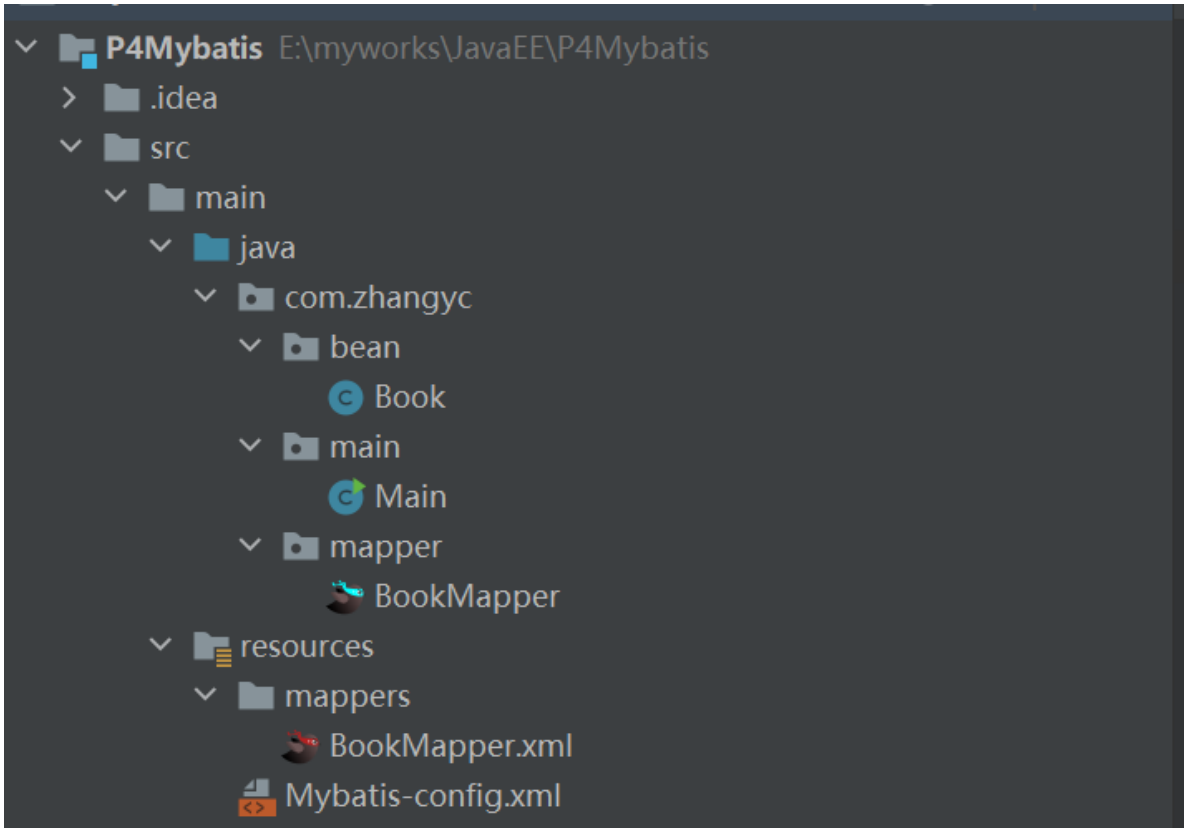
Java EE 第四次实习 MyBatis

课程	姓名	学号	班级
Java EE	张宇晨	2020012249	软工2001班

开发环境与工程结构

Intellij IDEA 2020.2

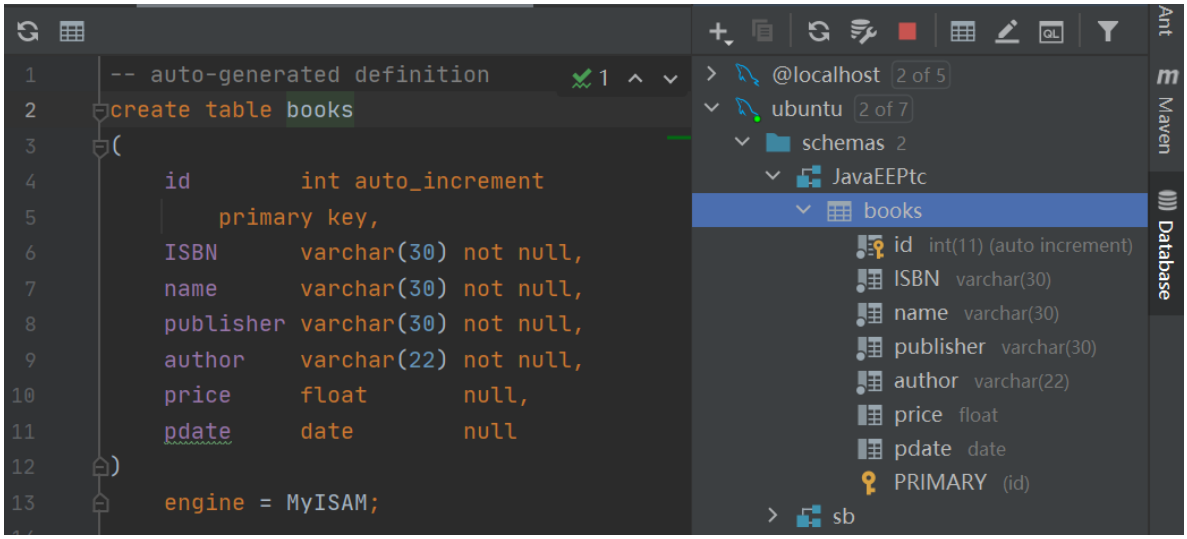
工程结构



实现内容

1. 创建book表

id自增，由数据库auto_increment方式实现

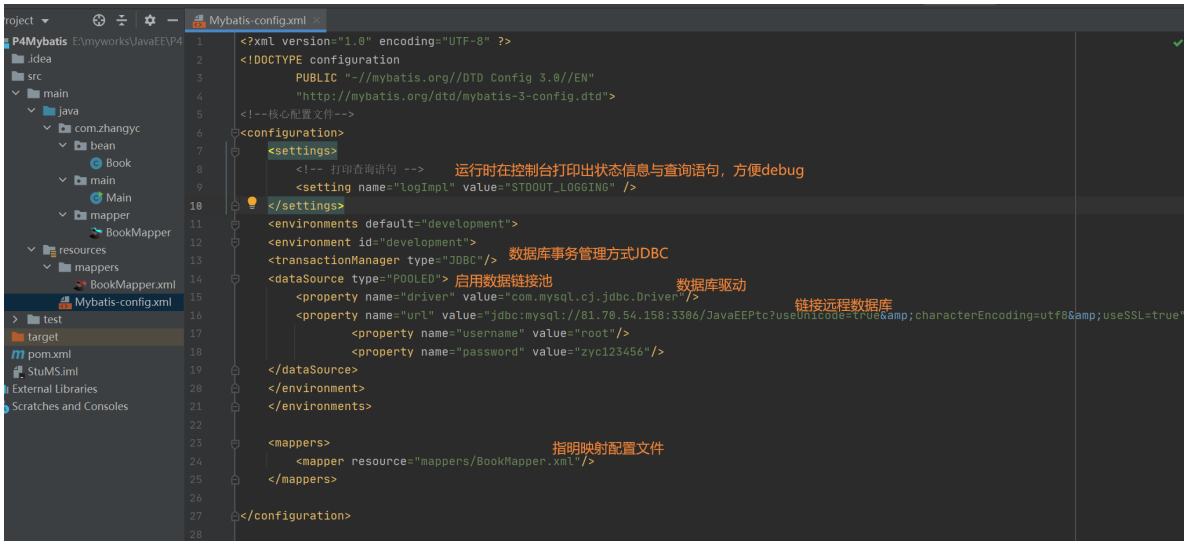


初始数据填充

<Filter Criteria>

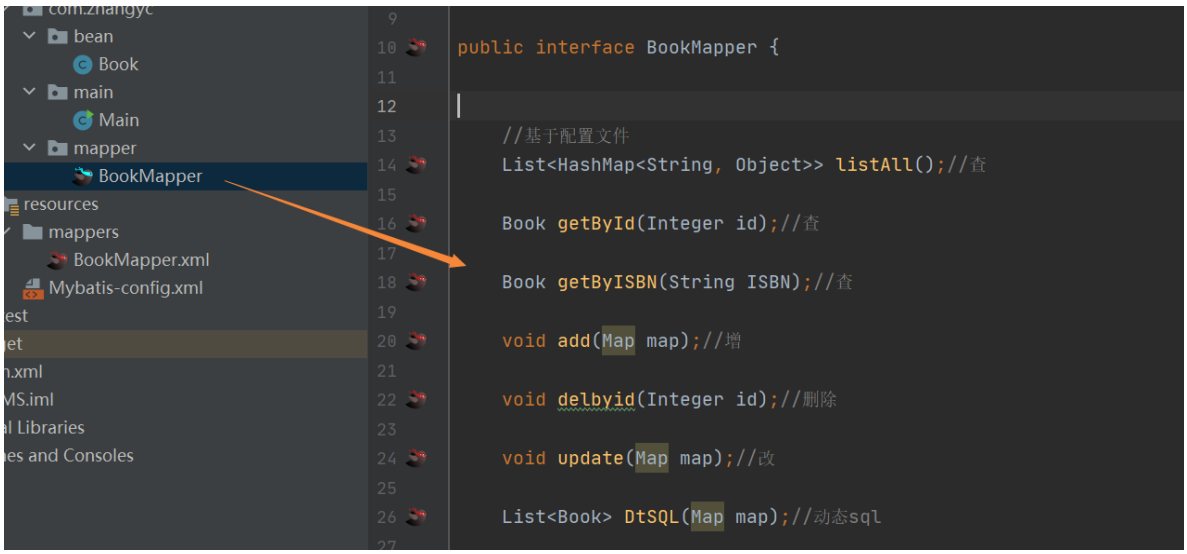
	id	ISBN	name	publisher	author	price	pdate
1	1	178-7-111	JavaEE开发技术	机械工业出版社	邹青	100	2021-11-05
2	2	321-981-702	智慧之城	须弥出版社	布椰尔	67	2020-11-24
3	3	221-98-702	机器学习	清华大学出版社	周志华	45	2020-06-18
4	4	97-66-123	提瓦特大陆史	璃月往生堂出版社	摩拉克斯	56	2021-11-10
5	5	331-98-702	自由之风	蒙德出版社	巴巴托斯	78	2021-07-23
6	6	331-98-702	计算机组成原理	清华大学出版社	战德臣	99	2021-12-15

2. 主配置文件

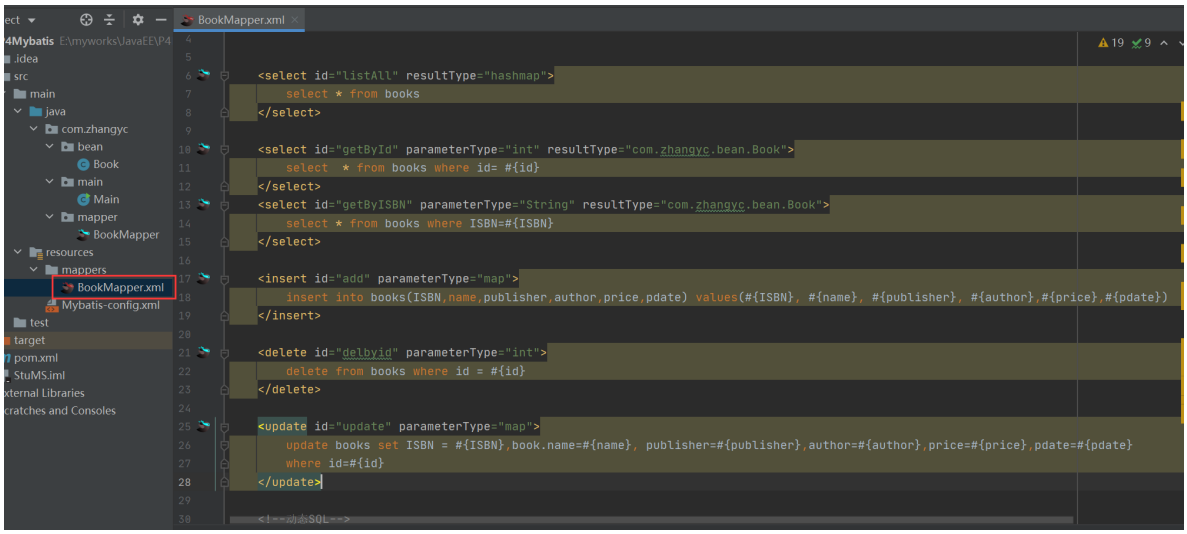


3. 映射文件增删改查

（1）声明——BookMapper.java



（2）定义——BookMapper.xml



（3）测试

测试代码：

```
public static void main(String argc[]) throws IOException {
        //主配置文件
        InputStream is = Resources.getResourceAsStream("Mybatis-config.xml");
        //创建SQL会话工厂，读取主配置文件建立数据库连接
        SqlSessionFactory ssf = new SqlSessionFactoryBuilder().build(is);
        //打开一个数据库会话
        SqlSession session = ssf.openSession();
        //通过getMapper方法获得mapper接口的的代理对象，通过mapper调用mapper接口中定义的方
法
        BookMapper mapper = session.getMapper(BookMapper.class);
     System.out.println("---基于配置文件的操作：");
        //查询
        System.out.println("查询书籍列表:");
        List<HashMap<String, Object>> bl = mapper.listAll();
        for(int i=0;i<bl.size();++i)
        {
            System.out.println(bl.get(i).toString());
        }
        System.out.println("查询id=1的书:");
        System.out.println(mapper.getById(1).toString());
        //增
        System.out.println("添加新书，ISBN=123，书名=测试一，出版社=西农出版社，作者=作
者一:");
        Map map = new HashMap();
        map.put("ISBN","123");
        map.put("name","测试一");
        map.put("publisher","西农出版社");
        map.put("author","作者一");
        map.put("price",89.0);
        map.put("pdate","2020-1-1");
        mapper.add(map);
        System.out.println("插入结果：");
        System.out.println(mapper.getByISBN("123").toString());
        //改
        System.out.println("修改新插入的书");
        map.put("ISBN","11-22-33");
        map.put("id",20);
        mapper.update(map);
        System.out.println("修改结果：");
        System.out.println(mapper.getByISBN("11-22-33").toString());
        //删除
        System.out.println("删除id=18的书籍：");
        mapper.delbyid(20);
        System.out.println("删除结果：");
        if(mapper.getById(20)==null)
        {
            System.out.println("删除成功");
        }
        session.close();
    }
```

测试结果：

4.注解增删改查

（1）定义

```
    //基于注解
    @Select("select * from books")
    List<Book> zjgetbooklist();

    @Select("select * from books where author = #{author}")
        //查询
    Book getByAuthor(String author);

    //增加书籍
    @Insert("insert into books(ISBN,name,publisher,author,price,pdate ) values
(#{ISBN},#{name},#{publisher},#{author},#{price},#{pdate})")
    void zjadd(@Param("ISBN") String ISBN, @Param("name") String name,
               @Param("publisher") String publisher, @Param("author") String
author, @Param("price") float price, @Param("pdate") String pdate);

    //更新书籍
    @Update("update books set ISBN=#{ISBN}, name=#{name}, publisher=#
{publisher}, author=#{author}, price=#{price}, pdate=#{pdate} \n" +
            "        where id=#{id}")
    void zjupdate(@Param("id") int id, @Param("name") String name,
@Param("ISBN") String ISBN, @Param("author") String author,
                  @Param("publisher") String publisher, @Param("price") float
price, @Param("pdate") String pdate);//改

    //删除
    @Delete("delete from books where name=#{name}")
    void zjdelbyname(String name);
```

（2）测试

测试代码：

```
public static void main(String argc[]) throws IOException {
        //主配置文件
        InputStream is = Resources.getResourceAsStream("Mybatis-config.xml");
        //创建SQL会话工厂，读取主配置文件建立数据库连接
        SqlSessionFactory ssf = new SqlSessionFactoryBuilder().build(is);
        //打开一个数据库会话
        SqlSession session = ssf.openSession();
```

```
        //通过getMapper方法获得mapper接口的的代理对象，通过mapper调用mapper接口中定义的方法
        BookMapper mapper = session.getMapper(BookMapper.class);
        System.out.println("---基于注解的操作：");
        System.out.println("查询书籍列表:");
                List<HashMap<String, Object>> bl = mapper.listAll();
        for(int i=0;i<bl.size();++i)
        {
            System.out.println(bl.get(i).toString());
        }
        System.out.println("按照作者查询书籍:");
        System.out.println(mapper.getByAuthor("邹青").toString());
        System.out.println("插入测试书籍__zjTest1,zjTest1,nwafu,zyc");
        mapper.zjadd("98-09-44","zjtest","nwafu","zyc",(float)78.2,"2020-10-12");
        System.out.println("查询插入结果");
        System.out.println(mapper.getByAuthor("zyc").toString());
        System.out.println("测试更新书籍__35,\"12345\",\"张宇晨\",\"西北农林科技大学\"");
        mapper.zjupdate(12,"注解测试书1","9090","张宇晨","西北农林科技大学",(float)78.2,"2020-10-12");
        System.out.println("查询更新结果：");
        System.out.println(mapper.getById(12).toString());
        System.out.println("测试删除");
        session.close();
    }
```

测试结果：

```
---基于注解的操作：
查询书籍列表:
{ISBN=178-7-111, author=邹青, price=100.0, pdate=2021-11-05, name=JavaEE开发技术与实践教程, publisher=机械工业出版社, id=1}
{ISBN=321-981-702, author=布椰尔, price=67.0, pdate=2020-11-24, name=智慧之城, publisher=须弥出版社, id=2}
{ISBN=221-98-702, author=周志华, price=45.0, pdate=2020-06-18, name=机器学习, publisher=清华大学出版社, id=3}
{ISBN=97-66-123, author=摩拉克斯, price=56.0, pdate=2021-11-10, name=提瓦特大陆史, publisher=璃月往生堂出版社, id=4}
{ISBN=331-98-702, author=巴巴托斯, price=78.0, pdate=2021-07-23, name=自由之风, publisher=蒙德出版社, id=5}
{ISBN=331-98-702, author=战德臣, price=99.0, pdate=2020-10-22, name=计算机组成原理, publisher=清华大学出版社, id=6}
{ISBN=902-34-76, author=作者一, price=70.0, pdate=2021-11-17, name=C++ primer plus, publisher=清华大学出版社, id=8}
按照作者查询书籍:
Book{id=1, ISBN='178-7-111', name='JavaEE开发技术与实践教程', publisher='机械工业出版社', author='邹青', price=100.0, pdate='2021-11-05'}
插入测试书籍__zjTest1,zjTest1,nwafu,zyc
查询插入结果
Book{id=12, ISBN='98-09-44', name='zjtest', publisher='nwafu', author='zyc', price=78.2, pdate='2020-10-12'}
测试更新书籍__35,"12345","张宇晨","西北农林科技大学"
查询更新结果：
Book{id=12, ISBN='9090', name='注解测试书1', publisher='西北农林科技大学', author='张宇晨', price=78.2, pdate='2020-10-12'}
测试删除
```

5.动态SQL

（1）定义

```
    <select id="DtSQL" parameterType="map" resultType="com.zhangyc.bean.Book">
        select  * from books where 1=1
        <if test="name!=null and name!=''">and book_name like concat('%',#
{name},'%')</if>
        <if test="author!=null and author!=''">and author like concat('%',#
{author},'%')</if>
        <if test="publisher!=null and publisher!=''">and publisher like
concat('%',#{publisher},'%')</if>
        <if test="ISBN!=null and ISBN!=''">and ISBN like concat('%',#{ISBN},'%')
</if>
        <if test="price!=null and price!=''">and price between #{price}-20.0 and
#{price}+20.0 </if>
        <if test="pdate!=null and pdate!=''">and pdate between
date_add('${pdate}',INTERVAL -365 DAY) and date_add('${pdate}',INTERVAL 365 DAY)
</if>

    </select>
```

(2) 测试

测试代码:

```
public static void main(String argc[]) throws IOException {
        //主配置文件
        InputStream is = Resources.getResourceAsStream("Mybatis-config.xml");
        //创建SQL会话工厂，读取主配置文件建立数据库连接
        SqlSessionFactory ssf = new SqlSessionFactoryBuilder().build(is);
        //打开一个数据库会话
        SqlSession session = ssf.openSession();
        //通过getMapper方法获得mapper接口的的代理对象，通过mapper调用mapper接口中定义的方
法
        BookMapper mapper = session.getMapper(BookMapper.class);
        System.out.println("执行动态SQL前的数据库状态");
        List<HashMap<String, Object>> bl = mapper.listAll();
        for(int i=0;i<bl.size();++i)
        {
            System.out.println(bl.get(i).toString());
        }

        Map map = new HashMap();
        System.out.println("加入时间2020-10-1");
        map.put("pdate","2020-10-1");
        List<Book> lb = mapper.DtSQL(map);
        for(int i=0;i<lb.size();++i)
        {
            System.out.println(lb.get(i).toString());
        }
        System.out.println("加入出版社模糊查询--清华");
        map.put("publisher","清华");
        lb = mapper.DtSQL(map);
        for(int i=0;i<lb.size();++i)
        {
            System.out.println(lb.get(i).toString());
        }
        System.out.println("加入出版社模糊查询--价格90");
```

```
        map.put("price",90.0);
        lb = mapper.DtSQL(map);
        for(int i=0;i<lb.size();++i)
        {
            System.out.println(lb.get(i).toString());
        }
        session.close();
    }
```

测试结果：

附录

JavaBean代码

```
package com.zhangyc.bean;

public class Book {
    private int id;
    private String ISBN;
    private String name;
    private String publisher;
    private String author;
    private float price;
    private String pdate;

    public int getId() {
        return id;
    }

    public float getPrice() {
        return price;
    }

    public String getPdate() {
        return pdate;
    }

    public void setPrice(float price) {
        this.price = price;
    }
```

```
    public void setId(int id) {
        this.id = id;
    }

    public String getISBN() {
        return ISBN;
    }

    public void setISBN(String ISBN) {
        this.ISBN = ISBN;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPublisher() {
        return publisher;
    }

    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }

    @Override
    public String toString() {
        return "Book{" +
                "id=" + id +
                ", ISBN='" + ISBN + '\'' +
                ", name='" + name + '\'' +
                ", publisher='" + publisher + '\'' +
                ", author='" + author + '\'' +
                ", price=" + price +
                ", pdate='" + pdate + '\'' +
                '}';
    }

    public void setPdate(String pdate) {
        this.pdate = pdate;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }



    public Book() {
    }
```

```
}
```

主配置文件

Mybatis-config.xml

```
<?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">
<!--核心配置文件-->
<configuration>
<!--    <settings>-->
<!--        &lt;!&ndash; 打印查询语句 &ndash;&gt;-->
<!--        <setting name="logImpl" value="STDOUT_LOGGING" />-->
<!--    </settings>-->
    <environments default="development">
    <environment id="development">
    <transactionManager type="JDBC"/>
    <dataSource type="POOLED">
        <property name="driver" value="com.mysql.cj.jdbc.Driver"/>
        <property name="url" value="jdbc:mysql://81.70.54.158:3306/JavaEEPtc?
useUnicode=true&amp;characterEncoding=utf8&amp;useSSL=true"/>
                <property name="username" value="root"/>
                <property name="password" value="zyc123456"/>
    </dataSource>
    </environment>
    </environments>

    <mappers>
        <mapper resource="mappers/BookMapper.xml"/>
    </mappers>

</configuration>
```

映射配置

BookMapper.java

```
package com.zhangyc.mapper;

import com.zhangyc.bean.Book;
import org.apache.ibatis.annotations.*;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

public interface BookMapper {


    //基于配置文件
    List<HashMap<String, Object>> listAll();//查

    Book getById(Integer id);//查
```

```
    Book getByISBN(String ISBN);//查

    void add(Map map);//增

    void delbyid(Integer id);//删除

    void update(Map map);//改

    List<Book> DtSQL(Map map);//动态sql


    //基于注解
    @Select("select * from books")
    List<Book> zjgetbooklist();

    @Select("select * from books where author = #{author}")
        //查询
    Book getByAuthor(String author);

    //增加书籍
    @Insert("insert into books(ISBN,name,publisher,author,price,pdate ) values
(#{ISBN},#{name},#{publisher},#{author},#{price},#{pdate})")
    void zjadd(@Param("ISBN") String ISBN, @Param("name") String name,
               @Param("publisher") String publisher, @Param("author") String
author, @Param("price") float price, @Param("pdate") String pdate);

    //更新书籍
    @Update("update books set ISBN=#{ISBN}, name=#{name}, publisher=#
{publisher}, author=#{author}, price=#{price}, pdate=#{pdate} \n" +
            "        where id=#{id}")
    void zjupdate(@Param("id") int id, @Param("name") String name,
@Param("ISBN") String ISBN, @Param("author") String author,
                  @Param("publisher") String publisher, @Param("price") float
price, @Param("pdate") String pdate);//改

    //删除
    @Delete("delete from books where name=#{name}")
    void zjdelbyname(String name);


}
```

BookMapper.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd" >
<mapper namespace="com.zhangyc.mapper.BookMapper">


    <select id="listAll" resultType="hashmap">
        select * from books
    </select>

    <select id="getById" parameterType="int" resultType="com.zhangyc.bean.Book">
        select  * from books where id= #{id}
```

```
    </select>
    <select id="getByISBN" parameterType="String"
resultType="com.zhangyc.bean.Book">
        select * from books where ISBN=#{ISBN}
    </select>

    <insert id="add" parameterType="map">
        insert into books(ISBN,name,publisher,author,price,pdate) values(#
{ISBN}, #{name}, #{publisher}, #{author},#{price},#{pdate})
    </insert>

    <delete id="delbyid" parameterType="int">
        delete from books where id = #{id}
    </delete>

    <update id="update" parameterType="map">
        update books set ISBN = #{ISBN},book.name=#{name}, publisher=#
{publisher},author=#{author},price=#{price},pdate=#{pdate}
        where id=#{id}
    </update>

    <!--动态SQL-->
    <select id="DtSQL" parameterType="map" resultType="com.zhangyc.bean.Book">
        select  * from books where 1=1
        <if test="name!=null and name!=''">and book_name like concat('%',#
{name},'%')</if>
        <if test="author!=null and author!=''">and author like concat('%',#
{author},'%')</if>
        <if test="publisher!=null and publisher!=''">and publisher like
concat('%',#{publisher},'%')</if>
        <if test="ISBN!=null and ISBN!=''">and ISBN like concat('%',#{ISBN},'%')
</if>
        <if test="price!=null and price!=''">and price between #{price}-20.0 and
#{price}+20.0 </if>
        <if test="pdate!=null and pdate!=''">and pdate between
date_add('${pdate}',INTERVAL -365 DAY) and date_add('${pdate}',INTERVAL 365 DAY)
</if>

    </select>

</mapper>
```

测试-Main

Main.java

```
package com.zhangyc.main;

import com.zhangyc.bean.Book;
import com.zhangyc.mapper.BookMapper;
import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;

import java.io.IOException;
```

```
import java.io.InputStream;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;

public class Main {
    public static void main(String argc[]) throws IOException {
        //主配置文件
        InputStream is = Resources.getResourceAsStream("Mybatis-config.xml");
        //创建SQL会话工厂，读取主配置文件建立数据库连接
        SqlSessionFactory ssf = new SqlSessionFactoryBuilder().build(is);
        //打开一个数据库会话
        SqlSession session = ssf.openSession();
        //通过getMapper方法获得mapper接口的的代理对象，通过mapper调用mapper接口中定义的方
法
        BookMapper mapper = session.getMapper(BookMapper.class);

////基于配置文件的数据库操作
        System.out.println("---基于配置文件的操作：");
        //查询
        System.out.println("查询书籍列表:");
        List<HashMap<String, Object>> bl = mapper.listAll();
        for(int i=0;i<bl.size();++i)
        {
            System.out.println(bl.get(i).toString());
        }
        System.out.println("查询id=1的书:");
        System.out.println(mapper.getById(1).toString());
        //插入
        System.out.println("添加新书，ISBN=123，书名=测试一，出版社=西农出版社，作者=作
者一:");
        Map map = new HashMap();
        map.put("ISBN","123");
        map.put("name","测试一");
        map.put("publisher","西农出版社");
        map.put("author","作者一");
        mapper.add(map);
        System.out.println("插入结果：");
        System.out.println(mapper.getByISBN("123").toString());
        //删除
        System.out.println("删除id=25的书籍：");
        mapper.delbyid(30);
        System.out.println("删除结果：");
        if(mapper.getById(30)==null)
        {
            System.out.println("删除成功");
        }
////基于注解的数据库操作
//
//        System.out.println("---基于注解的操作：");
//        System.out.println("查询书籍列表:");
//                List<HashMap<String, Object>> bl = mapper.listAll();
//        for(int i=0;i<bl.size();++i)
//        {
//            System.out.println(bl.get(i).toString());
//        }
//        System.out.println("按照作者查询书籍:");
```

```
//        System.out.println(mapper.getByAuthor("邹青").toString());
//        System.out.println("插入测试书籍__zjTest1,zjTest1,nwafu,zyc");
//        mapper.zjadd("98-09-44","zjtest","nwafu","zyc",(float)78.2,"2020-10-
12");
//        System.out.println("查询插入结果");
//        System.out.println(mapper.getByAuthor("zyc").toString());
//        System.out.println("测试更新书籍__35,\"12345\",\"张宇晨\",\"西北农林科技大学
\"");
//        mapper.zjupdate(12,"注解测试书1","9090","张宇晨","西北农林科技大学",
(float)78.2,"2020-10-12");
//        System.out.println("查询更新结果：");
//        System.out.println(mapper.getById(12).toString());
//        System.out.println("测试删除");
//        mapper.zjdelbyname("注解测试书1");

//动态SQL操作
//        System.out.println("执行动态SQL前的数据库状态");
//        List<HashMap<String, Object>> bl = mapper.listAll();
//        for(int i=0;i<bl.size();++i)
//        {
//            System.out.println(bl.get(i).toString());
//        }
//
//        Map map = new HashMap();
//        System.out.println("加入时间2020-10-1");
//        map.put("pdate","2020-10-1");
//        List<Book> lb = mapper.DtSQL(map);
//        for(int i=0;i<lb.size();++i)
//        {
//            System.out.println(lb.get(i).toString());
//        }
//        System.out.println("加入出版社模糊查询--清华");
//        map.put("publisher","清华");
//        lb = mapper.DtSQL(map);
//        for(int i=0;i<lb.size();++i)
//        {
//            System.out.println(lb.get(i).toString());
//        }
//        System.out.println("加入价格模糊查询--价格90");
//        map.put("price",90.0);
//        lb = mapper.DtSQL(map);
//        for(int i=0;i<lb.size();++i)
//        {
//            System.out.println(lb.get(i).toString());
//        }
        session.close();
    }
}
```