

实验 3 学生信息查询系统实验程序

一、学生信息查询系统需求描述

实验目标：要求利用本章所学知识完成一个学生信息查询系统，该系统要求实现2 个以下功能

1) 多条件查询

当用户输入的学生姓名不为空，则只根据学生姓名进行学生信息的查询；

当用户输入的学生姓名为空，而学生专业不为空，则只根据学生专业进行学生的查询；

当用户输入的学生姓名和专业都为空，则要求查询出所有学号不为空的学生信息。

2) 单条件查询，查询出所有id 值小于5 的学生的信息。

二、开发步骤

第 1 步：数据库开发

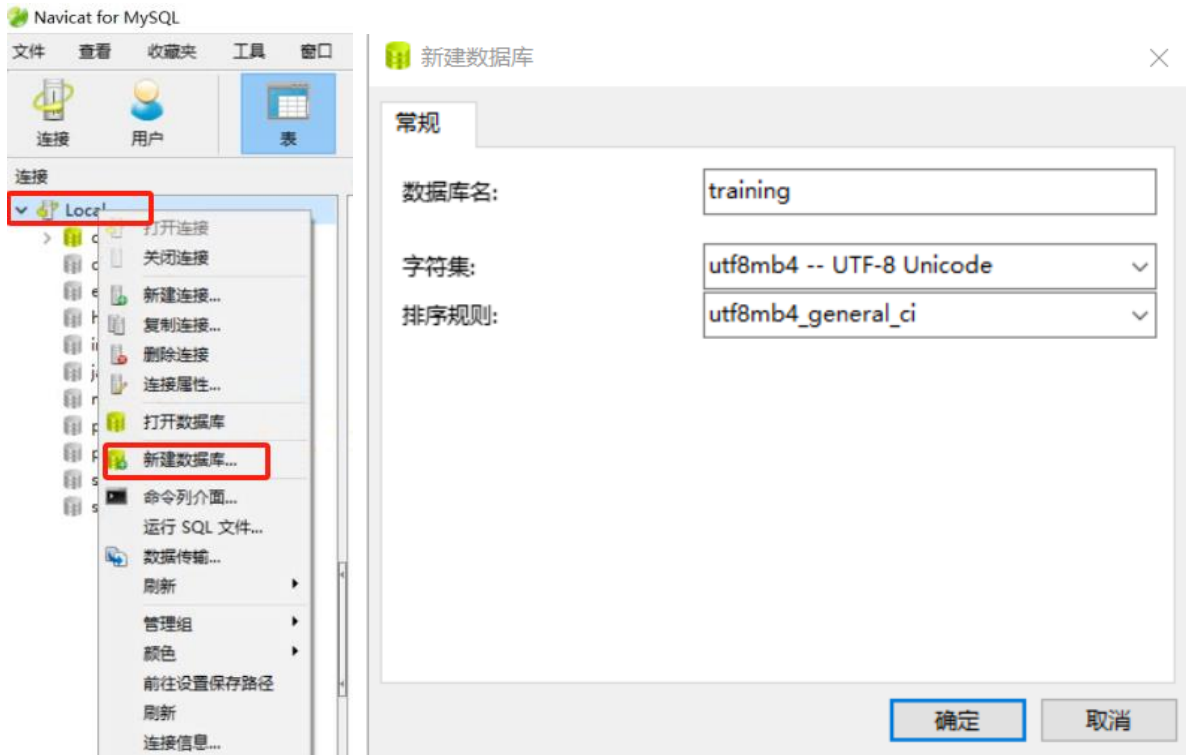
先使用 MySQL 数据库创建一个 **training** 数据库，然后创建一个学生表 **t_student**，方法如下：

通过使用 Navicat 工具，**连接上你安装的 MySQL 数据库**，输入用户名和密码进行**登录**。

假设建立的 MySQL 连接名为 **Local**，数据库登录用户名和密码为：**root/123456**。

登录后，手动创建数据库，需要创建的数据库名为：**training**，如图：选中 **Local** 连接名，

右键选择【**新建数据库**】



数据库创建好后，创建表 **t_student**，表结构如下。并按要求加入几条测试数据。

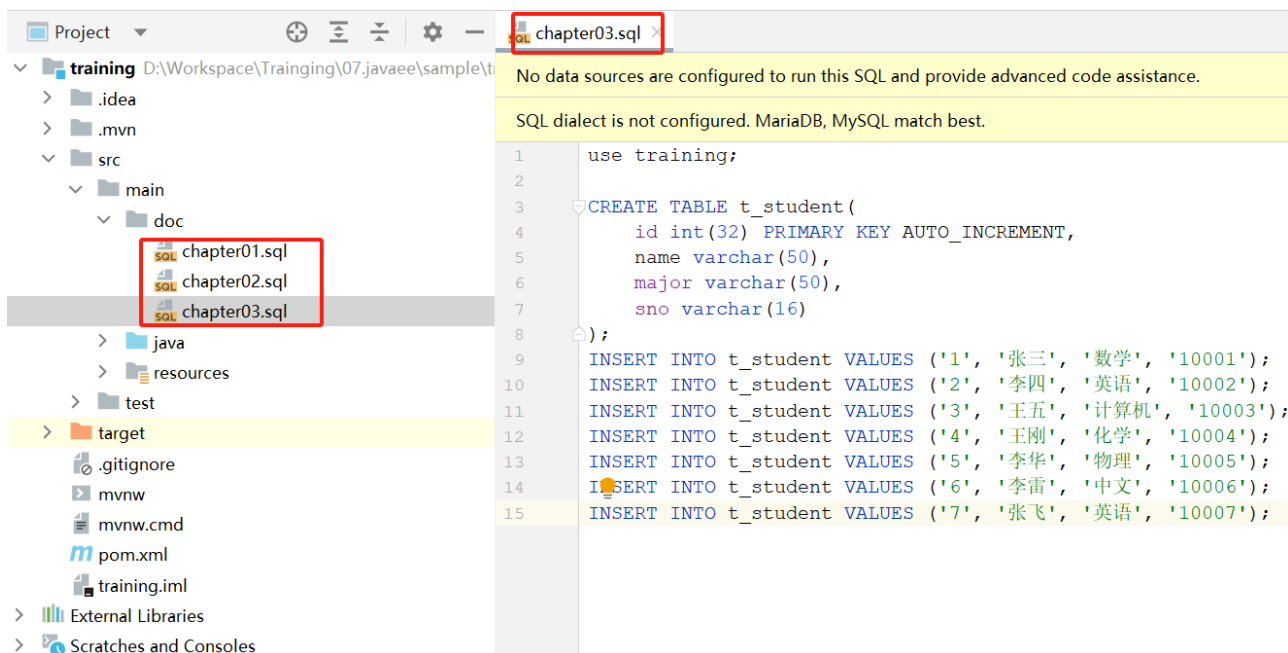
```
use training;

CREATE TABLE t_student(
    id int(32) PRIMARY KEY AUTO_INCREMENT,
    name varchar(50),
    major varchar(50),
    sno varchar(16)
);

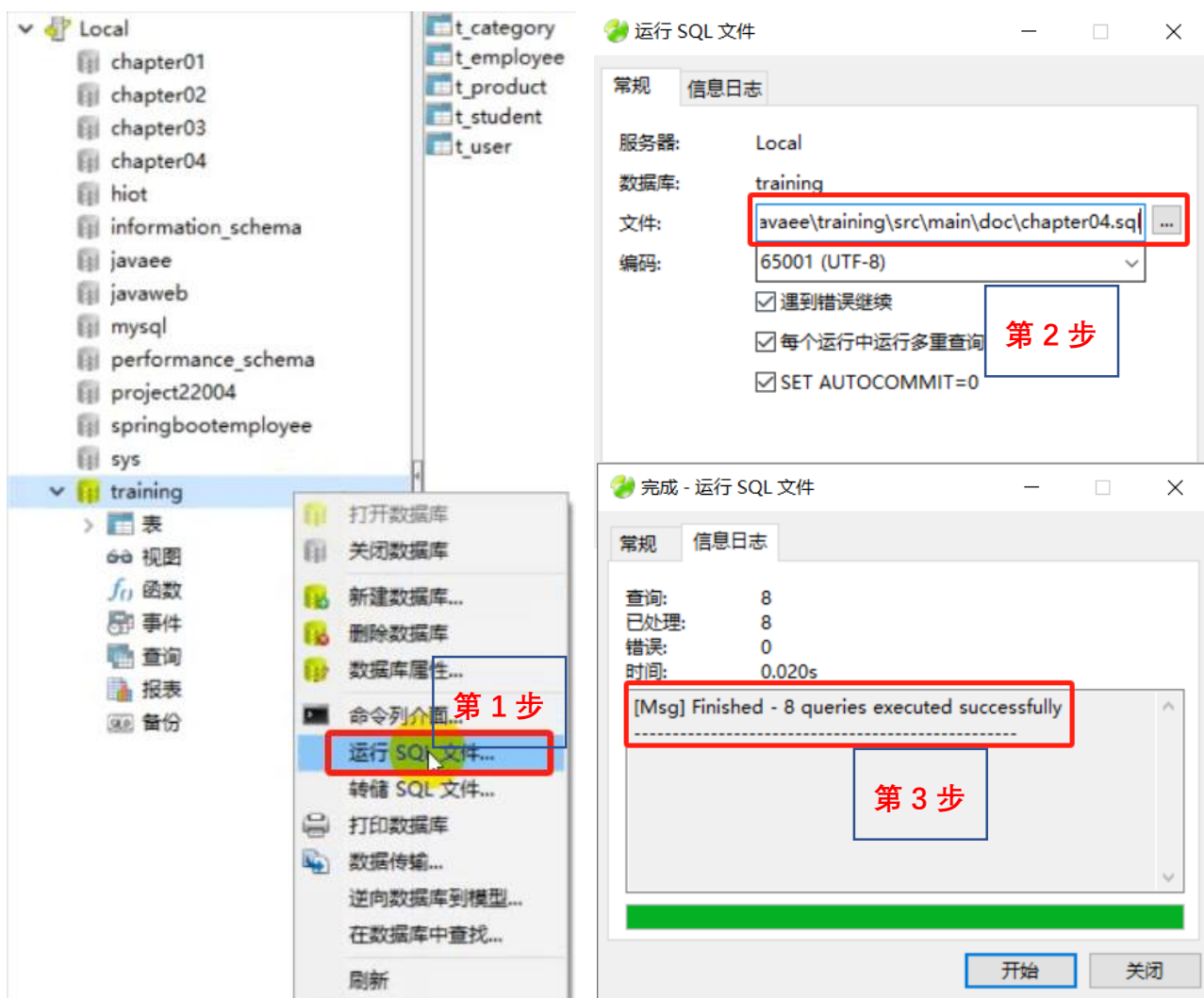
INSERT INTO t_student VALUES ('1', '张三', '数学', '10001');
INSERT INTO t_student VALUES ('2', '李四', '英语', '10002');
INSERT INTO t_student VALUES ('3', '王五', '计算机', '10003');
INSERT INTO t_student VALUES ('4', '王刚', '化学', '10004');
INSERT INTO t_student VALUES ('5', '李华', '物理', '10005');
INSERT INTO t_student VALUES ('6', '李雷', '中文', '10006');
INSERT INTO t_student VALUES ('7', '张飞', '英语', '10007');
```

至此，数据库部分开发完毕。

注：以上建表数据库脚本已放置工程的 **doc** 目录，文件路径如下图所示：



可以通过 Navicat 工具直接读取此脚本文件，以完成数据表的创建。

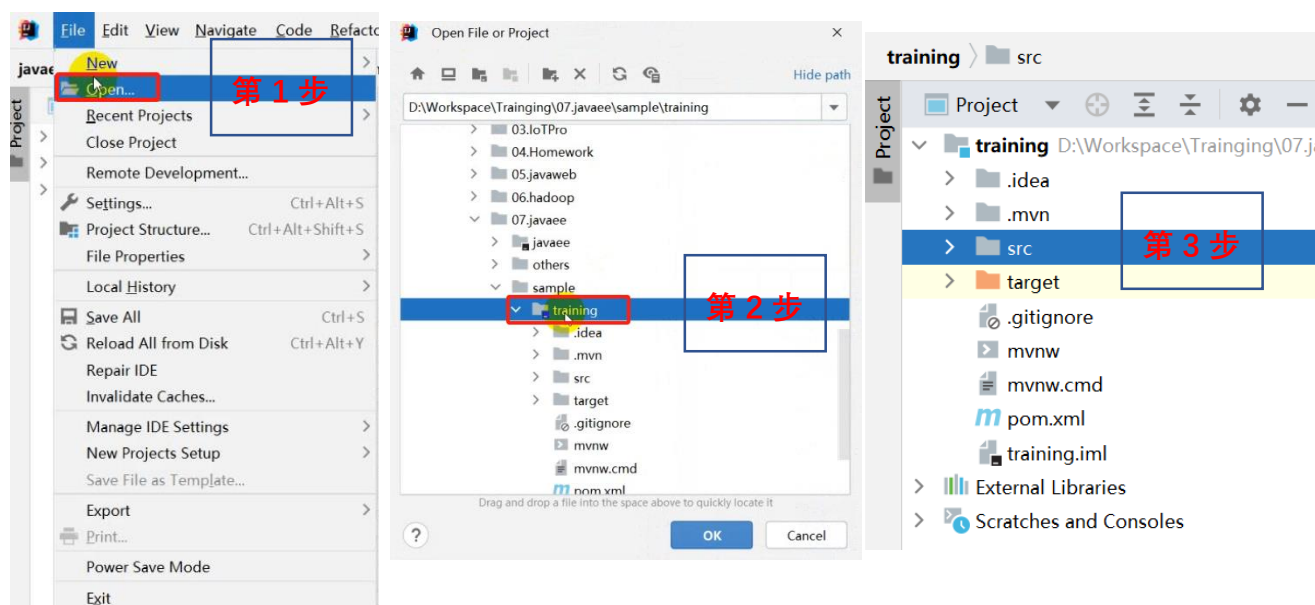


第 2 步：导入 training 工程，并配置本地 Maven 路径

Step1，下载并解压 training 工程。从【学习通】→【章节】→【第三章…】→【上机实验 3…】，找到 training.zip 压缩包，下载到本地，并解压成 training 目录。

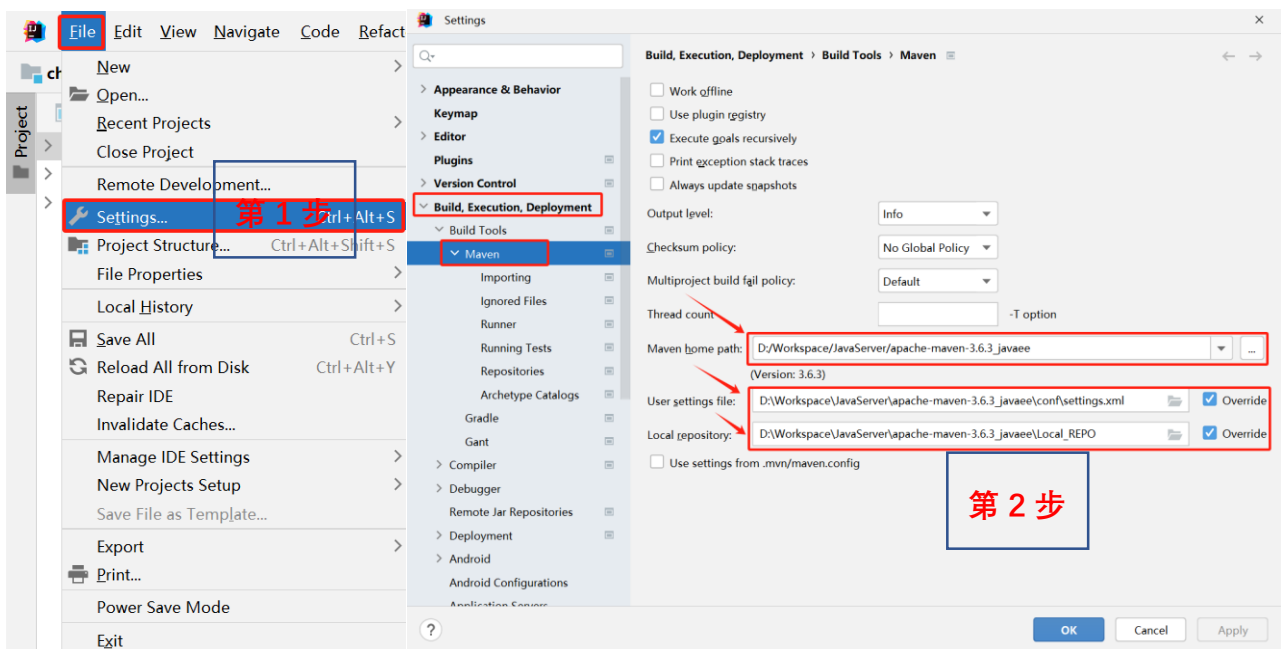
Step2，导入 training 工程。打开 IDEA 工具，点击【File】→【Open…】→选中上一步解压的 training 工程，如下图，确定即可导入。

注意，导入工程后，对于 IDEA 是 2023 的版本，要点击左上角边的文件夹小图标，即可查看到工程中的文件。



Step3，新工程需要重新设置 Maven 路径

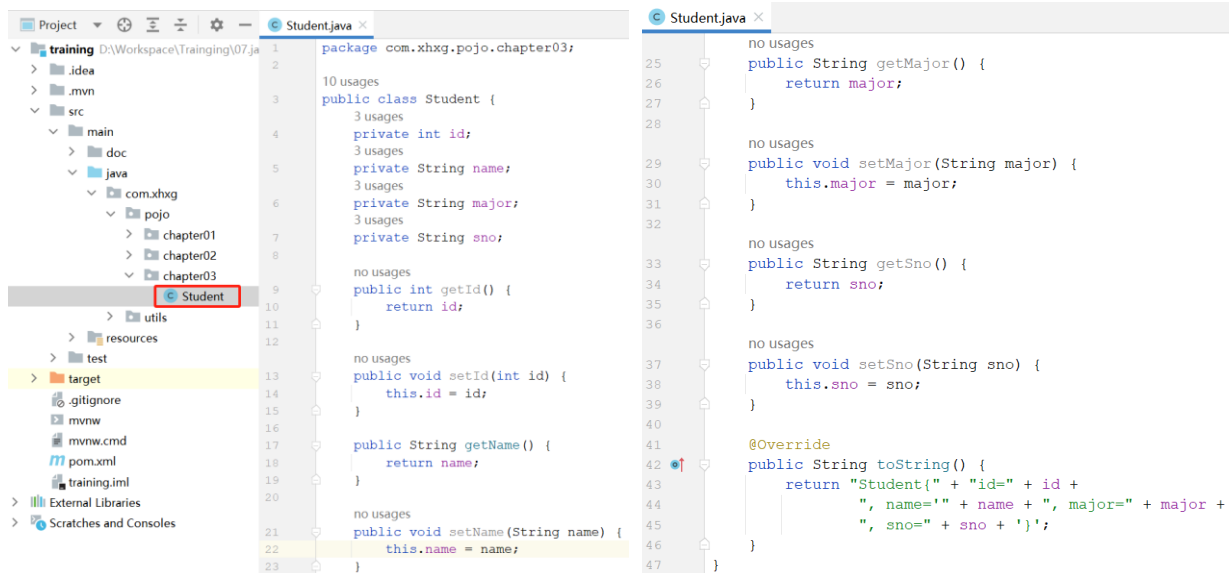
选择【File】→【Setting…】然后找到菜单单位【Build, Execution, Deployment】→【Build Tools】→【Maven】，按如下图所示，将你的本地 Maven 路径和配置文件设置好即可。



第 3 步：程序代码开发

1、创建 POJO 实体

打开 training 项目工程。创建代码包为：com.xhxx.pojo.chapter03，并且创建 Student 类



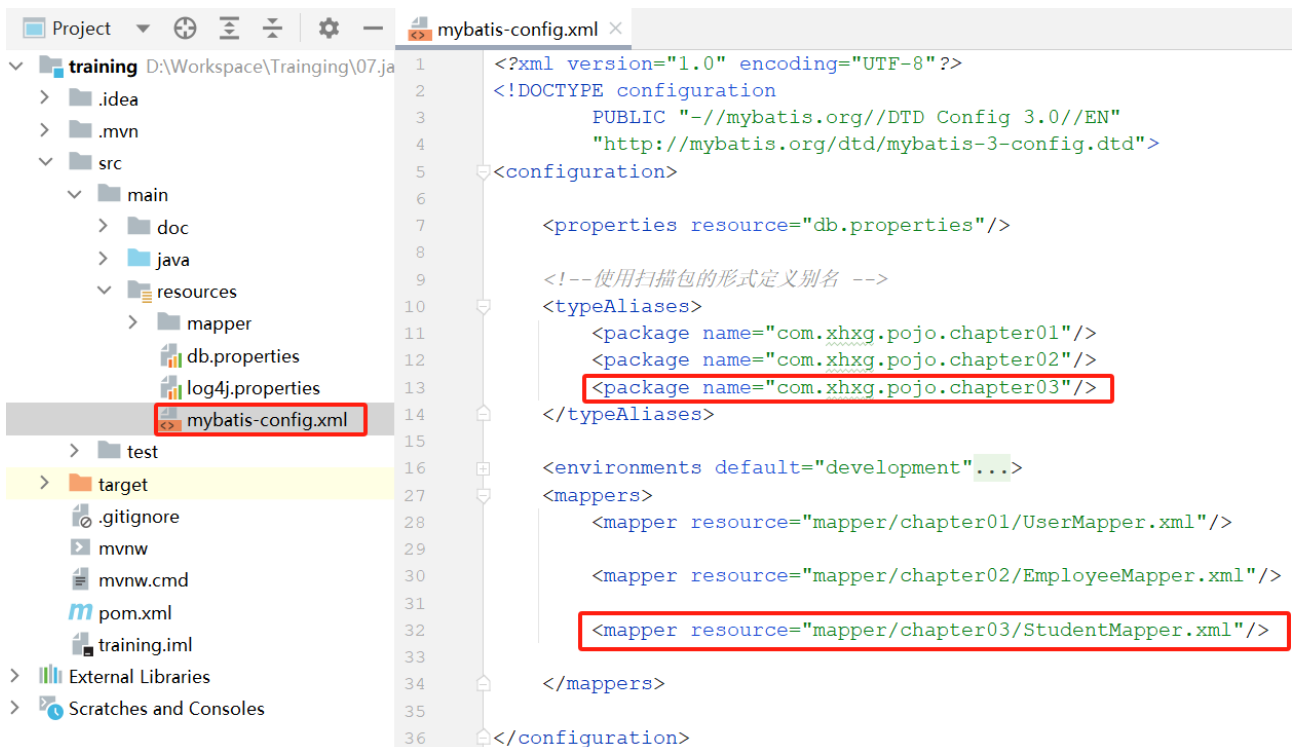
2、创建映射文件 StudentMapper.xml

在 src/main/resources 目录下创建目录 mapper/chapter03，然后创建一个 StudentMapper.xml 文件，内容如下。



3、创建 mybatis-config.xml 配置文件

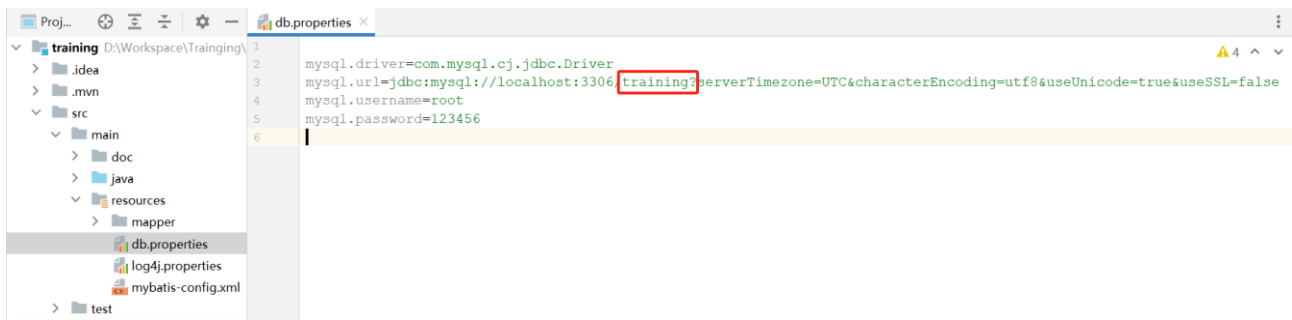
在 src/main/resources 目录下创建一个 mybatis-config.xml 文件，内容如下：



里面的配置项，除了数据库的连接信息外，还需要将前面的 CategoryMapper.xml 和 ProductMapper.xml 映射文件引入，另外还需要创建另一个配置文件 db.properties。

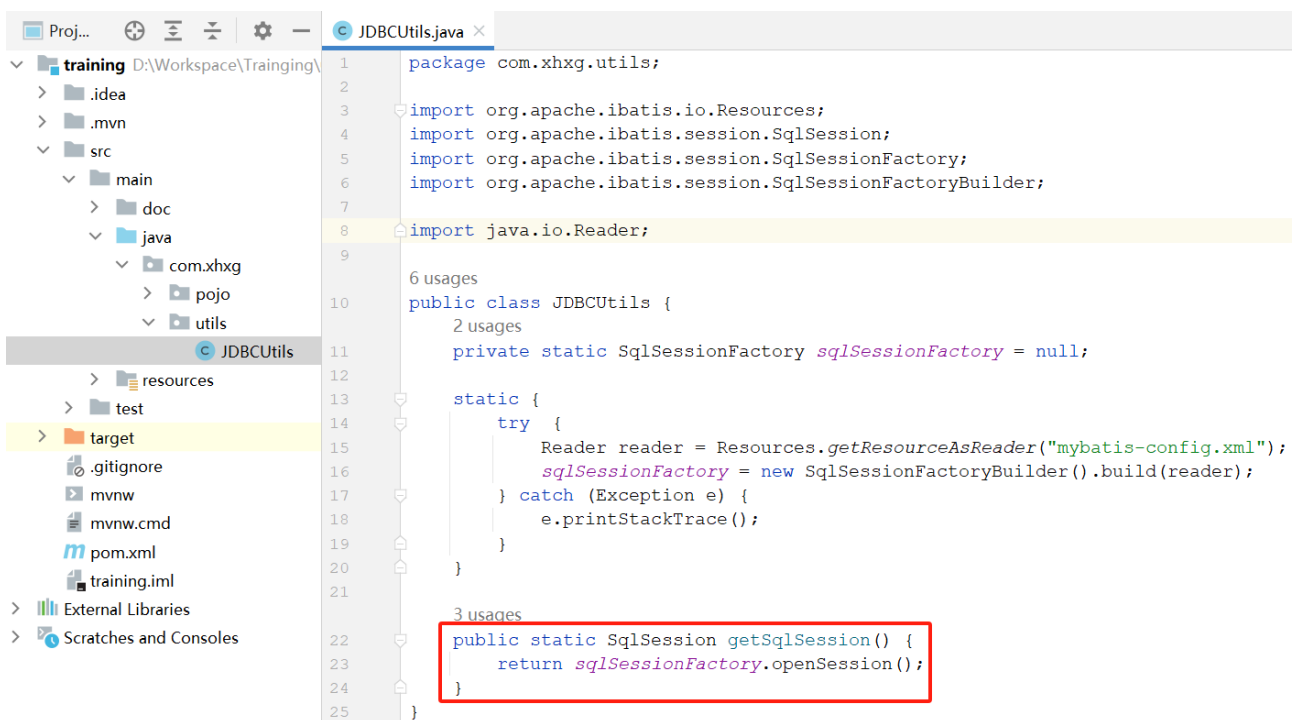
4、创建 db.properties 配置文件

在 src/main/resources 目录下创建一个 **db.properties** 文件，内容如下：



5、创建一个 Mybatis 连接工具类 JDBCUtils

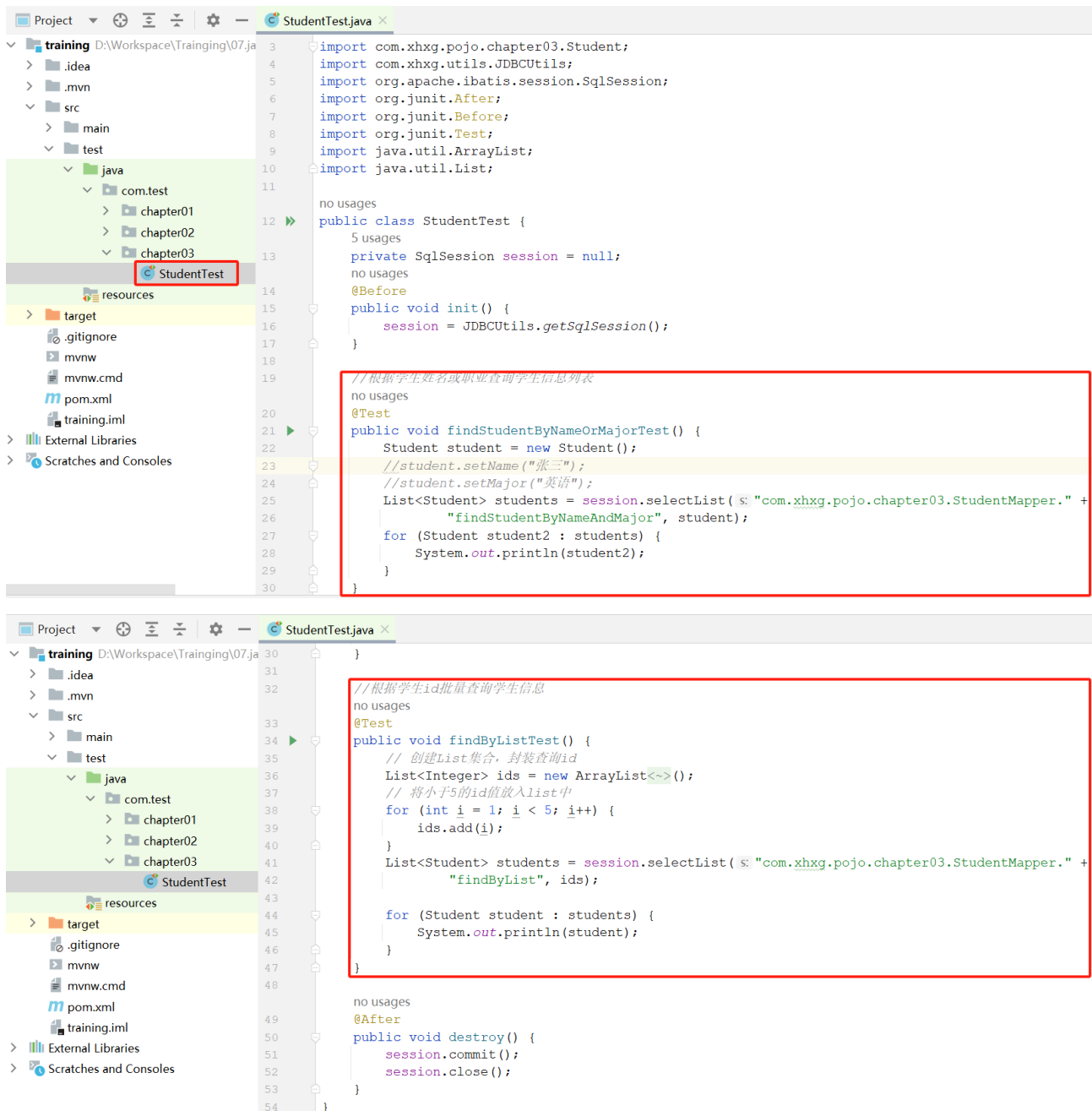
由于执行测试时，每个方法执行都需要读取配置文件，并根据配置文件的信息构建 `SqlSessionFactory` 对象、创建 `SqlSession` 对象、释放资源，这导致了大量的重复代码。为了简化开发，我们可以将读取配置文件和释放资源的代码封装到一个工具类 `JDBCUtils` 中（需要创建一个 `utils` 的 package 包，类路径：**com.xhxcg.utils.JDBCUtils**），然后通过工具类（`JDBCUtils`）创建 `SqlSession` 对象。代码如下：



6、编写测试类

以上代码和配置创建好后，就可以编写测试类进行验证测试。

我们在 src/test/java 目录下，建议测试包，路径为：com.test.chapter03，然后创建测类 StudentTest，测试的方法如下：



```
Project D:\Workspace\Training\07ja
> .idea
> .mvn
> src
  > main
  > test
    > java
      > com.test
        > chapter01
        > chapter02
        > chapter03
          > StudentTest
      > resources
    > target
    > .gitignore
    > mvnw
    > mvnw.cmd
    > pom.xml
    > training.iml
  > External Libraries
  > Scratches and Consoles

StudentTest.java
3  import com.xhgx.pojo.chapter03.Student;
4  import com.xhgx.utils.JDBCUtils;
5  import org.apache.ibatis.session.SqlSession;
6  import org.junit.After;
7  import org.junit.Before;
8  import org.junit.Test;
9  import java.util.ArrayList;
10 import java.util.List;
11
12 no usages
13 public class StudentTest {
14     5 usages
15     private SqlSession session = null;
16     no usages
17     @Before
18     public void init() {
19         session = JDBCUtils.getSqlSession();
20     }
21
22 // 根据学生姓名或职业查询学生信息列表
23 no usages
24 @Test
25 public void findStudentByNameOrMajorTest() {
26     Student student = new Student();
27     //student.setName("张三");
28     //student.setMajor("英语");
29     List<Student> students = session.selectList(s: "com.xhgx.pojo.chapter03.StudentMapper." +
30         "findStudentByNameAndMajor", student);
31     for (Student student2 : students) {
32         System.out.println(student2);
33     }
34 }
35
36 // 根据学生id批量查询学生信息
37 no usages
38 @Test
39 public void findByListTest() {
40     // 创建List集合，封装查询id
41     List<Integer> ids = new ArrayList<>();
42     // 将小于5的id值放入list中
43     for (int i = 1; i < 5; i++) {
44         ids.add(i);
45     }
46     List<Student> students = session.selectList(s: "com.xhgx.pojo.chapter03.StudentMapper." +
47         "findByList", ids);
48     for (Student student : students) {
49         System.out.println(student);
50     }
51 }
52
53 no usages
54 @After
55 public void destroy() {
56     session.commit();
57     session.close();
58 }
59 }
```

在不同的单元测试方法中，执行不同的单元测试，并输出相应的结果。

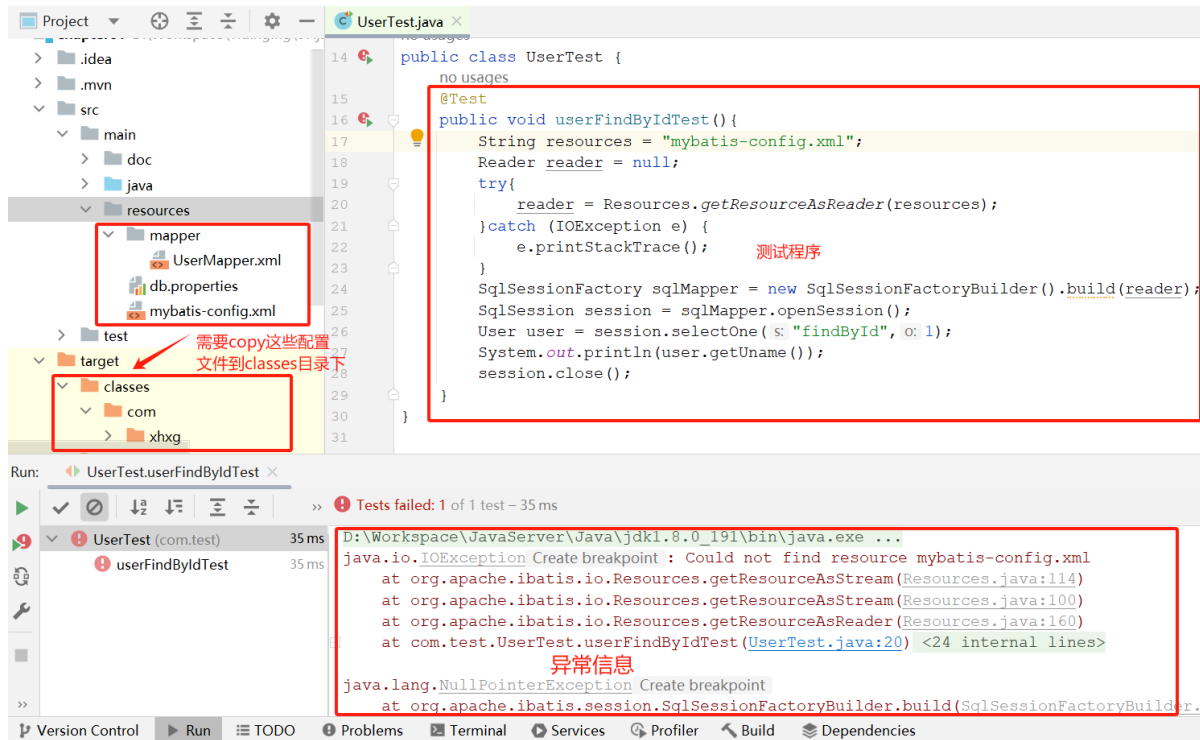
至此，本实验圆满成功，建立了 Java 代码与数据库的关联，验证了通过 Java 调用，正确获取数据库中的记录。

五、常见问题

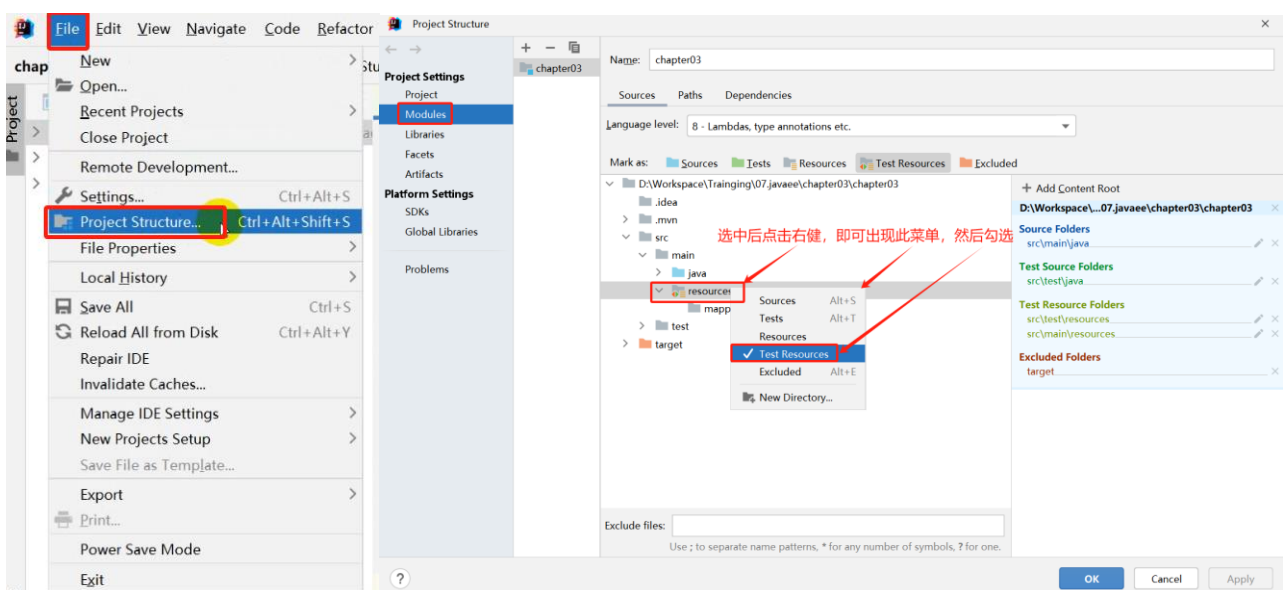
有时候 Maven 不能正确的复制配置文件到 target 目录中，此时运行测试程序就会报如下

异常：Could not find resource mybatis-config.xml

解决办法 1，就是手动将 src/main/resources/ 目录下的所有配置文件（包含目录）复制到 target/classes/ 目录下即可解决。（初学者建议采用此方法，增加你对项目工程的熟悉度）



解决办法 2，让 IDEA 自动编译复制，步骤如下截图。配置好后，IDEA 编译时会自动复制 resource 目录中配置文件到编译后的目录，就不需要再手工复制了。



注：若 Test Resources 不生效，可以换成勾选 Resources