1、MyBatis中#{}和${}的区别是什么？

#{}是预编译处理，${}是字符替换

在使用#{}时，MyBatis会将SQL中的#{}替换成”?”，配合PreparedStatement的set方法赋值，这样可以有效的防止SQL注入，保证程序的运行安全。

2、MyBatis有几种分页方式？

分页方式：逻辑分页和物理分页。

逻辑分页：使用MyBatis自带的RowBounds进行分页，它是一次性查询很多数据，然后在数据中再进行检索。

物理分页：自己手写SQL分页或使用分页插件PageHelper，去数据库查询指定条数的分页数据的形式。

3、RowBounds是一次性查询全部结果吗？为什么？

RowBounds表面是在“所有”数据中检索数据，其实并非是一次性查询出所有数据，因为MyBatis是对JDBC的封装，在JDBC驱动中有一个Fetch Size的配置，他规定了每次最多从数据库查询多少条数据，假如你要查询更多数据，它会在你执行nex()的时候，去查询更多的数据。就好比你去ATM机取10000元，但取款机每次最多能取2500元，所以你要取4次才能把钱取完。只要对于JDBC来说，当你调用next()的时候会自动帮你完成查询工作。这样做的好处可以有效的防止内存溢出。

4、MyBatis逻辑分页和物理分页的区别是什么？

逻辑分页是一次性查询很多数据，然后再在结果中检索分页的数据。这样做弊端是需要消耗大量的内存、有内存溢出的风险、对数据库压力较大。

物理分页是从数据库查询指定条数的数据，弥补了一次性全部查出的所有数据的种种缺点，比如需要大量的内存，对数据库查询压力较大等问题。

5、MyBatis是否支持延迟加载，延迟加载的原理是什么？

支持，设置lazyLoadingEnabled=true即可。

延迟加载的原理是调用的时候触发加载，而不是在初始化的时候就加载信息。比如调用a.getB().getName()，这个时候发现a.getB()的值为null，此时会单独触发事先保存好的关联B对象的SQL，先查询出来B，然后再调用a.setB(b)，而这时候再调用a.getB().getName()就有值了，这就是延迟加载的基本原理。

6、说一下MyBatis的一级缓存和二级缓存？

一级缓存：基于PerpetualCache的HashMap本地缓存，它的声明周期是和SQLSession一致的，有多个SQLSession或者分布式的环境中数据库操作，可能会出现脏数据。当Session flush或close之后，该Session中的所有Cache就将清空，默认一级缓存是开启的。

二级缓存：也是基于PerpetualCache的HashMap本地缓存，不同在于其存储作用域为Mapper级别的，如果多个SQLSession之间需要共享缓存，则需要使用到二级缓存，并且二级缓存可自定义存储源，如Ehcache。默认不打开二级缓存，要开启二级缓存，使用二级缓存属性类需要实现Serializable序列化接口（可用来保存对象的状态）。

开启二级缓存数据查询流程：二级缓存->一级缓存->数据库

缓存更新机制：当某一个作用域（一级缓存Session/二级缓存Mapper）进行了C/U/D操作后，默认该作用域下所有select中的缓存将被clear。

7、MyBatis和hibernate的区别有哪些？

灵活性：MyBatis更加灵活，自己可以写SQL语句，使用起来比较方便。

可移植性：MyBatis有很多自己写的SQL，因为每个数据库的SQL可以不相同，所以可移植性比较差。

学习和使用门槛：MyBatis入门比较简单，使用门槛也更低。

二级缓存：hibernate拥有更好的二级缓存，它的二级缓存可以自行更换为第三方的二级缓存。

8、MyBatis有哪些执行器（Executor）?

MyBatis有三种基本的Executor执行器：

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，用完后不关闭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。

9、MyBatis分页插件的实现原理是什么？

分页插件的基本原理是使用MyBatis提供的插件接口，实现自定义插件，在插件的拦截方法内拦截待执行的SQL，然后重写SQL，根据dialect方言，添加对应的物理分页语句和物理分页参数。