

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 指的是在 xml 映射文件或者注解中，以标签的形式标注，在进行 sql 操作时根据传入参数的条件动态生成不同 sql 的功能。

Mybatis 提供的动态 sql 标签有：

`<if>` `<where>` `<foreach>` `<trim>` `<set>` `<choose>` `<when>` `<otherwise>` `<include>`

执行原理：mybatis 在启动时通过 XMLConfigBuilder 解析 mapper.xml 配置文件，获取所有的 mapper 配置包括动态标签存放到 configuration 对象中；在执行过程中，executor 封装 boundsql 对象时根据 configuration 中的标签生成具体的 sql,再调用底层的 jdbc 进行数据库操作。

2、Mybatis 是否支持延迟加载？如果支持，它的实现原理是什么？

Mybatis 在使用 `<association>` 一对一查询、`<collection>` 一对多查询时支持懒加载，懒加载指的是在第一次查询时从单表查询返回结果，当使用到关联表的数据时再次进行查询的操作。

实现原理: Mybati 解析配置文件时会把 `<resultMap>` 标签映射为 ResultMap 对象，在处理返回结果时循环

<resultMap> 中的子标签，如果有配置懒加载的属性，则会创建一个代理对象，这个代理类继承了结果类并重写了结果类的所有方法。当调用懒加载的 get 方法时就会触发懒加载操作进行再次查询，懒加载查询所需要的参数及其他条件都封装在代理对象内部。

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

BaseExecutor: 抽象类，是 Executor 的默认实现，实现了 Executor 接口的大部分接口功能。

BatchExecutor: 批处理执行器，执行 update 操作时，将所有 sql 都添加到批处理中等待统一执行，它缓存了多个 statement 对象。

CachingExecutor: 缓存执行器，执行查询时先从缓存中查询结果，如果存在则直接返回，否则委托给 executor 去数据库中查询。

ReuseExecutor: 可重用执行器，执行 update 或者 select 操作时，以 sql 作为 key 查找 statement 对象，存在就使用，用完后放到 map 中缓存起来。

SimpleExecutor: mybatis 默认使用的执行器，每执行一次，就开启一个 statement 对象，用完立刻关闭 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）？

一级缓存：指 `Sqlsession` 对象的缓存，当执行完查询一个，查询结果会存入 `Sqlsession` 的 `map` 中，当再次以同样条件查询时先从该 `map` 中获取数据返回；一级缓存是 `sqlsession` 范围的缓存，当调用 `sqlsession` 的修改、添加、删除操作或者调用 `commit`、`close` 方法时会清空。

二级缓存：指 `SqlSessionFactory` 对象的缓存，由同一个 `SqlSessionFactory` 对象创建的所有 `sqlsession` 共享；二级缓存的读取优先级高于一级缓存，二级缓存的生命周期是整个应用，应用不结束就会一直存在内存中。

5、简述 **Mybatis** 的插件运行原理，以及如何编写一个插件？

`Mybatis` 仅可以编写针对 `ParameterHandler`、`ResultSetHandler`、`StatementHandler`、`Executor` 这 4 种接口的插件，`Mybatis` 使用 `JDK` 的动态代理，为需要拦截的接口生成代理对象以实现接口方法拦截功能，每当执行这 4 种接口对象的方法时，就会进入拦

截方法，具体就是 `InvocationHandler` 的 `invoke()`方法。

编写插件：实现 `Mybatis` 的 `Interceptor` 接口并复写 `intercept()`方法，然后在给插件编写注解，指定要拦截哪一个接口的哪些方法即可，另外需要在配置文件中配置编写的插件。