Java训练营

阶段一模块一：

一、简单题

1、Mybatis动态sql是做什么的？都有哪些动态sql？简述一下动态sql的执行原理？
动态SQL用于优化SQL编写，使得SQL可以根据条件自动进行拼接。例如<if><where><foreach>。在解析XML配置文件阶段，mybatis会根据SQL标签，判断是否动态SQL，如果是动态SQL则生成DynamicSqlSource对象存于MappedStatement对象，待执行阶段则根据传入的参数生成动态SQL的最终SQL语句

2、Mybatis是否支持延迟加载？如果支持，它的实现原理是什么？
支持延迟加载。实现原理是通过动态代理技术，在执行对象查询时首先不进行关联对象的查询，返回此对象的代理对象，当代理对象查询关联对象时，再执行查询操作

3、Mybatis都有哪些Executor执行器？它们之间的区别是什么？
SimpleExecutor：每次执行SQL都会创建PrepareStatement对象
ReuseExecutor：每次执行SQL时会先判断对象的成员变量statementMap是否存在相同sql的PrepareStatement对象，如果存在则复用，如果不存在则新建，并把新建的PrepareStatement对象存储起来留后续使用
BatchExecutor：执行更新操作时，通过PrepareStatement对象的addBatch添加到批处理中，在后续执行查询或者提交操作时，再循环执行批量更新操作
CachingExecutor：用于二级缓存中，内部包含一个委托对象，符合条件时首先在缓存中查询数据，查询不到则交给委托对象做查询，查询出的结果再缓存起来

4、简述下Mybatis的一级、二级缓存（分别从存储结构、范围、失效场景。三个方面来作答）？
一级缓存存储于BaseExecutor中的localCache，默认实现类是PerpetualCache，是通过HashMap进行存储。默认一级缓存是开启的。作用范围是同一个SqlSession，当对数据进行更新或者提交等操作时，会清空缓存。
二级缓存存储于Configuration中的MappedStatement，默认底层实现类也是PerpetualCache，并且通过LruCache、LoggingCache、SynchronizedCache等进行包装，最终也是存储于HashMap结构中，但默认情况下会进行序列化，储存的是对象的字节数组。作用范围是同一个命名空间，当执行update相关操作时二级缓存中的临时缓存(TransactionalCache.entriesToAddOnCommit)会清空，同时打上clearOnCommit标记，当执行commit时，二级缓存清空同时临时缓存写入二级缓存中

5、简述Mybatis的插件运行原理，以及如何编写一个插件？
Mybatis中的四大对象Executor、ParameterHandler、ResultSetHandler、StatementHandler在生成的时候并不是直接生成的，而是通过拦截器的包装，返回一个动态代理对象，当这些对象的被代理方法执行的时候会，生成代理对象时传入的InvocationHandler对象的invoke会被执行，在invoke方法中判断拦截器定义的拦截方法签名

包含当前方法签名时，会执行拦截器的拦截方法。
自定义插件：
a.实现接口Interceptor，通过注解定义需要拦截的四大对象方法
b.在Mybatis的配置文件中配置

二、编程题
请完善自定义持久层框架IPersistence，在现有代码基础上添加、修改及删除功能。【需要采用getMapper方式】

修改点：
1.增加标签解析
2.增加事物控制
3.增加插入/修改/删除功能

执行环境：
MySQL：8.0.18
JDK：pom.xml中定义1.8
Maven：3.6.1

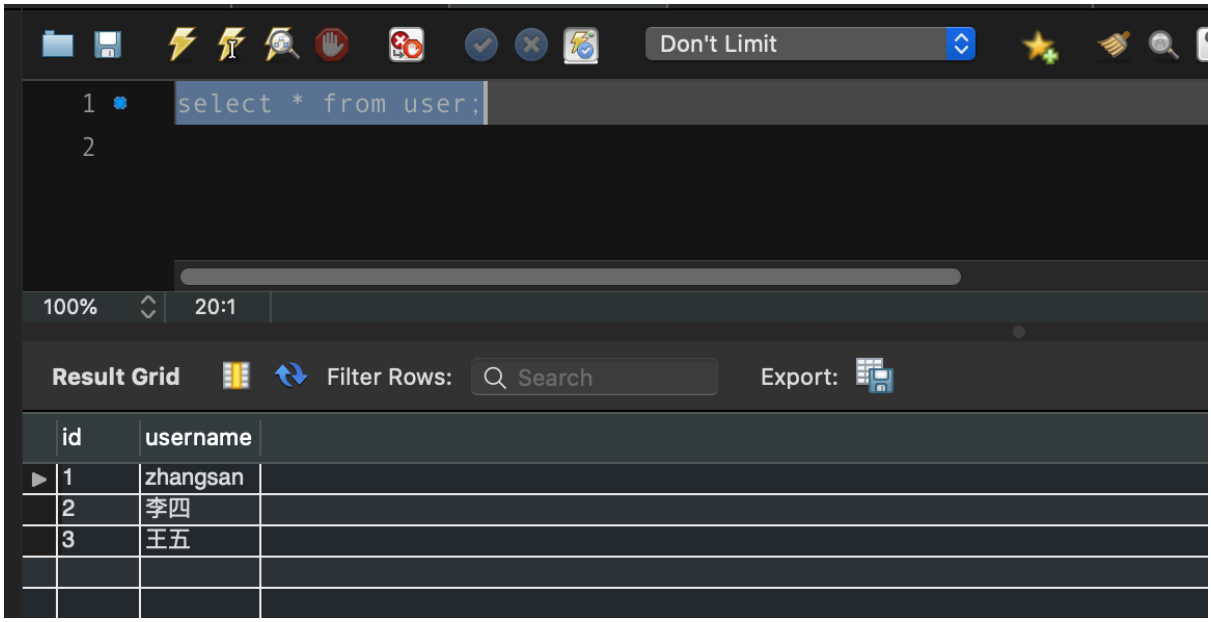
执行截图：
新增：

```java
//新增
User user = new User();
user.setId(3);
user.setUsername("王五");
int count = userDao.insertUser(user);

//修改
//  user.setUsername("王六");
//  int count = userDao.updateUser(user);

//删除
//  int count = userDao.deleteUser(3);

System.out.println("更新数据量: " + count);
sqlSession.commit();
sqlSession.close();
```

IPersistenceTest › test()

Tests passed: 1 of 1 test – 1 s 733 ms

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home/bin/java ...
objc[5679]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_45
  .jdk/Contents/Home/bin/java (0x10b5714c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_45
  .jdk/Contents/Home/jre/lib/libinstrument.dylib (0x10b5e74e0). One of the two will be used. Which one is unde
log4j:WARN No appenders could be found for logger (com.mchange.v2.log.MLog).
log4j:WARN Please initialize the log4j system properly.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'
  automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
更新数据量: 1
```



修改：

```
44          User user = new User();
45          user.setId(3);
46  //        user.setUsername("王五");
47  //        int count = userDao.insertUser(user);
48
49          //修改
50          user.setUsername("王六");
51          int count = userDao.updateUser(user);
52
53          //删除
54  //        int count = userDao.deleteUser(3);
55
56          System.out.println("更新数据量: " + count);
57          sqlSession.commit();
58          sqlSession.close();
59
60
```

IPersistenceTest › test()

Tests passed: 1 of 1 test – 1 s 537 ms

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home/bin/java ...
objc[5776]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_45
  .jdk/Contents/Home/bin/java (0x1064634c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_45
  .jdk/Contents/Home/jre/lib/libinstrument.dylib (0x1064d94e0). One of the two will be used. Which one is
log4j:WARN No appenders could be found for logger (com.mchange.v2.log.MLog).
log4j:WARN Please initialize the log4j system properly.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Dri
  automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
更新数据量: 1
```

```
1  select * from user;
2
```

100% 20:1

Result Grid Filter Rows: Search Export:

id	username
1	zhangsan
2	李四
3	王六

删除：

```
51
52          //删除
53          int count = userDao.deleteUser( id: 3);
54
55          System.out.println("更新数据量: " + count);
56          sqlSession.commit();
57          sqlSession.close();
58
59
```

IPersistenceTest › test()

Tests passed: 1 of 1 test – 1 s 532 ms

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home/bin/java ...
objc[5832]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_45
  .jdk/Contents/Home/bin/java (0x10c70f4c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_45
  .jdk/Contents/Home/jre/lib/libinstrument.dylib (0x10c7854e0). One of the two will be used. Which one
log4j:WARN No appenders could be found for logger (com.mchange.v2.log.MLog).
log4j:WARN Please initialize the log4j system properly.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.D
  automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
更新数据量: 1
```