# 项目需求分析

还是之前那个在线报名系统

项目越大越好，遇到的问题肯定越多。学习的时候时间是有限的，重点掌握方法

首页面 课程的展示

电商项目 首页，展示商品的信息

功能模块

前台功能 java基础 oop

前台功能 servlet+jsp mvc

个人中心 ssm+thymeleaf 三层架构dao service action

后台管理 vue+springboot 微服务架构模式

微服务模式 前后台分离的项目 整个项目 使用vue+springboot搭建 答辩项目

springboot 框架的整合工具，可以搭建微服务的一个框架

springcloud feign 远程调用，实现分布式模式的项目开发

面向服务模式

vue 可以直接使用，把之前从本地获取数据部分，改为使用axios 调用后台服务获取数据的

演示一下项目

如果要使用多种数据库，之间有兼容性问题

使用一个工具pd 设计

自动转换为mysql orcle sqlserver等主流的数据库

还可以生成对象模式

还可以生成一些业务代码

而且表与表之间的关系很清晰

1.需求分析

2.数据库设计

根据需求，设计数据库以及表，表之间的关系等

之前我们数据是保存到本地的。现在保存到数据库

course表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 备注 |
| Id | int |  | √ | 编号 |
| name | char | 20 |  | 名称 |

user表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

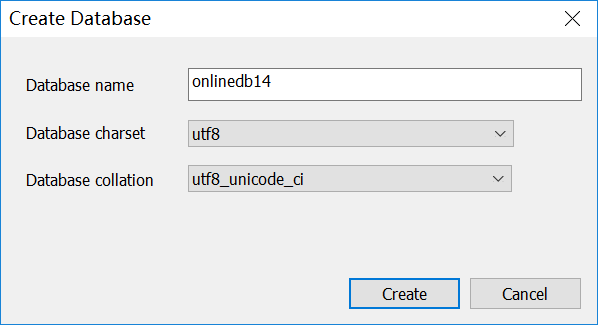
mycourse表

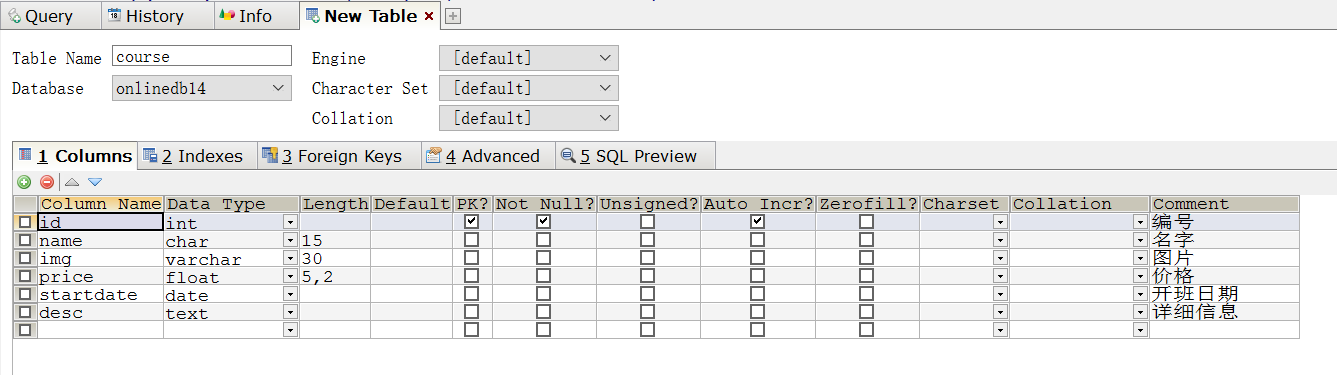
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

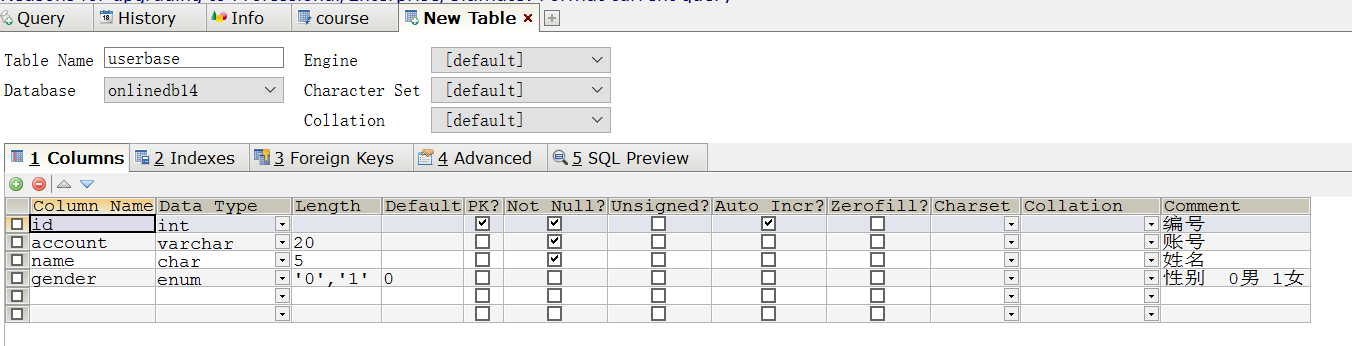
设计师很重要，从全局的角度去分析

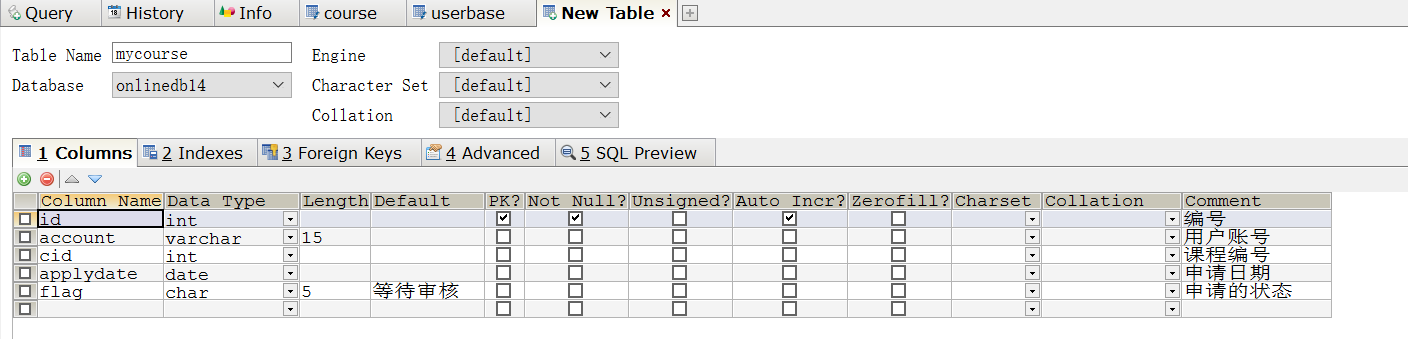
数据库设计师

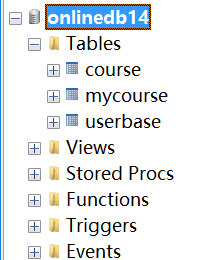
下面开始跟着我操作

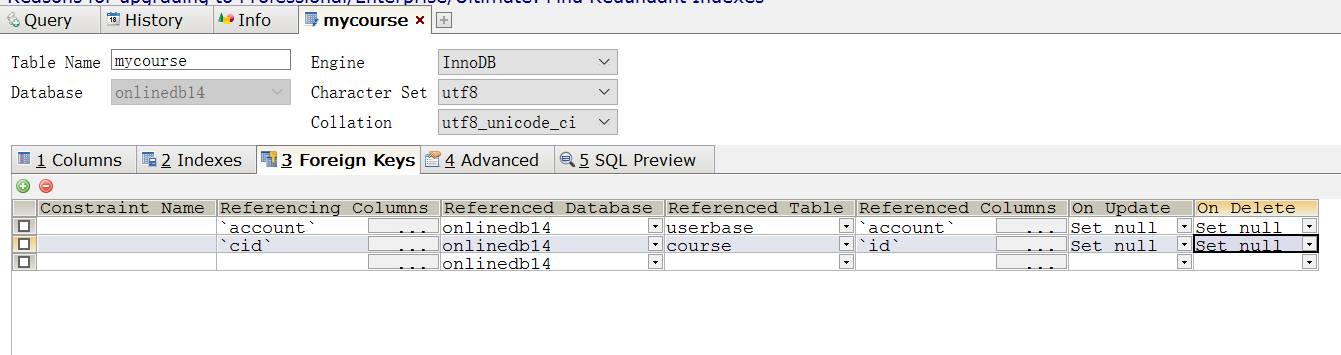












3.系统架构

三层架构设计模式

ui 界面层

service 业务层

dao 数据访问层

util 共同部分层

entity 试题层 对应数据库表的

vo 值对象层 视图对象层 对应页面上需要的信息，对应数据库多个表关联的信息

打开eclipse

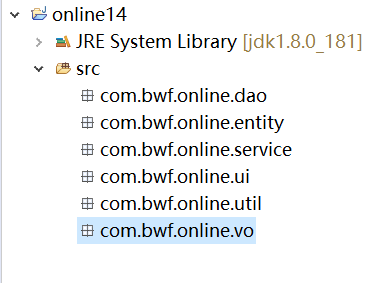
分层含义，就是为了以后层次切换，可以使用原来的代码

ui 换了，换成jsp 了如果业务写在ui层中，那么以后换成jsp，业务是重新写 如果放在独立的业务层，那么以后切换界面，就可以直接调用这个业务层了

jdbc 以后换框架 mybatis

不用dao层。

一句话，分层有很多优势



开发的顺序是什么？

ui

service

dao

没关系的，可以同时进行

一个团队

如果一个人，可以从ui开始

也可以从dao 开始

最好还是从dao底层开始设计

设计与开发

dao层也需要 共同的代码

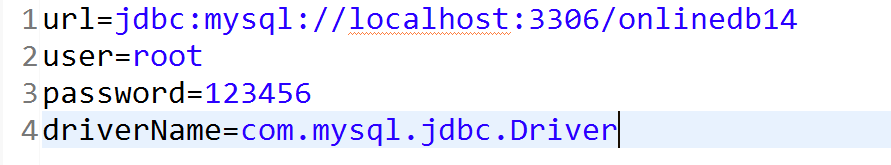
数据库驱动注册，连接，关闭等操作

再从头写一次。

编写 DBHelp类

数据库配置文件

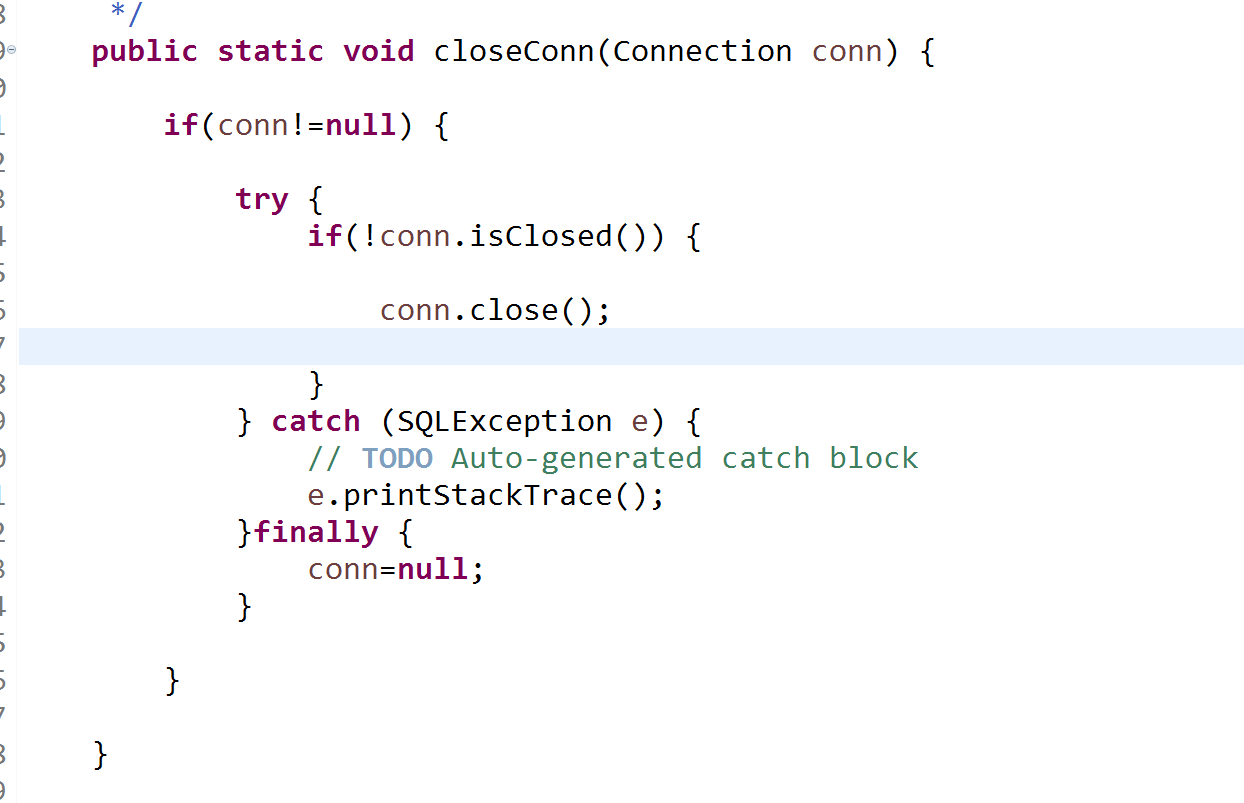
db.config.properties



编写好之后，测试

缺少驱动包

关闭连接对象

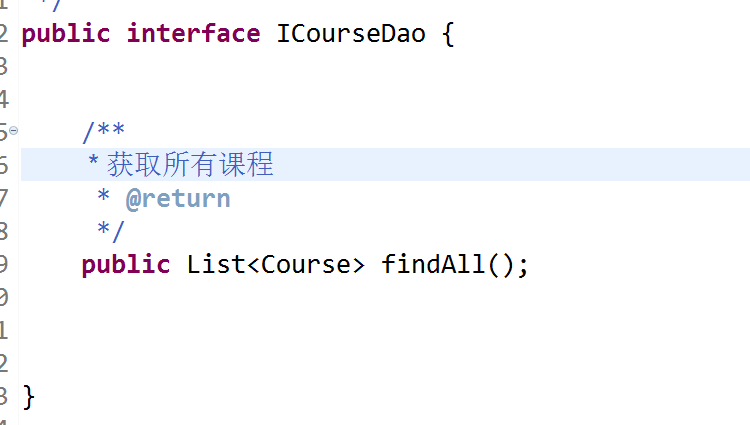


测试一下关闭连接

下面编写dao实现了

在数据库中录入数据

编写dao接口



实现类

还是完善接口中的方法定义，以及实现