链滴

MyBatis 面试题整理

作者：umeone

原文链接：https://ld246.com/article/1517109910928

来源网站：链滴

以下面试题通过网络整理，参考了很多文章，并结合自己的一点理解进行描述因参考文章较多，因此不再注明来源，望谅解。
<h4 id="JDBC编程有哪些不足-MyBatis是如何解决的-">JDBC 编程有哪些不足？MyBatis 是如何决的？</h4>

数据库的连接创建、释放频繁，从而影响性能，mybatis 通过配置 SqlMapConfig.xml 文件使连接池的方式解决了数据库连接创建和释放频繁所造成的性能影响。
大量的 sql 存在于代码之中，造成代码的可维护性低。mybatis 只用 xml 文件对 sql 进行统一管，方便维护。
jdbc 操作中存在参数时，需要准确的定位参数的位置和对应占位符的个数，否则会出错。mybati 通过提供参数对象的方式解决了该问题。
sql 语句在编写时，如果存在动态条件则不容易处理。mybatis 提供动态 sql 编写机制，时用户以根据自己传入参数的情况进行 sql 语句的动态编写。
对查询时返回的结果集处理是一件比较麻烦的事情，而随着查询结果的改变，处理结果集的代码随之改变。mybatis 通过把结果集映射成对应的 java 对象，解决了结果集处理麻烦的问题。

<h4 id="MyBatis编程步骤">MyBatis 编程步骤</h4>

创建 SqlSessionFactory
通过 SqlSessionFactory 获取 SqlSession
通过 SqlSession 执行数据库操作
提交事务
关闭会话

<h4 id="MyBatis和Hibernate的区别">MyBatis 和 Hibernate 的区别</h4>
总体来说 mybatis 和 hibernate 都属于持久层框架。就目前主流的 ORM 框架而言，hibernate 数据全自动 ORM 实现，而 mybatis 属于半自动 ORM 实现。他们最直观的区别在于：
hibernate 完整的实现了对象到数据库结构的映射，也就是说在使用 hibernate 的时候，你可以不需关心数据库结构，而仅仅只用知道数据库结构对应的对象即可，hibernate 提供了一系列的操作使你有的数据库操作转换成对象的操作，而 hibernate 在执行操作的时候会自动的帮你生成对应的 sql 并行，并把返回的结果集封装成对应的对象返回。
而 mybatis 在实现是仅仅是对结果集进行了对象的封装返回，而执行的 sql 需要开发人员自己去实现
虽然 hibernate 是全自动 ORM 框架，但是在某些情况下并不是那么好用，比如：

在某些情况下，数据库权限没有全部开放，只能执行部分 sql 操作的时候。
数据库查询操作十分繁琐，sql 执行很复杂，为了提供效率，需要对 sql 进行优化。
数据库查询需要用到存储过程时。

等等以上情况在使用 hibernate 时就不能应付了。这时使用 mybatis 则是一个不错的选择。但如果存在多数据库访问的情况下，hibernate 会是一个更好的选择，而 mybatis 在操作多数据库时，一个数据库操作需要实现不同的 sql。
<h4 id="---和---的区别">${}和#{}的区别</h4>
{}时property文件的变量占位符，可以运用在标签属性值和sq内部，属于静态文本替换，{}中的内容会直接被替换成变量对应的值。
#{}属于参数占位符，mybatis 会在预编译过程中把对应的参数替换成 sql 中？进行填充，并且在执行ql 前通过反射机制获取对应参数对象的值，并按照参数占位符进行参数的设置，这样参数占位符可以止 sql 注入。
<h4 id="xml映射文件中有哪些标签">xml 映射文件中有哪些标签</h4>

常见操作标签：select、insert、update、delete
动态 sql 标签：trim、where、set、foreach、if、choose、when、otherwise、bind
其他标签：、、(sql 片段标签)、(引入 sql 片段标签)、

<h4 id="mybatis如何确定唯一的操作sql-MappedStatement-">mybatis 如何确定唯一的操作 sql MappedStatement）</h4>
mybatis 通过 namespace 和 MappedStatement 的 id 值确定唯一的 MappedStatement。
<h4 id="mybatis的mapper接口中的接口方法可以通过不同参数的方式对方法进行重载吗-">mybatis 的 mapper 接口中的接口方法可以通过不同参数的方式对方法进行重载吗？</h4>
不可以。在 mybatis 在执行方法前是通过 namespace+id 进行 MappedStatement 的确定，参数没有关系，所以不能实现重载。
<h4 id="Mapper接口的工作原理">Mapper 接口的工作原理</h4>
mybatis 在运行时使用 JDK 动态代理生成代理对象，在调用接口方法时，代理对象会拦截接口法，转而运行 MappedStatement 的 sql 语句，并生成结果返回。
<h4 id="MyBatis是如何进行分页的-分页插件的原理-">MyBatis 是如何进行分页的？分页插件的原？</h4>
mybatis 是通过 RowBounds 对象进行分页的，他针对返回结果集 ResultSet 进行内存分页，非物理分页。
分页插件通过 mybatis 提供的分页插件接口进行实现，通过拦截器拦截需要执行的 sql 并重写 sql，此来添加对应的参数，最终实现分页效果。
<h4 id="MyBatis在批量插入时能返回数据库主键列表吗-">MyBatis 在批量插入时能返回数据库主列表吗？</h4>
可以。在 JDBC 执行批量插入时可以返回主键列表，而 MyBatis 是基于 JDBC 的实现，同样可返回主键列表。
<h4 id="MyBatis动态sql是什么-">MyBatis 动态 sql 是什么？</h4>
mybatis 在 xml 映射文件中提供了一套动态 sql 拼接的逻辑标签，通过这些标签我们可以根据际传入参数的情况进行 sql 的拼接，最终达到根据参数动态生成 sql 的效果。
<h4 id="MyBatis是否支持延迟加载-如果支持-他的实现原理是什么-">MyBatis 是否支持延迟加载如果支持，他的实现原理是什么？</h4>
mybatis 支持延迟加载，但有限制。他支持一对一、一对多的延迟加载。
要使 mybatis 支持延迟加载，需要在配置文件中配置 lazyLoadingEnabled 值为 true。mybatis 通过 CGLIB 创建目标对象的代理对象，在调用目标对象的 get 方法时，进行拦截并检查关联对象是为空，如果为空则调用 sql 进行相应对象的查询并通过 set 方法进行数据填充，最后在返回对应关联象。
<h4 id="在MyBatis的xml映射文件中-不同的xml文件id是否可以重复-">在 MyBatis 的 xml 映射文中，不同的 xml 文件 id 是否可以重复？</h4>
在 mybatis 实现中要获取一个 MappedStatement 是通过 namespace+id 进行唯一确定的，而 namespace 是最佳实践，但并不是必须的，如果没有 namespace 的存在，重复的 id 会造成 Mapp dStatement 被覆盖。所以结合最佳实践而言，不同的 xml 文件 id 建议别重复使用。
<h4 id="MyBatis中最终执行数据库操作的Executor执行器-在mybatis中有那些的执行器-他们的区是什么-">MyBatis 中最终执行数据库操作的 Executor 执行器，在 mybatis 中有那些 <code>基本</code> 的执行器？他们的区别是什么？</h4>
在 mybatis 中有三种基本的执行器，他们分别是：SimpleExecutor、ReuseExecutor、BatchEx cutor。他们区别如下：

SimpleExecutor：每执行一次 update 或 select 都开启一个 Statement 对象，执行完后关闭该象。
ReuseExecutor:执行 update 或 select 时，以 sql 作为 key 进行缓存查找 Statement，如果找直接执行，如果没找到则创建并缓存，执行完操作后不关闭该对象，以备下次执行时使用。
BatchExecutor:在执行 update 时，所有的 sql 都将添加到批处理器中进行统一处理。他缓存了个 Statement 对象，所有的对象都等待着屁处理器统一执行。

<h4 id="MyBatis是否支持枚举类的映射-">MyBatis 是否支持枚举类的映射？</h4>
支持。
mybatis 可以通过自定义 TypeHandler,并实现 <code>setParameter()</code> 和 <code>getRes lt()</code> 方法进行类型转换。