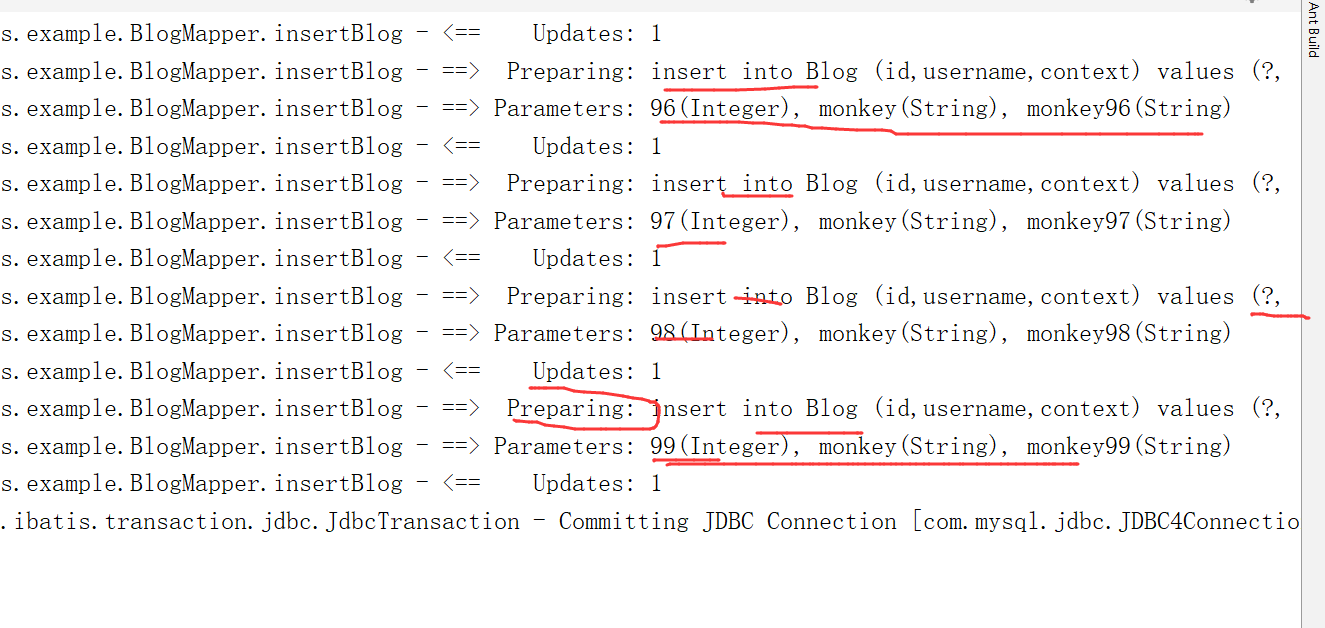
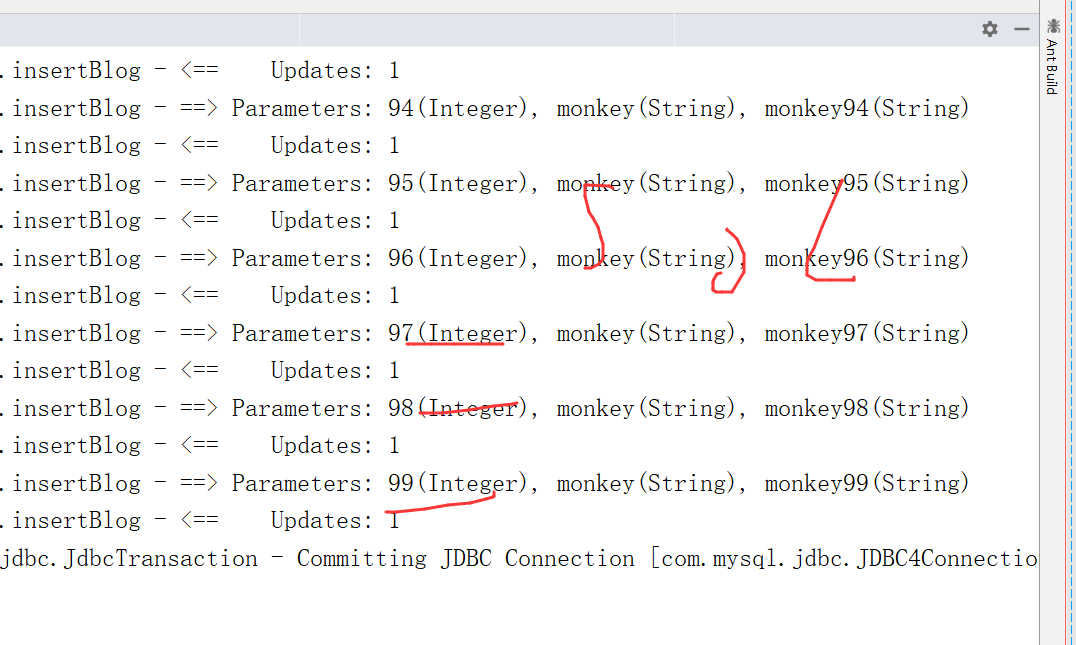
# Mybatis执行器：

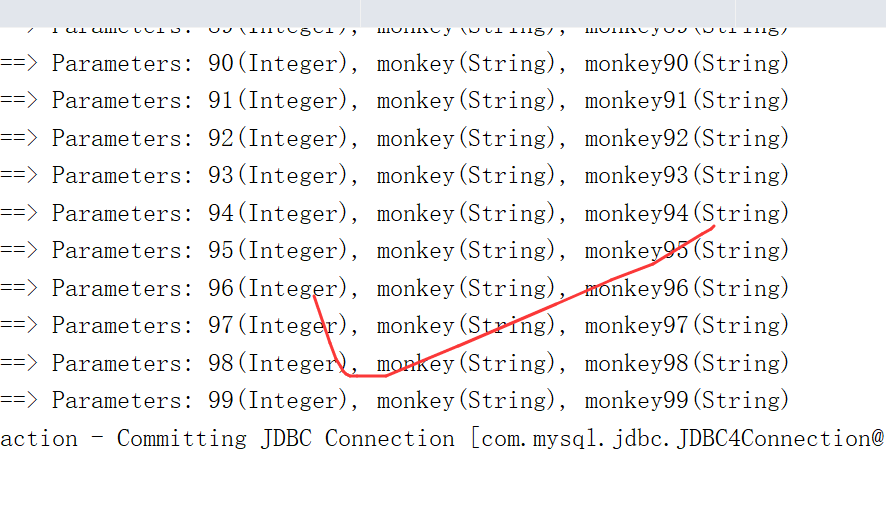
ExecutorType.SIMPLE：这个执行器类型不做特殊的事情。它为每个语句的每次执行创建一个新的预处理语句。



ExecutorType.REUSE：这个执行器类型会复用预处理语句。



ExecutorType.BATCH：这个执行器会批量执行所有更新语句，也就是jdbc addBatch API的facade模式。



所以这三种类型的执行器可以说时应用于不同的负载场景下，除了SIMPLE类型外，另外两种要求对系统有较好的架构设计，当然也提供了更多的回报。

### 执行期主要类总结

mybatis在执行期间，主要有四大核心接口对象：

一、执行器Executor，执行器负责整个SQL执行过程的总体控制。

二、参数处理器ParameterHandler，参数处理器负责PreparedStatement入参的具体设置。

三、语句处理器StatementHandler，语句处理器负责和JDBC层具体交互，包括prepare语句，执行语句，以及调用ParameterHandler.parameterize()设置参数。

四、结果集处理器ResultSetHandler，结果处理器负责将JDBC查询结果映射到java对象。

# Mybatis处理器：

### 参数处理器ParameterHandler

DefaultParameterHandler为具体实现类

org.apache.ibatis.scripting.defaults.DefaultParameterHandler#setParameters

### 语句处理器StatementHandler

比如设置超时时间、结果集每次提取大小等操作

org.apache.ibatis.executor.statement.BaseStatementHandler#prepare

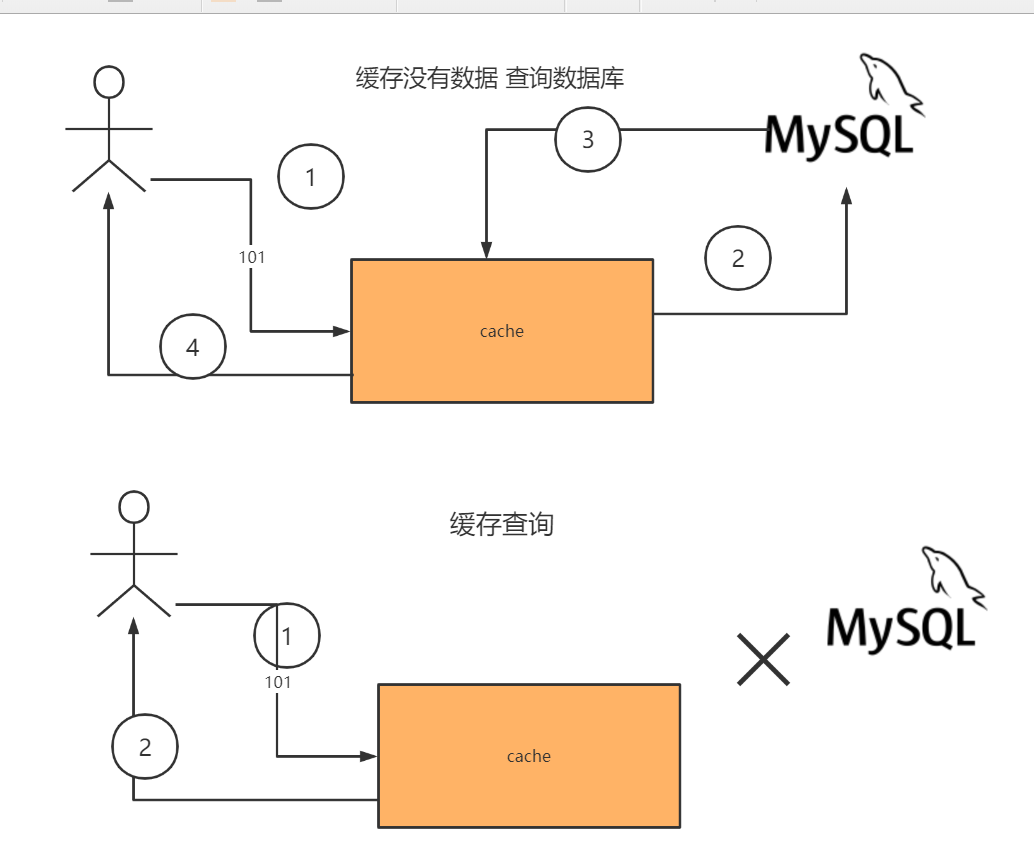
### 结果集处理器ResultSetHandler

　结果集处理器,顾名知义,就是用了对查询结果集进行处理的,目标是将JDBC结果集映射为业务对象

接口中定义的三个接口分别用于处理常规查询的结果集,游标查询的结果集以及存储过程调用的出参设置

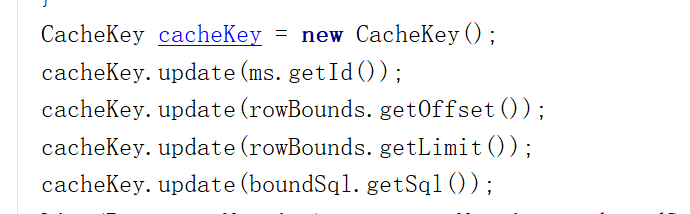
org.apache.ibatis.executor.resultset.DefaultResultSetHandler

# Mybatis缓存：



**一级缓存：**

mybatis提供了基本实现org.apache.ibatis.cache.impl.PerpetualCache，内部采用原始HashMap实现。第二个需要知道的方面是mybatis有一级缓存和二级缓存。一级缓存是SqlSession级别的缓存，不同SqlSession之间的缓存数据区域（HashMap）是互相不影响，MyBatis默认支持一级缓存，不需要任何的配置，默认情况下(一级缓存的有效范围可通过参数localCacheScope参数修改，取值为SESSION或者STATEMENT)，在一个SqlSession的查询期间，只要没有发生commit/rollback或者调用close()方法，那么mybatis就会先根据当前执行语句的CacheKey到一级缓存中查找，如果找到了就直接返回，不到数据库中执行。其实现在代码BaseExecutor.query()中



**二级缓存：**

二级缓存是mapper级别的缓存，多个SqlSession去操作同一个mapper的sql语句，多个SqlSession可以共用二级缓存，二级缓存是跨SqlSession。二级缓存默认不启用，需要通过在Mapper中明确设置cache，它的实现在CachingExecutor的query()方法中，如下所示：

|  |
| --- |
| @Override  public <E> List<E> query(MappedStatement ms, Object parameterObject, RowBounds rowBounds, ResultHandler resultHandler, CacheKey key, BoundSql boundSql)  throws SQLException {  Cache cache = ms.getCache();  if (cache != null) {  flushCacheIfRequired(ms);  if (ms.isUseCache() && resultHandler == null) {  ensureNoOutParams(ms, parameterObject, boundSql);  @SuppressWarnings("unchecked")  // 如果二级缓存中找到了记录就直接返回,否则到DB查询后进行缓存  List<E> list = (List<E>) tcm.getObject(cache, key);  if (list == null) {  list = delegate.<E> query(ms, parameterObject, rowBounds, resultHandler, key, boundSql);  tcm.putObject(cache, key, list); // issue #578 and #116  }  return list;  }  }  return delegate.<E> query(ms, parameterObject, rowBounds, resultHandler, key, boundSql);  } |

## Mybatis-Spring整合:

<http://mybatis.org/spring/zh/index.html>