

## 插入之后获取自增id

在插入操作完成之前或之后，可以配置<selectKey>标签获得生成的主键的值，获得插入之前还是之后的值，可以通过配置order属性来指定。

LAST\_INSERT\_ID：该函数是mysql的函数，获取自增主键的ID，它必须配合insert语句一起使用

这里因为使用的MySQL数据库，所以配置的order是after，如果使用的是Oracle数据库，就该配置成before，这是因为MySQL和Oracle生成主键不一样的原因

```
<insert id="add" parameterType="com.wxian.pojo.CheckGroup">
    <!-- 通过mybatis框架提供的selectKey标签获得自增产生的id
    keyProperty="id" 使用selectKey标签将执行完插入之后，
    自增的id值赋给checkgroup对象的id属性，因此这里可以使用getId得到id值
    AFTER在insert语句之后执行-->
    <selectKey resultType="java.lang.Integer" order="AFTER" keyProperty="id">
        select LAST_INSERT_ID()
    </selectKey>
    insert into t_checkgroup(code,name,sex,helpCode,remark,attention)
    values
    (#{code},#{name},#{sex},#{helpCode},#{remark},#{attention})
</insert>
```

## @Override

```
public void add(CheckGroup checkGroup, Integer[] checkitemIds) {
    // 新增检查组，操作t_checkgroup表
    checkGroupDao.add(checkGroup);
    // 设置检查组和检查项的关联关系，操作t_checkgroup_checkitem表
    // keyProperty="id" 使用selectKey标签将执行完插入之后，自增的id值赋给
    checkgroup对象的id属性，
    // 因此这里可以使用getId得到id值
    Integer checkGroupId = checkGroup.getId();
    if (checkitemIds != null && checkitemIds.length > 0){
        Map<String, Integer> map = null;
        for (Integer checkitemId : checkitemIds) {
            map = new HashMap<>();
            map.put("checkgroupId", checkGroupId); // 这里设置的key在mapper.xml
            文件中取值#{ }时要用到
            map.put("checkitemId", checkitemId);
            checkGroupDao.addCheckGroupAndCheckItem(map);
        }
    }
}
```

另一种写法：

```
<insert id="addUser" useGeneratedKeys="true" keyProperty="id">
  insert into user values (null,#{name},#{time},#{address})
</insert>
```

useGeneratedKeys: 开启自增长映射

keyProperty: 指定id所对应对象中的属性名

当执行完saveUser()方法后，其返回值依然是执行sql影响的行数，并不是要获取的自增ID！Mybatis会自动将返回的主键值赋值给对象User的属性id，因此可以通过属性的get方法获得插入的主键值： System.out.println(User.getId());