(1)mybatis 动态 SQL 是 mybatis 强大的的特性之一，在某些复杂的业务中，需要写复杂的 SQL 语句，这时就可以通过动态 SQL 提高 SQL 语
句的准确性，同时动态 SQL 也非常灵活，可以根据我们业务所传参数，灵活拼接 SQL
常用的动态 SQL 有：if choose when where foreach 等
动态 SQL 的基本原理：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
SQLsource：通过该接口可以获取 SQL 对象，它的实现类是 xmlSQLsource，表示通过 xml 文件生成 SQL 对象
Sql :该接口可以生成 sql 语句和获取 sql 相关的上下文环境(如 ParameterMap、ResultMap 等)，有三个实现类:
RawSql 表示为原生的 sql 语句，在初始化即可确定 sql 语句；
SimpleDynamicSql 表示简单的动态 sql，即 sql 语句中参数通过$property$方式指定，参数在 sql 生成过程中会被替换，
不作为 sql 执行参数；
DynamicSql 表示动态 sql，即 sql 描述文件中包含 isNotNull 等条件标签。
SqlChild:该接口表示 sql 抽象语法树的一个节点，包含 sql 语句的片段信息。该接口有两个实现类:
SqlTag 表示动态 sql 片段，即配置文件中的一个动态标签，内含动态 sql 属性值(如 prepend、property 值等)；
SqlText 表示静态 sql 片段，即为原生的 sql 语句。每条动态 sql 通过 SqlTag 和 SqlText 构成相应的抽象语法树
SqlTagHandler:该接口表示 SqlTag(即不同的动态标签)对应的处理方式。
比如实现类 IsEmptyTagHandler 用于处理<isEmpty>标签，IsEqualTagHandler 用于处理<isEqual>标签等
SqlTagContext:用于解释 sql 抽象语法树时使用的上下文环境。通过解释语法树每个节点，将生成的 sql 存入 SqlTagContext。
最终通过 SqlTagContext 获取完整的 sql 语句
动态 sql 的生成过程包含两步: 初始化过程用于解析 sql 配置，生成由 SqlChild 节点组成的抽象语法树；
请求处理过程通过运行期的参数对象解释抽象语法树，生成实际的 sql 语句
(2)mybatis 支持 association 关联对象和 collection 关联集合对象的延时加载，延迟加载是基于嵌套查询来实现的，主要是通过动态代理的形式实现，通过代理拦截到指定方法，执行数据加载
(3) mybatis 有三种执行器分别是 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 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。

(4)mybatis 一级，二级缓冲底层都是一个 hashmap，用于缓冲数据。

一级缓冲的范围是 SQLsession 级别，不同 SQLsession 缓冲区域之间的数据是不相互影响的

二级缓冲的范围是 namespace 级别，是一个全局缓冲，多个 SQLsession 操作一个 namespace 下的 SQL 语句，多个 SQLsession 是可以共享二级缓冲的。

一级缓冲的失效情况：

1 不同是 SQLsession

2SQLsession 相同，但是查询条件不一样

3SQLsession 相同，但是两次查询之间做了增删改操作

3SQLsession 相同，但是手动清除了缓冲

二级缓冲的失效情况：

1 不同的 namespace 空间

2 第一次的 SQLsession 未提交

3 手动清除掉二级缓冲

(5)mybatis 的插件运行原理：

mybatis 的插件就是拦截器，mybatis 利用动态代理机制，一层层包装目标对象，而实现在目标对象执行目标方法之前进行拦截

的效果。

插件开发步骤：

1 编写插件实现 interceptor 接口，并使用@Intercepts 完成插件签名

2 在全局文件中注册插件