# 查询分析器设计说明书

1. **设计背景**

在需要进行数据库管理的情况下，我们经常使用到数据库可视化管理工具帮助我们进行操作，使得我们脱离shell命令行，比较容易进行数据库管理。市面上常见的的数据库可视化管理工具有Navicat，Mysql Workbench等。这次主要仿照Navicat的界面实现我们常用的数据库查询功能。

1. **环境搭建**

1)操作系统: windows 10

2)编程语言: java(jdk1.8)

3)集成开发环境: IntelliJ IDEA 2020

5)jar依赖包(可使用maven进行下载):

I)springboot相关依赖

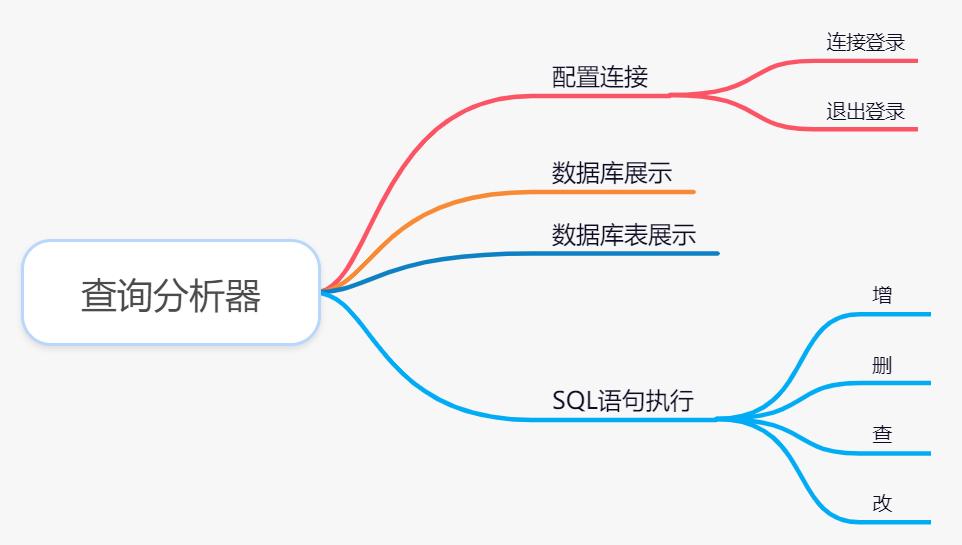
II)fastjson 1.2.75

III)lombok

IV)spring-boot-start-themleaf

V)spark jdbc

**三．功能设计**



1. 配置连接:配置连接的数据库信息，连接成功后才能进行数据库的查询功能。
2. 数据库展示:展示配置好的服务器连接下的所有数据库。
3. 数据库表展示:展示选中数据库中的所有表格。
4. SQL语句执行:在编辑框中编写SQL语句，执行SQL语句可以获取返回结果。

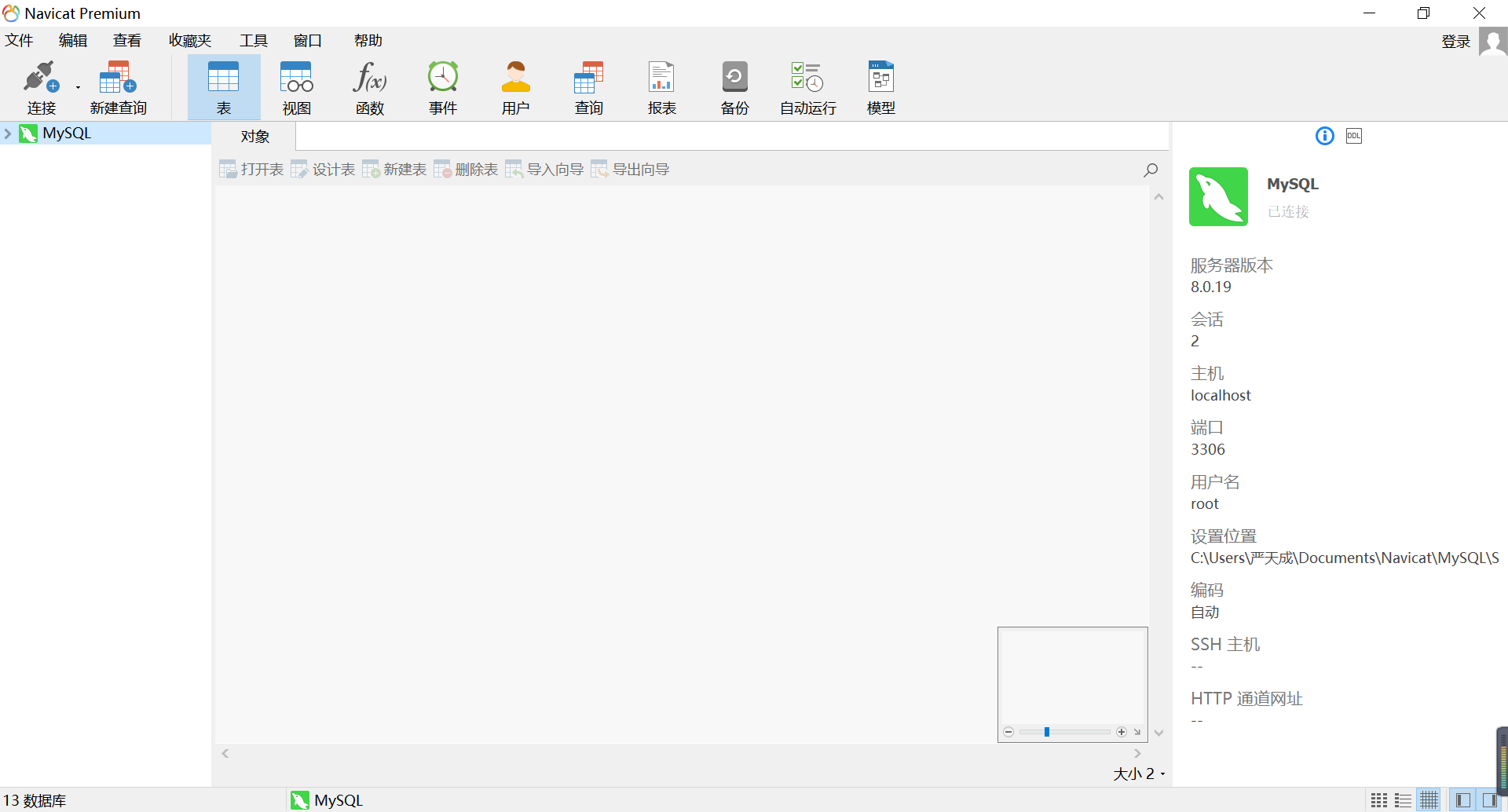
**四．程序实现**

1）本程序和Navicat不一样，使用的是B/S的架构。考虑到功能简单，为了简单项目的复杂度，不采用前后端分离的方式进行实现。

1. 前端使用了vue.JS和element ui框架，简化开发。前端的页面设计如下



功能区域主要是仿照Navicat的界面。



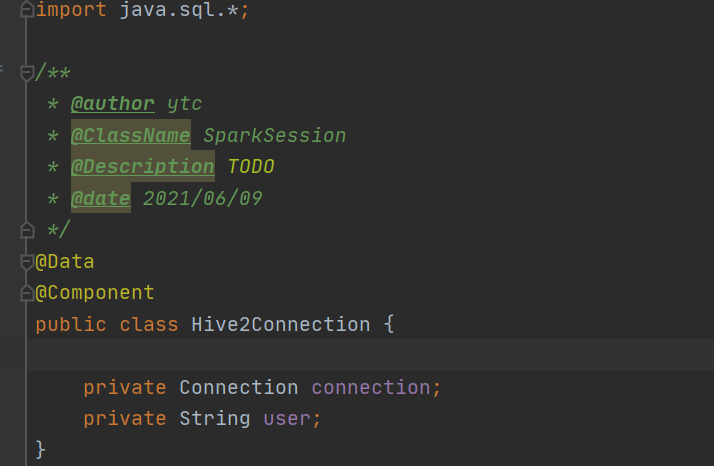
Navicat界面

1. 后端使用Springboot框架，参考后端常用的MVC分层架构，但是考虑到该程序与一般的应用程序不同，不需要Service层进行业务处理，因此舍弃了Service层，让Controller层直接调用Dao层进行数据库查询。



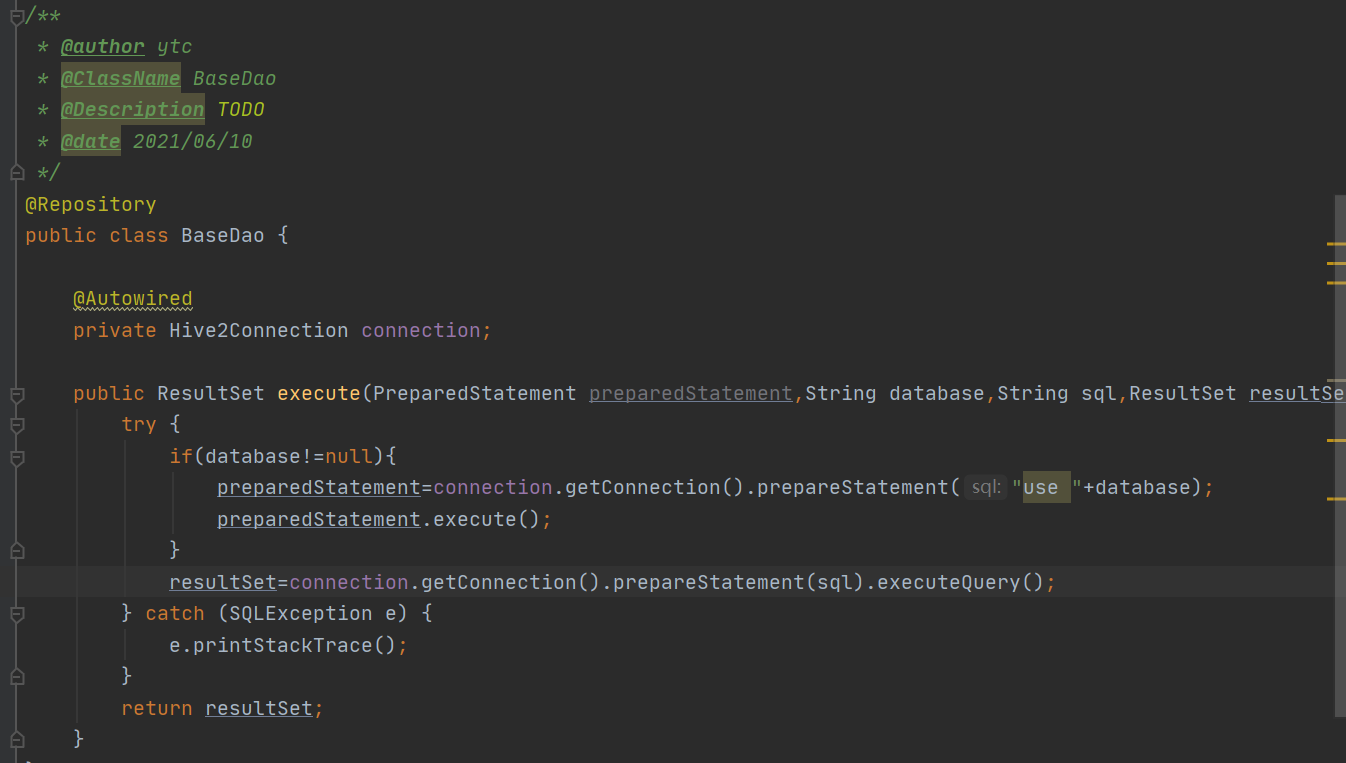
1. pojo包设计说明:

Pojo包下只有一个类Hive2Connection。其作用是保存用户连接的信息，因为数据库连接成功后，用户可以在同一个连接下进行长时间的查询，因此必须要保存用户连接，避免每次查询都要进行一次JDBC的连接。在实现保存连接时，是给Hive2Connection加上@Component的注解，这样就能让Spring IOC容器帮助我们管理Hive2Connection对象，Hive2Connection对象是全局唯一的，而且其数据域Connection可以保存JDBC连接。那么我们在进行数据库查询时，就能通过Hive2Connection对象获取到Connection，不需要每次都进行JDBC连接。



1. dao层设计说明:

dao层是程序和数据库直接打交道的层，由于本程序所有的SQL语句都在前端传来，不需要分区特定的增删查改功能，因此里面只有一个类BaseDao，通过execute(...)函数来进行统一的SQL查询。BaseDao里面依赖了Hive2Connection对象，由IOC容器自动注入。在查询过程中，使用Hive2Connection对象保存的JDBC连接进行数据库查询。Execute()参数有PreparedStatement，database，sql，ResultSet。其中PreparedStatement对象用于保存sql语句，它可以在程序运行时预编译，防止SQL注入，database指定使用的数据库，如果非空，则在数据库查询时想要执行use+database进行数据库转换。ReslutSet保存查询结果，然后将结果返回给API层(Controller层)。



1. api层设计说明:

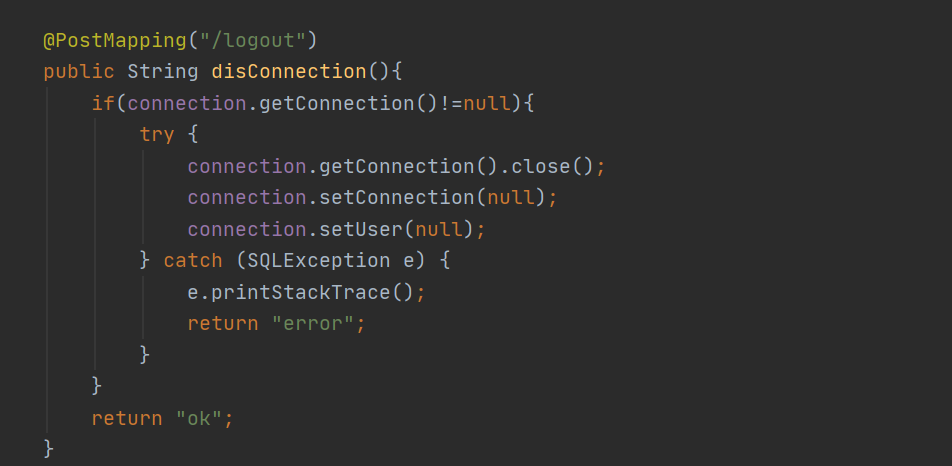
api层是前后端交互的接口。里面有3个类。

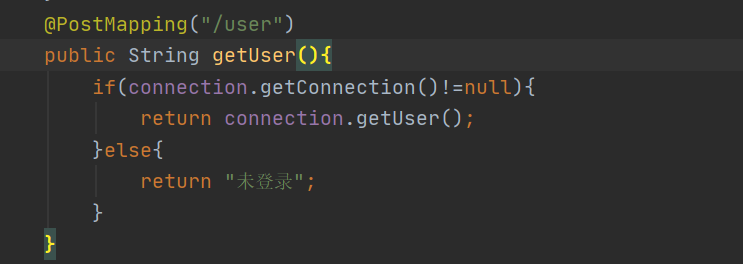
1. indexApi类:这个类主要是进行主页的跳转的，使用themeleaf模板引擎去进行页面跳转，当然本程序只有一个页面。



1. ConnectApi类:这个类是用于接受前端发送的连接请求相关信息。里面有3个方法。getConnection(...)，disConnection()，getUser()。getConnection(...)参数是前端传来的连接参数，首先它会验证连接是否已经建立，否则建立JDBC连接，并保存在Hive2Connection中。DisConnection()是处理前端发送的断开连接的请求，它会将Hive2Connection里面的JDBC连接关闭，同时清空连接信息。GetUser()是为了在前端没有断开连接时，重新打开页面可以直接获得登录信息。







1. SQLApi类:这个类主要用于处理前端传来的查询请求。主要有3个方法：showDatabases()，showTables(...)，query(...)。showDatabases()用于列出所有的数据库名字，返回前端渲染。showTables(...)接收前端传来的数据库名，然后去查出里面所有的表。query(...)的参数是指定的database和传来的sql语句，然后在方法体里面进行查询，以json数组的形势返回给前端。当然，这3个方法最后会关闭PrepareStatement，Result，释放资源。

