

轻松掌握jdbc操纵Oracle数据库lob字段 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/273/2021_2022__E8_BD_BB_E6_9D_BE_E6_8E_8C_E6_c102_273437.htm 在oracle数据库中

，lob（large object，大型对象）类型的字段使用的频率越来越高了。因为这种类型的字段，容量大（最多能容纳4gb的数据），且一个表中可以有多个这种类型的字段，很灵活，适用于数据量非常大的业务领域（如图象、档案等）。而long、long raw等类型的字段，虽然存储容量也不小（可达2gb），但由于一个表中只能有一个这样类型的字段的限制，现在已很少使用了。lob类型分为blob和clob两种：blob即二进制大型对象（binary large object），适用于存贮非文本的字节流数据（如程序、图象、影音等）。而clob，即字符型大型对象（character largeobject），则与字符集相关，适于存贮文本型的数据（如历史档案、大部头著作等）。下面以程序实例说明通过jdbc操纵oracle数据库lob类型字段。先建立如下两个测试用的数据库表，power designer pd模型如下：建表sql语句为：create table test_clob (id number(3), clobcol clob) create table test_blob (id number(3), blobcol blob) lob对象的存取 1、往数据库中插入一个新的clob对象 public static void clobinsert(string infile) throws exception { /* 设定不自动提交 */ boolean defaultcommit = conn.getautocommit(). conn.setautocommit(false). try { /* 插入一个空的clob对象 */ stmt.executeUpdate("insert into test_clob values (' 111 ' , empty_clob())"). /* 查询此clob对象并锁定 */ resultset rs = stmt.executeQuery("0select clobcol from test_clob where id= ' 111

```
' for 0update"). while (rs.next()) { /* 取出此clob对象 */
oracle.sql.clob clob = (oracle.sql.clob)rs.getclob("clobcol"). /*
向clob对象中写入数据 */ bufferedwriter out = new
bufferedwriter(clob.getcharacteroutputstream()). bufferedreader in
= new bufferedreader(new filereader(infile)). int c. while
((c=in.read())!=-1) { out.write(c). } in.close(). out.close(). } /* 正式
提交 */ conn.commit(). } catch (exception ex) { /* 出错回滚 */
conn.rollback(). throw ex. } /* 恢复原提交状态 */
conn.setautocommit(defaultcommit). } 100Test 下载频道开通 ,
各类考试题目直接下载。 详细请访问 www.100test.com
```