#### 1、#{}和${}的区别

${}是Properties文件中变量的占位符，它可以用于标签属性值和sql内部，属于静态文本替换，比如${driver}会被静态替换为com.mysql.jdbc.Driver。

#{}是sql的参数占位符，Mybatis会将sql中d

的#{}替换为?号，在sql执行前会使用preparedStatement的参数设置方法，按序给sql的?号占位符设置参数值，比如ps.setInt(0,parameterValue)，#{item.name}的取值方式为使用反射从参数对象中获取item对象的name属性值，相当于param.getItem().getName()。

#### 2、xml映射文件中有哪些常见的标签

select|insert|update|delete

<resultMap>、<parameterMap>、<sql>、<include>、<selectKey>，加上动态sql的9个标签，trim|where|set|foreach|if|choose|when|otherwise|bind等，其中为sql片段标签，通过<include>标签引入sql片段，<selectKey>为不支持自增的主键生成策略标签。

#### 3、通常一个xml映射文件，都会写一个Dao接口与之对应，这个Dao接口的工作原理是什么？接口里的方法，参数不同是，方法能重载吗？

Dao接口就是常说的，Mapper接口，

* 接口的全限名——>映射文件中的namespace的值，
* 接口的方法名——>映射文件中MappedStatement的id值，
* 接口方法内的参数——>传递给sql的参数。

Mapper接口是没有实现类的，当调用接口方法时，接口全限名+方法名拼接字符串作为 key 值，可唯一定位一个MappedStatement，举例：com.mybatis3.mappers.StudentDao.findStudentById，可以唯一找到 namespace为com.mybatis3.mappers.StudentDao下面id = findStudentById的MappedStatement。在 Mybatis 中，每一个<select>、<insert>、<update>、<delete>标签，都会被解析为一个MappedStatement对象。

Dao接口里的方法，是不能重载的，因为是全限名+方法名的保存和寻找策略。

Dao接口的工作原理是JDK动态代理，Mybatis 运行时会使用JDK动态代理为Dao接口生成代理proxy对象，代理对象proxy会拦截接口方法，转而执行MappedStatement所代表的sql，然后将sql执行结果返回。

#### 4、Mybatis是如何进行分页的？分页插件的原理是什么？

Mybatis使用RowBounds对象进行分页，它是针对ResultSet结果集执行的内存分页，而非物理分页，可以在sql内直接书写带有物理分页的参数来完成物理分页功能，也可以使用分页插件来完成物理分页。

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。

举例：select \_ from student，拦截sql后重写为：select t.\_ from (select \\* from student）t limit 0，10

#### 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，当然，只会拦截那些你指定需要拦截的方法。

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，记住，别忘了在配置文件中配置你编写的插件。

#### 6、Mybatis执行批量插入，能返回数据库主键列表吗

能

#### 7、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

Mybatis动态sql可以让我们在xml映射文件内，以标签的形式编写动态sql，完成逻辑判断和动态拼接sql的功能，Mybatis提供了9种动态sql标签。

trim|where|set|foreach|if|choose|when|otherwise|bind

其执行原理为，使用OGNL从sql参数对象中计算表达式的值，根据表达式的值动态拼接sql，以此来完成动态sql的功能。

#### 8、Mybatis是如何将sql执行结果封装为目标对象并返回的？都有哪些映射新形式

* 第一种是使用<resultMap>标签，逐一定义列名和对象属性名之间的映射关系。
* 第二种是使用sql列的别名功能，将列别名书写为对象属性名，比如 T\_NAME AS NAME，对象属性名一般是 name，小写，但是列名不区分大小写，Mybatis 会忽略列名大小写，智能找到与之对应对象属性名，你甚至可以写成 T\_NAME AS NaMe，Mybatis一样可以正常工作。

有了列名与属性名的映射关系后，Mybatis通过反射创建对象，同时使用反射给对象的属性逐一赋值并返回，那些找不到映射关系的属性，是无法完成赋值的。

#### 9、Mybatis有哪些方式实现一对一、一对多的关联查询。它们之间的区别。

Mybatis不仅可以执行一对一、一对多的关联查询，还可以执行多对一，多对多的关联查询，多对一查询，其实就是一对一查询，只需要把selectOne()修改为selectList()即可；多对多查询，其实就是一对多查询，只需要把 selectOne()修改为selectList()即可。

关联对象查询，有两种实现方式：

* 一种是单独发送一个sql去查询关联对象，赋给主对象，然后返回主对象。
* 另一种是使用嵌套查询，嵌套查询的含义为使用join查询，一部分列是A对象的属性值，另外一部分列是关联对象B的属性值，好处是只发一个sql查询，就可以把主对象和其关联对象查出来。

那么问题来了，join查询出来100条记录，如何确定主对象是5个，而不是100个？去重复的原理是<resultMap>标签内的<id>子标签，指定了唯一确定一条记录的id列，Mybatis根据列值来完成100条记录的去重复功能，<id>可以有多个，代表了联合主键的语意。

同样主对象的关联对象，也是根据这个原理去重复的，尽管一般情况下，只有主对象会有重复记录，关联对象一般不会重复。

举例：下面join查询出来6条记录，一、二列是Teacher对象列，第三列为Student对象列，Mybatis去重复处理后，结果为1个老师6个学生，而不是6个老师6个学生。

t\_id t\_name s\_id

| 1 | teacher | 38 | | 1 | teacher | 39 | | 1 | teacher | 40 | | 1 | teacher | 41 | | 1 | teacher | 42 | | 1 | teacher | 43 |

#### 10、Mybatis延迟加载的实现原理

mybatis:

configuration:  
 lazy-loading-enabled: true

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。

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。不光是Mybatis，几乎所有的包括Hibernate，支持延迟加载的原理都是一样的。

#### 11、Mybatis的xml映射文件中，不同的xml映射文件，id是否可以重复

不同的xml映射文件，如果配置了namespace，那么id可以重复；如果没有配置namespace，那么id不能重复；毕竟namespace不是必须的，只是最佳实践而已。原因就是namespace+id是作为Map<String, MappedStatement>的key使用的，如果没有namespace，就剩下id，那么，id重复会导致数据互相覆盖。

#### 12、Mybatis中如何执行批处理

使用BatchExecutor完成批处理。

#### 13、Mybatis都有哪些Executor执行器，它们之间的区别

Mybatis有三种基本的Executor执行器：

* SimpleExecutor
* ReuseExecutor
* BatchExecutor

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。

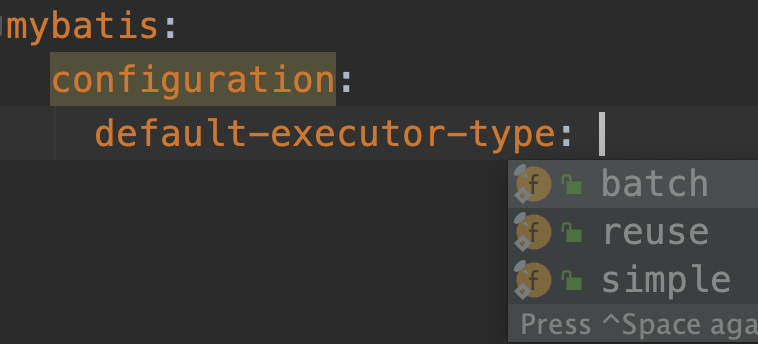
ReuseExecutor：执行update或select，以sql作为key查找Statement对象，存在就使用，不存在就创建，用完后，不关闭Statement对象，而是放置于Map<String, Statement>内，供下一次使用。

BatchExecutor：执行update（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。

作用范围：严格限制在SqlSession生命周期范围内。

#### 14、Mybatis中如何制定使用哪一种Executor执行器

在 Mybatis 配置文件中，可以指定默认的ExecutorType执行器类型：



也可以手动给DefaultSqlSessionFactory的创建SqlSession的方法传递ExecutorType类型参数。

#### 15、Mybatis是否可以映射Enum枚举类

Mybatis可以映射枚举类，不单可以映射枚举类，Mybatis可以映射任何对象到表的一列上。映射方式为自定义一个TypeHandler，实现TypeHandler的setParameter()和getResult()接口方法。

TypeHandler有两个作用：一是完成javaType至jdbcType的转换二是完成jdbcType至javaType的转换。体现为setParameter()和getResult()两个方法，分别代表设置sql问号占位符参数和获取列查询结果。

#### 16、Mybatis映射文件中，如果A标签通过include引用了B标签的内容，B标签你能否定义在A标签的后面，还是说必须定义在A标签的前面

虽然Mybatis解析xml映射文件是按照顺序解析的，但是，被引用的B标签依然可以定义在任何地方，Mybatis都可以正确识别。

原理是，Mybatis解析A标签，发现A标签引用了B标签，但是B标签尚未解析到，尚不存在，此时，Mybatis会将A标签标记为未解析状态，然后继续解析余下的标签，包含B标签，待所有标签解析完毕，Mybatis会重新解析那些被标记为未解析的标签，此时再解析A标签时，B标签已经存在，A标签也就可以正常解析完成了。

#### 17、简述Mybatis的xml映射文件和Mybatis内部数据结构之间的映射关系

Mybatis将所有xml配置信息都封装到All-In-One重量级对象Configuration内部。在xml映射文件中，

* <parameterMap>标签会被解析为ParameterMap对象，其每个子元素会被解析为ParameterMapping对象。
* <resultMap>标签会被解析为ResultMap对象，其每个子元素会被解析为ResultMapping对象。
* 每一个<select>、<insert>、<update>、<delete>标签均会被解析为MappedStatement对象，
* 标签内的sql会被解析为BoundSql对象。

#### 18、Mybatis为什么是半自动映射工具，与全自动的区别

Hibernate属于全自动ORM映射工具，使用Hibernate查询关联对象或者关联集合对象时，可以根据对象关系模型直接获取，所以它是全自动的。而Mybatis在查询关联对象或关联集合对象时，需要手动编写sql来完成，所以，称之为半自动ORM映射工具。