实验 2 员工管理系统实验程序

一、环境准备（工具及版本）

1、Java 版本：JDK 1.8

2、构建工具：Maven 3.6.3

3、开发工具：IntelliJ IDEA Ultimate 2022.3.3

4、数据库：MySQL 5.7+ 以上

5、数据库工具：Navicat 10+以上

二、Mybatis 环境搭建（略，可参考第一章实验）

搭建过程省略，遇问题请参考第一章的搭建过程。

三、Maven 项目工程创建（略，可参考第一章实验）

创建过程省略，遇问题请参考第一章的入门实验。

工程名称为：employee

Group：com.xhxg

Version：Java EE 8

四、员工管理系统实验案例

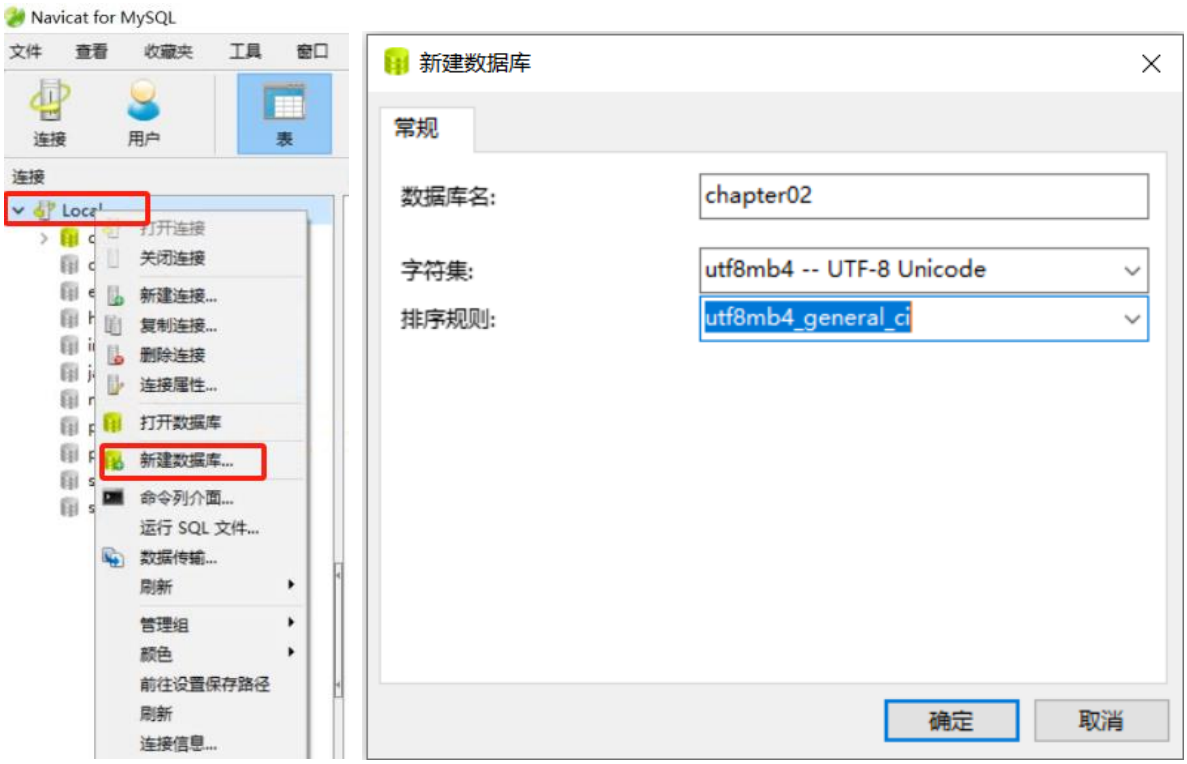
实验目标：完成一个**员工管理系统**，能够实现如下功能：**根据 id 查询、新增、修改、删除员工信息。**

第 1 步：数据库开发

先使用 MySQL 数据库创建一个 chapter02 数据库，然后创建一个用户表 t_employee，方法如下：

MySQL 安装过程，这里省略。安装完成后，我们通过使用 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，即可手动创建数据库，我们需要创建的数据

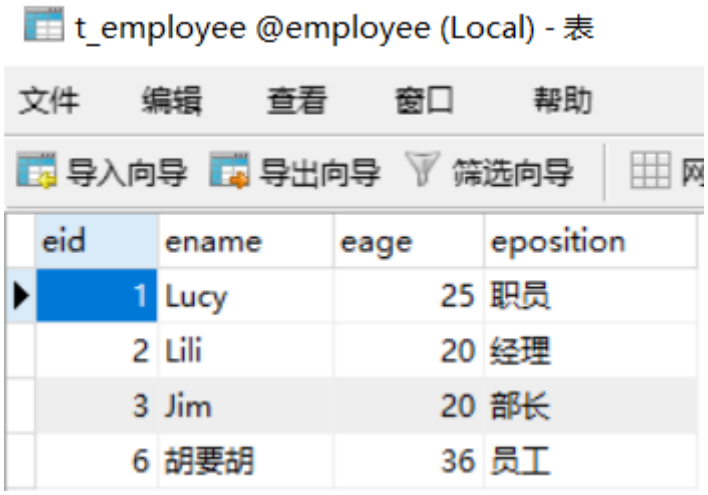
库名为：**chapter02，如图：**选中 **Local 连接名**，右键选择**【新建数据库】**



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

```
USE chapter02;
CREATE TABLE t_employee(
   eid INT PRIMARY KEY AUTO_INCREMENT,
   ename VARCHAR(50),
   eage INT,
   eposition varchar(50)
);
INSERT INTO t_employee(ename,eage,eposition) VALUES('Lucy',25,'职员');
INSERT INTO t_employee(ename,eage,eposition) VALUES('Lili',20,'经理');
INSERT INTO t_employee(ename,eage,eposition) VALUES('Jim',20,'部长');
```

全部创建完成后，效果如下：



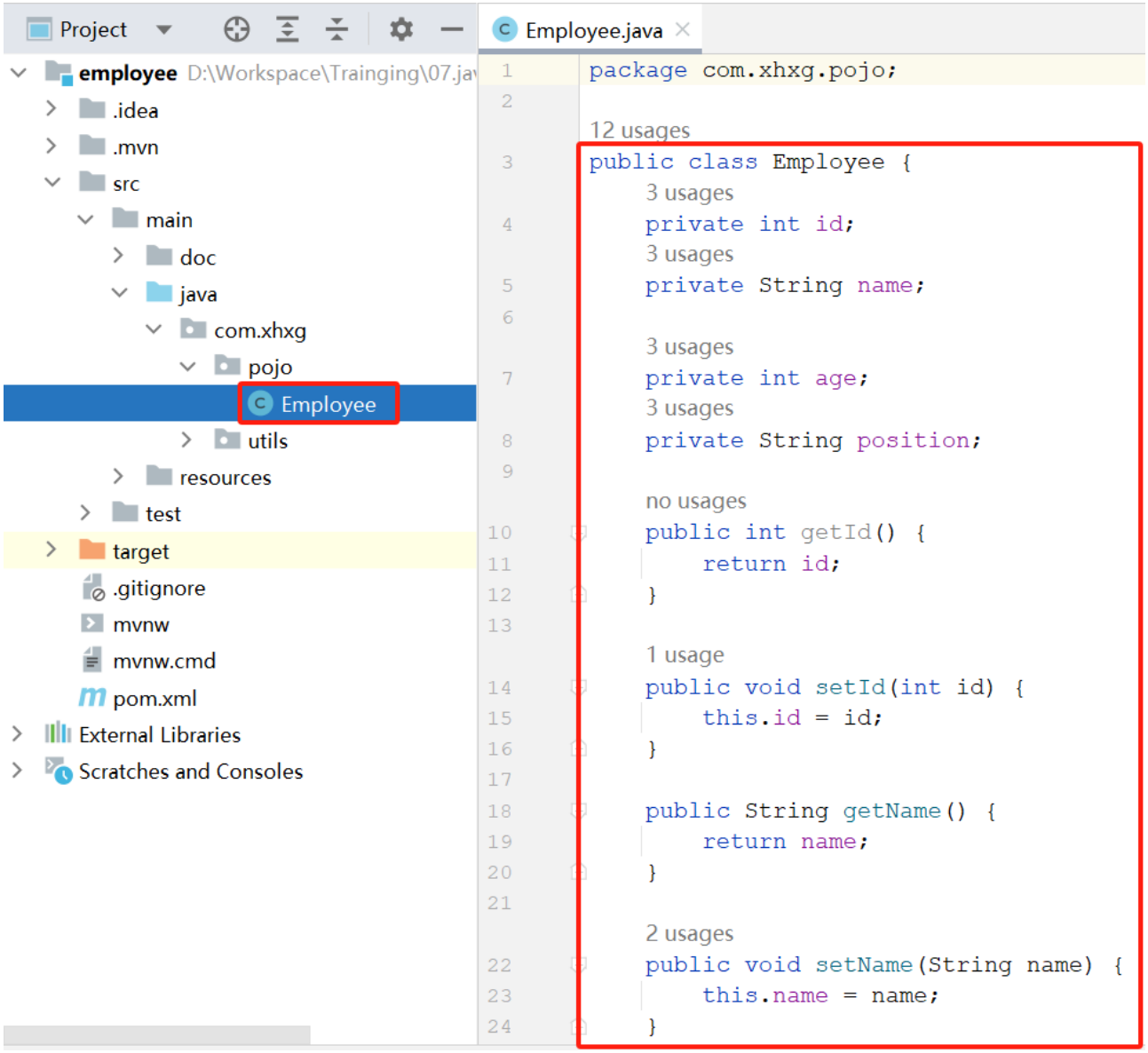
eid	ename	eage	eposition
1	Lucy	25	职员
2	Lili	20	经理
3	Jim	20	部长
6	胡要胡	36	员工

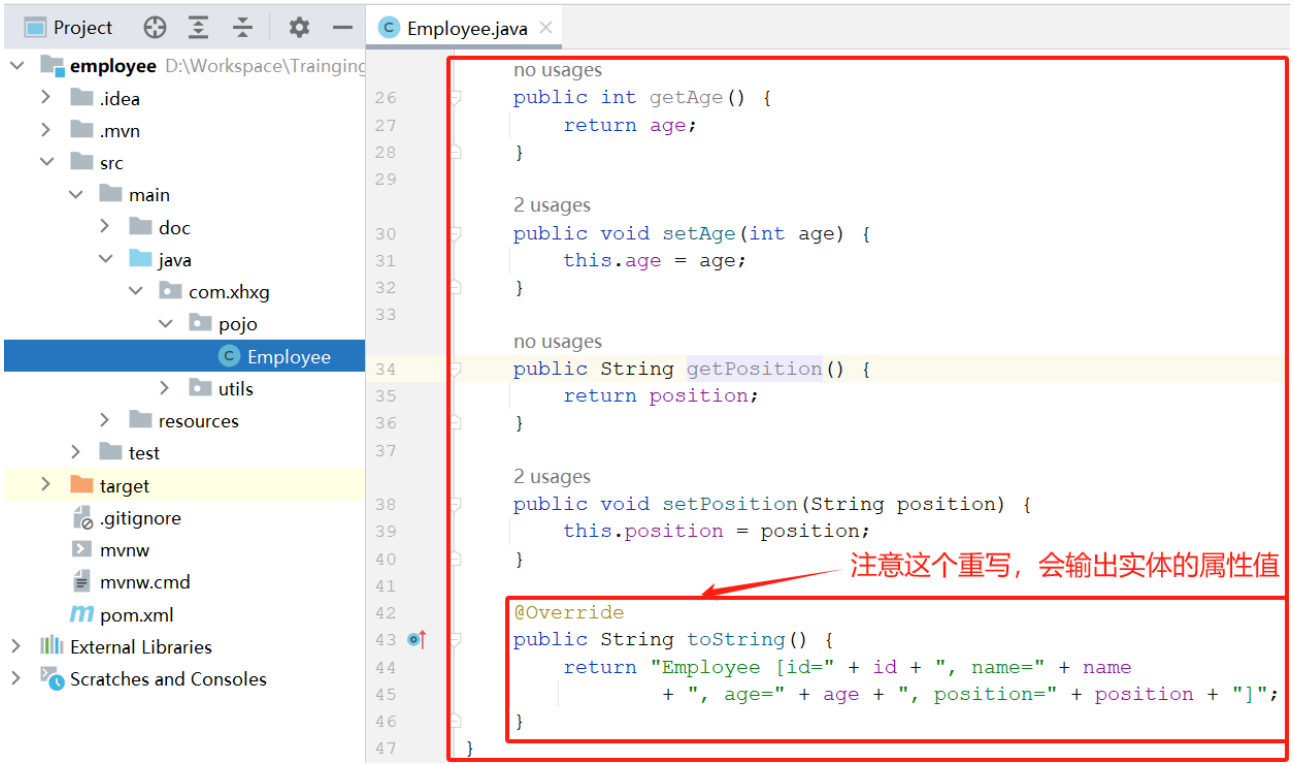
至此，数据库部分开发完毕。

第 2 步：程序代码开发

1、创建 POJO 实体

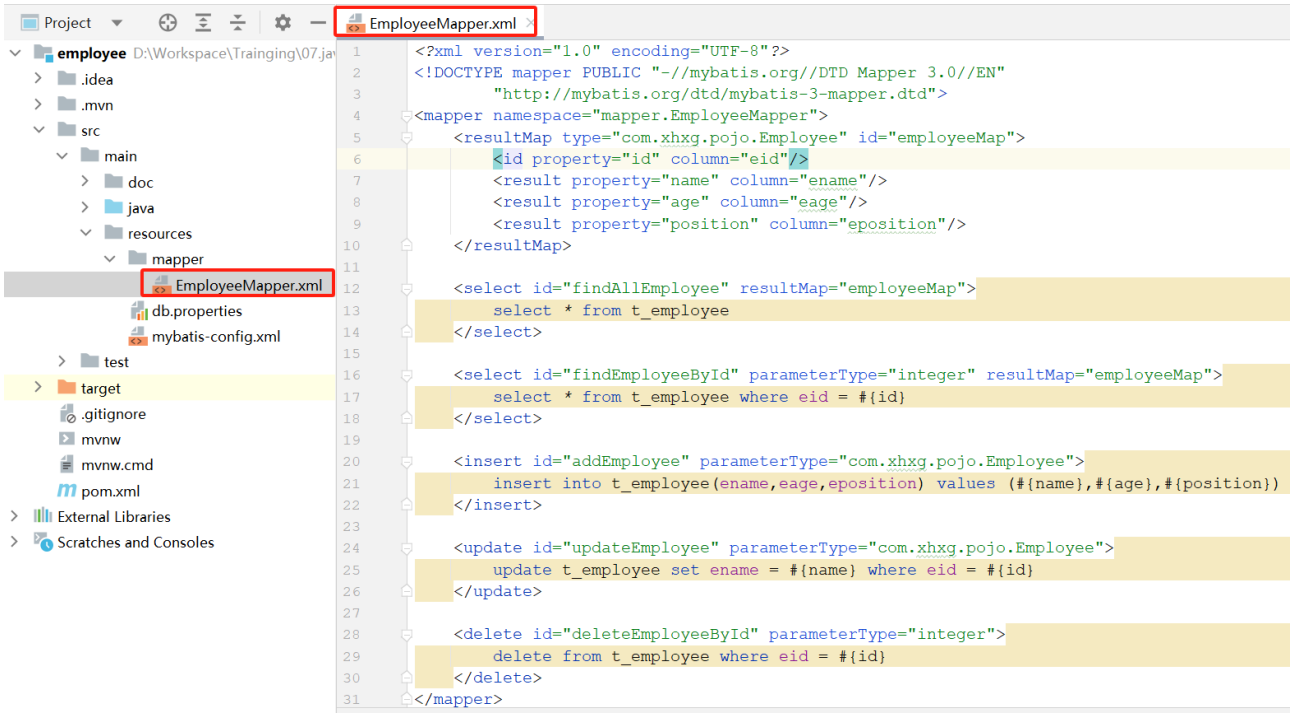
打开前面创建的 Mybatis 项目工程, 工程名称是 **employee**。创建代码包为：com.xhxg.pojo，并且创建 **Employee** 类


Project
employee D:\Workspace\Trainging\07.ja
.idea
.mvn
src
main
doc
java
com.xhxg
pojo
Employee
utils
resources
test
target
.gitignore
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
Employee.java
package com.xhxg.pojo;
12 usages
public class Employee {
3 usages
private int id;
3 usages
private String name;
3 usages
private int age;
3 usages
private String position;
no usages
public int getId() {
return id;
}
1 usage
public void setId(int id) {
this.id = id;
}
public String getName() {
return name;
}
2 usages
public void setName(String name) {
this.name = name;
}



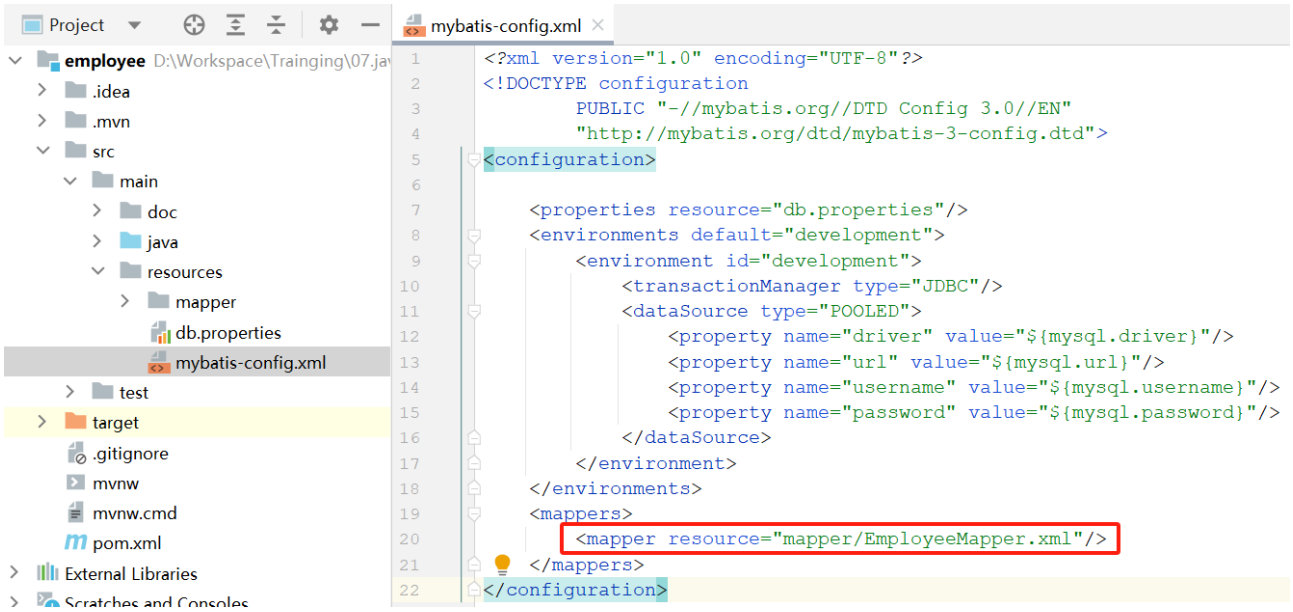
2、创建映射文件 EmployeeMapper.xml

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



3、创建 mybatis-config.xml 配置文件

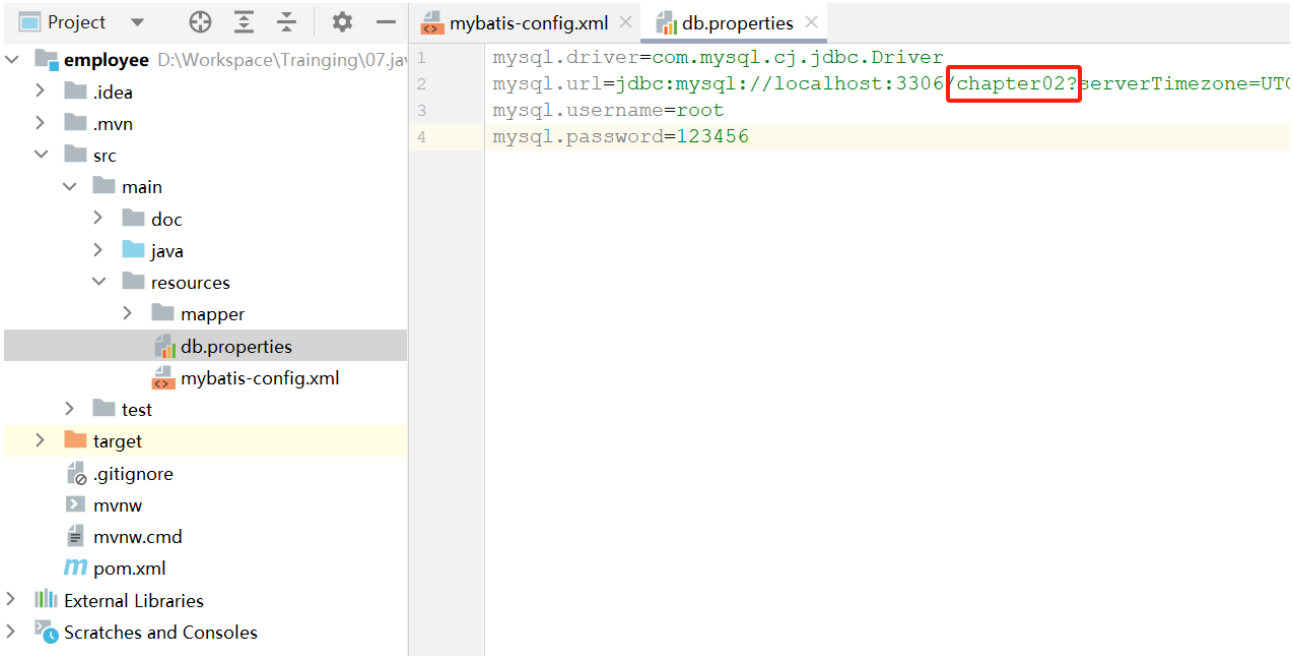
在 src/main/resources 目录下创建一个 **mybatis-config.xml** 文件，内容如下：



里面的配置项，需要创建另一个配置文件 db.properties 进行配置，所以我们跟着创建。

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

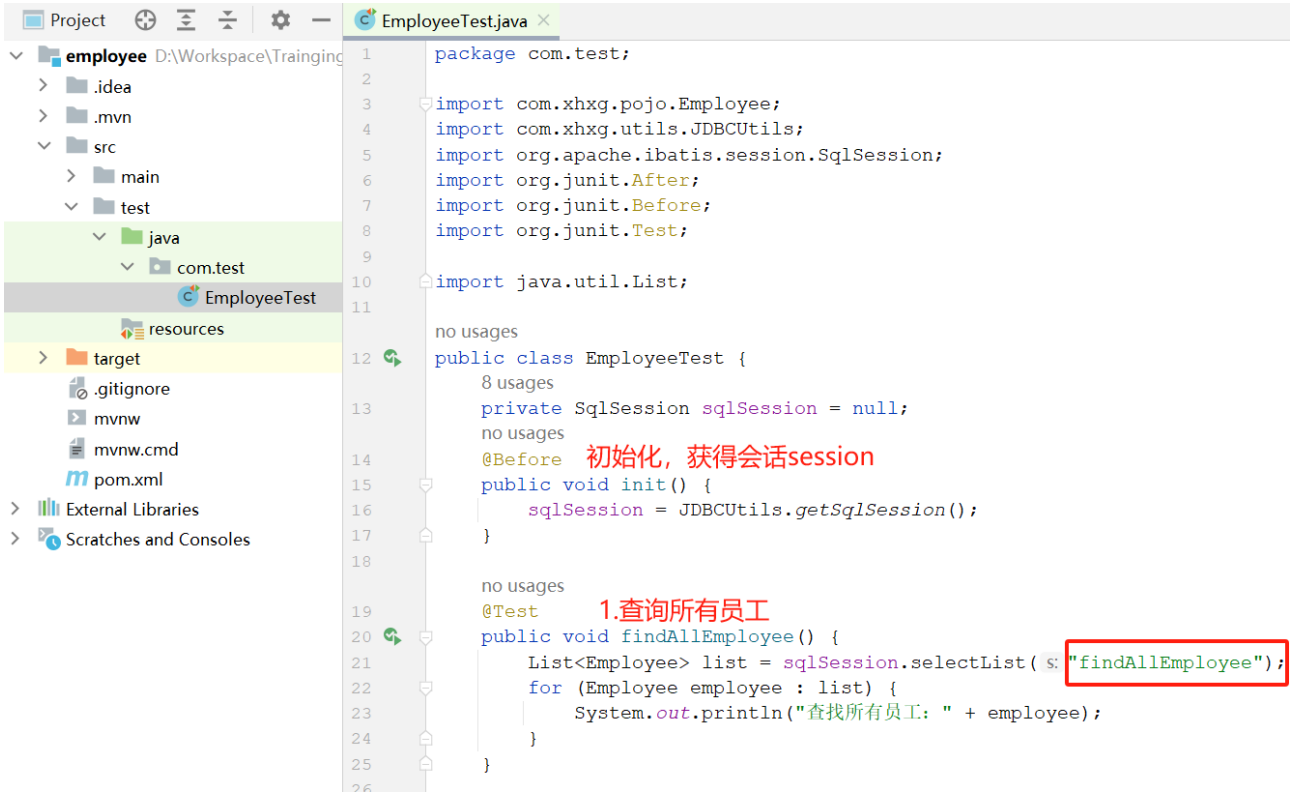
由于执行测试时，每个方法执行都需要读取配置文件，并根据配置文件的信息构建 SqlSessionFactory 对象、创建 SqlSession 对象、释放资源，这导致了大量的重复代码。为了

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.xhxg.utils.JDBCUtils,**），然后通过工具类（**JDBCUtils**）创建 SqlSession 对象。代码如下：

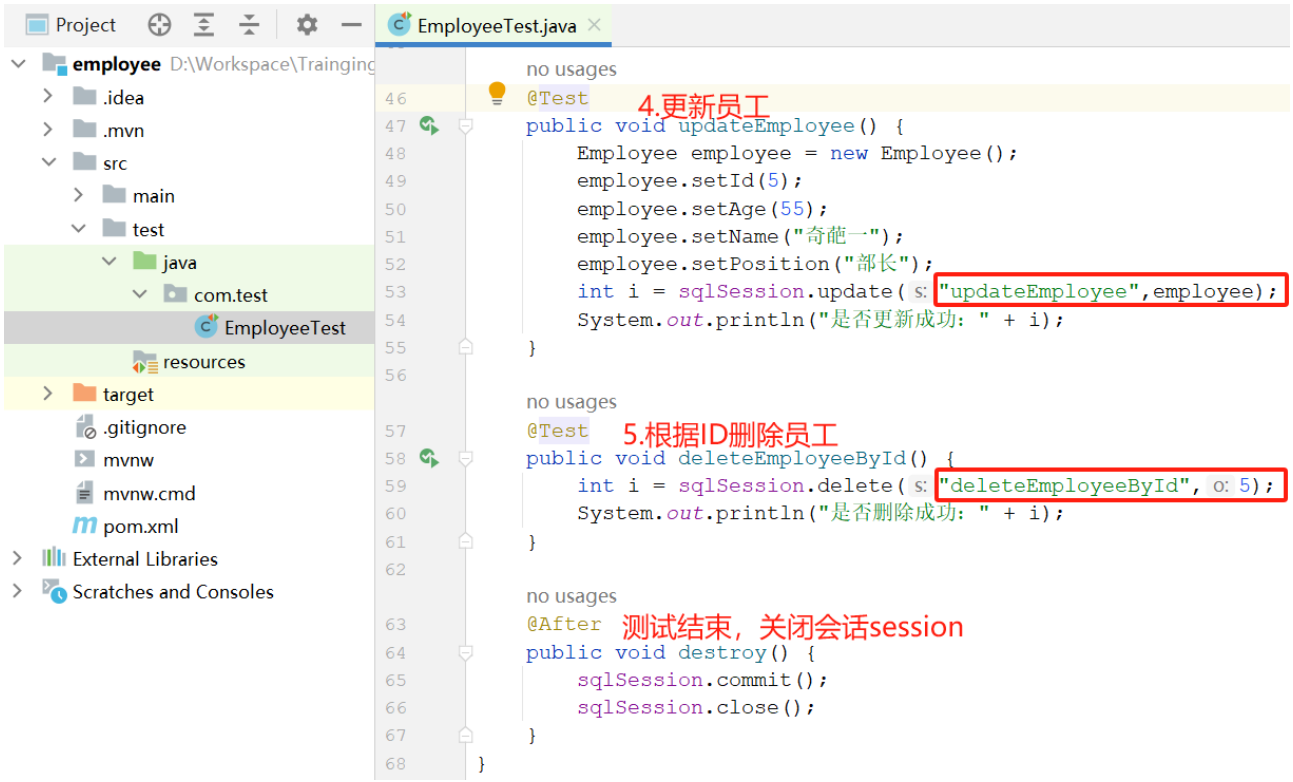
6、编写测试类

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。

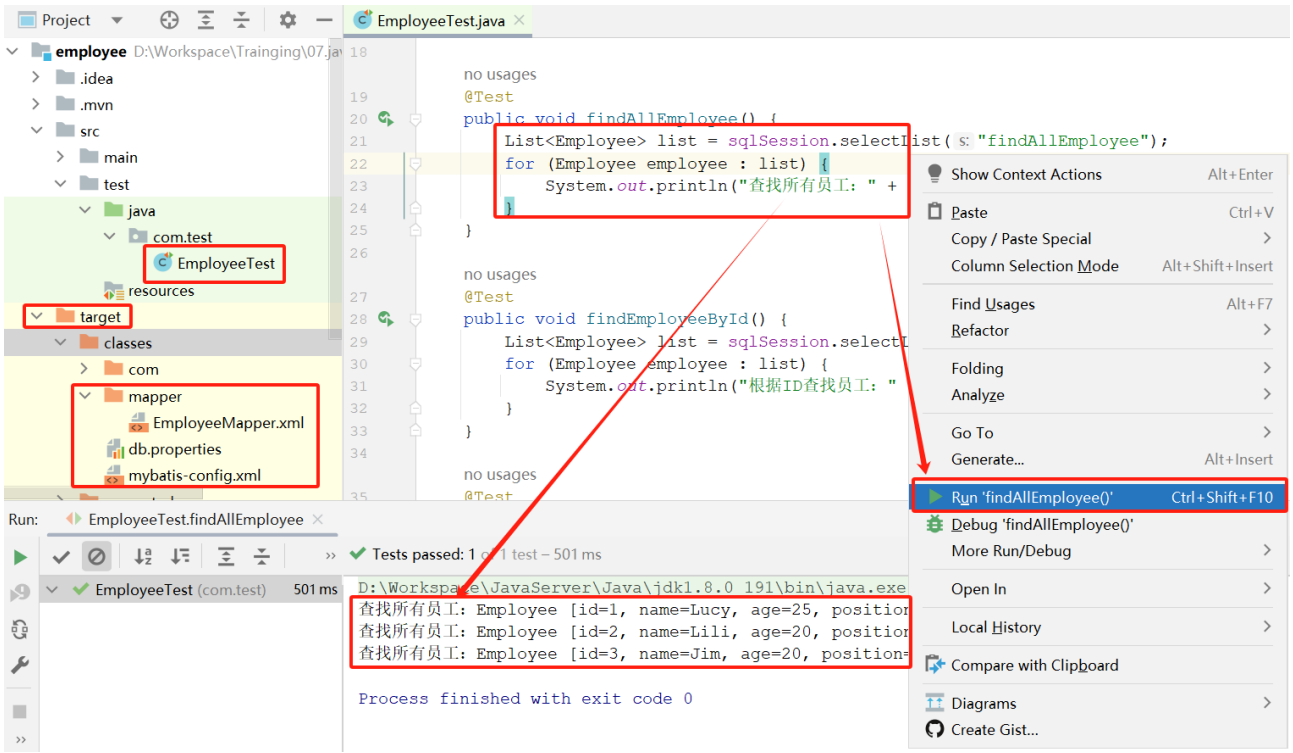
我们在 src/test/java 目录下，建议测试包，路径为：com.test，然后创建测类 EmployeeTest，测试的方法如下：


Project
employee D:\Workspace\Trainging
.idea
.mvn
src
main
test
java
com.test
EmployeeTest
resources
target
.gitignore
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
EmployeeTest.java
package com.test;
import com.xhxg.pojo.Employee;
import com.xhxg.utils.JDBCUtils;
import org.apache.ibatis.session.SqlSession;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import java.util.List;
no usages
public class EmployeeTest {
8 usages
private SqlSession sqlSession = null;
no usages
@Before
初始化，获得会话session
public void init() {
sqlSession = JDBCUtils.getSqlSession();
}
no usages
@Test
1.查询所有员工
public void findAllEmployee() {
List<Employee> list = sqlSession.selectList(s: "findAllEmployee");
for (Employee employee : list) {
System.out.println("查找所有员工: " + employee);
}
}


Project
employee D:\Workspace\Trainging
.idea
.mvn
src
main
test
java
com.test
EmployeeTest
resources
target
.gitignore
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
EmployeeTest.java
no usages
@Test
2.根据ID查询员工
public void findEmployeeById() {
List<Employee> list = sqlSession.selectList(s: "findEmployeeById", o: 1);
for (Employee employee : list) {
System.out.println("根据ID查找员工: " + employee);
}
}
no usages
@Test
3.添加员工
public void addEmployee() {
Employee employee = new Employee();
employee.setAge(36);
employee.setName("胡要胡");
employee.setPosition("员工");
int i = sqlSession.insert(s: "addEmployee",employee);
System.out.println("是否新增成功: " + i);
}



在不同的单元测试方法中，执行不同的单元测试。如在查询所有员工的测试方法中测试时，正确会输出查询结果，如下图：



至此，本实验完满成功，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，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。