```
MyBatis 允许你在已映射语句执行过程中的某一点进行拦截调用
                                                                                                                          Executor (update, query, flushStatements, commit, rollback, getTransaction, close, isClosed)
                                                                                                                          ParameterHandler (getParameterObject, setParameters)
                                                                                      允许使用插件来拦截的方法调用包括
                                                                                                                          ResultSetHandler (handleResultSets, handleOutputParameters)
                                                                                                                         StatementHandler (prepare, parameterize, batch, update, query)
                                                               插件 (plugins)
                                                                                                只需实现 Interceptor 接口,并指定了想要拦截的方法签名即可
                                                                                                            拦截在 Executor 实例中所有的 "update" 方法调用
                                                                                                           // ExamplePlugin.java
                                                                                                           @Intercepts({@Signature(
                                                                                                            type= Executor.class,
                                                                                     方法
                                                                                                            method = "update",
                                                                                                           args = {MappedStatement.class,Object.class})})
                                                                                                           public class ExamplePlugin implements Interceptor {
                                                                                                           public Object intercept(Invocation invocation) throws Throwable {
                                                                                                            return invocation.proceed();
                                                                                                           public Object plugin(Object target) {
                                                                                                            return Plugin.wrap(target, this);
                                                                                                           public void setProperties(Properties properties) {
                                                                                                           <!-- mybatis-config.xml -->
                                                                                                           <plugin interceptor="org.mybatis.example.ExamplePlugin">
                                                                                                            property name="someProperty" value="100"/>
                                                                                                            </plugin>
                                                                                                            </plugins>
                                                                                              可以根据不同的数据库厂商执行不同的语句,这种多厂商的支持是基于映射语句中的 databaseld 属性
                                                                                             MyBatis 会加载不带 databaseld 属性和带有匹配当前数据库 databaseld 属性的所有语句
                                                                                             <databaseIdProvider type="DB VENDOR" />
                                                                                             <databaseIdProvider type="DB VENDOR">
                                                              databaseldProvider
                                                                                              cproperty name="SQL Server" value="sqlserver"/>
                                                                                               oracle" />
                                                                                             </databaseIdProvider>
○10、XML 映射配置文件(三)
                                                                                          概念 🖯 用于引用定义好的映射定义,告诉mybatis去哪里找我们的sql定义配置
公众号java-mindmap
                                                                                                                                   <mapper resource="com/yu/res/UserMapper.xml" />
                                                                                                          11直接引用xml文件 ⊖
                                                                                                                                </mappers>
                                                             映射器 (mappers)
                                                                                                             通过绝对路径引用,注意
                                                                                                                                        <mapper url="file:///D:/workspace/mywork/MyBatis/src/com/yu/res/</pre>
                                                                                                            2 在绝对路径前加
                                                                                                                                       UserMapper.xml"/>
                                                                                                             上: "file:///"
                                                                                                                                       </mappers>
                                                                                            四中方式
                                                                                                                                            <mapper resource="com.yu.mapper.UserMapper" />
                                                                                                          3 引用mapper接口对象的方式
                                                                                                                                          </mappers>
                                                                                                                                         <package name="com.yu.mapper"/>
                                                                                                          4 引用mapper接口包的方式(
                                                                                                                                        </mappers>
                                                                                                            配置成适应多种环境,这种机制有助于将 SQL 映射应用于多种数据库之中
                                                                                                           ▶ 不过要记住:尽管可以配置多个环境,每个 SqlSessionFactory 实例只能选择其一。
                                                                                                               1)为了开发设置不同的数据库配置
                                                                                                  使用场景 ⊖ 2)测试和生产环境数据库不同
                                                                                                               3)有多个数据库却共享相同的模式,即对不同的数据库使用相同的SQL映射
                                                                                                              <!-- 对事务的管理和连接池的配置 -->
                                                                                                              <environments default="oracle_jdbc">
                                                                                                               <environment id="oracle_jdbc">
                                                                                                                 <transactionManager type="JDBC" />
                                                                                                                 <dataSource type="POOLED">
                                                                                                                 continue = "driver" value = "${jdbc.oracle.driverClassName}" />
                                                                                                                 contentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontentcontent<
                                                                                                                 operty name="username" value="${jdbc.oracle.username}" />
                                                                                                                 配置环境 (environments)
                                                                                                                 </dataSource>
                                                                                                                </environment>
                                                                                                                <environment id="mysql_jdbc">
                                                                                                                 <transactionManager type="JDBC" />
                                                                                                                 <dataSource type="POOLED">
                                                                                                                 cproperty name="driver" value="${jdbc.mysql.driverClassName}" />
                                                                                                                 comperty name="url" value="${jdbc.mysql.url}" />
                                                                                                                 coperty name="username" value="${jdbc.mysql.username}" />
                                                                                                                 operty name="password" value="${jdbc.mysql.password}" />
                                                                                                                 </dataSource>
                                                                                                                </environment>
                                                                                                              </environments>
                                                                                                  _________________用default指定默认的数据库链接::<environments default="oracle_jdbc">
                                                                                                                                            SqlSessionFactory factory = sqlSessionFactoryBuilder.build(reader, environment);
                                                                                                   根据数据库环境,获取SqlSessionFactory
                                                                                                                                            SqlSessionFactory factory = sqlSessionFactoryBuilder.build(reader, environment,properties);
                                                                    事务管理器
                                                                    例子 ⊝ <transactionManager type="JDBC" />
                                            transaction Manager
                                                                                  JDBC:简单的使用JDBC的提交和回滚设置,一览与从数据员得到的链接来管理事务范围
                                                                                  MANAGED:这个配置几乎什么都没做,它从来不提交或回滚一个链接,而它让容器来管理事务的整个生命周期
             environment中的配置
                                                                            <dataSource type="POOLED">
                                                                            cproperty name="driver" value="${jdbc.mysql.driverClassName}" />
                                                                            cproperty name="url" value="${jdbc.mysql.url}" />
                                                                            operty name="username" value="${jdbc.mysql.username}" />
                                               dataSource 🗎
                                                                            operty name="password" value="${jdbc.mysql.password}" />
                                                                 例子 🖯 </dataSource>
                                                                                UNPOOLED ⊙ 这个数据源的实现是每次被请求时打开和关闭连接。速度会有一些慢,适用于简单的应用程序。
                                                                               POOLED ⑤ 这是JDBC链接对象的数据源连接池的实现,用来避免创建新的链接实例时必要的连接和认证时间。适用于当前Web应用程序用来快速响应请求
                                                                                JNDI 🖯 这个数据源是为了使用如Spring或应用服务器这类的容器,容器可以集中或在外部配置数据源,然后设置JNDI上下文的引用
```