

一.懒加载

什么是懒加载？

对于页面有很多静态资源的情况下（比如网商购物页面），为了节省用户流量和提高页面性能，可以在**用户浏览到当前资源的时候，再对资源进行请求和加载。**

以下演示的时局部懒加载：

```
1  1. 全局配置,在mybatis配置文件中,lazyLoadingEnabled=true
2  2. 局部配合,在没有batis映射文件的resultMap中,fetchType="lazy"
3
4  配合resultMap中的collection,association,select,column
5
1 <mapper namespace="com.lanou.dao.BManDao">
2   <!-- 当前的mapper开启二级缓存-->
3   <cache/>
4
5   <resultMap id="BMan" type="com.lanou.pojo.BMan">
6     <id property="id" column="id"/>
7     <result property="name" column="name"/>
8     <result property="age" column="age"/>
9     <collection property="womanList" ofType="com.lanou.pojo.BWoman"
10      select="selectWoman" column="id"
11      fetchType="lazy">
12
13    </collection>
14  </resultMap>
15  <select id="selectWoman" resultType="com.lanou.pojo.BWoman">
16    select * from b_woman where man_id=#{id}
17  </select>
18
19  <select id="selectAllBMan" resultMap="BMan">
20    select * from b_man
21  </select>
22
23  <select id="selectById" resultType="com.lanou.pojo.BMan">
24    select * from b_man where id=#{id}
```

```
25 </select>
26 </mapper>
```

二.一级缓存,二级缓存

mybatis的缓存级别

1.一级缓存,sqlSession级别,默认开启,查询数据时,会先去一级缓存中获取数据,如果没有再去数据库中查(在一个sqlsession)

2.二级缓存,mapper级别,默认没有开启

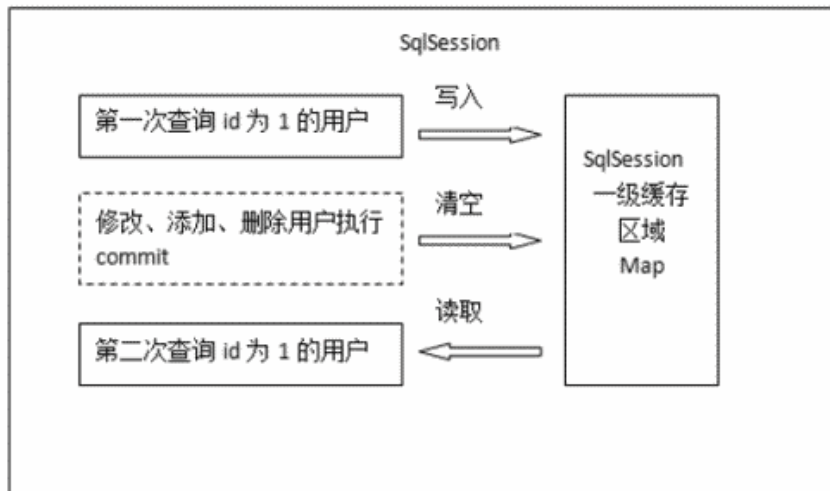
开启在MyBatis的配置中,添加cacheEnabled=true,在指定的映射文件中添加<cache/>

sqlSession相当于一个连接,二级缓存,将数据序列化

```
1 <!--开启二级缓存-->
2 <property name="cacheEnabled" value="true"/>
```

测试一级缓存:

11.2. 一级缓存原理



每次查询会先去缓存中找，如果找不到，再去数据库查询，然后把结果写到缓存中。Mybatis 的内部缓存使用一个 HashMap，key 为 hashCode+statementId+sql 语句。Value 为查询出来的结果集映射成的 java 对象。

SqlSession 执行 insert、update、delete 等操作 commit 后会清空该 SQLSession 缓存。

```
1 @Test
2 void test5() {
3     //用户查数据库,缓存只有一个session
4     SqlSession sqlSession = sqlSessionFactory.openSession();
5     BManDao bManDao = sqlSession.getMapper(BManDao.class);
6     BMan bMan = bManDao.selectByPrimaryKey(1);
7     System.out.println(bMan);
8
9     //sqlSession关闭,无法操作数据
10    //sqlSession.close();
11
12    //清除缓存
13    //sqlSession.clearCache();
14
15    //当执行增删改操作后,清空缓存
```

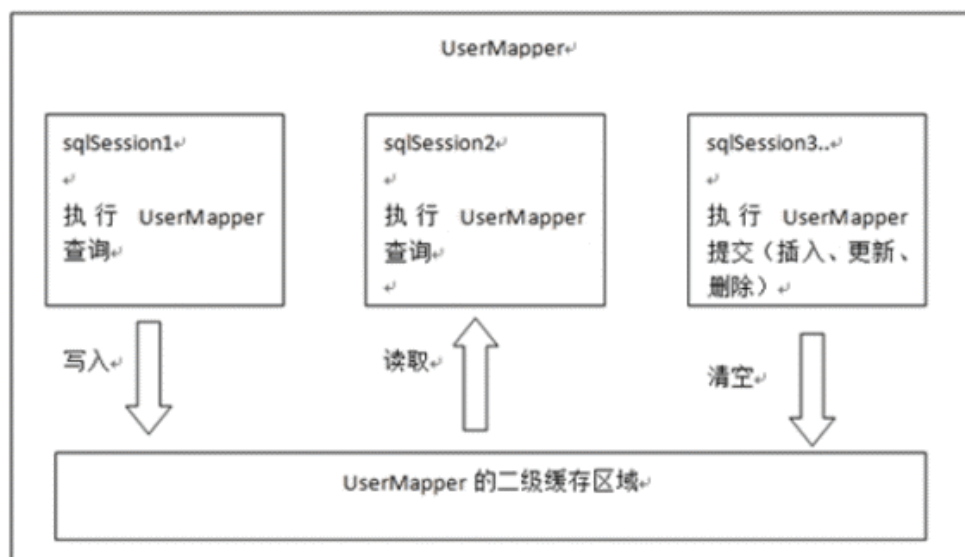
```

16  BMan b = new BMan();
17  b.setId(2);
18  b.setName("小明");
19
20  Example example = new Example(BMan.class);
21  Example.Criteria criteria = example.createCriteria();
22  criteria.andEqualTo("id",2);
23  bManDao.updateByExampleSelective(b,example);
24
25  System.out.println("????????");
26
27  sqlSession = sessionFactory.openSession();
28  bManDao = sqlSession.getMapper(BManDao.class);
29  BMan bMan1 = bManDao.selectByPrimaryKey(1);
30  System.out.println(bMan1);
31  }

```

测试二级缓存:

12.2. 二级缓存原理



二级缓存是 mapper 级别的。Mybatis 默认是没有开启二级缓存。

如果二级缓存开启, 首先从二级缓存查询数据, 如果二级缓存有则从二级缓存中获取数据, 如果二级缓存没有, 从一级缓存找是否有缓存数据, 如果一级缓存没有, 查询数据库。

12.1. 二级缓存概述

Mybatis 的二级缓存是指 mapper 映射文件。二级缓存的作用域是同一个 namespace 下的 mapper 映射文件内容，多个 `SqlSession` 共享。Mybatis 需要手动启动二级缓存。

在同一个 namespace 下的 mapper 文件中，执行相同的查询 SQL，第一次会去查询数据库，并写到缓存中；第二次直接从缓存中取。当执行 SQL 时两次查询中间发生了增删改操作，则二级缓存清空。

```
1 @Test
2 void test6() {
3
4     SqlSession sqlSession = sqlSessionFactory.openSession();
5     BManDao bManDao = sqlSession.getMapper(BManDao.class);
6     BMan bMan = bManDao.selectById(1);
7     System.out.println(bMan);
8
9     //没有关闭,数据存在一级缓存,关闭后,存入二级缓存
10    sqlSession.close();
11
12    System.out.println("//////////");
13
14    SqlSession sqlSession1 = sqlSessionFactory.openSession();
15    BManDao bManDao1 = sqlSession1.getMapper(BManDao.class);
16    BMan bMan1 = bManDao1.selectById(1);
17    System.out.println(bMan1);
18 }
```