

简答题

1. Mybatis 的动态 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?

答: 作用: 动态 sql 的作用是通过 mapper 中配置动态 sql 标签, 在构造 sql 时解析不同的传入的参数, 并对原有 sql 进行修饰, 动态生成最终执行的 sql 语句的功能。

执行原理: 1. 在 mybatis 初始化时, 在构建核心组件 configuration 中的 mappedStatement 时, 如果解析 mapper 中的 sql 标签包含了动态 sql 标签, 则会生成 DynamicSqlSource 并保存

2. 在执行 mapper 中包含动态 sql 的方法时, 分发到 BaseExecutor 中的 query 方法后, 会根据传入参数和对应 statement 中的 dynamicSqlSource, 调用 getBoundSql 方法, 构造出最终执行的 boundsql, 再进一步调用 statement 进行 sql 执行。

2. Mybatis 是否支持延迟加载? 如果支持, 它的实现原理是什么?

答: 支持

实现原理: 简要的说, 就是借助于代理方法, 将查询结果构造成代理对象, 在代理对象触发了懒加载需要去加载一些模块时, 执行对应 sql, 并更新代理对象

源码层面上讲, mybatis 初始化返回类型的时候, 会返回一个 cglib 代理对象, 这样的话, 该对象的关联对象相关信息就会在 loadpair 里边, 并且添加到 loadmap 中, cglib 对象会过滤 get, set, is, "equals", "clone", "hashCode", "toString" 触发方法, 然后才会调用 loadpair 来加载关联对象的值. 这部分操作是在 DefaultResultSetHandler 中的 getNestedQueryMappingValue 方法进行的, 如果开启了懒加载, 就会去生成代理对象代替原对象. 而在触发到懒加载时, 会调用 lazyLoader.load(property) 方法, 将懒加载记录模块 loadmap 中的对应 loadpair 删除, 并调用 load 方法, 执行 sql 并更新代理对象

3. Mybatis 都有哪些 Executor 执行器? 它们之间的区别是什么?

答: SimpleExecutor, ReuseExecutor, BatchExecutor, CachingExecutor, ClosedExecutor, 以及基类 BaseExecutor

BaseExecutor: 基类, 提供骨架方法

ClosedExecutor: 已经关闭的 Executor 的实现类 ResultLoaderMap 中记录当前已关闭的 Executor

CachingExecutor: 支持二级缓存的 Executor 的实现类, 其实是对真正的执行器的封装, 结合事务控制使用二级缓存

SimpleExecutor: 简单的 Executor 实现类, 实现常规的数据库操作方法

BatchExecutor: 在 SimpleExecutor 的基础上支持批量操作

ReuseExecutor: 可重用的 Executor 实现类, 1. 每次开始读或写操作, 优先从缓存中获取对应的 Statement 对象。如果不存在, 才进行创建。2. 执行完成后, 不关闭该 Statement 对象。

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）？

答：一级缓存：存储结构：HashMap<Object,Object> 范围：sqlSession 失效场景：

1.两个 cachekey 相同的操作之间如果进行了增删改操作,并 commit,则该 sqlSession 下的一级缓存会失效

2.手动清除,执行 clearCache 会导致一级缓存失效

二级缓存：存储结构：HashMap<Object,Object> 范围：mapper,失效场景，

1. mapper 中的某个 sqlSession 执行了增删改操作并 commit,则 mapper 层面的二级缓存失效

2. 如果使用了一些外部工具,比如 redis 做缓存,如果 redis 中的 key 过期了,或者某些原因导致 key 不存在了,则二级缓存也是失效了,简而言之缓存数据存储介质失效会导致二级缓存失效

3. 配置为软引用或弱引用的缓存被垃圾回收了,也是失效了

4. 到了 flushInterval 之后,缓存自动清除,则二级缓存失效了

5. 关联表的修改如果生效了,那存在一些关联查询会出现错误.需要用 cache-ref 来清除缓存影响.

暂时想到这么多.

5、简述 Mybatis 的插件运行原理，以及如何编写一个插件？

答：插件运行原理：基于代理机制,对 mybatis 的四大组件

(Executor,StatementHandler,ParameterHandler,ResultSetHandler)进行拦截并根据拦截器定义的操作,对原有代理对象的操作进行增强.

如何编写：编写实现类,实现 org.apache.ibatis.plugin. Interceptor 方法.

在 Intercepts 注解中定义需要被拦截的 Signature(可通过多个 Signature 实现对多个地方的拦截).每个 Signature 中,配置参数 type:拦截哪个接口 method:接口内方法名 args:拦截方法的入参

在实现类内部重写 intercept(核心拦截方法) plugin(生成代理并放入拦截器链)和 setProperties(获取插件配置参数)方法

写好拦截器实现类后,在 sqlMapConfig.xml 中配置拦截器及其所需参数,即可运行.