```
Java客户端 Jedis
    * Jedis: 一款java操作redis数据库的工具(类似于JDBC(可以使用java代码来操作mysql数据库),Jedis可以通过java代码来操作java代码)
    * 使用步骤:
        1. 下载jedis的jar包
        2. 使用
            //1. 获取连接
            Jedis jedis = new Jedis("localhost",6379);
            //2. 操作
            jedis.set("username","zhangsan");
            //3. 关闭连接
            jedis.close();


Jedis操作各种redis中的数据结构
        1) 字符串类型 string
                set
                get
             //1. 获取连接
            Jedis jedis = new Jedis();//如果使用空参构造，默认值"localhost",6379端口
            //2. 操作
            //存储
            jedis.set("username","zhangsan");
            //获取
            String username = jedis.get("username");
            System.out.println(username);
            //可以使用setex()方法存储可以指定过期时间的 key value
            jedis.setex("activecode",20,"hehe");//将activecode：hehe键值对存入redis，并且20秒后自动删除该键值对
            //3. 关闭连接
            jedis.close();


        2) 哈希类型 hash ： map格式
              hset
              hget
              hgetAll
            //1. 获取连接
            Jedis jedis = new Jedis();//如果使用空参构造，默认值"localhost",6379端口
            //2. 操作
```

```
            // 存储hash
            jedis.hset("user","name","lisi");
            jedis.hset("user","age","23");
            jedis.hset("user","gender","female");
            // 获取hash
            String name = jedis.hget("user", "name");
            System.out.println(name);//lisi
            // 获取hash的所有map中的数据
            Map<String, String> user = jedis.hgetAll("user");
            // keyset
            Set<String> keySet = user.keySet();
                for (String key : keySet) {
                    //获取value
                    String value = user.get(key);
                    System.out.println(key + ":" +
value);//gender:female    ;   name:lisi   ;      age:23
                }
             //3.关闭连接
             jedis.close();
```

3) 列表类型 list ： linkedlist格式。支持重复元素

```
            lpush / rpush
            lpop / rpop
            lrange start end : 范围获取
             //1. 获取连接
            Jedis jedis = new Jedis();//如果使用空参构造，默认值
"localhost",6379端口
            //2. 操作
            // list 存储
            jedis.lpush("mylist","a","b","c");//从左边存,先存a再存b再存
c
            jedis.rpush("mylist","a","b","c");//从右边存
            // list 范围获取
            List<String> mylist = jedis.lrange("mylist", 0, -1);
            System.out.println(mylist);//输出[c, b, a, a, b, c]
            // list 弹出
            String element1 = jedis.lpop("mylist");//c
            System.out.println(element1);
            String element2 = jedis.rpop("mylist");//c
            System.out.println(element2);
            // list 范围获取
```

```
            List<String> mylist2 = jedis.lrange("mylist", 0, -1);
            System.out.println(mylist2);//[b, a, a, b]
            //3. 关闭连接
            jedis.close();
```

4) 集合类型 set ： 不允许重复元素

sadd

smembers:获取所有元素

```
            //1. 获取连接
            Jedis jedis = new Jedis();//如果使用空参构造，默认值
"localhost",6379端口
            //2. 操作
            // set 存储
             jedis.sadd("myset","java","php","c++");
             // set 获取
             Set<String> myset = jedis.smembers("myset");
             System.out.println(myset);//输出[c++, java, php]
             //3. 关闭连接
             jedis.close();
```

5) 有序集合类型 sortedset：不允许重复元素，且元素有顺序

zadd

zrange

```
           //1. 获取连接
           Jedis jedis = new Jedis();//如果使用空参构造，默认值
"localhost",6379端口
           //2. 操作
           // sortedset 存储
           jedis.zadd("mysortedset",3,"亚瑟");
           jedis.zadd("mysortedset",30,"后裔");
           jedis.zadd("mysortedset",55,"孙悟空");
           // sortedset 获取
           Set<String> mysortedset = jedis.zrange("mysortedset", 0,
-1);
           System.out.println(mysortedset);//输出[亚瑟, 后裔, 孙悟空]
           //3. 关闭连接
           jedis.close();
```

6)Jedis连接池： JedisPool(与JDBD里面的连接池概念是一样的,但不同的是JDBC使用的是第三方提供的,但 JedisPool使用的Jedis的jar包自带的,即Jedis自带就有连接池)(将来获取Jedis客户端连接的时候直接从连接池里获取就行了,这样更便于管理与复用)

* 使用：

1. 创建JedisPool连接池对象
2. 调用方法 getResource()方法获取Jedis连接

```
//0.创建一个配置对象,可以设置一些参数
JedisPoolConfig config = new
JedisPoolConfig();//最大允许的连接数
config.setMaxTotal(50);//最大允许的连接数
config.setMaxIdle(10);//最大空闲数，数据库连接的最大空闲时间。超过空闲时间，数据库连接将被标记为不可用，然后被释放。
//1.创建Jedis连接池对象
JedisPool jedisPool = new
JedisPool(config,"localhost",6379);
//2.获取连接
Jedis jedis = jedisPool.getResource();
//3. 使用
jedis.set("hehe","heihei");
//4. 关闭 归还到连接池中
jedis.close();
```

7)连接池工具类(即把配置对象的参数给抽取出来,方便修改)

```
public class JedisPoolUtils {
    private static JedisPool jedisPool;
                    static{
        //读取配置文件
        InputStream is =
JedisPoolUtils.class.getClassLoader().getResourceAsStream("jedis.pr
operties");
        //创建Properties对象
        Properties pro = new Properties();
        //关联文件
        try {
            pro.load(is);
        } catch (IOException e) {
            e.printStackTrace();
        }
        //获取数据，设置到JedisPoolConfig中
        JedisPoolConfig config = new
JedisPoolConfig();

config.setMaxTotal(Integer.parseInt(pro.getProperty("maxTotal")));

config.setMaxIdle(Integer.parseInt(pro.getProperty("maxIdle")));
        //初始化JedisPool
```

```
                        jedisPool = new
JedisPool(config,pro.getProperty("host"),Integer.parseInt(pro.getPr
operty("port")));
                    }
                    /**
                     * 获取连接方法
                     */
                    public static Jedis getJedis(){
                        return jedisPool.getResource();
                    }
                }
                    //z再创建一个类，通过连接池工具类获取
                    Jedis jedis = JedisPoolUtils.getJedis();
                     jedis.set("hello","world");
                  jedis.close();;
```

案例需求（除了再IDEA里写代码，还得在本地导入一个数据库）：

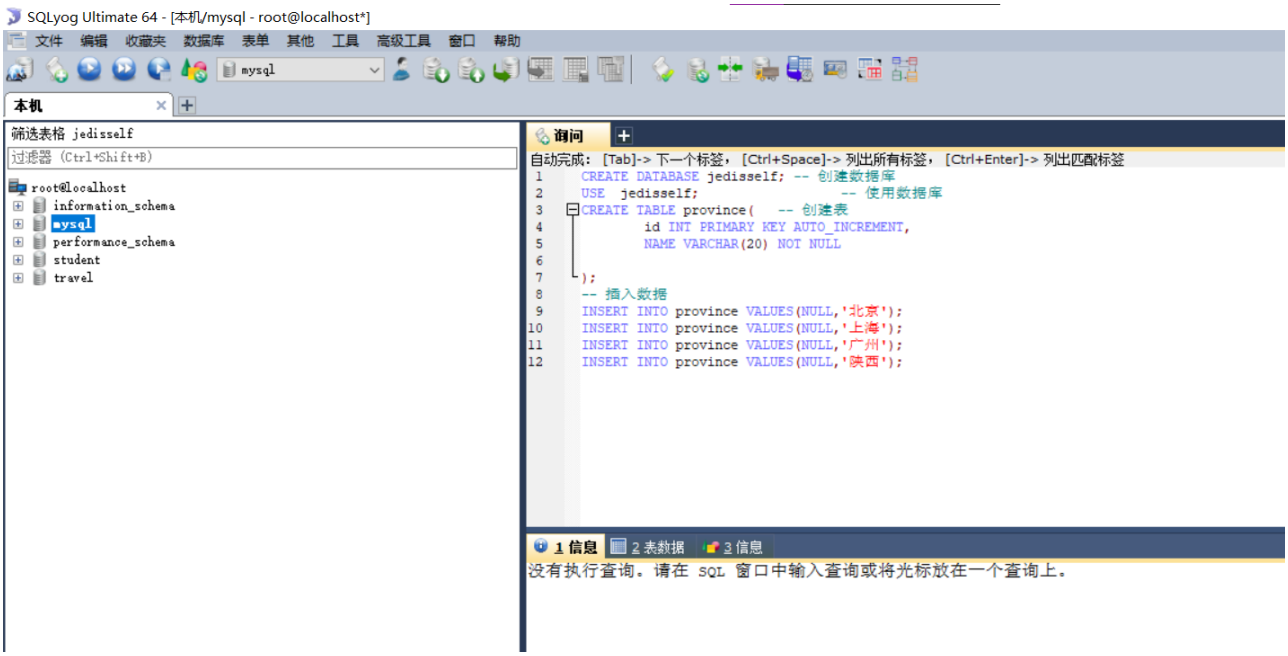
1. 提供index.html页面，页面中有一个省份 下拉列s表
2. 当 页面加载完成后 发送ajax请求，加载所有省份

* 注意：使用redis缓存一些不经常发生变化的数据。 * 数据库的数据一旦发生改变，则需要更新缓存。 * 数据库的表执行 增删改的相关操作，需要将redis缓存数据情况，再次存入 * 在service对应的增删改方法中，将redis数据删除。

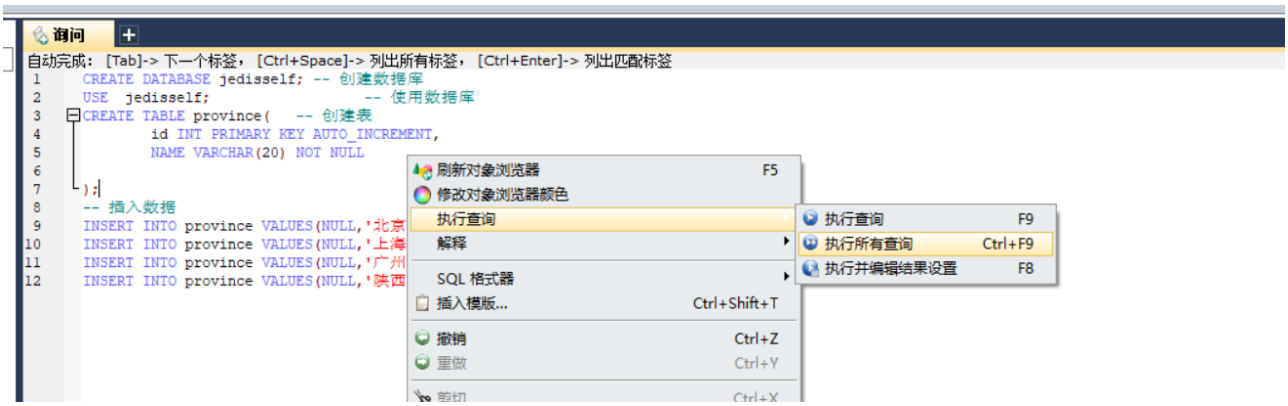
```
CREATE DATABASE jedisself; -- 创建数据库
USE  jedisself;          -- 使用数据库
CREATE TABLE province(   -- 创建表
    id INT PRIMARY KEY AUTO_INCREMENT,
    NAME VARCHAR(20) NOT NULL

);
-- 插入数据
INSERT INTO province VALUES(NULL,'北京');
INSERT INTO province VALUES(NULL,'上海');
INSERT INTO province VALUES(NULL,'广州');
INSERT INTO province VALUES(NULL,'陕西');
```

SQLyog导入数据库的另一种方法

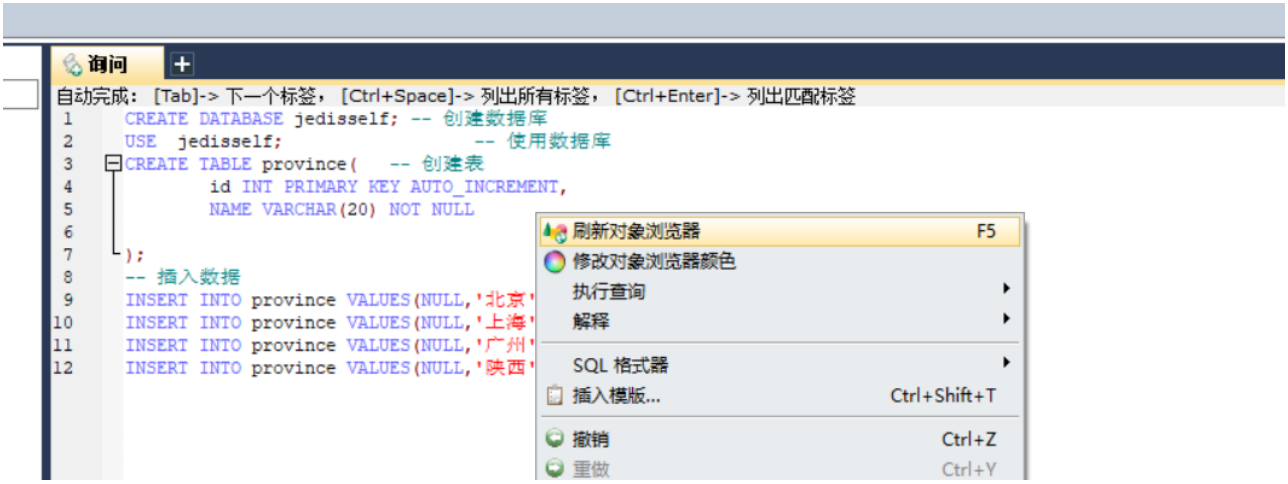
- 用记事本把视频给的sql给打开，把代码全部复制粘贴进去　SQLyog里面



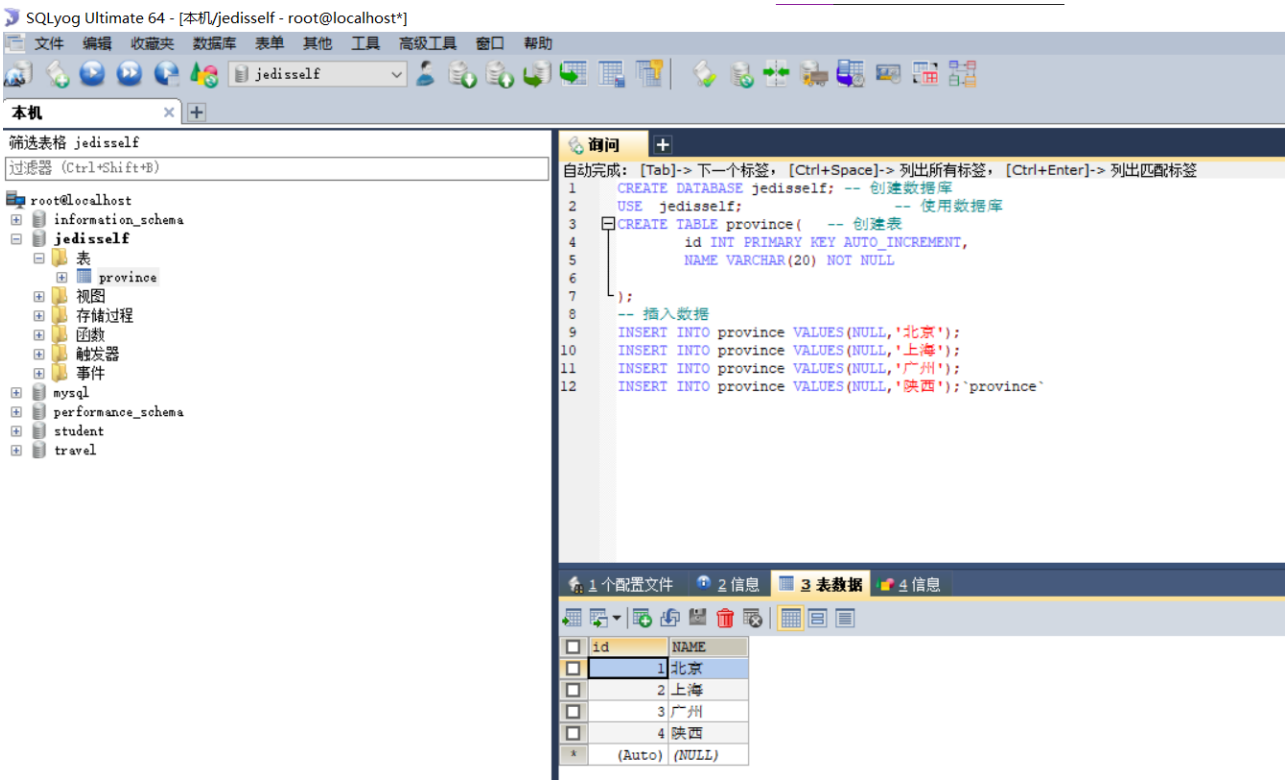
- 在代码里右键点击所有查询

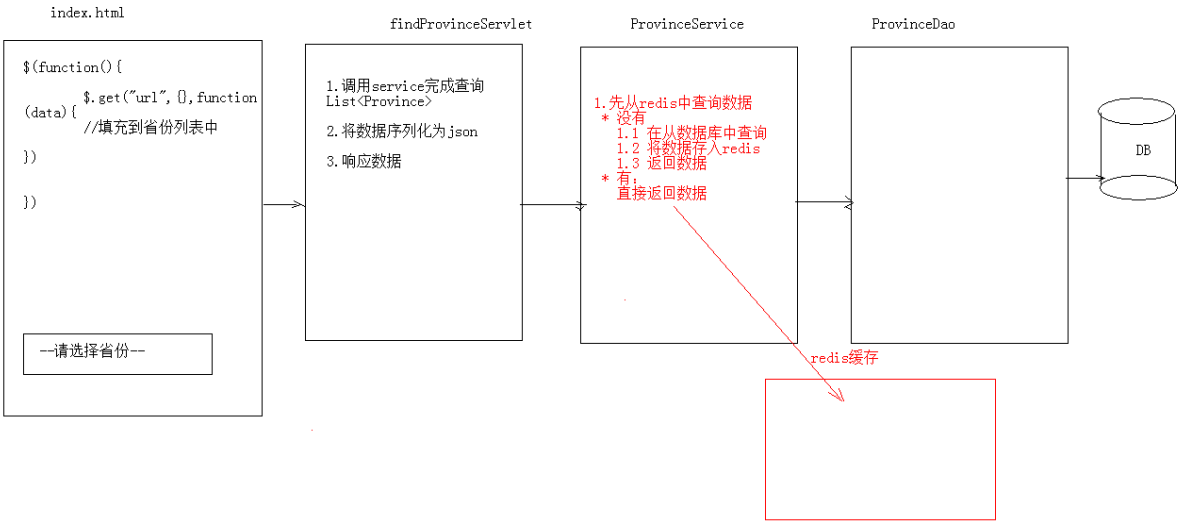


- 在代码里右键点击刷新对象浏览器


询问
自动完成： [Tab]-> 下一个标签， [Ctrl+Space]-> 列出所有标签， [Ctrl+Enter]-> 列出匹配标签
CREATE DATABASE jedisself; -- 创建数据库
USE jedisself; -- 使用数据库
CREATE TABLE province(-- 创建表
id INT PRIMARY KEY AUTO_INCREMENT,
NAME VARCHAR(20) NOT NULL
);
-- 插入数据
INSERT INTO province VALUES(NULL,'北京'
INSERT INTO province VALUES(NULL,'上海'
INSERT INTO province VALUES(NULL,'广州'
INSERT INTO province VALUES(NULL,'陕西'
刷新对象浏览器 F5
修改对象浏览器颜色
执行查询
解释
SQL 格式器
插入模版... Ctrl+Shift+T
撤销 Ctrl+Z
重做 Ctrl+Y

- 成功


SQLyog Ultimate 64 - [本机/jedisself - root@localhost*]
文件 编辑 收藏夹 数据库 表单 其他 工具 高级工具 窗口 帮助
jedisself
本机
筛选表格 jedisself
过滤器 (Ctrl+Shift+B)
root@localhost
information_schema
jedisself
表
province
视图
存储过程
函数
触发器
事件
mysql
performance_schema
student
travel
询问
自动完成： [Tab]-> 下一个标签， [Ctrl+Space]-> 列出所有标签， [Ctrl+Enter]-> 列出匹配标签
CREATE DATABASE jedisself; -- 创建数据库
USE jedisself; -- 使用数据库
CREATE TABLE province(-- 创建表
id INT PRIMARY KEY AUTO_INCREMENT,
NAME VARCHAR(20) NOT NULL
);
-- 插入数据
INSERT INTO province VALUES(NULL,'北京');
INSERT INTO province VALUES(NULL,'上海');
INSERT INTO province VALUES(NULL,'广州');
INSERT INTO province VALUES(NULL,'陕西');`province`
1 个配置文件
2 信息
3 表数据
4 信息
id NAME
1 北京
2 上海
3 广州
4 陕西
(Auto) (NULL)


index.html
$(function(){
$.get("url",{},function(data){
//填充到省份列表中
})
})
--请选择省份--
findProvinceServlet
1.调用service完成查询List<Province>
2.将数据序列化为json
3.响应数据
ProvinceService
1.先从redis中查询数据
* 没有
1.1 在从数据库中查询
1.2 将数据存入redis
1.3 返回数据
* 有:
直接返回数据
ProvinceDao
DB
redis缓存