二层:

第二层架构主要完成大多数的核心服务功能。如sql接口, 并完成缓存的查询。sql的分析和优化 以及部分内置函数的执行。所有跨存储引擎的功能也在这层实现, 如过程, 函数等。在该层, 服务器会解析查询并创建相应的内部解析树, 并对其完成相应的优化如确定查询表的顺序, 是否利用索引等。最后生成相应的执行操作。如select语句, 服务器还会查询内部的缓存。如果缓存空间足够大, 这样就解决大量读操作的环境中能够很好的提升系统的性能。

#### 3).存储引擎层:

存储引擎真正的负责MySQL中数据的存储和提取, 服务器通过API与存储引擎进行通信, 不同的存储引擎具有的功能不同, 这样我们可以根据自己的实际需求进行选择。

#### 4).数据存储层:

主要是将数据存储在运行于裸设备的文件系统之上, 并完成于存储引擎的交互。

具体内容:

Connectors

不同语言与SQL交互。

Management Services &Utilities

系统管理和控制工具。

Connection Pool

连接池, 管理缓冲用户连接, 线程处理所需要的缓存。

SQL Interface

接受用户的SQL命令, 返回用户查询结果, 比如 select from 就是调用 sql interface

Parser

解释器, SQL命令传递到解释器的时候会被解释器验证和解析, 解释器由Lex和YACC实现(一个很长的脚本), 主要功能, 将SQL语句分解成数据结构, 并将这个结构传递到后续步骤, 以后的SQL语句的传递和处理都是基于这个结构, 如果在分解构成中遇到错误, 说明这个sql语句是不合理的

Optimizer

查询优化器, SQL语句在查询之前会使用查询优化器对查询进行优化, 他使用的是“选取-投影-连接”进行查询, 用一个例子就可以理解: select uid,name from user where gender =1;这个select 查询先根据where语句进行选择, 而不是先将表全部查询出来以后再进行gender过滤, 这个select 查询先根据uid和name进行属性投影, 而不是将属性全部取出以后再进行过滤, 将这2个查询条件连接起来生成最终查询结果。

Cache和Buffer查询缓存

这个查询缓存有命中的查询结果, 查询语句就可以直接去查询缓存中取数据, 这个缓存机制由一系列小缓存组成, 比如表缓存, 记录缓存, key缓存, 权限缓存

Enginee

存储引擎, 存储引擎是MySQL和文件打交道的具体子系统, MySQL的存储引擎是插件式的, 它根据MySQL AB公司提供的文件访问层的一个抽象接口来定制一种文件访问机制(这个访问机制就叫做存储引擎), 一个库中不同表使用不同的存储引擎也是允许的。

## 2. mysql各个逻辑块简介

### 初始化模块

初始化模块就是在数据库启动的时候, 对整个数据库做的一些初始化操作, 例如: 各种系统环境变量的初始化, 各种缓存, 存储引擎初始化设置等

### 核心API

MySQL数据库核心API主要实现了数据库底层操作的优化功能, 其中主要包括IO操作、格式化输出、高性能存储数据结果算法的优化, 字符串的处理, 其中最重要的是内存管理。

### 网络交互模块

MySQL底层相互交互的模块抽象出接口, 对外提供可以接收和发送数据的API接口, 其他模块需要交互的时候, 可以通过API接口调用。

### 服务器客户端交互协议模块

mysql服务器采用C/S的形式访问数据库, 数据连接使用MySQL C/S交互协议模块, 实现了客户端与服务器端交互过程中所需要的一些独特的协议, 这些协议都是建立在现有的网络协议之上。

## 用户模块

用户模块主要功能是用于控制用户登录连接的权限和用户的授权管理

## 访问控制器模块

访问控制模块主要用于监控用户的每一个操作。访问控制模块实现的功能就是根据用户模块中不同的用户授权，以及根据其数据库的各种约束来控制用户对数据的访问。用户模块和访问控制模块结合起来，就组成了MySQL数据库的权限管理功能。

## 连接管理，连接线程与线程管理

连接管理模块负责监听MySQLServer的各种请求，根据不同的请求，然后转发到线程管理模块，每个客户请求都会被数据库自动分配一个独立的线程为其单独服务，而连接线程的主要工作就是负责MySQL Server与客户端通信，线程管理模块负责管理这些生成的线程。

## 转发模块

客户端连接MySQL之后会发送一些查询语句，在MySQL Server里面，连接线程接收到客户端的一个请求后，会直接将查询转发到各个对应的处理模块。转发模块主要就是根据查询语句语法分析，然后转发给不同的模块处理。

## 缓存模块

查询缓存模块主要功能是将客户端查询的请求返回的结果集到缓存中，与查询的一个HASH值对应。在查询的基表发生任何数据变化后，MySQL会自动将其查询的缓存失效。在读写比例非常高的应用系统中，查询缓存对性能的提高是非常显著的。

## 优化器模块

这个模块主要是将客户端发送的查询请求，在之前算法的基础上分析，计算出一个最优的查询策略，优化之后会提高查询访问的速度，最后根据其最优策略返回查询语句。

## 表变更管理模块

表变更管理模块主要负责完成DML和DDL的查询，例如，insert, update, delete, create table, alter table等语句的处理。

## 表维护模块

表维护模块主要用于检测表的状态，分析、优化表结构，以及修复表。

## 系统状态管理模块

在客户端请求系统状态的时候，系统状态模块主要负责将各种状态的数据返回给用户。最常用的一些查询状态的命令包括show status, show variables等，都是通过这个模块负责返回的。

## 表管理器

表管理器主要就是维护系统生成的表文件。例如，MyISAM 存储引擎类型表生成的是frm文件、MYD文件以及MYI文件，表管理器的工作就是维护这些文件，将各个表结构的信息缓存起来，另外该模块还管理表级别的锁。

## 日志记录模块

日志记录模块主要负责整个数据库逻辑层的日志文件，其中包含错误日志，二进制日志，以及慢查询日志等。

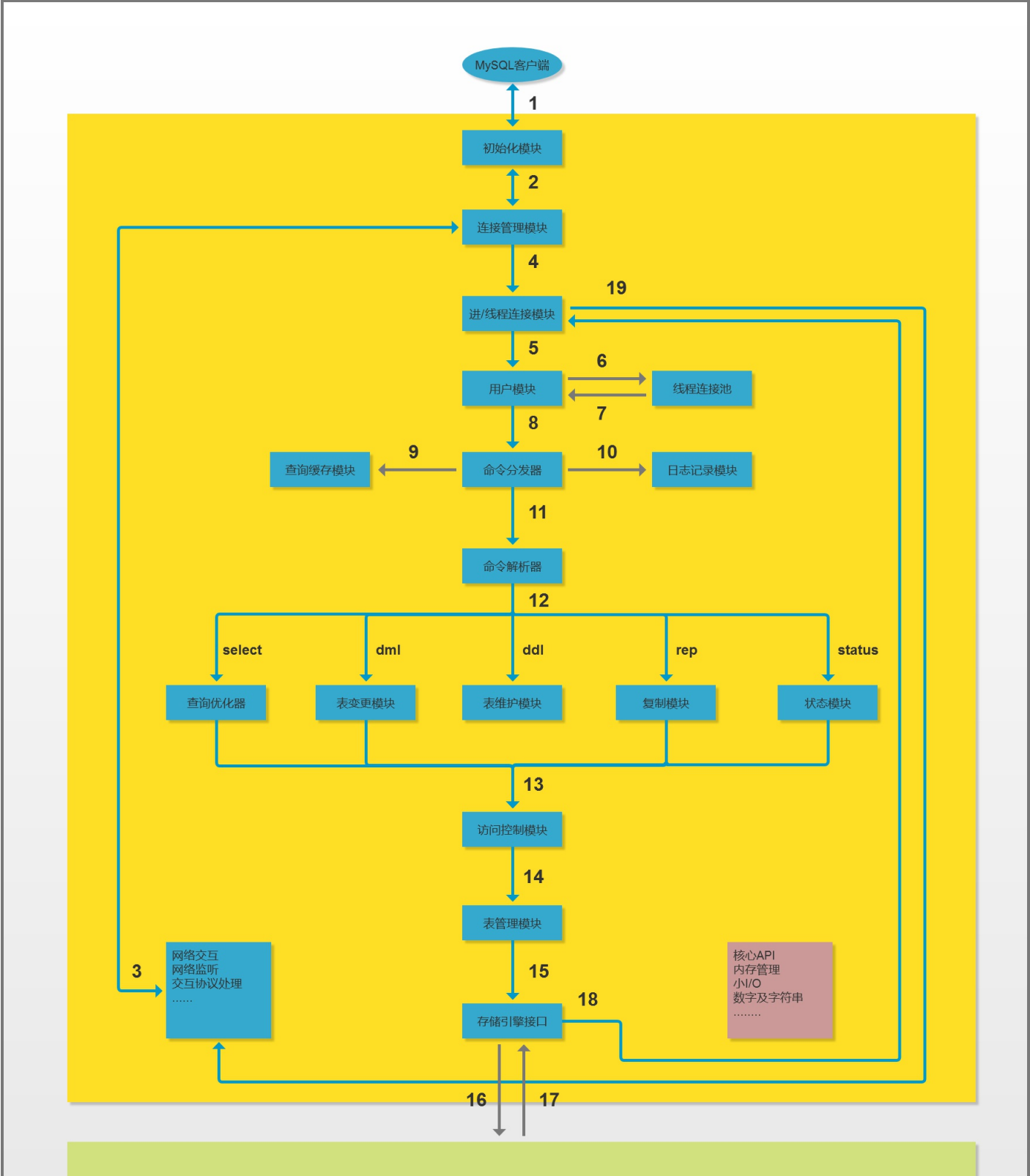
## 复制模块

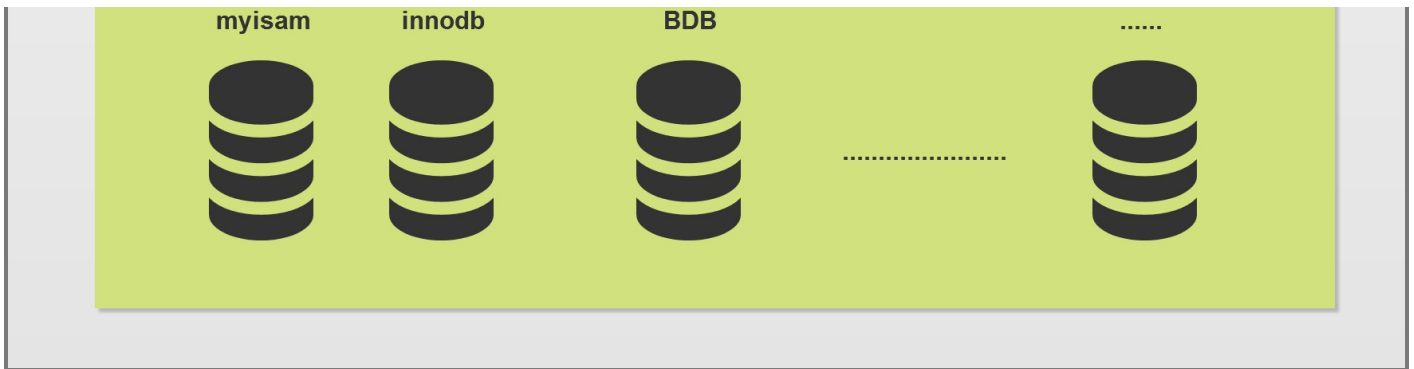
复制模块分为Master 模块和Slave 模块两部分。Master 模块主要负责复制环境中读取Master端的binary日志，以及Slave端的I/O线程交互等工作。Slave 模块主要有两个线程，一个负责从Master请求和接收binay日志，并写入本地I/O线程;另一个从relay log读取日志事件，然后解析成可以在Save端执行的命令，然后交给Slave端的SQL线程

MySQL实现了其数据库底层存储引擎的插件式管理，将各种数据处理高度抽象化。

3. sql执行流程

以下是SQL执行路线流程图：





### 3.1 执行流程分析

#### 3.1.1 启动

1. 通过命令 `net start mysql(windows)` / `service mysql start(linux)` 启动MySQL服务
2. 调用初始模块：初始化模块就是在数据库启动的时候，对整个数据库做的一些初始化操作，比如各种系统环境变量的初始化，各种缓存，存储引擎初始化设置等。

核心api：MySQL数据库核心api主要实现了数据库底层操作的优化功能，其中主要包括IO操作、格式化输出、高性能存储数据结果算法的优化，字符串的处理，其中最重要的是内存管理。

#### 3.1.2 连接

3. 用户发送一条SQL，这个时候会被网络交互模块监听到用户的操作请求，传递给‘连接管理模块’
4. 接收到请求转发到‘进/线程连接模块’
5. 调用‘用户模块’来进行权限检测（访问数据库的权限）
6. 通过检测之后就会去‘连接进/线程模块’从‘线程连接池’中查找空闲的被缓存的连接线程和客户端请求对接，如果失败则创建一个新的连接请求
7. 返回连接线程

网络交互模块：对外提供可以接收和发送数据的api接口，其他模块需要交互的时候，可以通过api接口调用

连接管理模块、进/线程连接模块、线程连接池：连接管理模块负责监听MySQL Server的各种请求，根据不同的请求，然后转发到线程管理模块，每个客户请求都会被数据库自动分配一个独立的线程为其单独服务，而连接线程的主要工作就是负责MySQL Server与客户端通信，线程管理模块负责管理这些生成的线程。

#### 3.1.3 处理

8. 在用户权限校验成功之后，并且获得新的连接池之后就会去‘命令分发器’，判断命令的类型如果是select就会去访问‘查询缓存’，如果没有就会往下执行；
9. 如果是select，并且开启‘查询缓存’之后就会去缓存中查询是否有与之相匹配的SQL,如果有就会校验用户访问该数据的权限,通过就返回不通过就会返回错误信息. 如果数据没有就会往下执行
10. 会记录过程中的SQL操作过程到日志文件中
11. 在第8,9步 没有满足相应条件之后往下执行进入 ‘命令解析器’,经过词法分析,语法分析后生成解析树
12. 根据操作转到对应的模块处理(预处理阶段), 根据SQL选择执行的模块
13. 模块收到请求后, 通过‘访问控制模块’检查所连接的用户是否有访问目标表和目标的权限（是指访问这些数据的权限）
14. 有权限‘表管理模块’先查看table cache中是否存在, 有则直接对应的表和获取锁, 负责重新打开表文件
15. 根据表的ENGINE数据, 获取表的存储引擎类型等信息
16. 通过接口调用对应的存储引擎处理
17. 返回查询之后数据内容

用户模块：主要功能是用于控制用户登入连接的权限和用户授权管理。

访问控制模块：主要用于监控用户的每一个操作。访问控制模块实现的功能就是根据用户模块中不同的用户授权，以及根据其数据库的各种约束来控制用户对数据的访问。用户模块和访问控制模块结合起来，就组成了MySQL数据库的权限管理功能。

查询优化器：这个模块主要是讲客户端发送的查询请求，在之前算法的基础上分析，计算出一个最优的查询策略，优化之后会提高查询访问的速度，最后根据其最优策略返回查询语句。

表变更管理模块：主要负责完成DML和DDI的查询，列如，insert, update, delete, create table, alter table等语句处理。

表维护模块：主要用于检测表的状态，分析，优化表结构，以及修复表。

mysql（master/slave）复制原理及配置 <https://www.cnblogs.com/jirgltp/3549047.html>

复制模块：复制模块分为Master模块和Slave模块两部分。Master模块主要负责复制环境中读取Master端的binary日志，以及Slave端的I/O线程交互等工作。Slave

模块主要有两个线程，一个负责从Master请求和接收binary日志，并写入本地I/O线程；另一个从relay log读取日志事件，然后解析可以在Slave端执行的命令，然后交给Slave端的SQL线程。

状态模块：在客户端请求系统状态的时候，系统状态模块主要负责将各种状态的数据返回给用户。最常用的一些查询状态的命令包括show status，show variable是等，都是通过这个模块负责返回的。

表管理模块：主要就是维护系统生成的表文件。列如MyISAM存储引擎就生成frm，myd，myi文件，维护这些文件，将哥哥表结构的信息缓存起来，另外该模块还管理表级别的锁。

存储引擎接口模块：MySQL实现了其数据库底层存储引擎的插件师管理，将各种数据处理高度抽象画。

#### 3.1.4 结果

18. 命令执行完了之后，将结果集返回给'理解进/线程模块'(返回的也可以是相应标识，成功失败)
19. '理解进/线程模块'进行后续的清理工作，并继续等待请求或断开与客户端的连接