动态sql是mybatis的主要特性之一，在mapper中定义的参数传到xml中之后，在查询之前mybatis会对其进行动态解析。

# 执行流程

## 创建SqlSessionFactory

SqlSessionFactoryBuilder去读取mybatis配置文件。然后build一个DefaultSqlSessionFactory。

## 返回SqlSession的实现类DefaultSqlSession对象

通常一系列的openSession方法最终都会调用本方法---openSessionFromDataSource，这个方法返回DefaultSqlSession。

## 返回一个MapperProxy的代理对象

通过MapperProxy动态代理Dao层，获取MapperProxy对象过程：

1. 通过SqlSession从Configuration中获取，
2. Configuration甩给MapperRegistry，
3. MapperRegistry又调用MapperProxyFactory

## 执行查询Excutor

拿到MapperProxy，每个MapperProxy对应一个dao接口，他直接调用mapperMethod的execute方法，在这个方法中判断CRUD类型，然后根据类型选择到底执行sqlsession中的哪个方法，又回到了sqlsession。

然后通过一层一层的调用来到doQuery方法然后将查询结果放到缓存中一份，这里看一下SimpleExecutor中的doQuery方法。

这个方法初始化了三个组件：

PreparedStatementHandler：

调用PreparedStatement来执行SQL语句，这样在第一次执行的时候会进行预编译，接下来执行相同的SQL语句时会提高数据库性能。

parameterHandler：

Statement使用prepare方法之后就会使用这个类设置参数。

resultSetHandler：

查询的结果集处理。

# MyBatis的缓存机制

Mybatis存在两个级别的缓存，会话级别(sqlsession一级缓存)，会话工厂级别(sqlsessionfactory二级缓存)。

## 一级缓存

一级缓存默认时开启的，如果只开启了一级缓存的话，首先会进入到BaseExecutor的query方法，这个方法中会根据传入的参数生成CacheKey，如果下列的值相同就会认为时相同的SQL：statementId、Offset、Limmit、Sql、Params。然后根据这个key去缓存中查询数据，如果查到的数据为空则去数据库中查找。向缓存中添加一份数据。

一级缓存又叫做本地缓存。

一级缓存又分为两种级别：Session和Statement。

1. session，针对同一个会话sqlsession中，执行多次条件相同的同一个sql，那么会共 享这一缓存。
2. Statement，缓存只针对当前执行的这一statement有效。

有多个sqlsession或者分布式的环境下，数据库写操作会引起脏数据，建议设定缓存级别为statement。相当于不实用一级缓存

一级缓存底层是HashMap实现的，会不会造成数据溢出？

1. 一般而言，使用一个sqlsession对象执行的操作不会太多。执行完就会消亡。
2. 对于一个sqlsssion而言，只要执行update操作(update，insert，delete)都会将 sqlsession中对应的一级缓存清空。
3. 可以手动的释放sqlsession中的缓存。

## 二级缓存

二级缓存的作用域是全局，也就是namespace级别的，并且是在sqlsession关闭或提交之后才会生效。

二级缓存的配置：

1. 全局配置文件中的setting的cacheEnable需要设置为true(不配置也行，默认为true)
2. Mapper配置文件中需要加入<cache>节点
3. Mapper配置文件中的select、节点需要加上属性useCache为true(不配置也行，默 认为true)

# Mybatis一级缓存在分布式下导致脏数据问题

分布式环境下，mybatis的一级缓存无法集中统一管理，也就是说无法搭建分布式缓存。再多个sqlsession环境下，虽然可以使用二级缓存，但是毕竟是本地存储，无法适应分布式的需求。

1. 设置mybatis一级缓存为statement级别的，因为在查询方法执行完毕后，会判断一级缓存的级别，如果是statement则会清空缓存，这样就相当于不使用一级缓存。
2. 创建sqlsession的时候，每次查询完毕后调用clearCache方法也可以清空缓存。
3. 在select标签里面设置flushCache=true
4. 使用EnCache等分布式缓存框架（redis）

https://blog.csdn.net/qq\_36236890/article/details/83964399

# 组件层次划分

https://baijiahao.baidu.com/s?id=1618660179878355395&wfr=spider&for=pc



# 核心流程



# #和$

#把传入的数据都当成一个字符串，会对传入的数据都加上一个双引号。

如果传入的值为111，那么就会解析称为order by “111”，如果传入的是id，解析成 order by “id”。

$将传入的值直接显示生成sql

Mybatis排序时使用order by动态参数时需要注意用$而不是#

如果用#就是 order by “id”，这是错误的语法，不应该有双引号

#号能够防止sql注入，因为对sql做了预编译处理。

$是未经过预编译的

#用于CRUD，${}用户模糊查询