### JAVA第三阶段—DAY10-JAVA案例

1. If的使用

演示通过多个条件查询书籍的方法，分析其中的问题，带出where

<select id="selectBooksByBook" parameterType="Book" resultType="Book">  
 select \* from tb\_book where  
 <if test="bookName != null and bookName != ''">  
 book\_name = #{bookName}  
 </if>  
 <if test="author != null and author != ''">  
 and author = #{author}  
 </if>  
 <if test="price != null and price != ''">  
 and price = #{price}  
 </if>  
</select>

1. Where的使用

演示通过多个条件查询书籍的方法，讲解where的作用

<select id="selectBooksByBook" parameterType="Book" resultType="Book">  
 select \* from tb\_book  
 <where>  
 <if test="bookName != null and bookName != ''">  
 book\_name = #{bookName}  
 </if>  
 <if test="author != null and author != ''">  
 and author = #{author}  
 </if>  
 <if test="price != null and price != ''">  
 and price = #{price}  
 </if>  
 </where>  
</select>

1. set案例

<update id="updateBook" parameterType="Book">  
 update tb\_book  
 <set>  
 <if test="bookName != null and bookName != ''">  
 book\_name=#{bookName},  
 </if>  
 <if test="price != null and price != 0">  
 price=#{price},  
 </if>  
 <if test="typeId != null and typeId != 0">  
 type\_id=#{typeId},  
 </if>  
 <if test="author != null and author != ''">  
 author=#{author},  
 </if>  
 <if test="publishOrg != null and publishOrg != ''">  
 publish\_org=#{publishOrg},  
 </if>  
 <if test="publishTime != null and publishTime != ''">  
 publish\_time=#{publishTime},  
 </if>  
 <if test="state != null and state != 0">  
 state=#{state},  
 </if>  
 <if test="bookImage != null and bookImage != ''">  
 book\_image=#{bookImage},  
 </if>  
 </trim>  
 where id=#{id}  
</update>

1. trim案例

用trim代替set，主要是讲解trim几个属性的作用

<update id="updateBook" parameterType="Book">  
 update tb\_book  
 <trim prefix="set" suffixOverrides=",">  
 <if test="bookName != null and bookName != ''">  
 book\_name=#{bookName},  
 </if>  
 <if test="price != null and price != 0">  
 price=#{price},  
 </if>  
 <if test="typeId != null and typeId != 0">  
 type\_id=#{typeId},  
 </if>  
 <if test="author != null and author != ''">  
 author=#{author},  
 </if>  
 <if test="publishOrg != null and publishOrg != ''">  
 publish\_org=#{publishOrg},  
 </if>  
 <if test="publishTime != null and publishTime != ''">  
 publish\_time=#{publishTime},  
 </if>  
 <if test="state != null and state != 0">  
 state=#{state},  
 </if>  
 <if test="bookImage != null and bookImage != ''">  
 book\_image=#{bookImage},  
 </if>  
 </trim>  
 where id=#{id}  
</update>

1. foreach案例

<select id="selectBooksByBookNameList" parameterType="java.util.List" resultType="Book">  
 select \* from tb\_book  
 <where>  
 book\_name in  
 <foreach collection="bookNames" item="bookName" open="(" close=")" separator=",">  
 #{bookName}  
 </foreach>  
 </where>  
</select>

1. 一对多关联查询

通过书籍类型查询相关书籍集合

<resultMap id="bookTypeMap" type="BookType" >

<id property="id" column="id"></id>

<result property="type" column="type"></result>

<!--配置集合 property是集合属性 select是查询方法 javaType是集合类型

ofType是集合的数据类型 column外建列作为参数传入查询方法-->

<collection property="books" select="com.blb.bookms.dao.IBookDAO.selectBooksByTypeId"

javaType="java.util.List" ofType="Book" column="id">

</collection>

</resultMap>

<select id="selectBookTypeById" resultMap="bookTypeMap" parameterType="java.lang.Integer">

select \* from tb\_book\_type where id = #{typeId}

</select>

1. 一对一关联查询

通过书籍查询书籍类型

<resultMap id="bookMap" type="Book">

<id property="id" column="id"></id>

<result property="bookName" column="book\_name"></result>

<result property="price" column="price"></result>

<result property="typeId" column="type\_id"></result>

<result property="author" column="author"></result>

<result property="publishOrg" column="publish\_org"></result>

<result property="publishTime" column="publish\_time"></result>

<result property="state" column="state"></result>

<result property="bookImage" column="book\_image"></result>

<!--映射一对一 类型-->

<association property="bookType" javaType="BookType"

select="com.blb.bookms.dao.IBookTypeDAO.selectBookTypeById" column="type\_id">

</association>

</resultMap>

<select id="selectBookById" parameterType="java.lang.Integer" resultMap="bookMap">

select \* from tb\_book where id = #{id}

</select>

1. 延迟加载

先只配置日志输出，看没有延迟加载，调用对象的情况

配置延迟加载后，通过日志，看对象查询时按需调用的

<settings>

<setting name="mapUnderscoreToCamelCase" value="true"/>

<!-- 打印查询语句 -->

<setting name="logImpl" value="STDOUT\_LOGGING" />

<!--配置懒加载-->

<setting name="lazyLoadingEnabled" value="true"/>

<!--执行所有属性按需加载-->

<setting name="aggressiveLazyLoading" value="false"/>

</settings>

1. 缓存机制

一级缓存，多次查询同一个对象，查看日志sql调用情况

关闭SQLSession后，再多次查询同一个对象，查看日志sql调用情况

配置二级缓存，再重复上面步骤，最后总结一级缓存和二级缓存的区别