

高级 Web 技术实验一：基于 MyBatis 的校级课程管理程序

1，任务目的

掌握 MyBatis 的环境搭建、核心配置文件、映射文件，学会使用基于 xml 和基于注解的 MyBatis 进行关系数据库的增删查改操作。

2，实验要求

1，技术选型：Java + Spring + MyBatis；推荐使用的数据库为 MySQL，引擎为 innoDB 存储引擎；2，实验结果在 Test 包中创建测试类，将实验结果直接输出到控制台，暂时不需要使用前端页面进行展示；3，提交实验报告，项目代码推荐提交至 Github 并在实验报告中给出项目的仓库地址。

注：请同学们把实验课的报告写成合集打印出来后交给学委收集齐全，在课程考试结束后一周内交至逸夫楼405。

3，实验题目

现有一个课程表 c_course 和一个学院表 s_school, 学院表和课程表之间是一对多的关系。课程表和学院表分别如表 1 和表 2 所示。

课程表 c_course 表 1:

课程 id (id)	课程名 (name)	课时 (hours)	开课学院(schools)
1	C 语言程序设计	70	1
2	Python 程序设计	70	1
3	大学英语	96	2
4	高级 Web 技术	32	1

学院表 s_school 表 2:

学院 id(id)	学院名称(school_name)
1	计算机学院
2	外国语学院

请根据以上表的字段及值完成数据库建库及数据初始化的过程，逐步完成以下子任务。

1. 查询 id=2 的课程信息；
2. 查询出所有计算机学院开设的课程信息；
3. 将 id=4 这门课程的课时数修改为 32+8=40；

4. 插入一条新的课程记录：names="大数据存储"，hours=32，schools=1；
5. 输出所有的学院开设的课程信息。

4, 实验指导

0. 环境搭建：安装 Java, Maven, IDEA 或者 Eclipse, MySQL等软件, 准备好开发环境。
1. 创建 Java 项目, 在 pom.xml 引入依赖, 参考：

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.hnust.ssm</groupId>
    <artifactId>exp1</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>
        <dependency>
            <groupId>org.mybatis</groupId>
            <artifactId>mybatis</artifactId>
            <version>3.5.2</version>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.11</version>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
            <scope>compile</scope>
        </dependency>
    </dependencies>
    <build>
        <resources>
            <resource>
                <directory>src/main/java</directory>
                <includes>
                    <include>/**/*.properties</include>
                    <include>/**/*.xml</include>
                </includes>
                <filtering>true</filtering>
            </resource>
        </resources>
    </build>
```

```
</project>
```

2. db.properties 参考:

```
mysql.driver=com.mysql.cj.jdbc.Driver
mysql.url=jdbc:mysql://localhost:3306/mybatis?
serverTimezone=UTC&characterEncoding=utf8&useUnicode=true&useSSL=false
mysql.username=root
mysql.password=root
```

3. mybatis-config.xml 参考:

```
<?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>
    <!-- 环境配置 -->
    <!-- 加载类路径下的属性文件 -->
    <properties resource="db.properties"/>
    <environments default="development">
        <environment id="development">
            <transactionManager type="JDBC"/>
            <!-- 数据库连接相关配置 ,db.properties文件中的内容-->
            <dataSource type="POOLED">
                <property name="driver" value="${mysql.driver}"/>
                <property name="url" value="${mysql.url}"/>
                <property name="username" value="${mysql.username}"/>
                <property name="password" value="${mysql.password}"/>
            </dataSource>
        </environment>
    </environments>
    <mappers>
        <mapper class="com.itheima.dao.IStudentMapper"/>
    </mappers>
</configuration>
```

4. MyBatisUtils 工具类:

```
import java.io.Reader;

import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;

/**
```

```

* 工具类
*/
public class MyBatisUtils {

    private static SqlSessionFactory sqlSessionFactory = null;

    // 初始化SqlSessionFactory对象
    static {
        try {
            // 使用MyBatis提供的Resources类加载MyBatis的配置文件
            Reader reader =
                Resources.getResourceAsReader("mybatis-config.xml");
            // 构建SqlSessionFactory工厂
            sqlSessionFactory =
                new SqlSessionFactoryBuilder().build(reader);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    // 获取SqlSession对象的静态方法
    public static SqlSession getSession() {
        return sqlSessionFactory.openSession();
    }
}

```

5. 推荐的建库脚本:

```

USE mybatis;
CREATE TABLE s_school (
    id int(32) PRIMARY KEY AUTO_INCREMENT,
    schoolname varchar(40)
);

INSERT INTO s_school VALUES (1, '计算机学院');
INSERT INTO s_school VALUES (2, '外国语学院');

CREATE TABLE s_student (
    id int(32) PRIMARY KEY AUTO_INCREMENT,
    name varchar(40),
    hours int,
    sid int(32) NOT NULL,
    FOREIGN KEY(sid) REFERENCES s_school(id)
);

INSERT INTO s_student VALUES (1, 'C语言程序设计', 70,1);
INSERT INTO s_student VALUES (2, 'Python程序设计', 70,1);
INSERT INTO s_student VALUES (3, '大学英语', 96,2);

```

```
INSERT INTO s_student VALUES (4, '高级web技术', 32,1);
```

版本历史

- 1, 起草, 陈向, 修改时间: 2022年4月5日21点。
- 2, 修改, 陈向, 修改时间: 2023年3月19日10点。