**Java程序实现oracle远程数据库的备份和恢复**

原创 2017年04月17日 10:59:06

* 1544

    最近项目到了收尾的阶段，为了使我们的系统功能更加完成，也为了使客户更加安心，决定添加一个oracle备份和还原的功能。现做一下分享。

一、导入导出命令

    oracle备份和还原除了使用工具比如PL/SQL进行导出导入还可以利用命令，在程序中也就只能调用外部命令了，首先我们得知晓oracle备份和还原的命令，根据需要我是需要备份和还原项目数据库的所有的表的，以下命令为备份和还原所有表命令。

   备份：exp username/password@TEST file=D:\export.dmp

   还原：imp username/password@TEST file=D:\export.dmp full=y ignore=y

二、Java代码实现

   调用runtime方法，因为是远程的数据库，所以SID为ip+"/"+数据库名

**[java]** [view plain](http://blog.csdn.net/ww130929/article/details/70208744) [copy](http://blog.csdn.net/ww130929/article/details/70208744)

1. /\*\*
2. \* 实现Oracle数据库导出
3. \* @author weiwenwen
4. \* userName 进入数据库所需要的用户名
5. \* password 进入数据库所需要的密码
6. \* SID 用户所在的数据库名
7. \* avePath 数据库导出文件保存路径
8. \* fileName 数据库导出文件文件名
9. \* @return 返回0表示导出成功
10. \* @throws Exception
11. \*/
12. **public**  **int** oraBackup() **throws** IOException{
13. //读取配置文件配置项
14. String userName=Global.getConfig("jdbc.username");
15. String password=Global.getConfig("jdbc.password");
16. String savePath=Global.getConfig("oraBackupPath");
17. String SID=Global.getConfig("SID");
18. String id=IdGen.uuid();
20. File saveFile = **new** File(savePath);
21. **if** (!saveFile.exists()) {// 如果目录不存在
22. saveFile.mkdirs();// 创建文件夹
23. }
24. //将记录插入数据库
25. PageData pd=**new** PageData();
26. pd.put("id", id);
27. pd.put("user\_name", UserUtils.getUser().getName());
28. pd.put("op\_dt", **new** Date());
30. oracleBackupDao.insertOraBackup(pd);
32. //所有空表
33. List<String> tablesList=oracleBackupDao.findAltTables();
34. //为所有空表执行语句
35. **for** (**int** i = 0; i < tablesList.size(); i++) {
36. oracleBackupDao.doAltSql(tablesList.get(i));
37. }
38. String command ="exp " + userName + "/" + password + "@" + SID + " file=" + savePath + id + ".dmp";
39. System.out.println("备份数据库命令操作:" + command);
41. Process process = Runtime.getRuntime().exec(command);
43. **final** InputStream is1 = process.getInputStream();
45. **new** Thread(**new** Runnable() {
46. **public** **void** run() {
47. BufferedReader br = **new** BufferedReader(**new** InputStreamReader(is1));
48. String info;
49. **try** {
50. **while** ((info=br.readLine()) != **null**){
51. System.out.println("info: "+info);
52. }
53. } **catch** (IOException e) {
54. e.printStackTrace();
55. }
56. }
57. }).start(); // 启动单独的线程来清空process.getInputStream()的缓冲区
58. InputStream is2 = process.getErrorStream();
59. BufferedReader br2 = **new** BufferedReader(**new** InputStreamReader(is2));
60. // 保存输出结果
61. StringBuilder buf = **new** StringBuilder();
62. String line = **null**;
63. **int** i=0;
64. **while** ((line = br2.readLine()) != **null**){
65. // 循环等待ffmpeg进程结束
66. System.out.println("info: " +line);
67. buf.append(line);
68. }
69. **try** {
70. **if**(buf.toString().contains("ORA-")&&buf.toString().contains("EXP-")){
71. System.err.println("备份失败！");
72. process.destroy();
73. }**else**{
74. i=process.waitFor();
75. System.out.println("over status: "+i);
76. }
77. } **catch** (InterruptedException e) {
78. e.printStackTrace();
79. }
80. System.out.println("备份结束...");
82. //备份完成将文件大小更新到表，备份的文件路径
83. String dmpPath = savePath + id + ".dmp";
84. //获取文件大小
85. pd.put("file\_size", **new** File(dmpPath).length());
86. oracleBackupDao.updateFileSize(pd);
87. **return** 0;
88. }

PS:因为是oracle11g,发现11G中新特性，当表无数据时，不分配segment，以节省空间。而使用exp命令时，无Segment的表不会被导出。所以如果单纯直接执行导出语句是不可以的，必须查询所有的空表后，执行alter命令。

**[sql]** [view plain](http://blog.csdn.net/ww130929/article/details/70208744) [copy](http://blog.csdn.net/ww130929/article/details/70208744)

1. **select** table\_name **from** user\_tables **where** NUM\_ROWS=0;

如果平时导出的话直接执行这句：

**[sql]** [view plain](http://blog.csdn.net/ww130929/article/details/70208744) [copy](http://blog.csdn.net/ww130929/article/details/70208744)

1. **select** 'alter table '||table\_name||' allocate extent;' **from** user\_tables **where** num\_rows=0

最后为所有的表执行alter命令

**[sql]** [view plain](http://blog.csdn.net/ww130929/article/details/70208744) [copy](http://blog.csdn.net/ww130929/article/details/70208744)

1. **alter** **table** '表名' allocate extent;

    导入还原的代码是类似的，根据命令自己拼字符串即可，不再累述

小结：

    1.看似一个小的功能，当简单代码的完成后的测试更是重头戏，要考虑表是不是全部导出，每个表的数据是不是完全，要考虑备份导出过程中的突发情况，比如网线断连接。

    2.导入还原时添加full=y ignore=y,否则导入会报错

    3.如果原数据库是有数据的直接导入的话数据会在原基础上添加，所以可能存在数据重复现象，试了很多办法不可以，所以我的处理是在还原时查询所有表，用truncate强制执行清表的操作，再重新导入。