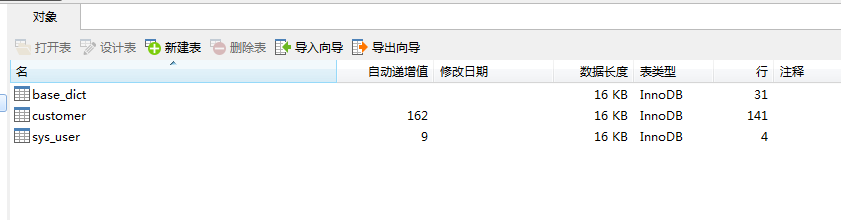
**Mybatis、Springmvc练习**

**CRM系统**

# 数据库

数据库使用mysql 数据库。



1. 创建crm数据库
2. 将参考资料中的sql脚本导入到数据库中

# 工程搭建

工程使用Springmvc、spring、mybatis框架整合完成。

## SqlMapConfig.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"* ?>  <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-config.dtd">  <configuration>  </configuration> |

## applicationContext-dao.xml

|  |
| --- |
| <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:mvc=*"http://www.springframework.org/schema/mvc"*  xmlns:context=*"http://www.springframework.org/schema/context"*  xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:task=*"http://www.springframework.org/schema/task"* xmlns:dubbo=*"http://code.alibabatech.com/schema/dubbo"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*  *http://www.springframework.org/schema/mvc*  *http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd*  *http://www.springframework.org/schema/context*  *http://www.springframework.org/schema/context/spring-context-4.0.xsd*  *http://www.springframework.org/schema/aop*  *http://www.springframework.org/schema/aop/spring-aop-4.0.xsd*  *http://www.springframework.org/schema/tx*  *http://www.springframework.org/schema/tx/spring-tx-4.0.xsd*  *http://www.springframework.org/schema/task*  *http://www.springframework.org/schema/task/spring-task-4.0.xsd*  *http://code.alibabatech.com/schema/dubbo*  *http://code.alibabatech.com/schema/dubbo/dubbo.xsd"*>  <!-- 配置 读取properties文件 jdbc.properties -->  <context:property-placeholder location=*"classpath:jdbc.properties"* />  <!-- 配置 数据源 -->  <bean id=*"dataSource"* class=*"com.alibaba.druid.pool.DruidDataSource"*>  <!-- 驱动 -->  <property name=*"driverClassName"* value=*"${jdbc.driver}"* />  <!-- url -->  <property name=*"url"* value=*"${jdbc.url}"* />  <!-- 用户名 -->  <property name=*"username"* value=*"${jdbc.username}"* />  <!-- 密码 -->  <property name=*"password"* value=*"${jdbc.password}"* />  </bean>  <!-- 配置 Mybatis的工厂 -->  <bean class=*"org.mybatis.spring.SqlSessionFactoryBean"*>  <!-- 数据源 -->  <property name=*"dataSource"* ref=*"dataSource"* />  <!-- 配置Mybatis的核心 配置文件所在位置 -->  <property name=*"configLocation"* value=*"classpath:SqlMapConfig.xml"* />  <!-- 配置pojo别名 -->  <property name=*"typeAliasesPackage"* value=*"cn.itcast.core.bean"*></property>  </bean>  <!-- 配置 1：原始Dao开发 接口实现类 Mapper.xml 三个 2：接口开发 接口 不写实现类 Mapper.xml 二个 （UserDao、ProductDao  、BrandDao。。。。。。。） 3：接口开发、并支持扫描 cn.itcast.core.dao(UserDao。。。。。） 写在此包下即可被扫描到 -->  <bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>  <property name=*"basePackage"* value=*"cn.itcast.core.dao"* />  </bean>  </beans> |

Jdbc.properties

|  |
| --- |
| jdbc.driver=com.mysql.jdbc.Driver  jdbc.url=jdbc:mysql://localhost:3306/crm?characterEncoding=utf-8  jdbc.username=root  jdbc.password=root |

## applicationContext-service.xml

|  |
| --- |
| <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:mvc=*"http://www.springframework.org/schema/mvc"*  xmlns:context=*"http://www.springframework.org/schema/context"*  xmlns:aop=*"http://www.springframework.org/schema/aop"*  xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:task=*"http://www.springframework.org/schema/task"*  xmlns:dubbo=*"http://code.alibabatech.com/schema/dubbo"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*  *http://www.springframework.org/schema/mvc*  *http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd*  *http://www.springframework.org/schema/context*  *http://www.springframework.org/schema/context/spring-context-4.0.xsd*  *http://www.springframework.org/schema/aop*  *http://www.springframework.org/schema/aop/spring-aop-4.0.xsd*  *http://www.springframework.org/schema/tx*  *http://www.springframework.org/schema/tx/spring-tx-4.0.xsd*  *http://www.springframework.org/schema/task*  *http://www.springframework.org/schema/task/spring-task-4.0.xsd*  *http://code.alibabatech.com/schema/dubbo*  *http://code.alibabatech.com/schema/dubbo/dubbo.xsd"*>      <!-- 配置 扫描 @Service -->  <context:component-scan base-package=*"cn.itcast.core.service"*/>        </beans> |

## applicationContext-trans.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:context=*"http://www.springframework.org/schema/context"* xmlns:p=*"http://www.springframework.org/schema/p"*  xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*  *http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd*  *http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.0.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd*  *http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.0.xsd"*>  <!-- 事务管理器 -->  <bean id=*"transactionManager"*  class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>  <!-- 数据源 -->  <property name=*"dataSource"* ref=*"dataSource"* />  </bean>  <!-- 通知 -->  <tx:advice id=*"txAdvice"* transaction-manager=*"transactionManager"*>  <tx:attributes>  <!-- 传播行为 -->  <tx:method name=*"save\*"* propagation=*"REQUIRED"* />  <tx:method name=*"insert\*"* propagation=*"REQUIRED"* />  <tx:method name=*"add\*"* propagation=*"REQUIRED"* />  <tx:method name=*"create\*"* propagation=*"REQUIRED"* />  <tx:method name=*"delete\*"* propagation=*"REQUIRED"* />  <tx:method name=*"update\*"* propagation=*"REQUIRED"* />  <tx:method name=*"find\*"* propagation=*"SUPPORTS"* read-only=*"true"* />  <tx:method name=*"select\*"* propagation=*"SUPPORTS"* read-only=*"true"* />  <tx:method name=*"get\*"* propagation=*"SUPPORTS"* read-only=*"true"* />  </tx:attributes>  </tx:advice>  <!-- 切面 -->  <aop:config>  <aop:advisor advice-ref=*"txAdvice"*  pointcut=*"execution(\* cn.itcast.core.service.\*.\*(..))"* />  </aop:config>  </beans> |

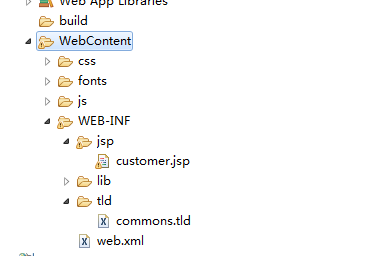
## Springmvc.xml

|  |
| --- |
| <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:mvc=*"http://www.springframework.org/schema/mvc"*  xmlns:context=*"http://www.springframework.org/schema/context"*  xmlns:aop=*"http://www.springframework.org/schema/aop"*  xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:task=*"http://www.springframework.org/schema/task"*  xmlns:dubbo=*"http://code.alibabatech.com/schema/dubbo"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*  *http://www.springframework.org/schema/mvc*  *http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd*  *http://www.springframework.org/schema/context*  *http://www.springframework.org/schema/context/spring-context-4.0.xsd*  *http://www.springframework.org/schema/aop*  *http://www.springframework.org/schema/aop/spring-aop-4.0.xsd*  *http://www.springframework.org/schema/tx*  *http://www.springframework.org/schema/tx/spring-tx-4.0.xsd*  *http://www.springframework.org/schema/task*  *http://www.springframework.org/schema/task/spring-task-4.0.xsd*  *http://code.alibabatech.com/schema/dubbo*  *http://code.alibabatech.com/schema/dubbo/dubbo.xsd"*>    <!-- 加载属性文件 -->  <context:property-placeholder location=*"classpath:resource.properties"*/>  <!-- 配置扫描 器 -->  <context:component-scan base-package=*"cn.itcast.core.web.controller"*/>  <!-- 配置处理器映射器 适配器 -->  <mvc:annotation-driven/>    <!-- 配置视图解释器 jsp -->  <bean id=*"jspViewResolver"* class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>  <property name=*"prefix"* value=*"/WEB-INF/jsp/"*/>  <property name=*"suffix"* value=*".jsp"*/>  </bean>    </beans> |

## Web.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app version=*"2.5"* xmlns=*"http://java.sun.com/xml/ns/javaee"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee*  *http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*>  <welcome-file-list>  <welcome-file>customer.action</welcome-file>  </welcome-file-list>  <!-- 上下文的位置 -->  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:applicationContext-\*.xml</param-value>  </context-param>  <!-- Spring的监听器 -->  <listener>  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  </listener>  <!-- POST提交过滤器 UTF-8 -->  <filter>  <filter-name>encoding</filter-name>  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>  <init-param>  <param-name>encoding</param-name>  <param-value>UTF-8</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>encoding</filter-name>  <url-pattern>\*.action</url-pattern>  </filter-mapping>  <!-- 前端控制器 -->  <servlet>  <servlet-name>crm</servlet-name>  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  <init-param>  <param-name>contextConfigLocation</param-name>  <!-- 此处不配置 默认找 /WEB-INF/[servlet-name]-servlet.xml -->  <param-value>classpath:springmvc.xml</param-value>  </init-param>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>crm</servlet-name>  <!-- 1:\*.do \*.action 拦截以.do结尾的请求 (不拦截 jsp png jpg .js .css) 2:/ 拦截所有请求  （不拦截.jsp) 建议使用此种 方式 （拦截 .js.css .png) (放行静态资源） 3:/\* 拦截所有请求（包括.jsp) 此种方式 不建议使用 -->  <url-pattern>\*.action</url-pattern>  </servlet-mapping>  </web-app> |

## 加入jsp及分页标签



# 查询条件初始化

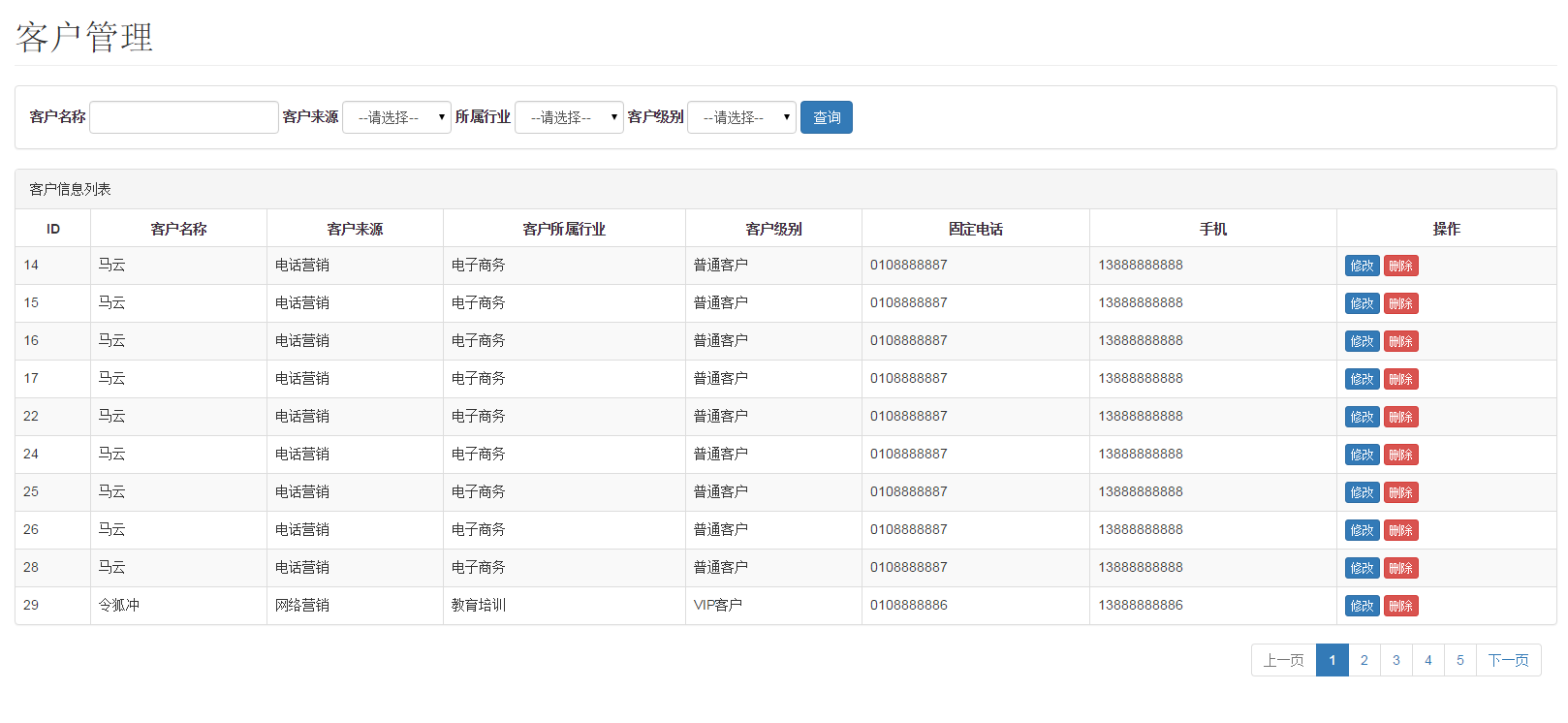
## 需求



初始化查询条件下拉列表。

# 客户列表展示

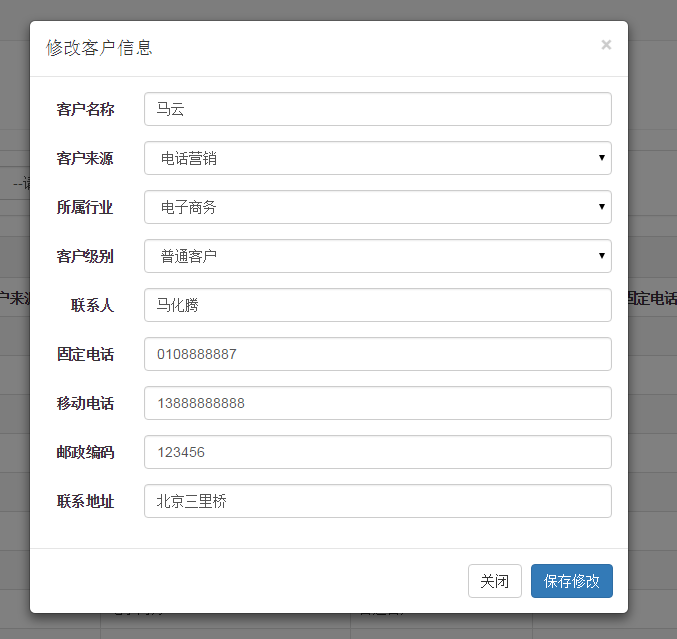
## 需求



展示商品列表，并且可以根据查询条件过滤查询结果，并且实现分页处理。

# 修改客户信息

## 需求



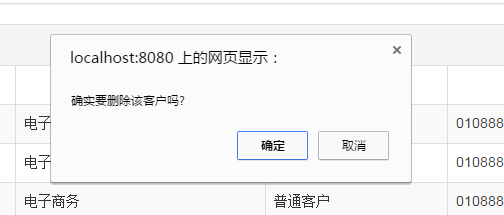
1. 点击客户列表中的“修改”按钮弹出客户信息修改对话框，并初始化客户信息
2. 点击“保存修改”按钮将修改后的结果保存到数据库中

# 删除客户

## 需求



点击客户列表中的删除按钮，提示“警告信息”



点击确定后删除用户信息，并刷新页面。