**1、什么是MyBatis？**

答：MyBatis是一个可以自定义SQL、存储过程和高级映射的持久层框架。

2、讲下MyBatis的缓存

答：MyBatis的缓存分为一级缓存和二级缓存,一级缓存放在session里面,默认就有,二级缓存放在它的命名空间里,默认是不打开的,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),可在它的映射文件中配置<cache/>

**3、Mybatis是如何进行分页的？分页插件的原理是什么？**

答：

1）Mybatis使用RowBounds对象进行分页，也可以直接编写sql实现分页，也可以使用Mybatis的分页插件。

2）分页插件的原理：实现Mybatis提供的接口，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。

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

**4、简述Mybatis的插件运行原理，以及如何编写一个插件？**

答：

1）Mybatis仅可以编写针对ParameterHandler、ResultSetHandler、StatementHandler、Executor这4种接口的插件，Mybatis通过动态代理，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。

2）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。

5、Mybatis动态sql是做什么的？都有哪些动态sql？能简述一下动态sql的执行原理不？

答：

1）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。

2）Mybatis提供了9种动态sql标签：trim|where|set|foreach|if|choose|when|otherwise|bind。

3）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。

6、#{}和${}的区别是什么？

答：

1）#{}是预编译处理，${}是字符串替换。

2）Mybatis在处理#{}时，会将sql中的#{}替换为?号，调用PreparedStatement的set方法来赋值；

3）Mybatis在处理${}时，就是把${}替换成变量的值。

4）使用#{}可以有效的防止SQL注入，提高系统安全性。

7、为什么说Mybatis是半自动ORM映射工具？它与全自动的区别在哪里？

答：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。

8、Mybatis是否支持延迟加载？如果支持，它的实现原理是什么？

答：

1）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lazyLoadingEnabled=true|false。

2）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。

9、MyBatis与Hibernate有哪些不同？

答：

1）Mybatis和hibernate不同，它不完全是一个ORM框架，因为MyBatis需要程序员自己编写Sql语句，不过mybatis可以通过XML或注解方式灵活配置要运行的sql语句，并将java对象和sql语句映射生成最终执行的sql，最后将sql执行的结果再映射生成java对象。

2）Mybatis学习门槛低，简单易学，程序员直接编写原生态sql，可严格控制sql执行性能，灵活度高，非常适合对关系数据模型要求不高的软件开发，例如互联网软件、企业运营类软件等，因为这类软件需求变化频繁，一但需求变化要求成果输出迅速。但是灵活的前提是mybatis无法做到数据库无关性，如果需要实现支持多种数据库的软件则需要自定义多套sql映射文件，工作量大。

3）Hibernate对象/关系映射能力强，数据库无关性好，对于关系模型要求高的软件（例如需求固定的定制化软件）如果用hibernate开发可以节省很多代码，提高效率。但是Hibernate的缺点是学习门槛高，要精通门槛更高，而且怎么设计O/R映射，在性能和对象模型之间如何权衡，以及怎样用好Hibernate需要具有很强的经验和能力才行。

总之，按照用户的需求在有限的资源环境下只要能做出维护性、扩展性良好的软件架构都是好架构，所以框架只有适合才是最好。

10、MyBatis的好处是什么？

答：

1）MyBatis把sql语句从Java源程序中独立出来，放在单独的XML文件中编写，给程序的维护带来了很大便利。

2）MyBatis封装了底层JDBC API的调用细节，并能自动将结果集转换成Java Bean对象，大大简化了Java数据库编程的重复工作。

3）因为MyBatis需要程序员自己去编写sql语句，程序员可以结合数据库自身的特点灵活控制sql语句，因此能够实现比Hibernate等全自动orm框架更高的查询效率，能够完成复杂查询。

11、简述Mybatis的Xml映射文件和Mybatis内部数据结构之间的映射关系？

答：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。

12、什么是MyBatis的接口绑定,有什么好处？

答：接口映射就是在MyBatis中任意定义接口,然后把接口里面的方法和SQL语句绑定,我们直接调用接口方法就可以,这样比起原来了SqlSession提供的方法我们可以有更加灵活的选择和设置.

13、接口绑定有几种实现方式,分别是怎么实现的?

答：接口绑定有两种实现方式,一种是通过注解绑定,就是在接口的方法上面加上@Select@Update等注解里面包含Sql语句来绑定,另外一种就是通过xml里面写SQL来绑定,在这种情况下,要指定xml映射文件里面的namespace必须为接口的全路径名.

14、什么情况下用注解绑定,什么情况下用xml绑定？

答：当Sql语句比较简单时候,用注解绑定；当SQL语句比较复杂时候,用xml绑定,一般用xml绑定的比较多

15、MyBatis实现一对一有几种方式?具体怎么操作的？

答：有联合查询和嵌套查询,联合查询是几个表联合查询,只查询一次,通过在resultMap里面配置association节点配置一对一的类就可以完成;嵌套查询是先查一个表,根据这个表里面的结果的外键id,去再另外一个表里面查询数据,也是通过association配置,但另外一个表的查询通过select属性配置。

16、Mybatis能执行一对一、一对多的关联查询吗？都有哪些实现方式，以及它们之间的区别？

答：能，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。

17、MyBatis里面的动态Sql是怎么设定的?用什么语法?

答：MyBatis里面的动态Sql一般是通过if节点来实现,通过OGNL语法来实现,但是如果要写的完整,必须配合where,trim节点,where节点是判断包含节点有内容就插入where,否则不插入,trim节点是用来判断如果动态语句是以and 或or开始,那么会自动把这个and或者or取掉。

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

答：

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。

19、Xml映射文件中，除了常见的select|insert|updae|delete标签之外，还有哪些标签？

答：还有很多其他的标签，<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<sql>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。

20、当实体类中的属性名和表中的字段名不一样，如果将查询的结果封装到指定pojo？

答：

1）通过在查询的sql语句中定义字段名的别名。

2）通过<resultMap>来映射字段名和实体类属性名的一一对应的关系。

21、模糊查询like语句该怎么写

答：

1）在java中拼接通配符，通过#{}赋值

2）在Sql语句中拼接通配符 （不安全 会引起Sql注入）

22、通常一个Xml映射文件，都会写一个Dao接口与之对应, Dao的工作原理，是否可以重载？

答：不能重载，因为通过Dao寻找Xml对应的sql的时候全限名+方法名的保存和寻找策略。接口工作原理为jdk动态代理原理，运行时会为dao生成proxy，代理对象会拦截接口方法，去执行对应的sql返回数据。

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

答：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。

24、Mybatis的Xml映射文件中，不同的Xml映射文件，id是否可以重复？

答：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。有了namespace，自然id就可以重复，namespace不同，namespace+id自然也就不同。

25、Mybatis中如何执行批处理？

答：使用BatchExecutor完成批处理。

26、Mybatis都有哪些Executor执行器？它们之间的区别是什么？

答：Mybatis有三种基本的Executor执行器，SimpleExecutor、ReuseExecutor、BatchExecutor。1）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。2）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Map3）BatchExecutor：完成批处理。

27、Mybatis中如何指定使用哪一种Executor执行器？

答：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。

28、Mybatis执行批量插入，能返回数据库主键列表吗？

答：能，JDBC都能，Mybatis当然也能。

29、Mybatis是否可以映射Enum枚举类？

答：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，一是完成从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。

30、如何获取自动生成的(主)键值？

答：配置文件设置usegeneratedkeys 为true

31、在mapper中如何传递多个参数？

答：

1）直接在方法中传递参数，xml文件用#{0} #{1}来获取

2）使用 @param 注解:这样可以直接在xml文件中通过#{name}来获取

32、resultType resultMap的区别？

答：

1）类的名字和数据库相同时，可以直接设置resultType参数为Pojo类

2）若不同，需要设置resultMap 将结果名字和Pojo名字进行转换

33、使用MyBatis的mapper接口调用时有哪些要求？

答：

1）Mapper接口方法名和mapper.xml中定义的每个sql的id相同

2）Mapper接口方法的输入参数类型和mapper.xml中定义的每个sql 的parameterType的类型相同

3）Mapper接口方法的输出参数类型和mapper.xml中定义的每个sql的resultType的类型相同

4）Mapper.xml文件中的namespace即是mapper接口的类路径。

34、Mybatis比IBatis比较大的几个改进是什么？

答：

1）有接口绑定,包括注解绑定sql和xml绑定Sql

2）动态sql由原来的节点配置变成OGNL表达式3） 在一对一,一对多的时候引进了association,在一对多的时候引入了collection节点,不过都是在resultMap里面配置

35、IBatis和MyBatis在核心处理类分别叫什么？

答：IBatis里面的核心处理类交SqlMapClient,MyBatis里面的核心处理类叫做SqlSession。

36、IBatis和MyBatis在细节上的不同有哪些？

答：

1）在sql里面变量命名有原来的#变量# 变成了#{变量}

2）原来的$变量$变成了${变量}

3）原来在sql节点里面的class都换名字交type

4）原来的queryForObject queryForList 变成了selectOne selectList5）原来的别名设置在映射文件里面放在了核心配置文件里