

MyBatis (二)

笔记本: mybatis

创建时间: 2018/12/23 19:47

更新时间: 2019/4/3 22:29

作者: qmys2018@163.com

动态SQL

■ <if>

```
select * from yg_ where 1=1
<if test="xm!=null">and xm_=#{xm}</if>
<if test="lxf!=null">and lxf_=#{lxf}</if>
```

■ <where>

```
select * from yg_
<where>
    <if test="xm!=null">and xm_=#{xm}</if>
    <if test="lxf!=null">and lxf_=#{lxf}</if>
</where>
```

■ <foreach>

```
select * from yg_ where bmbh_ in
<foreach item="item" index="index" collection="bmbh" open="(" separator=","
close=")">
    #{item}
</foreach>
```

SqlMapper.java

```
package mapper;
public interface SqlMapper {
    List<Map<String, Object>> find1(Map<String, Object> map);
    List<Map<String, Object>> find2(Map<String, Object> map);
    List<Map<String, Object>> find3(Map<String, Object> map);
}
```

UserMapper.xml

if

```
<select id="find1" resultType="map">
    select * from yg_ where 1=1
    <if test="xm!=null">
        and xm_=#{xm}
    </if>
    <if test="lxf!=null">
        and lxf_=#{lxf}
    </if>
</select>
```

UserMapper.xml

where

```
<select id="find2" resultType="map">
    select * from yg_
    <where>
        <if test="xm!=null">
            and xm_=#{xm}
        </if>
        <if test="lxf!=null">
            and lxf_=#{lxf}
        </if>
    </where>
```

```
        </if>
    </where>
</select>
```

UserMapper.xml

foreach

```
<select id="find3" resultType="map">
    select * from yg_
    <where>
        <if test="xm!=null">
            and xm_=#{xm}
        </if>
        <if test="lxfs!=null">
            and lxfs_=#{lxfs}
        </if>
        <if test="bmbh!=null">
            and bmbh_in
            <foreach item="item" index="index" collection="bmbh"
open="(" separator="," close=")">
                #{item}
            </foreach>
        </if>
    </where>
</select>
```

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">
<configuration>
    <environments default="mysql">
        <environment id="mysql">
            <transactionManager type="JDBC" />
            <dataSource type="POOLED">
                <property name="driver" value="com.mysql.jdbc.Driver" />
                <property name="url" value="jdbc:mysql://localhost:3306/mydb" />
                <property name="username" value="root" />
                <property name="password" value="123456" />
            </dataSource>
        </environment>
    </environments>
    <mapppers>
        <mapper resource="mapper/mysql/SqlMapper.xml" />
    </mapppers>
</configuration>
```

高级结果映射

■在MyBatis框架中，用于完成连接查询或嵌套查询的结果映射。

- 关联映射（多对一）

- <association>

- 关联的嵌套查询（多对一）

- <association select>

- 集合映射（一对多）

- <collection>

- 集合嵌套查询（一对多）

Bm.java

```
public class Bm {
    private String bh;
    private String mc;
    private String sjbh;
    private List<Yg> ygs;
    public String getBh() {return bh;}
    public void setBh(String bh) {this.bh = bh;}
    public String getMc() {return mc;}
    public void setMc(String mc) {this.mc = mc;}
    public String getSjbh() {return sjbh;}
    public void setSjbh(String sjbh) {this.sjbh = sjbh;}
    public List<Yg> getYgs() {return ygs;}
    public void setYgs(List<Yg> ygs) {this.ygs = ygs;}
}
```

Yg.java

```
public class Yg {
    private String bh;
    private String xm;
    private String bmbh;
    private String lxfs;
    private Bm bm;
    public String getBh() {return bh;}
    public void setBh(String bh) {this.bh = bh;}
    public String getXm() {return xm;}
    public void setXm(String xm) {this.xm = xm;}
    public String getBmbh() {return bmbh;}
    public void setBmbh(String bmbh) {this.bmbh = bmbh;}
    public String getLxfs() {return lxfs;}
    public void setLxfs(String lxfs) {this.lxfs = lxfs;}
    public Bm getBm() {return bm;}
    public void setBm(Bm bm) {this.bm = bm;}
}
```

Mapper.java

```
public interface BmMapper1 {  
    List<Bm> find();  
}
```

```
public interface BmMapper2 {  
    List<Bm> find();  
}
```

```
public interface YgMapper1 {  
    List<Yg> find();  
}
```

```
public interface YgMapper2 {  
    List<Yg> find();  
}
```

关联映射 (YgMapper1.xml)

```
<mapper namespace="mapper.YgMapper1">  
    <resultMap id="YgMap" type="model.Yg">  
        <id column="BH_" property="bh" jdbcType="VARCHAR" />  
        <result column="XM_" property="xm" jdbcType="VARCHAR" />  
        <result column="BMBH_" property="bmbh" jdbcType="VARCHAR" />  
        <result column="LXFS_" property="lxfs" jdbcType="VARCHAR" />  
        <association property="bm" javaType="model.Bm">  
            <id column="BM_BH" property="bh" jdbcType="VARCHAR" />  
            <result column="MC_" property="mc" jdbcType="VARCHAR" />  
        </association>  
    </resultMap>  
    <select id="find" resultMap="YgMap">  
        select t1.*,t2.bh_ bm_bh,t2.mc_  
        from yg_ t1,bm_ t2  
        where t1.bmbh_=t2.bh_  
    </select>  
</mapper>
```

关联映射 (Demo0204.java)

```
InputStream in = Resources.getResourceAsStream("mybatis-config.xml");  
SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);  
SqlSession session = factory.openSession();  
YgMapper1 ygMapper1 = session.getMapper(YgMapper1.class);  
List<Yg> list = ygMapper1.find();  
for (Yg yg : list) {  
    System.out.print(yg.getBh() + "\t");  
    System.out.print(yg.getXm() + "\t");  
    System.out.print(yg.getBm().getBh() + "\t");  
    System.out.println(yg.getBm().getMc());  
}  
session.close();
```

关联嵌套查询 (YgMapper2.xml)

```
<mapper namespace="mapper.YgMapper2">
  <resultMap id="YgMap" type="model.Yg">
    <id column="BH_" property="bh" jdbcType="VARCHAR" />
    <result column="XM_" property="xm" jdbcType="VARCHAR" />
    <result column="BMBH_" property="bmbh" jdbcType="VARCHAR" />
    <result column="LXFS_" property="lxfs" jdbcType="VARCHAR" />
    <association column="BMBH_" property="bm" select="findBm" />
  </resultMap>
  <resultMap id="BmMap" type="model.Bm">
    <id column="BH_" property="bh" jdbcType="VARCHAR" />
    <result column="MC_" property="mc" jdbcType="VARCHAR" />
  </resultMap>
  <select id="find" resultMap="YgMap">
    select * from yg_
  </select>
  <select id="findBm" resultMap="BmMap">
    select * from bm_ where bh_=#{bmbh}
  </select>
</mapper>
```

关联嵌套查询 (Demo0205.java)

```
InputStream in = Resources.getResourceAsStream("mybatis-config.xml");
SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);
SqlSession session = factory.openSession();
YgMapper2 ygMapper2 = session.getMapper(YgMapper2.class);
List<Yg> list = ygMapper2.find();
for (Yg yg : list) {
    System.out.print(yg.getBh() + "\t");
    System.out.print(yg.getXm() + "\t");
    System.out.print(yg.getBm().getBh() + "\t");
    System.out.println(yg.getBm().getMc());
}
session.close();
```

集合映射 (BmMapper1.xml)

```
<mapper namespace="mapper.BmMapper1">
  <resultMap id="BmMap" type="model.Bm">
    <id column="BH_" property="bh" jdbcType="VARCHAR" />
    <result column="MC_" property="mc" jdbcType="VARCHAR" />
    <result column="SJBH_" property="sjbh" jdbcType="VARCHAR" />
    <collection property="ygs" ofType="model.Yg">
      <id column="YG_BH_" property="bh" jdbcType="VARCHAR" />
      <result column="XM_" property="xm" jdbcType="VARCHAR" />
    </collection>
  </resultMap>
  <select id="find" resultMap="BmMap">
    select t1.*,t2.bh_ yg_bh,t2.xm_
    from bm_ t1,yg_ t2
    where t1.bh_=t2.bmbh_
  </select>
</mapper>
```

```
</select>
</mapper>
```

集合映射 (Demo0206.java)

```
InputStream in = Resources.getResourceAsStream("mybatis-config.xml");
SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);
SqlSession session = factory.openSession();
BmMapper1 bmMapper1 = session.getMapper(BmMapper1.class);
List<Bm> list = bmMapper1.find();
for (Bm bm : list) {
    System.out.println(bm.getBh() + "\t" + bm.getMc());
    for (Yg yg : bm.getYgs()) {
        System.out.println(yg.getBh() + "\t" + yg.getXm());
    }
}
session.close();
```

集合嵌套查询 (BmMapper2.xml)

```
<mapper namespace="mapper.BmMapper2">
    <resultMap id="BmMap" type="model.Bm">
        <id column="BH_" property="bh" jdbcType="VARCHAR" />
        <result column="MC_" property="mc" jdbcType="VARCHAR" />
        <result column="SJBH_" property="sjbh" jdbcType="VARCHAR" />
        <collection column="BH_" property="ygs" select="findYg" />
    </resultMap>
    <resultMap id="YgMap" type="model.Yg">
        <id column="BH_" property="bh" jdbcType="VARCHAR" />
        <result column="XM_" property="xm" jdbcType="VARCHAR" />
    </resultMap>
    <select id="find" resultMap="BmMap">
        select * from bm_
    </select>
    <select id="findYg" resultMap="YgMap">
        select * from yg_ where bmbh_=#{bh}
    </select>
</mapper>
```

集合嵌套查询 (Demo0207.java)

```
InputStream in = Resources.getResourceAsStream("mybatis-config.xml");
SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);
SqlSession session = factory.openSession();
BmMapper2 bmMapper2 = session.getMapper(BmMapper2.class);
List<Bm> list = bmMapper2.find();
for (Bm bm : list) {
    System.out.println(bm.getBh() + "\t" + bm.getMc());
    for (Yg yg : bm.getYgs()) {
        System.out.println(yg.getBh() + "\t" + yg.getXm());
    }
}
session.close();
```

MybatisGenerator

■ **MyBatisGenerator (MBG)** 是一个Mybatis的代码生成器。它可以内省数据库的表，然后生成可以用来访问表的基础对象。这样和数据库表进行交互时不需要创建对象和配置文件。MBG的解决了对数据库操作有最大影响的一些简单的CRUD（插入，查询，更新，删除）操作。

- **insert** (插入)
 - **update by primary key** (根据主键更新记录)
 - **update by example** (根据条件更新记录)
 - **delete by primary key** (根据主键删除记录)
 - **delete by example** (根据条件删除记录)
 - **select by primary key** (根据主键查询记录)
 - **select by example** (根据条件查询记录集)
- **count by example** (根据条件查询记录总数)

MybatisGenerator

- **mybatis-generator-core-1.3.3.jar**
- **generator.xml**
- **Generator.java**

generator.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE generatorConfiguration
PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
"http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
<generatorConfiguration>
  <context id="mybatis3">
    <commentGenerator>
      <property name="suppressAllComments" value="true" />
    </commentGenerator>
    <jdbcConnection driverClass="" connectionURL="" userId="" password="" />
    <javaModelGenerator targetPackage="" targetProject="d:\MBG" />
    <sqlMapGenerator targetPackage="" targetProject="d:\MBG" />
    <javaClientGenerator type="XMLMAPPER" targetPackage=""
targetProject="d:\MBG" />
    <table schema="" tableName="%">
      <property name="ignoreQualifiersAtRuntime" value="true" />
    </table>
  </context>
</generatorConfiguration>
```

Generator.java

```
File directory = new File("d:/MBG");
if (!directory.exists()) {
    directory.mkdir();
}
ConfigurationParser cp = new ConfigurationParser(null);
File file = new File(Generator.class.getResource("generator.xml").toURI());
Configuration config = cp.parseConfiguration(file);
DefaultShellCallback callback = new DefaultShellCallback(true);
MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config, callback, null);
myBatisGenerator.generate(null);
```

总结

■动态SQL

■高级结果映射

- 准备工作
- 关联映射
- 关联嵌套查询
- 集合映射
- 集合嵌套查询

· ■MybatisGenerator