

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？ if,where 等，类似脚本语言，可以动态拼接 sql。 第一部分：在启动加载解析 xml 配置文件的时候进行解析，根据关键标签封装成对应的 handler 处理对象，封装成 sqlSource 对象存在 mappedStatement。 调用流程： I、SqlSessionFactoryBuilder 对 builder 对象的时候，调用 XMLConfigBuilder 解析 sqlMapConfig.xml 配置文件，在解析过程中使用到了私有的 mapperElement(XNode parent)方法 II、上面方法中通过构建 XMLMapperBuilder,获取到所有的配置 mapper 配置，在调用 private void configurationElement(XNode context)方法进行解析 mapper.xml,通过 void buildStatementFromContext(List list, String requiredDatabaseId)方法解析 mapper.xml 内的每一个标签 III、循环中构建 XMLStatementBuilder 对象，调用 parseStatementNode()方法来封装 mappedStatment 对象， IIII、在过程中需要构建 sqlSource 对象，通过 XMLLanguageDriver 对象进行处理，在 XMLLanguageDriver 中构建解析动态标签对象 XMLScriptBuilder 第二部分：在执行过程中获取 sqlSource 中获取 bondSql 对象时，执行相应的标签 handler 调用查询执行到 BaseExecutor 的 query 方法时候会去 getBoundSql 并且将参数传进去， 在 sqlSource 接口 DynamicSqlSource 实现类中，调用 getBoundSql 方法执行过程共创建 DynamicContext 对象进行判定解析封装成 SqlSource 对象返回。

2、Mybatis 是否支持延迟加载？如果支持，它的实现原理是什么？ Mybatis 仅支持 association 关联对象和 collection 关联集合对象的延迟加载，association 指的就是一对一，collection 指的就是一对多查询。在 Mybatis 配置文件中，可以配置是否启用延迟加载 lazyLoadingEnabled=true|false。它的原理是，使用 CGLIB 创建目标对象的代理对象，当调用目标方法时，进入拦截器方法，比如调用 a.getB().getName()，拦截器 invoke()方法发现 a.getB()是 null 值，那么就会单独发送事先保存好的查询关联 B 对象的 sql，把 B 查询上来，然后调用 a.setB(b)，于是 a 的对象 b 属性就有值了，接着完成 a.getB().getName()方法的调用。这就是延迟加载的基本原理。

3、Mybatis 都有哪些 Executor 执行器？它们之间的区别是什么？ SimpleExecutor: 每执行一次 update 或 select, 就开启一个 Statement 对象, 用完立刻关闭 Statement 对象。 ReuseExecutor: 执行 update 或 select, 以 sql 作为 key 查找 Statement 对象, 存在就使用, 不存在就创建, 用完不关闭 Statement 对象, 而是放置于 Map 内, 供下一次使用。简言之, 就是重复使用 Statement 对象。 BatchExecutor: 执行 update (没有 select, JDBC 批处理不支持 select), 将所有 sql 都添加到批处理中 (addBatch()), 等待统一执行 (executeBatch()), 它缓存了多个 Statement 对象, 每个 Statement 对象都是 addBatch()完毕后, 等待逐一执行 executeBatch()批处理。与 JDBC 批处理相同。

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：它指的是 Mybatis 中 sqlSession 对象的缓存，当我们执行查询以后，查询的结果会同时存入到 sqlSession 为我们提供的一块区域中，该区域的结构是一个 Map，当我们再次查询同样的数据，mybatis 会先去 sqlSession 中查询是否有，的话直接拿出来用，当 sqlSession 对象消失时，mybatis 的一级缓存也就消失了，同时一级缓存是 sqlSession 范围的缓存，当调用 sqlSession 的修改、添加、删除、commit(),close 等方法时，就会清空一级缓存。 二级缓存：他值得是 Mybatis 中 SqlSessionFactory 对象的缓存，由同一个 SqlSessionFactory 对象创建的 sqlSession 共享其缓存，但是其中缓存的是数据而不是对象，所以从二级缓存再次查询出得结果的对象与 第一次存入的对象是不一样的。

5、简述 Mybatis 的插件运行原理，以及如何编写一个插件？ 1、每个创建出来的对象不是直接返回的，而是 interceptorChain.pluginAll(parameterHandler); 2、获取到所有的 Interceptor（拦截器）（插件需要实现的接口）；调用 interceptor.plugin(target);返回 target 包装后的对象 3、插件机制，我们可以使用插件为目标对象创建一个代理对象；AOP（面向切面）我们的插件可以为四大对象创建出代理对象；代理对象就可以拦截到四大对象的每一个执行；