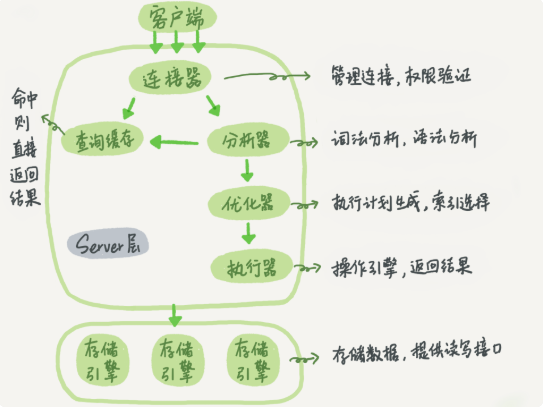
第一贴Mysql 执行流程



1-1Mysql 分为 Server 和存储引擎两部分

1-1-1 Server 层包括连接器、查询缓存、分析器、优化器、执行器涵盖 MySQL 的大多数核心服务功能，以及所有的内置函数（如日期，时间，数学和加密函数等）

所有夸存储引擎的功能都在这一层实现，比如存储过程，触发器，视图等。

1-2-1 存储引擎负责数据的存储和提取，支持InnoDB,MyISAM,Memory等多个存储引擎，最常用的InnoDB,它从5.5.5开始成为默认的存储引擎

**第1步 连接器**

#### 会先连接到数据库。负责跟客户端建立连接，获取权限，维持和管理连接，连接命令一般这么写

Mysql -h $Localhost -p $Port -u$User -p

连接命令中的mysql 是客户端工具，用来跟服务端建立连接，完成TCP握手后，连接器默认开始认证你的身份，用的用户名和密码

如果用户名或密码不正确，提示Access denied for user 错误，

如果认证通过，连机器会到权限表里面查出用户的权限，此链接权限的判断逻辑，都依赖与此是读到的权限 ，即使管理员修改权限，也不会影响当前连接的权限。

连接完成后，如果没有后续动作，这个连接就处于空闲状态，可以在show processlist命令中看到，客户端如果长时间没动静，连接器户自动断开，有参数wait\_timeout控制的，默认是8小时。如果连接被断开后，再次请求，提示‘LOST connection to MySQL server during Query’，此时要继续需要重连。尽量使用长连接。

问题：长连接可能导致内存占用太大，被系统强行杀掉

1. 定期断开长连接，使用一段时间，或者程序里面判断执行过一个占用内存的大查询后，断开连接，之后要查询再重连
2. 如果使用的5.7或者更新的版本，可以在每执行一次比较大的操作后，通过执行

mysql\_reset\_connection来重新初始化链接资源，这个过程不需要重连和做权限验证，

但是会将连接恢复到刚刚连接的状态

**第二步 查询缓存**

之前执行的语句可能户以key-value对的形式，被缓存的内存中，key是查询语句。

value 是查询结果,如果能知道key,那么value就会被直接进返回给客户端，如果没有则执行后面的操作。这样的效率会很高。

**但是大多数不要是用查询缓存，为什么呢？因为缓存弊大于利？**

查询缓存的失效率非常高，只要有对一个表单的更新，这个表上的缓存就会被清空，因此很可能缓存的结果还没用就被清空了，对于更新压力大的数据库来说，查询缓存的命中率会非常低，除非是一张静态表，会很长时间才更新一次，比如一个系统配置表。那这张表的查询才适合会用查询缓存。

好在MySQL可以“按需使用”的方式，可以将参数query\_cache\_type设置为DEMAND,这样对于默认的SQL语句都不使用查询缓存，对于确定使用查询缓存的语句，可以用SQL\_CACHE显示指定，

Select sql\_cache \* from t where id = 10;

Mysql 8.0 版本直接删除了该功能。

**第三步 分析器**

如果缓存没中，会对sql做“词法分析”，判断输入的字符串和空格代表的什么意思，然后做“语法分析”，根据词法分析的结果，语法分析器会根据语法规则判断输入的SQL是否满足MySQL语法

**第四步 优化器**

经过分析器，优化器是在表里面有多个索引时候，决定使用哪个索引，或者在一个语句多表关联（join）的时候，决定各个表的连接顺序，哪个是驱动表，哪个是被驱动表，逻辑方法是一样的，可以执行的效率会有不同，而优化器的作用就是决定选择使用哪一个方案

**第五步 执行器**

执行器知道要做什么，要先判断对表T时候有执行查询的权限，如果没有，就会返回权限的错误，如下所示（在工程上实现上，如果命中查询缓存，会在查询缓存返回的结果时候，做权限验证，查询也会在优化器之前调用precheck验证权限）

如果有权限，打开表继续执行，执行器会根据表的引擎定义，去使用这个引擎接口

Select \* from T where ID = 10;

1.调用InnoDB引擎接口去这个表的第一行，判断ID值是不是10，如果不是则跳过，如果是则将这行存在结果集中

2.调用引擎接口取“下一行”，重复相同的逻辑判断，直到去这个表的最后一行

3.执行器将上述的遍历过程中所有满足的行记录集作为结果返回该客户端

至此，这个语句的执行完成。

对于有索引的表，执行逻辑也差不多，第一次调用的是“取满足条件的第一行”这个接口，之后循环取“满足条件的下一行”这个接口，这些接口都是引擎中已经定义好的。

慢查询日志中rows\_examined字段，表示这个语句执行过程中扫描了多少行，这个值就是执行器每次调用引擎获取数据行的累加的

**在有些的场景下，执行器调用一次，在引擎内部扫描多少行，因此引擎扫描行数跟rows\_examined并不是完全相同的，**

问题

如果表T没有K字段，而执行了这个语句 select \* from t where k1=1 那么肯定报错

Unknow column ‘k’ in where clause ，这个错误是在哪个阶段报出来的

分析器