

CRM - 客户管理

学习目标

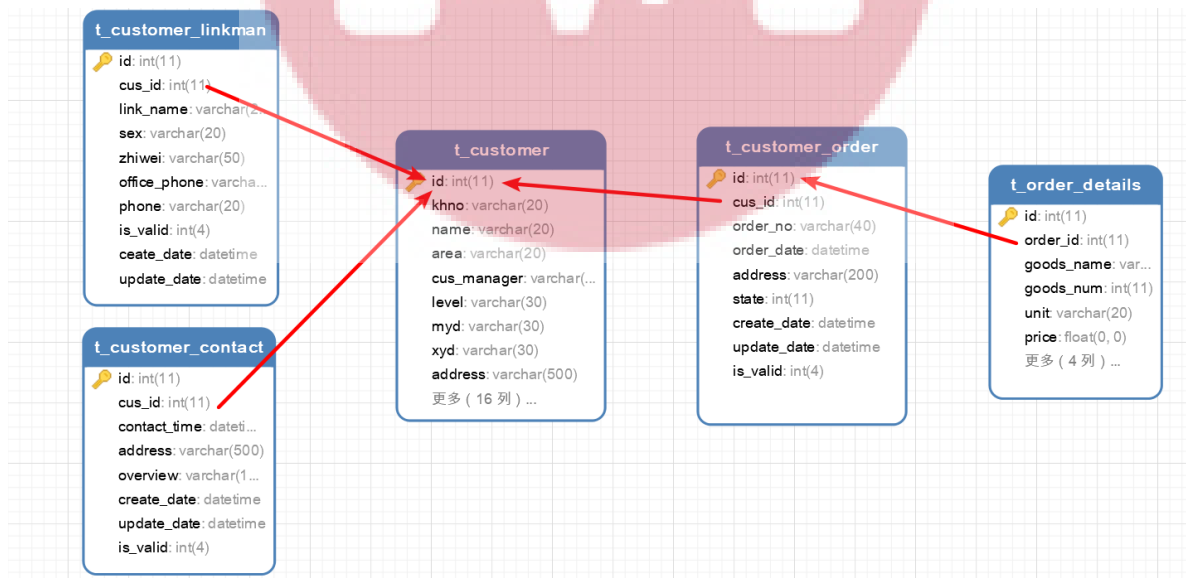


表结构设计

客户信息管理表结构

这里主要涉及表:

t_customer 客户表、t_customer_contact 客户交往记录表、t_customer_linkman 客户联系人表、t_customer_order 客户订单表、t_order_details 订单详情表





t_customer	客户信息表			
	字段	字段类型	字段限制	字段描述
主键	id	int(11)	自增	id主键
	khno	varchar(20)	可空	客户编号
	name	varchar(20)	可空	客户姓名
	area	varchar(20)	可空	客户所属地区
	cus_manager	varchar(20)	可空	客户经理
	level	varchar(30)	可空	客户级别
	myd	varchar(30)	可空	客户满意度
	xyd	varchar(30)	可空	客户信用度
	address	varchar(500)	可空	客户地址
	post_code	varchar(50)	可空	邮编
	phone	varchar(20)	可空	联系电话
	fax	varchar(20)	可空	传真
	web_site	varchar(20)	可空	网址
	yyzzzch	varchar(50)	可空	营业执照注册号
	fr	varchar(20)	可空	法人代表
	zczj	varchar(20)	可空	注册资金
	nyye	varchar(20)	可空	年营业额
	khyh	varchar(50)	可空	开户银行
	khzh	varchar(50)	可空	开户账号
	dsdjh	varchar(50)	可空	地税登记号
	gsdjh	varchar(50)	可空	国税登记号
	state	int(11)	可空	流失状态
	is_valid	int(4)	可空	有效状态
	create_date	datetime	可空	创建时间
	update_date	datetime	可空	更新时间



t_customer_contact	客户交往记录表			
	字段	字段类型	字段限制	字段描述
主键	id	int(11)	自增	id主键
	cus_id	int(11)	可空	客户id
	contact_time	datetime	可空	交往时间
	address	varchar(500)	可空	交往地址
	overview	varchar(100)	可空	地址
	create_date	datetime	可空	创建时间
	update_date	datetime	可空	更新时间
	is_valid	int(4)	可空	有效状态

t_customer_linkman	客户联系人表			
	字段	字段类型	字段限制	字段描述
主键	id	int(11)	自增	id主键
	cus_id	int(11)	可空	客户id
	link_name	varchar(20)	可空	联系人姓名
	sex	varchar(20)	可空	性别
	zhiwei	varchar(50)	可空	职位
	office_phone	varchar(50)	可空	办公电话
	phone	varchar(20)	可空	手机号
	is_valid	int(4)	可空	有效状态
	create_date	datetime	可空	创建时间
	update_date	datetime	可空	更新时间



t_customer_order	客户订单			
	字段	字段类型	字段限制	字段描述
主键	id	int(11)	自增	id主键
	cus_id	int(11)	可空	客户id
	order_no	varchar(40)	可空	订单编号
	order_date	datetime	可空	下单时间
	address	varchar(200)	可空	地址
	state	int(11)	可空	状态
	create_date	datetime	可空	创建时间
	update_date	datetime	可空	更新时间
	is_valid	int(4)	可空	有效状态

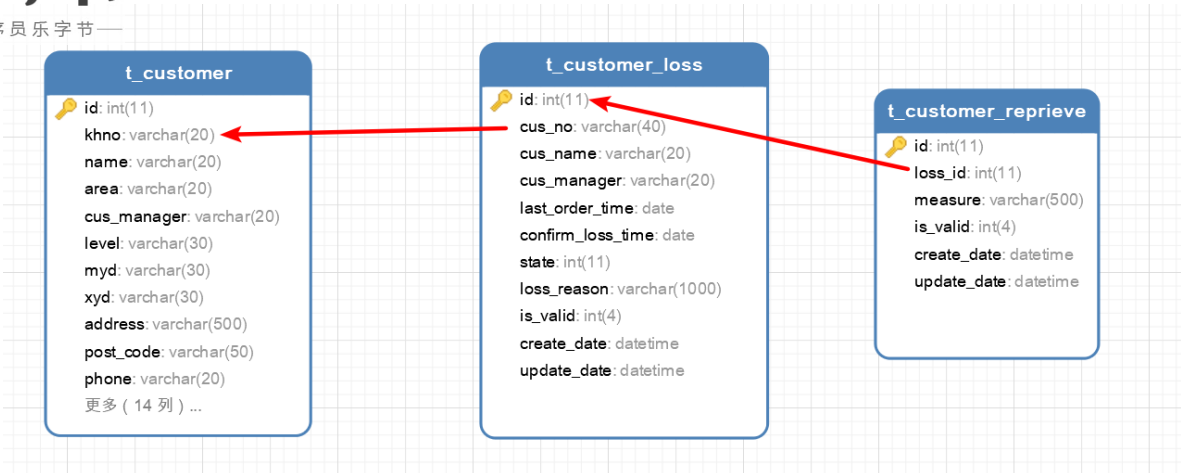
t_order_details	订单详情表			
	字段	字段类型	字段限制	字段描述
主键	id	int(11)	自增	id主键
	order_id	int(11)	可空	订单id
	goods_name	varchar(100)	可空	商品名称
	goods_num	int(11)	可空	商品数量
	unit	varchar(20)	可空	商品单位
	price	float	可空	单价
	sum	float	可空	总金额
	is_valid	int(4)	可空	有效状态
	create_date	datetime	可空	创建时间
	update_date	datetime	可空	更新时间

客户流失管理表结构

这里主要涉及表有

t_customer_loss 客户流失表

t_customer_reprieve 客户流失暂缓表



t_customer_loss	客户流失表			
	字段	字段类型	字段限制	字段描述
主键	id	int(11)	自增	id主键
	cus_no	varchar(40)	可空	客户编号
	cus_name	varchar(20)	可空	客户姓名
	cus_manager	varchar(20)	可空	客户经理
	last_order_time	date	可空	最后下单时间
	confirm_loss_time	date	可空	确认流失时间
	state	int(11)	可空	流失状态
	loss_reason	varchar(1000)	可空	流失原因
	is_valid	tinyint(4)	可空	有效状态
	create_date	datetime	可空	创建时间
	update_date	datetime	可空	更新时间

t_customer_reprieve	客户流失暂缓表			
	字段	字段类型	字段限制	字段描述
主键	id	int(11)	自增	id主键
	loss_id	int(11)	可空	流失id
	measure	varchar(500)	可空	措施
	is_valid	tinyint(4)	可空	有效状态
	create_date	datetime	可空	创建时间
	update_date	datetime	可空	更新时间

客户信息管理



客户信息管理查询

客户信息管理页面效果

客户名

客户编号

请选择...

搜索

+

添加

+

联系人管理

+

交往记录

+

订单查看

☰

🏠

📄

<input type="checkbox"/>	编号	客户名	法人	客户编号	地区	客户经理	满意度	客户级别	信用度	详细地址	邮	操作
<input type="checkbox"/>	1	北京大...	赵飞翔	KH2132...	北京	test	☆☆☆	战略合...	☆☆☆	北京海...	100	<div>编辑删除</div>
<input type="checkbox"/>	2	风驰科技	码云	KH2015...	北京	test	☆☆☆☆	大客户	☆☆☆☆	321	2	<div>编辑删除</div>
<input type="checkbox"/>	20	腾讯	赵飞翔	KH2017...	测试	test	☆☆☆...	大客户	☆☆☆☆			<div>编辑删除</div>
<input type="checkbox"/>	21	阿里巴巴	码云	KH2017...	北京	test01	☆☆☆...	战略合...	☆☆☆...	浙江杭州	324	<div>编辑删除</div>
<input type="checkbox"/>	22	中国工...	吴三强	KH2017...	上海	test	☆☆☆...		☆☆☆...	浦东	201	<div>编辑删除</div>
<input type="checkbox"/>	23	百度	李彦宏	KH2018...	北京	test	☆☆☆☆	战略合...	☆☆☆...	北京西...	100	<div>编辑删除</div>
<input type="checkbox"/>	24	小米科技	雷军	KH2018...	上海	test	☆☆☆...	重点开...	☆☆☆...	北京市...	100	<div>编辑删除</div>

客户信息查询后端代码实现

layui 框架通过表格展示后端表数据,数据格式见[官网](#)测试数据地址。

接口定义与SQL配置

- CustomerMapper.java

```
public interface CustomerMapper extends BaseMapper<Customer,Integer> {  
    /*  
        由于考虑到多个模块均涉及多条件查询  
        这里对于多条件分页查询方法由父接口BaseMapper定义  
    */  
}
```

- CustomerMapper.xml

```
<select id="selectByParams" parameterType="com.xxxx.crm.query.CustomerQuery"  
resultMap="BaseResultMap">  
    select  
    <include refid="Base_Column_List"/>  
    from t_customer  
    <where>  
        is_valid=1  
        <!--  
            state 流失状态  
            0 未流失  
            1 已流失  
        -->  
        and state =0  
        <if test="null !=cusName and cusName !=''">  
            and name like concat('%',{cusName},'%')  
        </if>  
        <if test="null !=cusNo and cusNo !=''">  
            and khno =#{cusNo}  
        </if>  
        <if test="null !=level and level !=''">  
            and level =#{level}  
        </if>  
</select>
```



```
<if test="null !=myd and myd !='">
    and myd =#{myd}
</if>

</where>
</select>
```

- CustomerQuery.java

在crm.query 包下创建CustomerQuery.java 查询类，设置查询条件

```
public class CustomerQuery extends BaseQuery {
    private String cusName;

    private String cusNo;

    private String level;

    private String myd;

    // 1 0-1000 2 1000-3000 3 3000-5000 4 50000
    private String type;

    private String time;
}
```

CustomerService 定义

- CustomerService .java

```
@Service
public class CustomerService extends BaseService<Customer, Integer> {
    /*
     * 由于考虑到多个模块均涉及多条件查询
     * 这里对于多条件分页查询方法由父类BaseService定义并实现
     */
}
```

- BaseService.java 分页查询方法定义与实现

```
public Map<String, Object> queryByParamsForTable(BaseQuery baseQuery) {
    Map<String, Object> result = new HashMap<String, Object>();
    PageHelper.startPage(baseQuery.getPage(), baseQuery.getLimit());
    PageInfo<T> pageInfo = new PageInfo<T>(selectByParams(baseQuery));
    result.put("count", pageInfo.getTotal());
    result.put("data", pageInfo.getList());
    result.put("code", 0);
    result.put("msg", "");
    return result;
}
```

CustomerController.java

```
@Controller
@RequestMapping("customer")
```



```
public class CustomerController extends BaseController {

    @Resource
    private CustomerService customerService;

    @Resource
    private CustomerOrderService orderService;

    @RequestMapping("index")
    public String index(){
        return "customer/customer";
    }

    @RequestMapping("list")
    @ResponseBody
    public Map<String,Object> queryCustomersByParams(CustomerQuery
customerQuery){
        return customerService.queryByParamsForTable(customerQuery);
    }
}
```

客户信息管理前端核心代码

客户信息管理主页面模板

resources/views/customer 目录创建 customer.ftl 模块文件,模板内容如下(模板依赖的 layui 文件由 common.ftl 文件提供), layui 表格数据展示模板文件实现参考该[地址](#)

- customer.ftl

```
<!DOCTYPE html>
<html>
<head>
    <title>客户管理</title>
    <#include "../common.ftl">
</head>
<body class="childrenBody">

<form class="layui-form" >
    <blockquote class="layui-elem-quote quoteBox">
        <form class="layui-form">
            <div class="layui-inline">
                <div class="layui-input-inline">
                    <input type="text" name="name"
                        class="layui-input
searchVal" placeholder="客户名" />
                </div>
                <div class="layui-input-inline">
                    <input type="text" name="khno" class="layui-input
searchVal" placeholder="客户编号" />
                </div>
                <div class="layui-input-inline">
                    <select name="level" id="level">
                        <option value="">请选择...</option>
                    </select>
                </div>
            </div>
        </form>
    </blockquote>
</form>
```




```
<option value="战略合作伙伴">战略合作伙伴</option>
<option value="大客户">大客户</option>
<option value="重点开发客户">重点开发客户</option>
</select>

</div>
<a class="layui-btn search_btn" data-type="reload"><i
    class="layui-icon">&#xe615;</i> 搜索</a>

</div>
</form>
</blockquote>
<table id="customerList" class="layui-table" lay-filter="customers"></table>

<!--操作-->
<script id="customerListBar" type="text/html">
    <a class="layui-btn layui-btn-xs" id="edit" lay-event="edit">编辑</a>
    <a class="layui-btn layui-btn-xs layui-btn-danger" lay-event="del">删除</a>
</script>

</form>
<script type="text/javascript" src="{ctx}/static/js/customer/customer.js">
</script>

</body>
</html>
```

客户信息管理主页面模板核心js

static/js/customer目录下创建customer.js 文件,初始化layui表格数据,layui表格数据展示模板文件实现参考该[地址](#)

- customer.js

```
layui.use(['table', 'layer', 'form'], function() {
    var layer = parent.layer === undefined ? layui.layer : top.layer,
        $ = layui.jquery,
        table = layui.table,
        form = layui.form;

    //客户列表展示
    var tableIns = table.render({
        elem: '#customerList',
        url: ctx + '/customer/list',
        cellMinwidth: 95,
        page: true,
        height: "full-125",
        limits: [10, 15, 20, 25],
        limit: 10,
        toolbar: "#toolbarDemo",
        id: "customerListTable",
        cols: [[
            {type: "checkbox", fixed: "center"},
            {field: "id", title: '编号', fixed: "true"},
            {field: 'name', title: '客户名', align: "center"},
            {field: 'fr', title: '法人', align: "center"},
        ]]
```



```
{field: 'khno', title: '客户编号', align: 'center'},
{field: 'area', title: '地区', align: 'center'},
{field: 'cusManager', title: '客户经理', align: 'center'},
{field: 'myd', title: '满意度', align: 'center'},
{field: 'level', title: '客户级别', align: 'center'},
{field: 'xyd', title: '信用度', align: 'center'},
{field: 'address', title: '详细地址', align: 'center'},
{field: 'postCode', title: '邮编', align: 'center'},
{field: 'phone', title: '电话', align: 'center'},
{field: 'webSite', title: '网站', align: 'center'},
{field: 'fax', title: '传真', align: 'center'},
{field: 'zczj', title: '注册资金', align: 'center'},
{field: 'yyzzzch', title: '营业执照', align: 'center'},
{field: 'khyh', title: '开户行', align: 'center'},
{field: 'khzh', title: '开户账号', align: 'center'},
{field: 'gsdjh', title: '国税', align: 'center'},
{field: 'dsdjh', title: '地税', align: 'center'},
{field: 'createDate', title: '创建时间', align: 'center'},
{field: 'updateDate', title: '更新时间', align: 'center'},
{title: '操作',
templet: '#customerListBar', fixed: "right", align: "center", minWidth: 150}
    ]
  });

});
```

客户信息多条件查询事件添加

客户信息数据表格数据展示成功后,接下来考虑添加多条件查询点击事件,这里使用layui表格reload重载基础方法实现, [点击这里参考官网介绍](#)。

```
<script type="text/html" id="toolbarDemo">
  <div class="layui-btn-container">
    <a class="layui-btn layui-btn-normal addNews_btn" lay-event="add">
      <i class="layui-icon">#xe608;</i>
      添加
    </a>
    <a class="layui-btn layui-btn-normal" lay-event="link">
      <i class="layui-icon">#xe608;</i>
      联系人管理
    </a>
    <a class="layui-btn layui-btn-normal addNews_btn" lay-
event="recode">
      <i class="layui-icon">#xe608;</i>
      交往记录
    </a>
    <a class="layui-btn layui-btn-normal addNews_btn" lay-event="order">
      <i class="layui-icon">#xe608;</i>
      订单查看
    </a>
  </div>
</script>
```

- customer.js 添加搜索点击事件



// 多条件搜索

```
$(".search_btn").on("click",function(){
    table.reload("customerListTable",{
        page: {
            curr: 1 //重新从第 1 页开始
        },
        where: {
            cusName: $("input[name='name']").val(), //客户名
            cusNo: $("input[name='khno']").val(), //客户编号
            level: $("#level").val() //客户等级
        }
    })
});
```

• 搜索效果

小米科技

KH2018050411200330

重点开发客户

搜索

添加

联系人管理

交往记录

订单查看

列表

新增

删除

<input type="checkbox"/>	编号	客户名	法人	客户编号	地区	客户经理	满意度	客户级别	信用度	详细地址	邮	操作
<input type="checkbox"/>	24	小米科技	雷军	KH2018...	上海	test	☆☆☆...	重点开...	☆☆☆...	北京市...	100	<div>编辑删除</div>

客户管理数据添加

后端核心代码

机会数据添加实现思路

```
/**
 * 1.参数校验
 *    客户名称 name 非空 不可重复
 *    phone 联系电话 非空 格式符合规范
 *    法人 非空
 * 2.默认值设置
 *    isValid state cteatDate updatdeDate
 *    khno 系统生成 唯一 (uuid| 时间戳 | 年月日时分秒 雪花算法)
 * 3.执行添加 判断结果
 */
```

机会数据添加核心代码

```
@Transactional(propagation = Propagation.REQUIRED)
public void saveCustomer(Customer customer){
    checkParams(customer.getName(),customer.getPhone(),customer.getFr());

    AssertUtil.isTrue(null!=customerMapper.queryCustomerByName(customer.getName()),
        "该客户已存在!");
    customer.setIsValid(1);
    customer.setState(0);
    customer.setCreateDate(new Date());
    customer.setUpdateDate(new Date());
    String khno = "KH_"+new SimpleDateFormat("yyyyMMddHHmmss").format(new
        Date());
```



```
customer.setKhno(khno);
AssertUtil.isTrue(insertSelective(customer)<1,"客户添加失败!");
}
private void checkParams(String name, String phone, String fr) {
    AssertUtil.isTrue(StringUtils.isBlank(name),"请指定客户名称!");
    AssertUtil.isTrue(!(PhoneUtil.isMobile(phone)),"手机号格式非法!");
    AssertUtil.isTrue(StringUtils.isBlank(fr),"请指定公司法人!");
}
```

后端页面转发&添加方法调用

对于机会数据添加与更新表单页可以实现共享,这里在转发机会数据添加与更新页面时共用一套代码即可(考虑更新时涉及到机会数据显示操作,这里根据机会id查询机会记录并放入到请求域中)。

```
@RequestMapping("addOrUpdateSaleChancePage")
public String addOrUpdateSaleChancePage(Integer id,Model model){
    model.addAttribute("customer",customerService.selectByPrimaryKey(id));
    return "customer/add_update";
}

@RequestMapping("save")
@ResponseBody
public ResultInfo saveCustomer(Customer customer){
    customerService.saveCustomer(customer);
    return success("客户添加成功");
}
```

前端核心代码

客户数据添加工具栏事件

这里对于对话框打开方法代码请参考官网[页面](#)

```
//头工具栏事件
table.on('toolbar(customers)', function(obj){
    var checkStatus = table.checkStatus(obj.config.id);
    switch(obj.event){
        case "add":
            openAddOrUpdateCustomerDialog();
            break;
    };
});

// 打开添加客户数据页面
function openAddOrUpdateCustomerDialog(id){
    var url = ctx+"/customer/addOrUpdateSaleChancePage";
    var title="客户管理-添加";
    if(id){
        url = url+"?id="+id;
        title="客户管理-更新";
    }
    layui.layer.open({
        title : title,
        type : 2,
        area:["700px","500px"],

```



```
maxmin:true,  
content : url
```

```
});
```

```
}
```

add_update.ftl 表单模板文件

views/customer目录下添加add_update.ftl 表单模板

```
<!DOCTYPE html>  
<html>  
<head>  
    <#include "../common.ftl">  
</head>  
<body class="childrenBody">  
<form class="layui-form" style="width:80%;">  
    <input name="id" type="hidden" value="${(customer.id)!}"/>  
  
    <div class="layui-form-item layui-row">  
        <div class="layui-col-xs6">  
            <label class="layui-form-label">客户名称</label>  
            <div class="layui-input-block">  
                <input type="text" class="layui-input"  
                    name="name" id="name" lay-verify="required"  
value="${(customer.name)!}" placeholder="请输入客户名">  
            </div>  
        </div>  
        <div class="layui-col-xs6">  
            <label class="layui-form-label">法人</label>  
            <div class="layui-input-block">  
                <input type="text" class="layui-input"  
                    name="fr" id="fr" lay-verify="required"  
value="${(customer.fr)!}" placeholder="请输入法人">  
            </div>  
        </div>  
    </div>  
  
    <div class="layui-form-item layui-row">  
        <div class="layui-col-xs6">  
            <label class="layui-form-label">区域</label>  
            <div class="layui-input-block">  
                <input type="text" class="layui-input"  
                    name="area" lay-verify="required"  
value="${(customer.area)!}" placeholder="请输入区域">  
            </div>  
        </div>  
        <div class="layui-col-xs6">  
            <label class="layui-form-label">客户经理</label>  
            <div class="layui-input-block">  
                <input type="text" class="layui-input"  
                    name="cusManager" value="${(customer.cusManager)!}"  
placeholder="请输入客户经理">  
            </div>  
        </div>  
    </div>
```



```
<div class="layui-form-item layui-row">
  <div class="layui-col-xs6">
    <label class="layui-form-label">客户级别</label>
    <div class="layui-input-block">
      <select name="level" id="level">
        <option value="" >请选择</option>
        <option value="普通客户">普通客户</option>
        <option value="重点开发客户" >重点开发客户</option>
        <option value="大客户" >大客户</option>
        <option value="合作伙伴" >合作伙伴</option>
      </select>
    </div>
  </div>
  <div class="layui-col-xs6">
    <label class="layui-form-label">信用度</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="xyd" value="{{(customer.xyd)!}}" placeholder="请输入
客户信用级别">
    </div>
  </div>
</div>

<div class="layui-form-item layui-row">
  <div class="layui-col-xs6">
    <label class="layui-form-label">邮编</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="postCode" value="{{(customer.postCode)!}}"
placeholder="请输入客户邮编">
    </div>
  </div>
  <div class="layui-col-xs6">
    <label class="layui-form-label">联系电话</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="phone" value="{{(customer.phone)!}}"
placeholder="请输入客户联系电话">
    </div>
  </div>
</div>

<div class="layui-form-item layui-row">
  <div class="layui-col-xs6">
    <label class="layui-form-label">客户地址</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="address" value="{{(customer.address)!}}"
placeholder="请输入客户地址">
    </div>
  </div>
  <div class="layui-col-xs6">
    <label class="layui-form-label">传真</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="fax" value="{{(customer.fax)!}}" placeholder="请输入
客户传真">
    </div>
  </div>
</div>
```



```
</div>
</div>
</div>

<div class="layui-form-item layui-row">
  <div class="layui-col-xs6">
    <label class="layui-form-label">网站</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="webSite" value="${(customer.webSite)!}"
placeholder="请输入客户网站地址">
    </div>
  </div>
  <div class="layui-col-xs6">
    <label class="layui-form-label">注册资金</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="zczj" value="${(customer.zczj)!}" placeholder="请输
入注册资金">
    </div>
  </div>
</div>

<div class="layui-form-item layui-row">
  <div class="layui-col-xs6">
    <label class="layui-form-label">开户行</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="khyh" value="${(customer.khyh)!}" placeholder="请输
入客户开户行">
    </div>
  </div>
  <div class="layui-col-xs6">
    <label class="layui-form-label">开户账号</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="khzh" value="${(customer.khzh)!}" placeholder="请输
入开户账号">
    </div>
  </div>
</div>

<div class="layui-form-item layui-row">
  <div class="layui-col-xs6">
    <label class="layui-form-label">国税</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
        name="gsdjh" value="${(customer.gsdjh)!}"
placeholder="请输入国税">
    </div>
  </div>
  <div class="layui-col-xs6">
    <label class="layui-form-label">地税</label>
    <div class="layui-input-block">
      <input type="text" class="layui-input"
```



```
name="dsdjh" value="${(customer.dsdjh)!}"
placeholder="请输入地税">
    </div>
</div>
</div>

<div class="layui-form-item layui-row">

    <div class="layui-col-xs6">
        <label class="layui-form-label">年营业额</label>
        <div class="layui-input-block">
            <input type="text" class="layui-input"
                name="nyye" value="${(customer.nyye)!}" placeholder="请输入客户年营业额">
        </div>
    </div>
</div>

<br/>
<div class="layui-form-item layui-row layui-col-xs12">
    <div class="layui-input-block">
        <button class="layui-btn layui-btn-lg" lay-submit=""
            lay-filter="addOrUpdateCustomer">确认
        </button>
        <button class="layui-btn layui-btn-lg layui-btn-normal">取消</button>
    </div>
</div>
</form>
<script type="text/javascript" src="${ctx}/js/customer/add.update.js"></script>
</body>
</html>
```

add.update.js 文件添加

js/customer目录下添加 add.update.js 文件,完成机会数据添加与更新表单提交操作

```
layui.use(['form', 'layer'], function () {
    var form = layui.form,
        layer = parent.layer === undefined ? layui.layer : top.layer,
        $ = layui.jquery;

    form.on("submit(addOrUpdateCustomer)", function (data) {
        var index = top.layer.msg('数据提交中, 请稍候', {icon: 16, time: false,
            shade: 0.8});
        //弹出loading
        var url=ctx + "/customer/save";
        if($("#input[name='id']").val()){
            url=ctx + "/customer/update";
        }
        $.post(url, data.field, function (res) {
            if (res.code == 200) {
                setTimeout(function () {
                    top.layer.close(index);
                    top.layer.msg("操作成功!");
                    layer.closeAll("iframe");
                    //刷新父页面
                }, 1000);
            }
        });
    });
});
```




```
        parent.location.reload();
    }, 500);
} else {
    layer.msg(
        res.msg, {
            icon: 5
        }
    );
}
});
return false;
});
});
```

客户信息数据更新

后端核心代码

实现思路

```
/**
 * 1. 参数校验
 *    记录存在校验
 *    客户名称 name 非空 不可重复
 *    phone 联系电话 非空 格式符合规范
 *    法人 非空
 * 2. 默认值设置
 *    updateDate
 * 3. 执行更新 判断结果
 */
```

核心代码

```
@Transactional(propagation = Propagation.REQUIRED)
public void updateCustomer(Customer customer){
    AssertUtil.isTrue(null==customer.getId() || null==selectByPrimaryKey(customer.getId()), "待更新记录不存在!");
    checkParams(customer.getName(), customer.getPhone(), customer.getFr());
    Customer temp = customerMapper.queryCustomerByName(customer.getName());
    AssertUtil.isTrue(null != temp && !
        (temp.getId().equals(customer.getId())),"该客户已存在!");
    customer.setUpdateDate(new Date());
    AssertUtil.isTrue(updateByPrimaryKeySelective(customer)<1, "客户更新失败!");
}
```

页面转发 & 更新方法调用

这里客户信息数据更新页面视图与添加操作实现视图代码共用。



```
@RequestMapping("update")
@ResponseBody
public ResultInfo updateCustomer(Customer customer){
    customerService.updateCustomer(customer);
    return success("客户更新成功");
}
```

前端核心代码

客户信息数据表格行监听事件添加

- sale.chance.js

表格添加行监听事件,行监听事件参考[这里](#)

```
/**
 * 行监听
 */
table.on("tool(customers)", function(obj){
    var layEvent = obj.event;
    if(layEvent === "edit") {
        openAddOrUpdateCustomerDialog(obj.data.id);
    }
});

// 打开添加客户数据页面
function openAddOrUpdateCustomerDialog(id){
    var url = ctx+"/customer/addOrUpdateSaleChancePage";
    var title="客户管理-添加";
    if(id){
        url = url+"?id="+id;
        title="客户管理-更新";
    }
    layui.layer.open({
        title : title,
        type : 2,
        area:["700px","500px"],
        maxmin:true,
        content : url
    });
}
```

客户信息数据删除

后端核心代码



客户信息后端删除核心代码

- CustomerService.java

```
@Transactional(propagation = Propagation.REQUIRED)
public void deleteCustomer(Integer cid){
    Customer customer= selectByPrimaryKey(cid);
    AssertUtil.isTrue(null==cid||null==cid,"待删除记录不存在!");
    /**
     * 如果客户被删除
     *      级联 客户联系人 客户交往记录 客户订单 被删除
     *
     * 如果客户被删除
     *      如果子表存在记录 不支持删除
     */
    customer.setIsValid(0);
    AssertUtil.isTrue(updateByPrimaryKeySelective(customer)<1,"客户删除失败!");
}
```

- CustomerController.java

```
@RequestMapping("delete")
@ResponseBody
public ResultInfo deleteCustomer(Integer id){
    customersService.deleteCustomer(id);
    return success("客户删除成功");
}
```

前端核心代码

行监听添加删除事件

```
/**
 * 行监听
 */
table.on("tool(customers)", function(obj){
    var layEvent = obj.event;
    if(layEvent === "edit") {
        openAddOrUpdateCustomerDialog(obj.data.id);
    }else if(layEvent === "del") {
```



```
function (index) {
    $.post(ctx+"/customer/delete",{id:obj.data.id},function (data) {
        if(data.code==200){
            layer.msg("操作成功! ");
            tableIns.reload();
        }else{
            layer.msg(data.msg, {icon: 5});
        }
    });
}
});
```

客户订单查看

客户订单数据查询

页面效果



前端事件添加

当点击订单查看时,此时添加客户信息表头工具栏事件

```
//头工具栏事件
table.on('toolbar(customers)', function(obj){
    var checkStatus = table.checkStatus(obj.config.id);
    switch(obj.event){
        case "add":
            openAddOrUpdateCustomerDialog();
            break;
        case "order":
            openOrderInfoDialog(checkStatus.data);
            break;
    };
});
```



// 打开订单查看页面

```
function openOrderInfoDialog(datas) {
    if(datas.length==0){
        layer.msg("请选择待查看订单对应客户!", {icon: 5});
        return;
    }
    if(datas.length>1){
        layer.msg("暂不支持批量查看!", {icon: 5});
        return;
    }
    var url = ctx+"/customer/orderInfoPage?cid="+datas[0].id;
    layui.layer.open({
        title : "客户管理-订单信息展示",
        type : 2,
        area:["700px","500px"],
        maxmin:true,
        content : url
    });
}
```

- 客户信息管理控制器添加页面转发方法

```
@RequestMapping("orderInfoPage")
public String showOrderInfo(Integer cid, Model model){
    model.addAttribute("customer",customerService.selectByPrimaryKey(cid));
    return "customer/customer_order";
}
```

后端核心代码

- CustomerOrderMapper.xml

```
<select id="selectByParams"
parameterType="com.xxxx.crm.query.CustomerOrderQuery"
resultType="com.xxxx.crm.vo.CustomerOrder">
    select
    <include refid="Base_Column_List"/>
    from t_customer_order
    <where>
        is_valid=1 and cus_id=#{cid}
        <if test="null != orderNo and orderNo !=''">
            and order_no=#{orderNo}
        </if>
        <if test="null !=state">
            and state=#{state}
        </if>
    </where>
</select>
```

- CustomerOrderController.java



```
@RequestMapping("list")
@ResponseBody
public Map<String,Object> queryCustomerOrdersByParams(CustomerOrderQuery
customerOrderQuery){
    return customerOrderService.queryByParamsForTable(customerOrderQuery);
}
```

前端核心代码

- customer_order.ftl 模板文件添加
views/customer 目录下创建customer_order.ftl 文件

```
<!DOCTYPE html>
<html>
<head>
    <title>客户订单查看</title>
    <#include "../common.ftl">
</head>
<body class="childrenBody">
    <div class="layui-col-md12">
        <div class="layui-card">
            <div class="layui-card-body">
                <form class="layui-form">
                    <input name="id" type="hidden" value="${(customer.id)!}" />
                    <div class="layui-form-item layui-row">
                        <div class="layui-col-xs6">
                            <label class="layui-form-label">客户名称</label>
                            <div class="layui-input-block">
                                <input type="text" class="layui-input"
                                    name="name" id="name"
                                    value="${(customer.name)!}" readonly="readonly">
                            </div>
                        </div>
                        <div class="layui-col-xs6">
                            <label class="layui-form-label">法人</label>
                            <div class="layui-input-block">
                                <input type="text" class="layui-input"
                                    name="fr" id="fr"
                                    value="${(customer.fr)!}" readonly="readonly">
                            </div>
                        </div>
                    </div>
                    <div class="layui-form-item layui-row">
                        <div class="layui-col-xs6">
                            <label class="layui-form-label">客户地址</label>
                            <div class="layui-input-block">
                                <input type="text" class="layui-input"
                                    name="address"
                                    value="${(customer.address)!}" readonly="readonly">
                            </div>
                        </div>
                        <div class="layui-col-xs6">
                            <label class="layui-form-label">联系电话</label>

```



```
<div class="layui-input-block">
  <input type="text" class="layui-input"
    name="phone"
    value="${(customer.customer)!}" id="phone" readonly="readonly">
</div>
</div>
</div>
</form>
</div>
</div>
</div>

<div class="layui-col-md12">
  <table id="customerOrderList" class="layui-table" lay-
filter="customerOrders"></table>
</div>
<!--操作-->
<script id="customerOrderListBar" type="text/html">
  <a class="layui-btn layui-btn-xs" lay-event="info">订单详情</a>
</script>
<script type="text/javascript" src="${ctx}/js/customer/customer.order.js">
</script>
</body>
</html>
```

- customer.order.js 核心js

```
layui.use(['table','layer'],function(){
  var layer = parent.layer === undefined ? layui.layer : top.layer,
      $ = layui.jquery,
      table = layui.table;

  //订单列表展示
  var tableIns = table.render({
    elem: '#customerOrderList',
    url: ctx+'/order/list?cid='+$("#input[name='id']").val(),
    cellMinwidth: 95,
    page: true,
    height: "full-125",
    limits: [10,15,20,25],
    limit: 10,
    toolbar: "#toolbarDemo",
    id: "customerOrderListTable",
    cols: [[
      {type: "checkbox", fixed:"center"},
      {field: "id", title:'编号',fixed:"true"},
      {field: 'orderNo', title: '订单编号',align:"center"},
      {field: 'orderDate', title: '下单日期',align:"center"},
      {field: 'address', title: '收货地址',align:"center"},
      {field: 'state', title: '支付状态',align:"center",templet:function
(d) {
          if(d.state==1){
            return "已支付;"
          }else{
            return "未支付";
          }
        }
      },
      {field: 'createDate', title: '创建时间',align:"center"},
    ]
  });
```



```
{field: 'updateDate', title: '更新时间',align:"center"},
{title: '操作',fixed:"right",align:"center",
minwidth:150,templet:"#customerOrderListBar"}
    ]
});
});
```

客户订单详情查看

页面效果



前端事件添加

客户订单页添加行监听事件, customer.order.js

```
/**
 * 行监听
 */
table.on("tool(customerOrders)", function(obj){
    var layEvent = obj.event;
    if(layEvent == "info") {
        openOrderDetailDialog(obj.data.id);
    }
});
// 打开添加计划项数据页面
function openOrderDetailDialog(id){
    var url = ctx+"/customer/orderDetailPage?orderId="+id;
    layui.layer.open({
        title : "订单详情查看",
        type : 2,
        area:["700px","400px"],
        maxmin:true,
        content : url
    });
}
```

后端核心代码

- CustomerController.java

```
@RequestMapping("orderDetailPage")
public String orderDetailPage(Integer orderId, Model model){
    model.addAttribute("order",orderService.queryOrderDetailById(orderId));
    return "customer/customer_order_detail";
}
```

- OrderDetailsController.java

```
@Controller
```




```
@RequestMapping("order_details")
public class OrderDetailsController extends BaseController {

    @Resource
    private OrderDetailsService orderDetailsService;

    @RequestMapping("list")
    @ResponseBody
    public Map<String, Object> queryOrderDetailsByParams(OrderDetailsQuery
orderDetailsQuery) {
        return orderDetailsService.queryByParamsForTable(orderDetailsQuery);
    }
}
```

前端核心代码

views/customer 目录下新建 customer_order_detail.ftl 文件

```
<!DOCTYPE html>
<html>
<head>
    <title>订单详情查看</title>
    <#include "../common.ftl">
</head>
<body class="childrenBody">
<div class="layui-col-md12">
    <div class="layui-card">
        <div class="layui-card-body">
            <form class="layui-form">
                <input name="id" type="hidden" value="${(order.id)!}"/>
                <div class="layui-form-item layui-row">
                    <div class="layui-col-xs6">
                        <label class="layui-form-label">订单编号</label>
                        <div class="layui-input-block">
                            <input type="text" class="layui-input"
                                name="orderNo" id="orderNo"
                                value="${(order.orderNo)!}" readonly="readonly">
                        </div>
                    </div>
                    <div class="layui-col-xs6">
                        <label class="layui-form-label">总金额(¥)</label>
                        <div class="layui-input-block">
                            <input type="text" class="layui-input"
                                name="total" id="total"
                                value="${(order.total)!}" readonly="readonly">
                        </div>
                    </div>
                </div>
                <div class="layui-form-item layui-row">
                    <div class="layui-col-xs6">
                        <label class="layui-form-label">物流地址</label>
                        <div class="layui-input-block">
                            <input type="text" class="layui-input"
                                value="" id="address">
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>
</div>
```



js/customer 目录下新建customer.order.details.js

11



对于客户联系人管理，客户交往记录管理两个模块功能实现，模仿前面模块实现。

客户流失管理

客户流失管理页面效果

客户编号

客户名

请选择

搜索

编号

客户编号

客户名称

客户经理

最后下单...

流失原因

确认流失...

创建时间

更新时间

操作

383

KH21321...

北京大牛...

admin

公司转型

2020-02-1...

2020-01-1...

2020-01-1...

详情

384

KH20150...

风驰科技

test

2020-01-1...

2020-01-1...

添加暂缓

客户流失管理后端实现

客户流失规则定义

客户自创建超过六个月以来未与企业产生任何订单或者客户最后下单日期距离现在超过六个月的客户定义为流失客户

流失客户数据查询分析与实现

这里对于客户流失规则的定义包含两种情况

- 创建的客户数据距离当前时间超过了6个月
- 客户没有下单 或者最后一次下单时间距离当前时间超过6个月

在实现流失客户数据查询时采用逆向查询方式 先将有效客户查出来然后按照排除法获取无效客户

```
SELECT
    *
FROM
    t_customer c
WHERE
    c.is_valid = 1
    AND c.state = 0
    AND DATE_ADD(c.create_date,INTERVAL 6 MONTH) < NOW()
    AND c.id NOT IN (
        SELECT
            o.cus_id
        FROM
            t_customer_order o
        WHERE
            o.state = 1
            AND o.is_valid = 1
            AND DATE_ADD(o.order_date,INTERVAL 6 MONTH) > NOW()
    )
```



客户流失业务逻辑实现

```
@Transactional(propagation = Propagation.REQUIRED)
public void updateCustomerState(){
    List<Customer> lossCustomers = customerMapper.queryLossCustomers();
    if(null !=lossCustomers && lossCustomers.size()>0){
        List<CustomerLoss> customerLosses= new ArrayList<CustomerLoss>();
        List<Integer> lossCusIds=new ArrayList<Integer>();
        lossCustomers.forEach(customer->{
            CustomerLoss customerLoss=new CustomerLoss();
            // 设置最后下单时间
            CustomerOrder lastCustomerOrder =
customerOrderMapper.queryLastCustomerOrderByCusId(customer.getId());
            if(null !=lastCustomerOrder){
                customerLoss.setLastOrderTime(lastCustomerOrder.getOrderDate());
            }
            customerLoss.setCreateDate(new Date());
            customerLoss.setCusManager(customer.getCusManager());
            customerLoss.setCusName(customer.getName());
            customerLoss.setCusNo(customer.getKhno());
            customerLoss.setIsValid(1);
            // 设置客户流失状态为暂缓流失状态
            customerLoss.setState(0);
            customerLoss.setUpdateDate(new Date());
            customerLosses.add(customerLoss);
            lossCusIds.add(customer.getId());
        });
        AssertUtil.isTrue(customerLossMapper.insertBatch(customerLosses)
<customerLosses.size(),"客户流失数据流转失败!");
        AssertUtil.isTrue(customerMapper.updateCustomerStateByIds(lossCusIds)
<lossCusIds.size(),"客户流失数据流转失败!");
    }
}
```

流失客户定时任务处理添加

当实现了客户数据转移业务逻辑代码后，这里需要思考一个问题:客户数据量的问题随着时间的积累，流失的客户数据可能就比较大了，如果数据的获取在用户查询时进行，此时后端对于数据的查询就会变得很慢，此时可以使用我们之前讲到的定时任务来处理，后台通过定时器来对流失客户数据定时进行转移处理，从而当前端用户查询时只需到客户流失表查询流失数据即可。



```
public class TaskService {

    @Resource
    private CustomerService customerService;

    @Scheduled(cron = "0/2 * * * * ?")
    public void job(){
        System.out.println("定时任务开始执行-->" + new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss").format(new Date()));
        customerService.updateCustomerState();
    }
}
```

Starter开启定时任务环境配置

```
@SpringBootApplication
@MapperScan("com.xxxx.crm.dao")
@EnableScheduling
public class Starter {

    public static void main(String[] args) {
        SpringApplication.run(Starter.class);
    }
}
```

客户流失管理页面展示

- CustomerLossController.java

```
@Controller
@RequestMapping("customer_loss")
public class CustomerLossController extends BaseController {

    @Resource
    private CustomerLossService customerLossService;

    @RequestMapping("index")
    private String index(){
        return "customerLoss/customer_loss";
    }

    @RequestMapping("list")
    @ResponseBody
    public Map<String, Object> queryCustomerLossByParams(CustomerLossQuery
customerLossQuery){
        return customerLossService.queryByParamsForTable(customerLossQuery);
    }
}
```

- customer_loss.ftl 模板文件



```
<!DOCTYPE html>
<html>
<head>
    <title>流失客户管理</title>
    <#include "../common.ftl">
</head>
<body class="childrenBody">

<form class="layui-form" >
    <blockquote class="layui-elem-quote quoteBox">
        <form class="layui-form">
            <div class="layui-inline">
                <div class="layui-input-inline">
                    <input type="text" name="cusNo"
                        class="layui-input
searchVal" placeholder="客户编号" />
                </div>
                <div class="layui-input-inline">
                    <input type="text" name="cusName" class="layui-input
searchVal" placeholder="客户名" />
                </div>
                <div class="layui-input-inline">
                    <select name="state" id="state" >
                        <option value="" >请选择</option>
                        <option value="0">暂缓流失</option>
                        <option value="1" >确认流失</option>
                    </select>
                </div>
                <a class="layui-btn search_btn" data-type="reload"><i
class="layui-icon">&#xe615;</i> 搜索</a>
            </div>
        </form>
    </blockquote>
    <table id="customerLossList" class="layui-table" lay-
filter="customerLosses"></table>

    <script type="text/html" id="toolbarDemo">
</script>

    <!--操作-->
    <script id="op" type="text/html" >
        {{# if (d.state=== 0 ) { }}
            <a href="javascript:;" class="layui-btn layui-btn-warm layui-btn-xs"
lay-event="add">添加暂缓</a>
        {{# } else { }}
            <a href="javascript:;" class="layui-btn layui-btn-normal layui-btn-xs"
lay-event="info">详情</a>
        {{# } }}
    </script>

</form>
<script type="text/javascript" src="{ctx}/js/customerLoss/customer_loss.js">
</script>
```



```
</body>
</html>
```

- customer.loss.js 文件添加

js/customerLoss 目录下添加customer.loss.js 文件

```
layui.use(['table', 'layer', "form"], function() {
    var layer = parent.layer === undefined ? layui.layer : top.layer,
        $ = layui.jquery,
        table = layui.table;

    //用户列表展示
    var tableIns = table.render({
        elem: '#customerLossList',
        url: ctx + '/customer_loss/list',
        cellMinwidth: 95,
        page: true,
        height: "full-125",
        limits: [10, 15, 20, 25],
        limit: 10,
        toolbar: "#toolbarDemo",
        id: "customerLossListTable",
        cols: [[
            {type: "checkbox", fixed: "center"},
            {field: "id", title: '编号', fixed: "true"},
            {field: 'cusNo', title: '客户编号', align: "center"},
            {field: 'cusName', title: '客户名称', align: "center"},
            {field: 'cusManager', title: '客户经理', align: "center"},
            {field: 'lastOrderTime', title: '最后下单时间', align: "center"},
            {field: 'lossReason', title: '流失原因', align: "center"},
            {field: 'confirmLossTime', title: '确认流失时间', align: "center"},
            {field: 'createDate', title: '创建时间', align: "center"},
            {field: 'updateDate', title: '更新时间', align: "center"},
            {title: '操作', fixed: "right", align: "center"},
        ]],
        minWidth: 150, templet: "#op"
    });

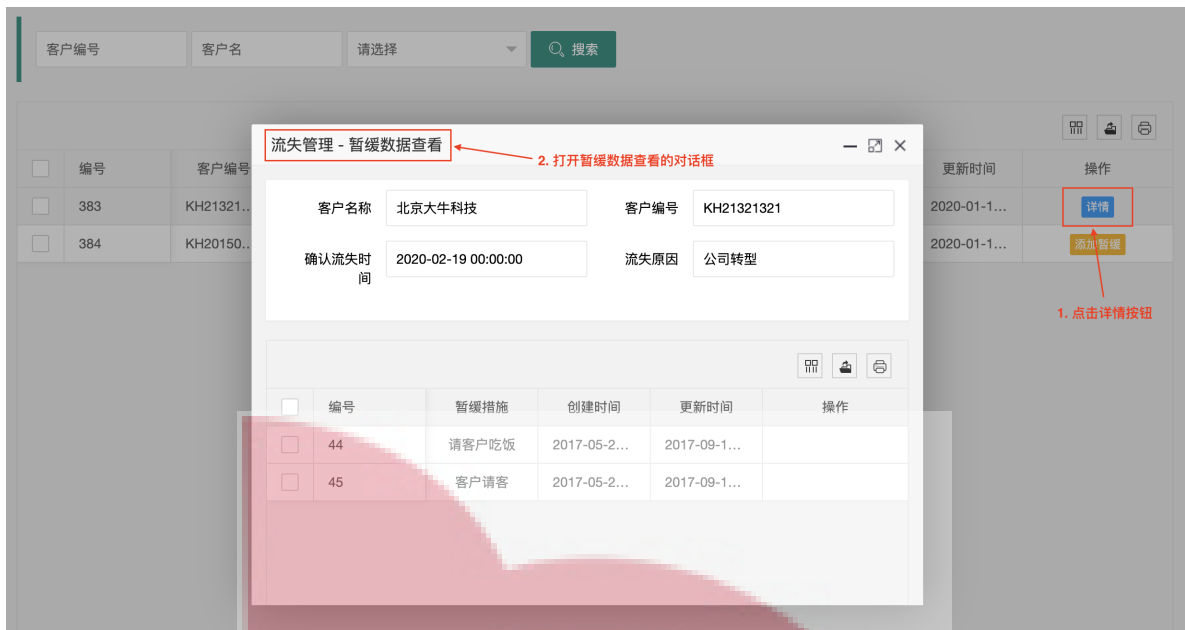
    $(".search_btn").on("click", function() {
        table.reload("customerLossListTable", {
            page: {
                curr: 1 //重新从第 1 页开始
            },
            where: {
                cusNo: $("input[name='cusNo']").val(), //用户名
                cusName: $("input[name='cusName']").val(), //邮箱
                state: $("#state").val() //手机号
            }
        });
    });
});
```

流失客户暂缓处理



这里流失客户数据展示成功后,并不代表客户已确认流失,需要作下一步流失暂缓处理,所以这里在展示客户数据时有显示详情展示的,有显示添加暂缓的处理,对流失客户做进一步确认操作实现过程可以参看计划数据开发操作这里不再提供具体实现流程,页面效果如下:

已确认流失客户



暂缓客户处理

