Ibatis读写CLOB数据

|  |
| --- |
| Ibatis是一个高效，方便，易于学习的数据访问组件，在性能上比hibernate高，学习难度也比hibernate和jdo要低，而且它比直接使用jdbc方便和易于维护。所以Ibatis深入大家的喜爱，一些对性能有更高的要求的系统（如保险，金融行业系统），或改造遗留系统时，Ibatis是数据访问组件的首选。 在使用Oracle数据库时，读取CLOB和BLOB等大类型的数据一直是个比较犯难的事，一般都是通过JDBC代码来实现对CLOB和BLOB数据的读写，效果和性能都是最好的，但是代码也相当复杂，且代码难以重用。 在使用ibatis作为数据访问组件，也经常会遇到要读取CLOB，BLOB大类型数据。怎样使用Ibatis读取CLOB,BLOB数据也是一个难题，并且Oracle在这方面一直没有解决好。 公司的项目正好有这方面的需要，要求我给予解决。在网上找了很多的资料，都没有一个比较简单易用的解决办法，通过不断的验实，终于得出了比较好的解决办法，所以写成文字，大家可以分享。如果大家以后有这方面的需要就可以直接使用，少走弯路，当然如果大家有更好的办法，希望能告诉我，我当不胜感激。  准备工作： 1． 测试数据库 CREATE TABLE USERINFO(USERID VARCHAR2(5),  USERNAME VARCHAR2(20), MEMO CLOB, constraint PK\_USERINFO primary key(USERID)); 2.域模型对象 UserInfoDTO.java  3.数据访问对象接口 UserInfoDao.java  一、 读取CLOB，BLOB类型数据的几种方法 1. jdbc实现 采用jdbc来读写是最原始，也是最直接的方法 UserInfoDaoImpl.java  2. 使用Spring的org.springframework.jdbc.support.lob.OracleLobHandler类处理 2.1 sql-map-config.xml的配置 <typeHandler jdbcType="CLOB" javaType="java.lang.String" callback="org.springframework.orm.ibatis.support.ClobStringTypeHandler"/> 2.2 sqlMapClient的配置 <bean id="sqlMapClient" class="org.springframework.orm.ibatis.SqlMapClientFactoryBean"> <property name="configLocation"><value>/sql-map-config.xml</value></property> <property name="dataSource"><ref local="dataSource"/></property> <property name="lobHandler"><ref local="oracleLobHandler"/></property> </bean>  <bean id="oracleLobHandler" class="org.springframework.jdbc.support.lob.OracleLobHandler" lazy-init="true">  特别注意：采用这种方法只对数据源是直接连接Oracle 的JDBC驱动方式有效，如果你采用数据连接池作为数据源，则这种办法无效。但是一般大型的项目都是使用连接池的，所以这个方法有很大的局限性。  3. 通过实现Ibatis的TypeHandlerCallback接口来实现 下面是读取的方法：  sqlMap 的配置 <resultMap id="userResult" class="com.prs.application.ehld.sample.common.dto.UserInfoDTO"> <result property="userID" column="USERID" columnIndex="1"/> <result property="userName" column="USERNAME" columnIndex="2"/> <result property="memo" column="memo" jdbcType="CLOB" javaType = "java.lang.String" typeHandler =" OracleClobTypeHandlerCallback "/> </resultMap>  二、 存在的问题 上面三种方法都存在的问题: 1. 使用jdbc来实现，就失去了使用ibatis带来的便利，同时也失去了自动事务管理的能力，还有需要自己去手工管理连接对象。代码量也相当复杂。 2. 使用spring提供的处理器，不支持采用数据连接池的数据源，有很大的局限性。  3. 通过实现ibatis的回调接口来实现，也有一定的局限性，需要新增一个类，配置也不方便，还可能会出错。  难道ibatis读写LOB大类型数据就没辙了吗？通过试验，发现可以配置ParameterMap和ResultMap就可以方便的实现对LOB的读写了，而且不用去实现任何类。只需要配置就可以了  三、 通过配置ParameterMap和ResultMap来实现对LOB类型的读写 3.1 sqlMap的配置 <resultMap id="userResult" class="com.prs.application.ehld.sample.common.dto.UserInfoDTO"> <result property="userID" column="USERID" columnIndex="1"/> <result property="userName" column="USERNAME" columnIndex="2"/> <result property="memo" column="memo" jdbcType="CLOB" javaType = "java.lang.String" /> </resultMap>  <parameterMap id="userPara" class = "com.prs.application.ehld.sample.common.dto.UserInfoDTO"> <parameter property="userID" jdbcType="VARCHAR" javaType ="java.lang.String"/> <parameter property="userName" jdbcType="VARCHAR" javaType ="java.lang.String"/> <parameter property="memo" jdbcType="CLOB" javaType ="java.lang.String"/>  </parameterMap> 3.2 插入和读取语句的配置 <select id="getUserInfoList" resultMap="userResult" > SELECT USERINFO.USERID , USERINFO.USERName , USERINFO.MEMO FROM USERINFO </select>  <insert id="insertUserInfo" parameterMap = "userPara" > INSERT INTO USERINFO(USERID, USERName, memo)  VALUES(?,?,? ) </insert> 注意：因为使用了ParameterMap作为输入参数，在插入语句中用?号来代替属性值(如：#userId#)  而不是常见的： INSERT INTO USERINFO(USERID, USERName, memo)  VALUES(#userId#, #userName#, #memo# ) 但是当paramaterMap的class属性是java.util.Map类时，应该使用#userId#类似的参数，不能用?来代替。  但是这又有一个问题，就是插入的字符串不能超过4000个字符，而CLOB类型的字段可以存4GB大小的字符。只要对userInfo对象的memo成员设置字符串超过了4000个字符，就提示“不能创建更多的套接字”,为什么会报这个错，暂是没有弄清楚。  而需求是要CLOB字段要存6000个汉字，相当于12000个英文字符。可以说是白忙了一场，没有达到目的。  四、 使用Oracle 10g的jdbc驱动程序 因为字符串只要超过了4000个字符就不能插入，所以不得不试着换一下驱动程序看看。一直以来都认为oracel的jdbc 驱动对处理LOB对象有一些问题，想看看10g出来后是不是有所改变。于是上网下载10g的驱动(class14.jar)，一阵痛苦的等待后，问题解决，我把字符串设为12万个字符也没有问题了。  另外采用10g 的驱动就算不使用parameterMap也可以成功的插入字符串到CLOB类型字段去，请要注意的是，这样只能插入的字符一定要小于32767个。也就是说我把memo 属性设置多于32766个字符，照样插不进去。这个原因主要是jdbc驱动限制了String的长度的原因。 <insert id="insertUserInfo" parameterClass = "UserInfoDTO" > INSERT INTO USERINFO(USERID, USERName, memo)  VALUES(#userID#, #userName#, #memo# ) </insert> 使用10g的驱动，这样能写入32766个字符  五、 怎么读写BLOB 上面都一直在说CLOB，其实把CLOB实现了，那么BLOB也同样简单，只是注意它的java类型，如果一个字段为BLOB类型，那么在parameterMap中jdbcType 为BLOB, 把javaType设为：[]byte就可以了。 例： <parameterMap id="userPara" class = "com.prs.application.ehld.sample.common.dto.UserInfoDTO"> <parameter property="userID" jdbcType="VARCHAR" javaType ="java.lang.String"/> <parameter property="userName" jdbcType="VARCHAR" javaType ="java.lang.String"/> <parameter property="memo" jdbcType="BLOB" javaType ="[]byte"/>  </parameterMap>  假设memo在数据库中为BLOB 类型  那么在javaBean中memo的java类型为[]byte  六、 总结 采用10g的驱动，和通过配置parameterMap和resultMap能够轻松和完美的解决LOB大型数据的读写，无需要编写新的java来实现，也没有局限性。 |