

图文 02 为了执行SQL语句，你知道MySQL用了什么样的架构设计吗？

2935 人次阅读 2020-01-14 07:00:00

详情 评论

为了执行SQL语句，你知道MySQL用了什么样的架构设计吗？

如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑

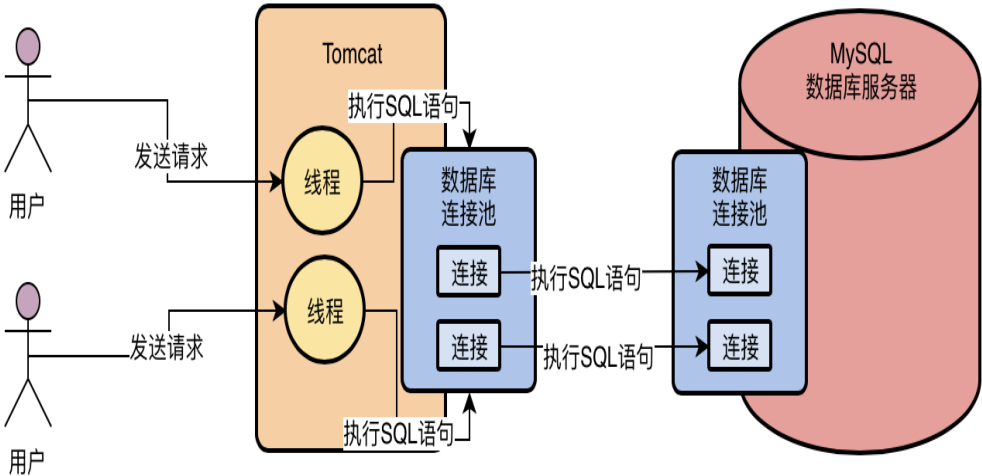
如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《MySQL专栏付费用户如何加群》（购买后可见）

1、把MySQL当个黑盒子一样执行SQL语句

上一讲我们已经说到，我们的系统采用数据库连接池的方式去并发访问数据库，然后数据库自己其实也会维护一个连接池，其中管理了各种系统跟这台数据库服务器建立的所有连接

我们先看下图回顾一下



当我们的系统只要能从数据库连接池获取到一个数据库连接之后，我们就可以执行增删改查的SQL语句了

从上图其实我们就可以看到，我们可以通过数据库连接把要执行的SQL语句发送给MySQL数据库。

然后呢？大部分同学了解到这个程度就停下来了，然后大家觉得要关注的可能主要就是数据库里的表结构，建了哪些索引，然后就按照SQL语法去编写增删改查SQL语句，把MySQL当个黑盒子去执行SQL语句就可以了。

我们只知道执行了insert语句之后，在表里会多出来一条数据；执行了update语句之后，会对表里的数据进行更改；执行了delete语句之后，会把表里的一条数据删除掉；执行了select语句之后，会从表里查询一些数据出来。

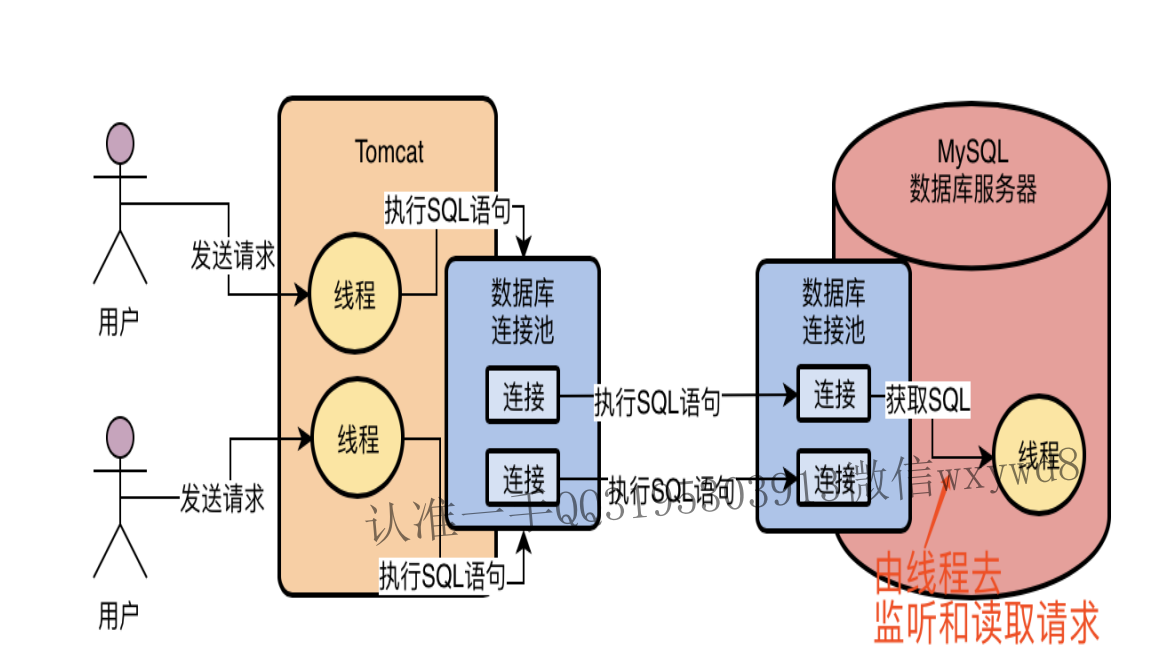
如果语句性能有点差？没关系，在表里建几个索引就可以了！可能这就是目前行业内很多工程师对数据库的一个认知，完全当他是黑盒子，来建表以及执行SQL语句。

但是大家既然跟着我开始学习了，从现在开始就要打破这种把数据库当黑盒子的认知程度，要深入底层，去探索数据库的工作原理以及生产问题的优化手段！

2、一个不变的原则：网络连接必须让线程来处理

现在假设我们的数据库服务器的连接池中的某个连接接收到了网络请求，假设就是一条SQL语句，那么大家先思考一个问题，谁负责从这个连接中去监听网络请求？谁负责从网络连接里把请求数据读取出来？

我想很多人恐怕都没思考过这个问题，但是如果大家对计算机基础知识有一个简单了解的话，应该或多或少知道一点，那就是网络连接必须得分配给一个线程去进行处理，由一个线程来监听请求以及读取请求数据，比如从网络连接中读取和解析出来一条我们的系统发送过去的SQL语句，如下图所示：



3、SQL接口：负责处理接收到的SQL语句

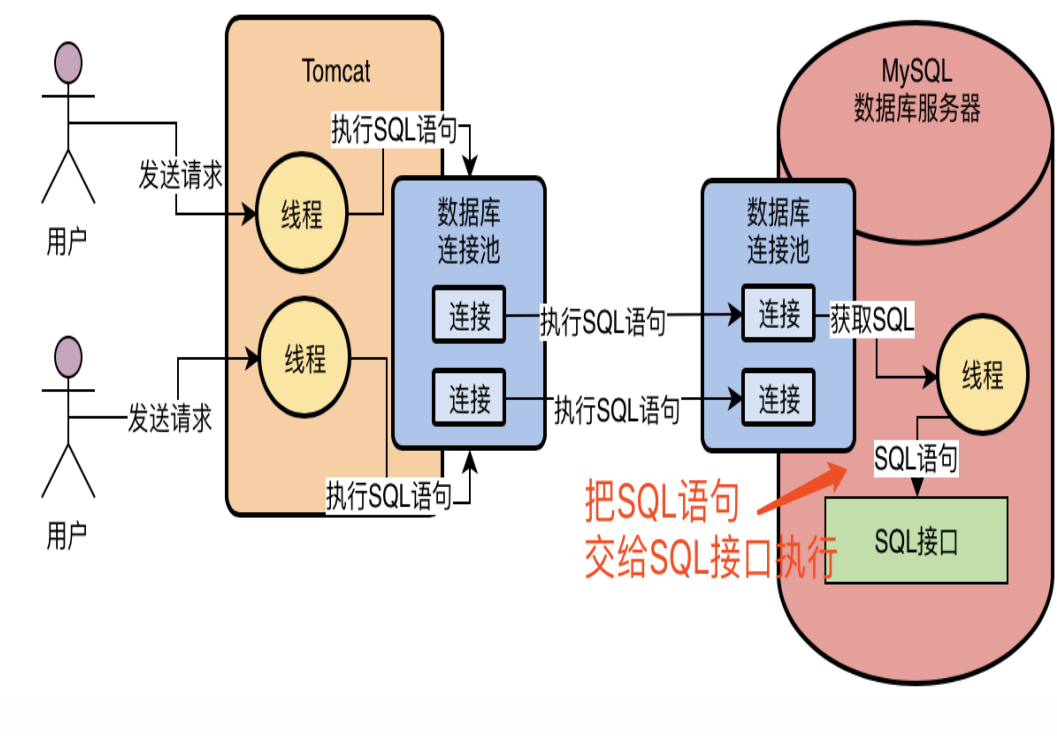
接着我们来思考一下，当MySQL内部的工作线程从一个网络连接中读取出来一个SQL语句之后，此时会如何来执行这个SQL语句呢？

其实SQL是一项伟大的发明，他发明了简单易用的数据读写的语法和模型，哪怕是个产品经理，或者是运营专员，甚至是销售专员，即使他不会技术，他也能轻松学会使用SQL语句。

但如果你要去执行这个SQL语句，去完成底层数据的增删改查，那这就是一项极度复杂的任务了！

所以MySQL内部首先提供了一个组件，就是SQL接口（SQL Interface），他是一套执行SQL语句的接口，专门用于执行我们发送给MySQL的那些增删改查的SQL语句

因此MySQL的工作线程接收到SQL语句之后，就会转交给SQL接口去执行，如下图。



4、查询解析器：让MySQL能看懂SQL语句

接着下一个问题来了，SQL接口怎么执行SQL语句呢？你直接把SQL语句交给MySQL，他能看懂和理解这些SQL语句吗？

比如我们来举一个例子，现在有这么一个SQL语句：
select id,name,age from users where id=1

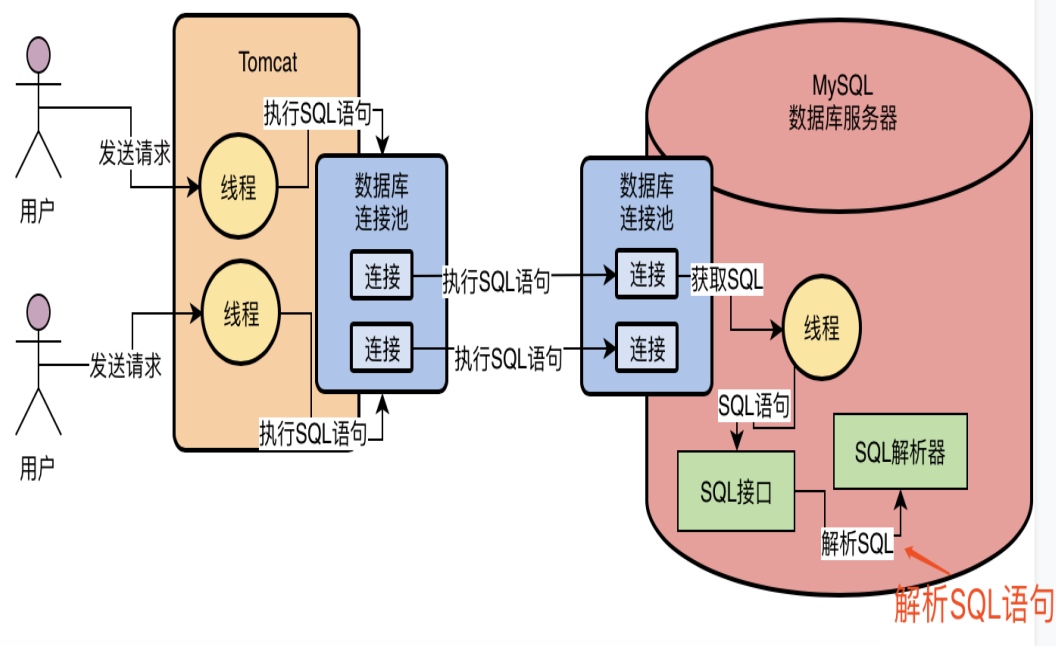
这个SQL语句，我们用人脑是直接就可以处理一下，只要懂SQL语法的人，立马大家就知道他是什么意思，但是MySQL自己本身也是一个系统，是一个数据库管理系统，他没法直接理解这些SQL语句！

所以此时有一个关键的组件要出场了，那就是**查询解析器**

这个查询解析器（Parser）就是负责对SQL语句进行解析的，比如对上面那个SQL语句进行一下拆解，拆解成以下几个部分：

- 我们现在要从“users”表里查询数据
- 查询“id”字段的值等于1的那行数据
- 对查出来的那行数据要提取里面的“id,name,age”三个字段。

所谓的SQL解析，就是按照既定的SQL语法，对我们按照SQL语法规则编写的SQL语句进行解析，然后理解这个SQL语句要干什么事情，如下图所示：



5、查询优化器：选择最优的查询路径

当我们通过解析器理解了SQL语句要干什么之后，接着会找查询优化器（Optimizer）来选择一个最优的查询路径。

可能有同学这里就不太理解什么是最优的查询路径了，这个看起来确实很抽象，当然，这个查询优化器的工作原理，后续将会是我们分析的重点，大家现在不用去纠结他的原理。

但是我们可以用一个极为通俗简单的例子，让大家理解一下所谓的最优查询路径是什么。

就用我们刚才讲的那个例子好了，我们现在理解了一个SQL想要干这么一个事儿：我们现在要从“users”表里查询数据，查询“id”字段的值等于1的那行数据，对查出来的那行数据要提取里面的“id,name,age”三个字段。

事是明白了，但是到底应该怎么来实现呢？

你看，要完成这个事儿我们有以下几个查询路径（纯属用于大家理解的例子，不代表真实的MySQL原理，但是通过这个例子，大家肯定能理解所谓最优查询路径的意思）：

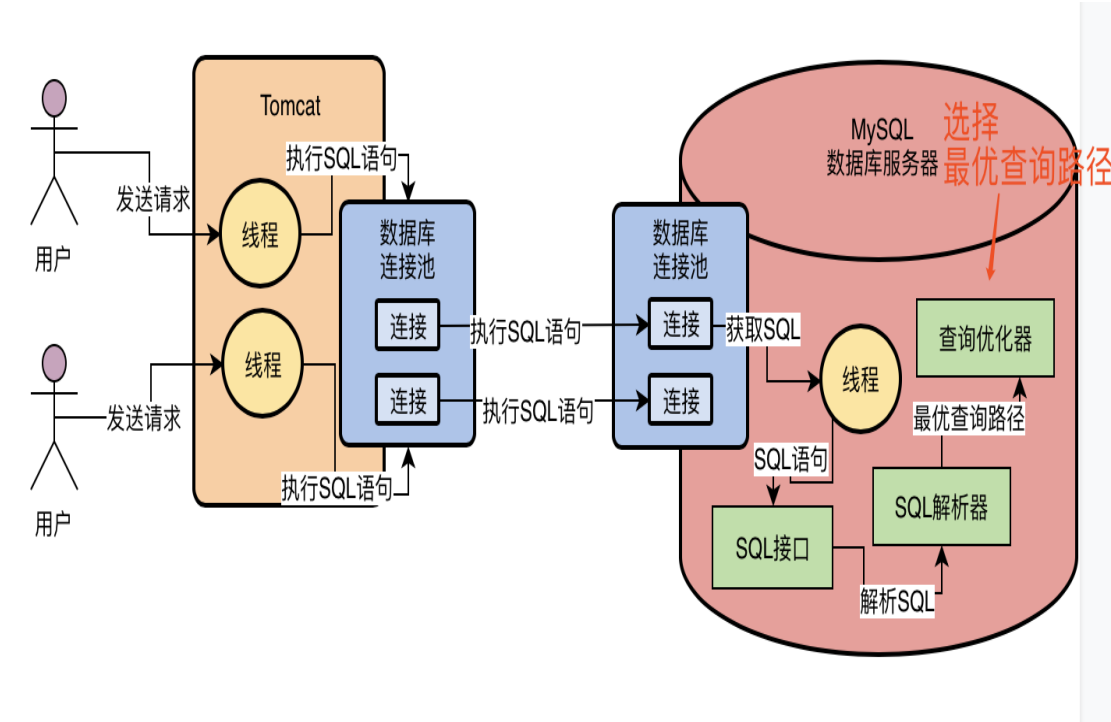
直接定位到“users”表中的“id”字段等于1的一行数据，然后查出来那行数据的“id,name,age”三个字段的值就可以了
先把“users”表中的每一行数据的“id,name,age”三个字段的值都查出来，然后从这批数据里过滤出来“id”字段等于1的那行数据的“id,name,age”三个字段

上面这就是一个最简单的SQL语句的两种实现路径，其实我们会发现，要完成这个SQL语句的目标，两个路径都可以做到，但是哪一种更好呢？显然感觉上是第一种查询路径更好一些。

所以查询优化器大概就是干这个的，他会针对你编写的几十行、几百行甚至上千行的复杂SQL语句生成查询路径树，然后从里面选择一条最优的查询路径出来。

相当于他会告诉你，你应该按照一个什么样的步骤和顺序，去执行哪些操作，然后一步一步的把SQL语句就给完成了。

我们来一起看看下面的图：



6、调用存储引擎接口，真正执行SQL语句

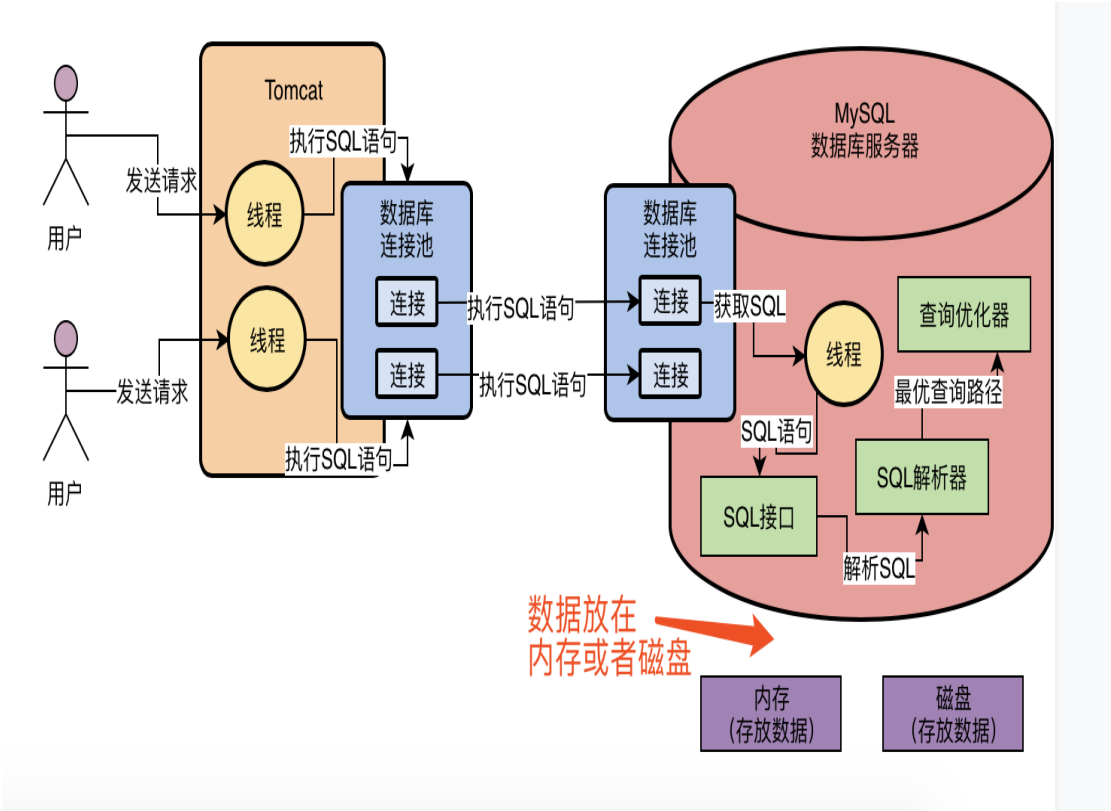
最后一步，就是把查询优化器选择的最优查询路径，也就是你到底应该按照一个什么样的顺序和步骤去执行这个SQL语句的计划，把这个计划交给底层的存储引擎去真正的执行。这个存储引擎是MySQL的架构设计中很有特色的一个环节。

不知道大家是否思考过，真正在执行SQL语句的时候，要不然是更新数据，要不然是查询数据，那么数据你觉得存放在哪里？

说白了，数据库也不是什么神秘莫测的东西，你可以把他理解为一个类似你平时写的图书馆管理系统、电信计费系统、电商订单系统之类的系统罢了。

数据库自己就是一个编程语言写出来的系统而已，然后启动之后也是一个进程，执行他里面的各种代码，也就是我们上面所说的那些东西。所以对数据库而言，我们的数据要不然是放在内存里，要不然是放在磁盘文件里，没什么特殊的地方！

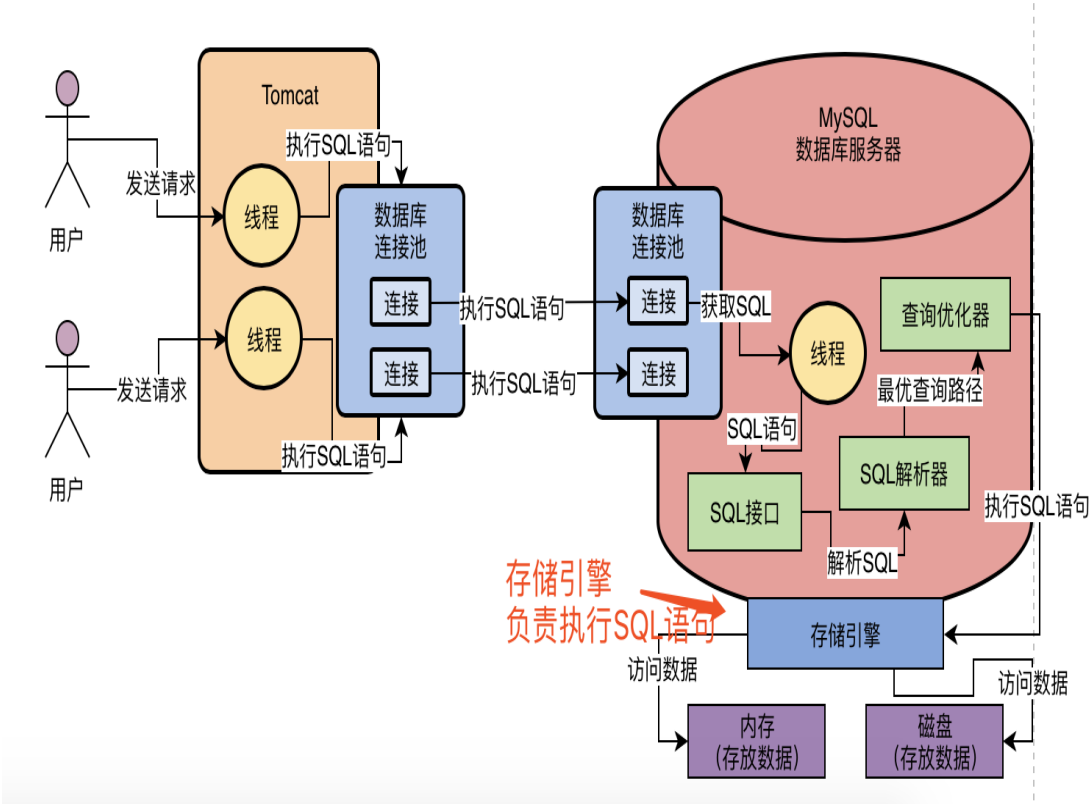
所以我们来思考一下，假设我们的数据有的存放在内存里，有的存放在磁盘文件里，如下图所示。



那么现在问题来了，我们已经知道一个SQL语句要如何执行了，但是我们现在怎么知道哪些数据在内存里？哪些数据在磁盘里？我们执行的时候是更新内存的数据？还是更新磁盘的数据？我们如果更新磁盘的数据，是先查询哪个磁盘文件，再更新哪个磁盘文件？

是不是感觉一头雾水

所以这个时候就需要存储引擎了，存储引擎其实就是执行SQL语句的，他会按照一定的步骤去查询内存缓存数据，更新磁盘数据，查询磁盘数据，等等，执行诸如此类的一系列的操作，如下图所示。



MySQL的架构设计中，SQL接口、SQL解析器、查询优化器其实都是通用的，他就是一套组件而已。

但是存储引擎的话，他是支持各种各样的存储引擎的，比如我们常见的InnoDB、MyISAM、Memory等等，我们是可以选择使用哪种存储引擎来负责具体的SQL语句执行的。

当然现在MySQL一般都是使用InnoDB存储引擎的，至于存储引擎的原理，后续我们也会深入一步一步分析，大家不必着急。

7、执行器：根据执行计划调用存储引擎的接口

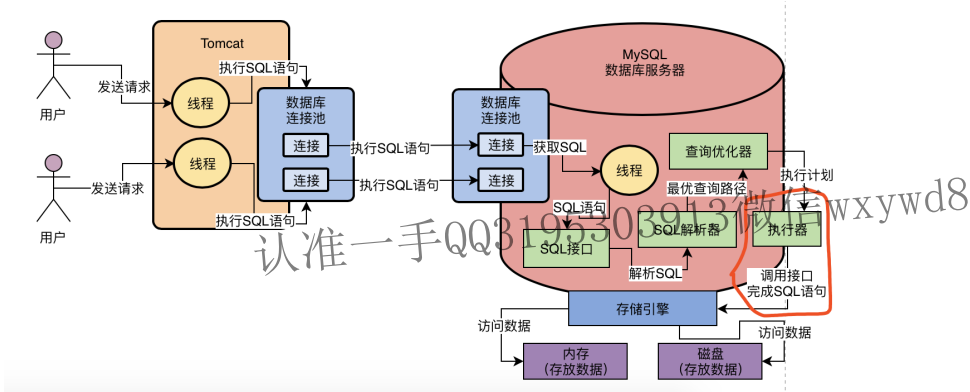
那么看完存储引擎之后，我们回过头来思考一个问题，存储引擎可以帮助我们去访问内存以及磁盘上的数据，那么是谁来调用存储引擎的接口呢？

其实我们现在还漏了一个执行器的概念，这个执行器会根据优化器选择的执行方案，去调用存储引擎的接口按照一定的顺序和步骤，就把SQL语句的逻辑给执行了。

举个例子，比如执行器可能会先调用存储引擎的一个接口，去获取“users”表中的第一行数据，然后判断一下这个数据的“id”字段的值是否等于我们期望的一个值，如果不是的话，那就继续调用存储引擎的接口，去获取“users”表的下一行数据。

就是基于上述的思路，**执行器就会去根据我们的优化器生成的一套执行计划，然后不停的调用存储引擎的各种接口去完成SQL语句的执行计划**，大致就是不停的更新或者提取一些数据出来

我们看下图的示意



8、小思考题：打开脑洞，你觉得不同的存储引擎是用来干什么的？

今天给大家留一个小的思考题，就是你先别管MySQL有哪些存储引擎，你就从业务场景来出发考虑，有的场景可能是高并发的更新，有的场景可能是大规模数据查询，有的场景可能是允许丢失数据的

那么你觉得如果让你来设计存储引擎，你觉得应该有哪些存储引擎，分别适用于什么场景？

希望大家在评论区留言，踊跃的思考和回复。

End

专栏版权归公众号狸猫技术窝所有

未经许可不得传播，如有侵权将追究法律责任