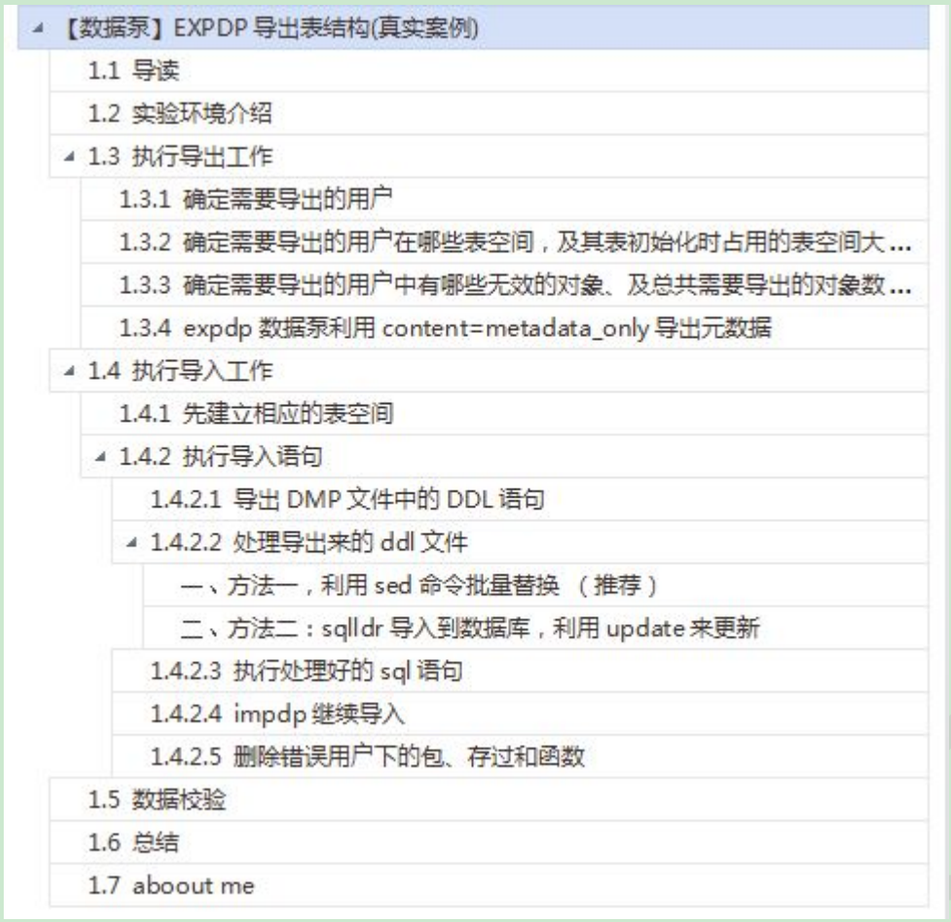
【数据泵】EXPDP 导出表结构(真实案例)

BLOG 文档结构图



因工作需要现需要把一个生产库下的元数据（表定义，索引定义，函数定义，包定义，存储过程）导出到测试库上，本来以为很简单的，可是做的过程发现很多的问题，现记录如下，希望有同样需要的朋友不要再走弯路了。

1.1 导读

各位技术爱好者，看完本文后，你可以掌握如下的技能，也可以学到一些其它你所不知道的知识，~O(∩_∩)O~：

① EXPDP 和 IMPDP 如何导出导入元数据，包括表定义，索引定义，函数定义，包定义，存储过程（重点）

② 表的初始化定义参数 initial，及如何批量修改该参数

③ 如何导出 DMP 文件中的 DDL 语句（重点）

④ 10g 和 11g 默认情况下有哪些用户及其作用

⑤ linux 中的批量替换 sed 命令

⑥ sqlldr 和 spool 命令

本文如有错误或不完善的地方请大家多多指正，ITPUB 留言或 QQ 皆可，您的批评指正是我写作的最大动力。

1.2 实验环境介绍

源库：10.2.0.1　AIX

目标库：11.2.0.3　RHEL6.5

1.3 执行导出工作

1.3.1　　确定需要导出的用户

oracle 安装好后有很多的系统默认用户，比如 sys 和 system，对于这 2 个用户里的元数据我们就没有必要再重新导出嘛，不然导入的时候还提示错误，看着实在不好。

官网信息：

All databases created by the Database Configuration Assistant (DBCA) include the SYS, SYSTEM, SYSMAN, and DBSNMP database accounts. In addition, Oracle Database provides several other administrative accounts. Before using these accounts, you must unlock them and reset their passwords.

11g 默认用户比较多：

User Name	Description	See Also
ANONYMOUS	Enables HTTP access to Oracle XML DB.	Oracle XML DB Developer's Guide
APEX_030200	The account owns the Application Express schema and metadata.	Oracle Application Express Application Builder User's Guide

APEX_PUBLIC_USER	The minimally privileged account used for Application Express configuration with Oracle HTTP Server and mod_plsql.	Oracle Application Express Application Builder User's Guide
APPQOSSYS	Used for storing and managing all data and metadata required by Oracle Quality of Service Management.	None
BI	The account that owns the Business Intelligence schema included in the Oracle Sample Schemas. It is available only if you loaded the sample schemas.	Oracle Database Sample Schemas
CTXSYS	The Oracle Text account.	Oracle Text Reference
DBSNMP	The account used by the Management Agent component of Oracle Enterprise Manager to monitor and manage the database.	Oracle Enterprise Manager Grid Control Installation and Basic Configuration
DIP	The account used by the Directory Integration Platform (DIP) to synchronize the changes in Oracle Internet Directory with the applications in the database.	None
DVSYS	There are two roles associated with this account. The Database Vault owner role manages the Database Vault roles and configurations. The Database Vault Account Manager is used to manage database user accounts. Note: Part of Oracle Database Vault user interface text is stored in database tables in the DVSYS schema. By default, only the English language is loaded into these tables. You can use Oracle Database Vault Configuration Assistant to add more languages to Oracle Database Vault. For the necessary steps, see Appendix C in Oracle Database Vault Administrator's Guide	Oracle Database Vault Administrator's Guide
EXFSYS	The account owns the Expression Filter schema.	None
FLOWS_FILES	The account owns the Application Express uploaded files.	Oracle Application Express Application Builder User's Guide
HR	The account that owns the Human Resources schema included in the Oracle Sample Schemas. It is available only if you loaded the sample schemas.	Oracle Database Sample Schemas
IX	The account that owns the Information Transport schema included in the Oracle Sample Schemas. It is available only if you loaded the sample schemas.	Oracle Database Sample Schemas
LBACSYS	The Oracle Label Security administrator account.	Oracle Label Security Administrator's Guide
MDDATA	The schema used by Oracle Spatial for storing geocoder and router data.	Oracle Spatial Developer's Guide

MDSYS	The Oracle Spatial and Oracle Multimedia Locator administrator account.	Oracle Spatial Developer's Guide
MGMT_VIEW	An account used by Oracle Enterprise Manager Database Control.	None
OE	The account that owns the Order Entry schema included in the Oracle Sample Schemas. It is available only if you loaded the sample schemas.	Oracle Database Sample Schemas
ORDPLUGINS	The Oracle Multimedia user. Plug-ins supplied by Oracle and third-party plug-ins are installed in this schema.	Oracle Multimedia Reference
ORDSYS	The Oracle Multimedia administrator account.	Oracle Multimedia Reference
ORDDATA	This account contains the Oracle Multimedia DICOM data model.	Oracle Multimedia DICOM Developer's Guide
OUTLN	The account that supports plan stability. Plan stability enables you to maintain the same execution plans for the same SQL statements. OUTLN acts as a role to centrally manage metadata associated with stored outlines.	Oracle Database Concepts
ORACLE_OCM	This account contains the instrumentation for configuration collection used by the Oracle Configuration Manager.	Oracle Configuration Manager Installation and Administration Guide
OWBSYS	The account used by Oracle Warehouse Builder as its default repository. You must unlock this account after installing the Oracle Database and before launching the Warehouse Builder Repository Assistant.	Oracle Warehouse Builder Installation and Administration Guide
OWBSYS_AUDIT	This account is used by the Warehouse Builder Control Center Agent to access the heterogeneous execution audit tables in the OWBSYS schema.	Oracle Warehouse Builder Installation and Administration Guide
PM	The account that owns the Product Media schema included in the Oracle Sample Schemas. It is available only if you loaded the sample schemas.	Oracle Database Sample Schemas
SCOTT	An account used by Oracle sample programs and examples.	Oracle Database Administrator's Guide
SH	The account that owns the Sales History schema included in the Oracle Sample Schemas. It is available only if you loaded the sample schemas during an Enterprise Edition installation.	Oracle Database Administrator's Guide
SI_INFORMTN_SCHEMA	The account that stores the information views for the SQL/MM Still Image Standard.	Oracle Multimedia Reference
SPATIAL_CSW_ADMIN_USR	The Catalog Services for the Web (CSW) account. It is used by the Oracle Spatial CSW cache manager to load all record type metadata, and record instances from the database into the main memory for the record types that are cached.	Oracle Spatial Developer's Guide

SPATIAL_WFS_ADMIN_USR	The Web Feature Service (WFS) account. It is used by the Oracle Spatial WFS cache manager to load all feature-type metadata, and feature instances from the database into main memory for the feature types that are cached.	Oracle Spatial Developer's Guide
SYS	The account used to perform database administration tasks.	Oracle Database Administrator's Guide
SYSMAN	The account used to perform Oracle Enterprise Manager database administration tasks.	Oracle Enterprise Manager Grid Control Installation and Basic Configuration
SYSTEM	Another account used to perform database administration tasks.	Oracle Database Administrator's Guide
WMSYS	The account used to store the metadata information for Oracle Workspace Manager.	Oracle Database Workspace Manager Developer's Guide
XDB	The account used for storing Oracle XML DB data and metadata.	Oracle XML DB Developer's Guide

10g 下比较少：

CTXSYS	CTXSYS	The Oracle Text account	*Oracle Text Reference*
DBSNMP	DBSNMP	The account used by the Management Agent component of Oracle Enterprise Manager to monitor and manage the database	*Oracle Enterprise Manager Grid Control Installation and Basic Configuration*
LBACSYS	LBACSYS	The Oracle Label Security administrator account	*Oracle Label Security Administrator's Guide*
MDDATA	MDDATA	The schema used by Oracle Spatial for storing Geocoder and router data	*Oracle Spatial User's Guide and Reference*
MDSYS	MDSYS	The Oracle Spatial and Oracle *inter*Media Locator administrator account	*Oracle Spatial User's Guide and Reference*
DMSYS	DMSYS	The Oracle Data Mining account.	*Oracle Data Mining Administrator's Guide* *Oracle Data Mining Concepts*
OLAPSYS	MANAGER	The account used to create OLAP metadata structures. It owns the OLAP Catalog (CWMLite).	*Oracle OLAP Application Developer's Guide*

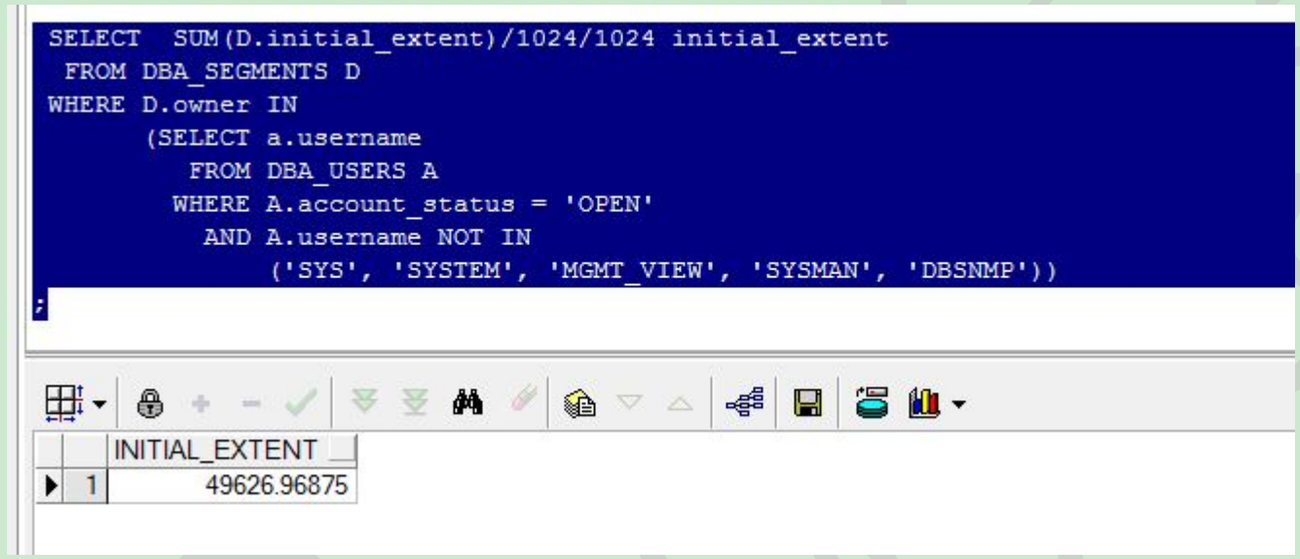
ORDPLUGINS	ORDPLUGINS	The Oracle *inter*Media user. Plug-ins supplied by Oracle and third party format plug-ins are installed in this schema.	*Oracle interMedia User's Guide*
ORDSYS	ORDSYS	The Oracle *inter*Media administrator account	*Oracle interMedia User's Guide*
OUTLN	OUTLN	The account that supports plan stability. Plan stability enables you to maintain the same execution plans for the same SQL statements. OUTLN acts as a role to centrally manage metadata associated with stored outlines.	*Oracle Database Performance Tuning Guide*
SI_INFORMTN_SCHEMA	SI_INFORMTN_SCHEMA	The account that stores the information views for the SQL/MM Still Image Standard	*Oracle interMedia User's Guide*
SYS	CHANGE_ON_INSTALL	The account used to perform database administration tasks	*Oracle Database Administrator's Guide*
SYSMAN	CHANGE_ON_INSTALL	The account used to perform Oracle Enterprise Manager database administration tasks. Note that SYS and SYSTEM can also perform these tasks.	*Oracle Enterprise Manager Grid Control Installation and Basic Configuration*
SYSTEM	MANAGER	Another account used to perform database administration tasks.	*Oracle Database Administrator's Guide*

1.3.2 确定需要导出的用户在哪些表空间，及其表初始化时占用的表空间大小

```
SELECT D.tablespace_name, SUM(D.initial_extent)/1024/1024 initial_extent
  FROM DBA_SEGMENTS D
 WHERE D.owner IN
       (SELECT a.username
          FROM DBA_USERS A
         WHERE A.account_status = 'OPEN'
           AND A.username NOT IN
               ('SYS', 'SYSTEM', 'MGMT_VIEW', 'SYSMAN', 'DBSNMP'))
 GROUP BY D.tablespace_name
 ORDER BY initial_extent desc ;
```

	TABLESPACE_NAME	INITIAL_EXTENT
1	SD_DPA_F_01	19131.3125
2	DWII_SOR_F_01	16981.5625
3	DWII_DPA_F_01	6005.6875
4	SD_CNY_F_01	5168.1875
5	DWII_FXDM_F_01	685.03125
6	USERS	540.5
7	DW_USER	289.875
8	SD_SOR_T_01	246.6875
9	SD_DPA_D_01	245.375
10	DWII_DPA_I_01	243.875
11	SD_CNY_D_01	84.9375
12	SQCHECK	2.375
13	DWII_CNY_BK_F_01	1.4375
14	DWII_DPA_S_01	0.0625
15	DWII_SOR_I_01	0.0625

```
SELECT  SUM(D.initial_extent)/1024/1024 initial_extent
  FROM DBA_SEGMENTS D
 WHERE D.owner IN
      (SELECT a.username
         FROM DBA_USERS A
        WHERE A.account_status = 'OPEN'
          AND A.username NOT IN
              ('SYS', 'SYSTEM', 'MGMT_VIEW', 'SYSMAN', 'DBSNMP'))
;
```



由此可以知道，创建这些元数据大约需要 49G 的空间，如果涉及到数据的话，还需要判断数据占用空间，这里一定需要判断这个，不然执行导入的时候会因为表空间不足而不能导入，我第一次导入的时候就是因为这里没有判断导致花费了很长的时间，我一直扩展表空间，但是就是就表空间不足的错误(ORA-01659),想想建表不会花这么大的空间的吧，最后查看了表的定义才知道，原来表初始化的时候就很大，这个问题后边还需要再处理一下的，不然测试库没法导入，当然存储够的话就另当别论了。

ORA-39171: Job is experiencing a resumable wait.

ORA-01659: unable to allocate MINEXTENTS beyond 4 in tablespace DWII_SOR_F_01

1.3.3 确定需要导出的用户中有哪些无效的对象、及总共需要导出的对象数量

这一步也很重要，决定着最终导出结果的正确性验证。

```
SELECT d.OWNER,count(1)
FROM    dba_objects d
WHERE   d.OWNER in (SELECT a.username
        FROM DBA_USERS A
       WHERE A.account_status = 'OPEN'
         AND A.username NOT IN
             ('SYS', 'SYSTEM', 'MGMT_VIEW', 'SYSMAN', 'DBSNMP'))
group by  d.OWNER;


SELECT d.OWNER,
       d.status,count(1)
FROM    dba_objects d
WHERE   d.OWNER in (SELECT a.username
        FROM DBA_USERS A
       WHERE A.account_status = 'OPEN'
         AND A.username NOT IN
             ('SYS', 'SYSTEM', 'MGMT_VIEW', 'SYSMAN', 'DBSNMP'))
group by  d.OWNER, d.status;

SELECT d.OWNER,
       d.OBJECT_NAME,
       d.OBJECT_TYPE,
       d.status
FROM    dba_objects d
WHERE   d.status = 'INVALID'
and d.owner in (SELECT a.username
        FROM DBA_USERS A
       WHERE A.account_status = 'OPEN'
         AND A.username NOT IN
             ('SYS', 'SYSTEM', 'MGMT_VIEW', 'SYSMAN', 'DBSNMP'));
```

1.3.4 expdp 数据泵利用 content=metadata_only 导出元数据

导出命令，注意这里不导出数据只导出定义我们采用 content=metadata_only 来处理：

expdp lhr/lhr directory=DATA_PUMP_DIR dumpfile=lhrsql20150515.dmp logfile=lhrsql20150515.log content=metadata_only schemas=TEST,SQCHECK,DWUSER,DPA,CNYDM,ONL1,LHR,TEST1,FXDM,DWII_ETL,DWUSER1,SOR,DW_ETL,NRDM,NRDM_ETL,FXDM_ETL,LCM2,CNY_ETL

由于是事后写文档，所以这里只贴出导出元数据的日志：

```
;;;
Export: Release 10.2.0.1.0 - 64bit Production on Friday, 15 May, 2015 13:05:54

Copyright (c) 2003, 2005, Oracle.  All rights reserved.
;;;
Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options
Starting "LHR"."SYS_EXPORT_SCHEMA_01":  lhr/******** directory=DATA_PUMP_DIR dumpfile=lhrsql20150515.dmp logfile=lhrsql20150515.log content=metadata_only
schemas=TEST,SQCHECK,DWUSER,DPA,CNYDM,ONL1,LHR,TEST1,FXDM,DWII_ETL,DWUSER1,SOR,DW_ETL,NRDM,NRDM_ETL,FXDM_ETL,LCM2,CNY_ETL
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/DB_LINK
Processing object type SCHEMA_EXPORT/SEQUENCE/SEQUENCE
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/COMMENT
Processing object type SCHEMA_EXPORT/PACKAGE/PACKAGE_SPEC
Processing object type SCHEMA_EXPORT/FUNCTION/FUNCTION
Processing object type SCHEMA_EXPORT/FUNCTION/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
Processing object type SCHEMA_EXPORT/PROCEDURE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/PACKAGE/COMPILE_PACKAGE/PACKAGE_SPEC/ALTER_PACKAGE_SPEC
Processing object type SCHEMA_EXPORT/FUNCTION/ALTER_FUNCTION
Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
Processing object type SCHEMA_EXPORT/VIEW/VIEW
Processing object type SCHEMA_EXPORT/VIEW/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/VIEW/COMMENT
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Master table "LHR"."SYS_EXPORT_SCHEMA_01" successfully loaded/unloaded
******************************************************************************
Dump file set for LHR.SYS_EXPORT_SCHEMA_01 is:
  /oracle/product/10.2.0/db_1/rdbms/log/lhrsql20150515.dmp
Job "LHR"."SYS_EXPORT_SCHEMA_01" successfully completed at 13:09:49
```

接下来就是把导出来的文件利用 ftp 工具或 scp 传递到目标端，我这里就直接传到我的虚拟机上吧。

1.4 执行导入工作

1.4.1 先建立相应的表空间

根据上边的工作我们知道需要在测试库上建立以下的表空间，我这里都设置的是 20M，实际情况下应该根据上边查询出来的结果来设置相应的大小，**表空间应用包含用户的默认表空间及用户下对象所在的表空间**：

```
create tablespace  DWII_CNY_BK_F_01 datafile '+DATA' size 20M;
create tablespace  DWII_DPA_F_01 datafile '+DATA' size 20M;
create tablespace  DWII_DPA_I_01 datafile '+DATA' size 20M;
create tablespace  DWII_DPA_S_01 datafile '+DATA' size 20M;
create tablespace  DWII_SOR_F_01 datafile '+DATA' size 20M;
create tablespace  DWII_SOR_I_01 datafile '+FRA' size 20M;
create tablespace  DW_USER datafile '+FRA' size 20M;
create tablespace  SQCHECK datafile '+FRA' size 20M;
create tablespace  SD_CNY_D_01 datafile '+FRA' size 20M;
create tablespace  SD_CNY_F_01 datafile '+FRA' size 20M;
create tablespace  SD_DPA_D_01 datafile '+FRA' size 20M;
create tablespace  SD_DPA_F_01 datafile '+FRA' size 20M;
create tablespace  SD_SORT_T_01 datafile '+FRA' size 20M;
create tablespace  DWII_FXDM_F_01 datafile '+FRA' size 20M;
create tablespace  SD_SOR_T_01 datafile '+FRA' size 20M;
```

如果空间不够，我们可以追加数据文件：alter tablespace DWII_DPA_F_01 add datafile '+FRA' size 50M;

1.4.2 执行导入语句

一般情况下，如果存储够的话，我们把相应的表空间设置大一点之后这里直接执行导入语句就可以了，但是我是在本机的虚拟机里执行的，由前边的情况我们可以知道大约需要 49G 的空间，这个显然不太合适，哪该怎么办呢？我能想到的办法只有如下 2 种，如果大家还有好的办法可以给我留言。

① 在源库上修改表的定义后然后再执行导出命令

② 从已经导出来的 dmp 文件中抽取其中的 DDL 语句，然后将 DDL 语句导入到数据库中，update 掉其中的 STORAGE(INITIAL 参数后再将语句导出到 sql 文本中执行 sql 语句，这样可以解决表的定义问题。

③ 从已经导出来的 dmp 文件中抽取其中的 DDL 语句，然后利用 linux 的 sed 批量替换功能替换掉不正确的参数。

显然，第一种比较方便，也比较快，但是不实用，由于是生产库，参数不能随便修改，我们就采用第二或第三种办法，也可以多演示一种 impdp 的用法。

1.4.2.1 导出 DMP 文件中的 DDL 语句

我们在 impdp 的导入命令中添加 sqlfile 参数后执行导入并不会真正将数据导入到数据库，而会抽取出 dmp 文件中的 DDL 语句，如下：

```
[oracle@rhel6_lhr dpdump]$ impdp  lhr/lhr  directory=DATA_PUMP_DIR dumpfile=lhrsql20150515.dmp  logfile=imp_exptest.log sqlfile=lhrsql20150515.sql

Import: Release 11.2.0.3.0 - Production on Fri May 15 15:08:03 2015

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options


Master table "LHR"."SYS_SQL_FILE_FULL_01" successfully loaded/unloaded
Starting "LHR"."SYS_SQL_FILE_FULL_01":  lhr/******** directory=DATA_PUMP_DIR dumpfile=lhrsql20150515.dmp logfile=imp_exptest.log sqlfile=lhrsql20150515.sql
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/DB_LINK
Processing object type SCHEMA_EXPORT/SEQUENCE/SEQUENCE
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/COMMENT
Processing object type SCHEMA_EXPORT/PACKAGE/PACKAGE_SPEC
Processing object type SCHEMA_EXPORT/FUNCTION/FUNCTION
Processing object type SCHEMA_EXPORT/FUNCTION/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
Processing object type SCHEMA_EXPORT/PROCEDURE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/PACKAGE/COMPILE_PACKAGE/PACKAGE_SPEC/ALTER_PACKAGE_SPEC
Processing object type SCHEMA_EXPORT/FUNCTION/ALTER_FUNCTION
Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
Processing object type SCHEMA_EXPORT/VIEW/VIEW
Processing object type SCHEMA_EXPORT/VIEW/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/VIEW/COMMENT
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Job "LHR"."SYS_SQL_FILE_FULL_01" successfully completed at 15:21:50

[oracle@rhel6_lhr dpdump]$ ll lhrsql20150515.sql
-rw-r--r-- 1 oracle asmadmin 65707967 May 15 15:21 lhrsql20150515.sql
[oracle@rhel6_lhr dpdump]$ more lhrsql20150515.sql
-- CONNECT LHR
ALTER SESSION SET EVENTS '10150 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10904 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '25475 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10407 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10851 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '22830 TRACE NAME CONTEXT FOREVER, LEVEL 192 ';
-- new object type path: SCHEMA_EXPORT/USER
-- CONNECT SYSTEM
 CREATE USER "CNY_ETL" IDENTIFIED BY VALUES '4686A1050F638F44'
      DEFAULT TABLESPACE "DW_USER"
      TEMPORARY TABLESPACE "TEMP";

 CREATE USER "LCM2" IDENTIFIED BY VALUES '48BCFDF435352212'
      DEFAULT TABLESPACE "DWII_SOR_F_01"
      TEMPORARY TABLESPACE "TEMP";

 CREATE USER "FXDM_ETL" IDENTIFIED BY VALUES 'EA010AEA839BFA14'
      DEFAULT TABLESPACE "DW_USER"
      TEMPORARY TABLESPACE "TEMP";

 CREATE USER "NRDM_ETL" IDENTIFIED BY VALUES '54A4A046AEE8B31E'
      DEFAULT TABLESPACE "DW_USER"
```

```
      TEMPORARY TABLESPACE "TEMP";

 CREATE USER "NRDM" IDENTIFIED BY VALUES '1AE3DF7368DF560D'
      DEFAULT TABLESPACE "SD_CNY_F_01"
      TEMPORARY TABLESPACE "TEMP";

 CREATE USER "DW_ETL" IDENTIFIED BY VALUES '91635F9C0744E7EC'
      DEFAULT TABLESPACE "DW_USER"
      TEMPORARY TABLESPACE "TEMP";

 CREATE USER "SOR" IDENTIFIED BY VALUES 'BA3A6C912E6BFF14'
      DEFAULT TABLESPACE "DWII_SOR_F_01"
      TEMPORARY TABLESPACE "TEMP";

。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。省略



[oracle@rhel6_lhr dpdump]$ tail -n 50 lhrsql20150515.sql
  c := 'SPOT_EXCHNG_RATE_SRC';
  EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
               2,.5,2,156,0,2.65784513562818E+35,2.65784513872303E+35,5,0,nv,nv,
               TO_DATE('2015-05-14 22:00:18',df),'33303031','33303032',nv,2,nv;
  c := 'SPOT_EXCHNG_RATE_SRC_NM';
  EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
               2,.5,2,156,0,1.18758942587854E+36,1.20817519861257E+36,12,0,nv,nv,
               TO_DATE('2015-05-14 22:00:18',df),'E4B8ADE997B4E4BBB7','E8AFA2E4BBB7E59D87E580BC',nv,2,nv;
  c := 'FLEG_SWAP_PNT';
  EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
               14,.0714285714285714,14,156,0,-64.01,16.5,4,0,nv,nv,
               TO_DATE('2015-05-14 22:00:18',df),'3E256466','C11133',nv,2,nv;
  c := 'SRC_SYS_LBL';
  EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
               1,1,1,156,0,3.44097282552972E+35,3.44097282552972E+35,5,0,nv,nv,
               TO_DATE('2015-05-14 22:00:18',df),'42454E4D','42454E4D',nv,2,nv;

END;
/

DECLARE
  c varchar2(60);
  nv varchar2(1);
  df varchar2(21) := 'YYYY-MM-DD:HH24:MI:SS';
  s varchar2(60) := 'DPA';
  t varchar2(60) := 'BNCHMK_OPTN_DLT_PARAM_F';
  p varchar2(1);
  sp varchar2(1);
  stmt varchar2(300) := 'INSERT INTO "SYS"."IMPDP_STATS" (type,version,c1,c2,c3,c4,c5,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,d1,r1,r2,ch1,flags,cl1) VALUES
(:1,6,:2,:3,:4,:5,:6,:7,:8,:9,:10,:11,:12,:13,:14,:15,:16,:17,:18,:19,:20,:21,:22,:23)';
BEGIN
  NULL;
  c := 'CRT_TMST';
  EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
               1,1,1,156,0,2457157.57244213,2457157.57244213,11,0,nv,nv,
               TO_DATE('2015-05-14 22:00:18',df),'7873050E0E2D14','7873050E0E2D14',nv,2,nv;
  c := 'PPLN_WKDT';
  EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
               1,1,1,156,0,20150422,20150422,6,0,nv,nv,
               TO_DATE('2015-05-14 22:00:18',df),'C415100517','C415100517',nv,2,nv;
  c := 'PPLN_TMST';
  EXECUTE IMMEDIATE stmt USING 'C',t,p,sp,c,s,
               1,1,1,156,0,2457157.57244213,2457157.57244213,11,0,nv,nv,
               TO_DATE('2015-05-14 22:00:18',df),'7873050E0E2D14','7873050E0E2D14',nv,2,nv;

  DBMS_STATS.IMPORT_TABLE_STATS('"DPA"','"BNCHMK_OPTN_DLT_PARAM_F"',NULL,'"IMPDP_STATS"',NULL,NULL,'"SYS"');
  DELETE FROM "SYS"."IMPDP_STATS";
END;
```

```
/


[oracle@rhel6_lhr dpdump]$
```

可以看到生成的 SQL 语句开始部分是 ddl 的建用户语句，最后是导入表的统计信息部分。

1.4.2.2 处理导出来的 ddl 文件

一、 方法一，利用 sed 命令批量替换 （推荐）

这一步其实如果 linux 的 sed 命令熟悉的话可以直接替换掉 sql 语句中的相关行，我今天也是临时网上搜了下，经过大量实验，然后居然成功了，如下：

```
[oracle@rhel6_lhr dpdump]$ sed ’s/^  STORAGE(INITIAL.*/STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645/g’ lhrsql20150515.sql > lhrsql20150515.sql_bk2
[oracle@rhel6_lhr dpdump]$ more lhrsql20150515.sql_bk2  | grep ″STORAGE(INITIAL″
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
[oracle@rhel6_lhr dpdump]$ more lhrsql20150515.sql_bk2  | grep ″STORAGE(INITIAL 131072″
[oracle@rhel6_lhr dpdump]$  more lhrsql20150515.sql  | grep ″STORAGE(INITIAL 131072″
  STORAGE(INITIAL 131072 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  STORAGE(INITIAL 131072 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  STORAGE(INITIAL 131072 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  STORAGE(INITIAL 131072 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  STORAGE(INITIAL 131072 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  STORAGE(INITIAL 131072 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
。。。。。。。。。。。。。
```

可以看到替换已经成功，如果不熟悉该命令，那么先熟悉该命令就麻烦了。

二、 方法二：sqlldr 导入到数据库，利用 update 来更新

都是搞数据库的，这样做虽然麻烦点，但是绝对不会出错，而且也是一种技巧，如果有的文本很大又很难处理的话我们就可以导入到数据库中，然后处理。

首先建表：

```
create table imp_sql_lhr (id number ,text  varchar2(4000))  ;
```

sqlldr 的控制文件内容:sqlldr_table.ctl ：

```
UNRECOVERABLE
load data
LENGTH CHARACTER
infile 'lhrsql20150515.sql'
APPEND imp_sql_lhr
trailing nullcols
(
id SEQUENCE(1,1),
text  char(4000) "TRIM(:text)"

)
```

sqlldr 命令：

```
sqlldr lhr/lhr  control=sqlldr_table.ctl  log=a.log   parallel=y  readsize=4194304 streamsize=10485760 date_cache=5000 direct=true
```

导入到数据库后，我们就可以非常方便的来处理表中的数据了，如下：

```
SELECT * FROM imp_sql_lhr a ;
```

	ID	TEXT
1	1	ALTER SESSION SET EVENTS '10150 TRACE NAME CONTEXT FOREVER, LEVEL 1';
2	2	ALTER SESSION SET EVENTS '10904 TRACE NAME CONTEXT FOREVER, LEVEL 1';
3	3	ALTER SESSION SET EVENTS '25475 TRACE NAME CONTEXT FOREVER, LEVEL 1';
4	4	ALTER SESSION SET EVENTS '10407 TRACE NAME CONTEXT FOREVER, LEVEL 1';
5	5	ALTER SESSION SET EVENTS '10851 TRACE NAME CONTEXT FOREVER, LEVEL 1';
6	6	ALTER SESSION SET EVENTS '22830 TRACE NAME CONTEXT FOREVER, LEVEL 192 ';
7	7	-- new object type path: SCHEMA_EXPORT/USER
8	8	-- CONNECT SYSTEM
9	9	CREATE USER "CNY_ETL" IDENTIFIED BY VALUES '4686A1050F638F44'
10	10	DEFAULT TABLESPACE "DW_USER"
11	11	TEMPORARY TABLESPACE "TEMP";
12	12	
13	13	CREATE USER "LCM2" IDENTIFIED BY VALUES '48BCFDF435352212'
14	14	DEFAULT TABLESPACE "DWII_SOR_F_01"
15	15	TEMPORARY TABLESPACE "TEMP";
16	16	
17	17	CREATE USER "FXDM_ETL" IDENTIFIED BY VALUES 'EA010AEA839BFA14'
18	18	DEFAULT TABLESPACE "DW_USER"
19	19	TEMPORARY TABLESPACE "TEMP";
20	20	
21	21	CREATE USER "NRDM_ETL" IDENTIFIED BY VALUES '54A4A046AEE8B31E'
22	22	DEFAULT TABLESPACE "DW_USER"

```
SELECT * FROM imp_sql_lhr a where a.text like '%STORAGE(INITIAL%'     and a.text not like '%STORAGE(INITIAL 65536%'   ;
```

```
SELECT * FROM imp_sql_lhr a where a.text like '%STORAGE(INITIAL%'   and a.text not like '%STORAGE(INITIAL 65536 %' ;
```

	ID	TEXT
1	25274	STORAGE(INITIAL 163840 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
2	27431	STORAGE(INITIAL 223346688 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
3	27505	STORAGE(INITIAL 3145728 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
4	27600	STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
5	27623	STORAGE(INITIAL 10485760 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
6	27807	STORAGE(INITIAL 632291328 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
7	27926	STORAGE(INITIAL 555745280 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
8	28016	STORAGE(INITIAL 1212153856 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
9	28073	STORAGE(INITIAL 890241024 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
10	28198	STORAGE(INITIAL 955252736 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
11	28323	STORAGE(INITIAL 735051776 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
12	28446	STORAGE(INITIAL 3268411392 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
13	28565	STORAGE(INITIAL 2537553920 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
14	28651	STORAGE(INITIAL 513802240 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
15	28766	STORAGE(INITIAL 1238368256 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
16	28791	STORAGE(INITIAL 1872756736 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
17	28818	STORAGE(INITIAL 196608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
18	28911	STORAGE(INITIAL 393216 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
19	29285	STORAGE(INITIAL 10485760 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
20	29412	STORAGE(INITIAL 100663296 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
21	29488	STORAGE(INITIAL 176160768 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
22	29509	STORAGE(INITIAL 256901120 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
23	29622	STORAGE(INITIAL 17825792 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
24	29732	STORAGE(INITIAL 5242880 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
25	29845	STORAGE(INITIAL 214958080 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
26	29863	STORAGE(INITIAL 1349517312 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
27	29923	STORAGE(INITIAL 4194304 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
28	29984	STORAGE(INITIAL 25165824 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
29	30079	STORAGE(INITIAL 925892608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
30	30139	STORAGE(INITIAL 6291456 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
31	30230	STORAGE(INITIAL 1004535808 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
32	30279	STORAGE(INITIAL 131072 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
33	30361	STORAGE(INITIAL 131072 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

执行更新语句：

```
UPDATE imp_sql_lhr t
SET       t.text = 'STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645'
WHERE    t.text LIKE '%STORAGE(INITIAL%'
AND       t.text NOT LIKE '%STORAGE(INITIAL 65536 %';
```

最后利用 spool 来导出到 sql 文本中：

```
set echo on
set trimspool on
set trimout on
set linesize 4000
```

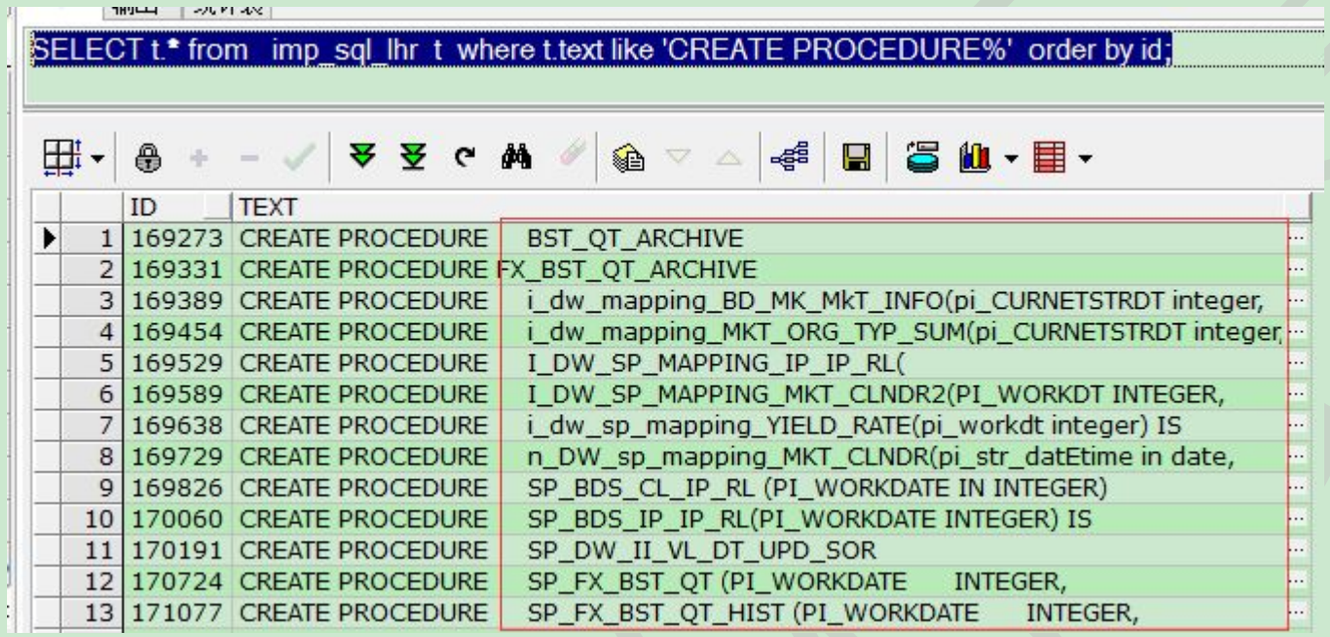
```
set pagesize 0
set sqlblanklines on
set feedback off
set serveroutput off
set term off
set echo off


spool  lhrsql20150515.sql_bk2
SELECT t.text from   imp_sql_lhr  t order by id;
spool off
```

1.4.2.3 执行处理好的 sql 语句

该步骤比较简单，就是把处理好的 ddl 语句提前执行一下，让数据库中包含相应的对象，这样再执行 impdp 导入的时候就不会再创建这些表了。

需要注意的是：**DDL 语句中创建存过、函数、包的语句中是不包含对象所属的 schema 的，这样的话如果那个用户执行 sql 脚本的话，这些对象就创建在那个用户下了，这个显然是错误的，不是我们期望的，那么如何处理这个问题呢，想了想，很简单的嘛，我们 sql 脚本执行完毕后，再执行一次 impdp 的命令就可以把这些对象重建，然后把错误的存过删除就可以了。**

```
SELECT t.* from     imp_sql_lhr    t    where t.text like 'CREATE PROCEDURE%'    order by id;
```



SELECT t.* from imp_sql_lhr t where t.text like 'CREATE PROCEDURE%' order by id;

	ID	TEXT
1	169273	CREATE PROCEDURE BST_QT_ARCHIVE
2	169331	CREATE PROCEDURE FX_BST_QT_ARCHIVE
3	169389	CREATE PROCEDURE i_dw_mapping_BD_MK_MkT_INFO(pi_CURNETSTRDT integer,
4	169454	CREATE PROCEDURE i_dw_mapping_MKT_ORG_TYP_SUM(pi_CURNETSTRDT integer,
5	169529	CREATE PROCEDURE I_DW_SP_MAPPING_IP_IP_RL(
6	169589	CREATE PROCEDURE I_DW_SP_MAPPING_MKT_CLNDR2(PI_WORKDT INTEGER,
7	169638	CREATE PROCEDURE i_dw_sp_mapping_YIELD_RATE(pi_workdt integer) IS
8	169729	CREATE PROCEDURE n_DW_sp_mapping_MKT_CLNDR(pi_str_datEtime in date,
9	169826	CREATE PROCEDURE SP_BDS_CL_IP_RL (PI_WORKDATE IN INTEGER)
10	170060	CREATE PROCEDURE SP_BDS_IP_IP_RL(PI_WORKDATE INTEGER) IS
11	170191	CREATE PROCEDURE SP_DW_II_VL_DT_UPD_SOR
12	170724	CREATE PROCEDURE SP_FX_BST_QT (PI_WORKDATE INTEGER,
13	171077	CREATE PROCEDURE SP_FX_BST_QT_HIST (PI_WORKDATE INTEGER,

```
[oracle@rhel6_lhr dpdump]$ sqlplus lhr/lhr

SQL*Plus: Release 11.2.0.3.0 Production on Fri May 15 16:32:08 2015

Copyright (c) 1982, 2011, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
```

```
16:32:09 SQL> set echo off;
16:32:21 SQL> set serveroutput off;
16:32:30 SQL> set timing on;
16:32:40 SQL> set time on;
16:32:45 SQL> set timing off;
16:32:50 SQL> set time off;
SQL>
SQL>
SQL>
SQL> @lhrsql20150515.sql_bk2;

Session altered.


Session altered.


Session altered.

。。。。。。。。。。。。。。。。。。。。。。。。。。。 省略



SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
```

1.4.2.4 impdp 继续导入

```
[oracle@rhel6_lhr dpdump]$ impdp lhr/lhr directory=DATA_PUMP_DIR dumpfile=lhrsql20150515.dmp logfile=lhrsql20150515_imp.log parallel=4;

Import: Release 11.2.0.3.0 - Production on Fri May 15 19:05:29 2015

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
Master table "LHR"."SYS_IMPORT_FULL_02" successfully loaded/unloaded
Starting "LHR"."SYS_IMPORT_FULL_02":  lhr/******** directory=DATA_PUMP_DIR dumpfile=lhrsql20150515.dmp logfile=lhrsql20150515_imp.log parallel=4
Processing object type SCHEMA_EXPORT/USER
ORA-31684: Object type USER:"CNY_ETL" already exists
ORA-31684: Object type USER:"LCM2" already exists
ORA-31684: Object type USER:"FXDM_ETL" already exists
ORA-31684: Object type USER:"NRDM_ETL" already exists
ORA-31684: Object type USER:"NRDM" already exists
ORA-31684: Object type USER:"DW_ETL" already exists
ORA-31684: Object type USER:"SOR" already exists
ORA-31684: Object type USER:"DWUSER1" already exists
ORA-31684: Object type USER:"DWII_ETL" already exists
ORA-31684: Object type USER:"FXDM" already exists
ORA-31684: Object type USER:"TEST1" already exists
ORA-31684: Object type USER:"LHR" already exists
ORA-31684: Object type USER:"ONL1" already exists
ORA-31684: Object type USER:"CNYDM" already exists
```

```
ORA-31684: Object type USER:"DPA" already exists
ORA-31684: Object type USER:"DWUSER" already exists
ORA-31684: Object type USER:"SQCHECK" already exists
ORA-31684: Object type USER:"TEST" already exists
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/DB_LINK
ORA-31684: Object type DB_LINK:"SOR"."COG_DB" already exists
ORA-31684: Object type DB_LINK:"DPA"."COG_DB" already exists
Processing object type SCHEMA_EXPORT/SEQUENCE/SEQUENCE
ORA-31684: Object type SEQUENCE:"SOR"."SEQ_BDS_CL_IP_RL_ID" already exists
ORA-31684: Object type SEQUENCE:"SOR"."SEQ_GOLD_RATE_INFO" already exists
ORA-31684: Object type SEQUENCE:"SOR"."SEQ_MBR_API_ELGBLTY" already exists
ORA-31684: Object type SEQUENCE:"SOR"."SEQ_MBR_MKT_ELGBLTY" already exists
ORA-31684: Object type SEQUENCE:"SOR"."SEQ_MBR_MMKNG_ROLE" already exists
ORA-31684: Object type SEQUENCE:"SOR"."SEQ_ORG_CD_HSTRY" already exists
ORA-31684: Object type SEQUENCE:"SOR"."SQ_DW_IP_IP_RL" already exists
ORA-31684: Object type SEQUENCE:"SOR"."SQ_MKT_CLNDR_ID" already exists
ORA-31684: Object type SEQUENCE:"FXDM"."SEQ_GOLD_MBR_MMKNG_ROLE_D" already exists
ORA-31684: Object type SEQUENCE:"FXDM"."SEQ_GOLD_MINAMNT_CONFIG" already exists
ORA-31684: Object type SEQUENCE:"DPA"."SEQ_BDS_MBR_D_ID" already exists
ORA-31684: Object type SEQUENCE:"DPA"."SEQ_DL_MKT_INFO" already exists
ORA-31684: Object type SEQUENCE:"DPA"."SEQ_GOLD_RATE_INFO" already exists
ORA-31684: Object type SEQUENCE:"DPA"."SEQ_MBR_AUTH_BY_API_D" already exists
ORA-31684: Object type SEQUENCE:"DPA"."SEQ_MBR_MMKNG_ROLE_D" already exists
ORA-31684: Object type SEQUENCE:"DPA"."SEQ_ORG_D" already exists
ORA-31684: Object type SEQUENCE:"DPA"."SQ_DW_BEST_QUOTE" already exists
ORA-31684: Object type SEQUENCE:"DPA"."SQ_DW_BOND_MID_QUOTE_ID" already exists
ORA-31684: Object type SEQUENCE:"DPA"."SQ_DW_MEMBER_D_ID" already exists
Processing object type SCHEMA_EXPORT/TABLE/TABLE
ORA-39151: Table "DPA"."FX_DL_MKT_INFO" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_FWD_CPI_QT" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_FWD_DL_BY_SF_CP" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_FWD_QT_BY_DIR" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_SPOT_CPI_QT" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_SPOT_DL_BY_SF_CP" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_SPOT_QT_BY_DIR" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_SWAP_CPI_QT" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_SWAP_DL_BY_SF_CP" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_SWAP_DL_LEG_MKT_INFO" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."FX_SWAP_QT_BY_DIR" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."BOND_TURNOVER_RATE_F" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."MONI_TORT_CRCLTN_AMNT_F" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "DPA"."MONI_TURNOVER_RATE_F" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "SOR"."BST_QT_HIST" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
ORA-39151: Table "SOR"."BST_QT" exists. All dependent metadata and data will be skipped due to table_exists_action of skip
。。。。。。。。。。。。。。。。。。。。。。。。。。省略
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."QUE_BOND_DTL" creation failed
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."QUE_FX_FWD_SWAP_DTL" creation failed
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."QUE_CCS_DTL" creation failed
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."QUE_CREPO_DTL" creation failed
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."QUE_FCO_DTL" creation failed
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."QUE_FRA_DTL" creation failed
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."QUE_IBO_DTL" creation failed
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."QUE_IRS_DTL" creation failed
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."QUE_RIBD_DTL" creation failed
ORA-39112: Dependent object type TABLE_STATISTICS skipped, base object type TABLE:"SOR"."FX_BM_FXO_DL_DAY_HIST" creation failed
Job "LHR"."SYS_IMPORT_FULL_02" completed with 2165 error(s) at 19:16:19

[oracle@rhel6_lhr dpdump]$
[oracle@rhel6_lhr dpdump]$
[oracle@rhel6_lhr dpdump]$
```

1.4.2.5 删除错误用户下的包、存过和函数

执行如下的脚本来删除相应的错误对象：

```
BEGIN

    FOR CUR IN (SELECT 'DROP ' || D.OBJECT_TYPE || '   ' || D.OBJECT_NAME SQLT
                FROM    dba_objects d
                WHERE   d.OWNER = 'LHR'
                AND     d.CREATED >=
                        to_date('2015-05-15 09:25:27',
                                'YYYY-MM-DD HH24:MI:SS')
                AND     D.OBJECT_TYPE IN ('FUNCTION', 'PROCEDURE', 'PACKAGE')) LOOP

        EXECUTE IMMEDIATE CUR.SQLT;

    END LOOP;

END;
```

1.5 数据校验

执行如下脚本和源库作比较，查看数据是否完整。

```
SELECT d.OWNER,count(1)
FROM    dba_objects d
WHERE   d.OWNER in (SELECT a.username
        FROM DBA_USERS A
       WHERE A.account_status = 'OPEN'
         AND A.username NOT IN
             ('SYS', 'SYSTEM', 'MGMT_VIEW', 'SYSMAN', 'DBSNMP'))
group by  d.OWNER;


SELECT d.OWNER,
       d.status,count(1)
FROM    dba_objects d
WHERE   d.OWNER in (SELECT a.username
        FROM DBA_USERS A
       WHERE A.account_status = 'OPEN'
         AND A.username NOT IN
             ('SYS', 'SYSTEM', 'MGMT_VIEW', 'SYSMAN', 'DBSNMP'))
```

```
group by   d.OWNER, d.status;

SELECT d.OWNER,
       d.OBJECT_NAME,
       d.OBJECT_TYPE,
       d.status
FROM    dba_objects d
WHERE   d.status = 'INVALID'
and d.owner in (SELECT a.username
         FROM DBA_USERS A
        WHERE A.account_status = 'OPEN'
          AND A.username NOT IN
              ('SYS', 'SYSTEM', 'MGMT_VIEW', 'SYSMAN', 'DBSNMP'));
```

1.6 总结

到此所有的处理算是基本完毕，过程很简单，但是不同的场景处理方式有很多种，我们应该学会灵活变通，核心即 expdp 和 impdp 但是需要做很多的处理。

1.7 aboout me

.....

本文作者：小麦苗，只专注于数据库的技术，更注重技术的运用

ITPUB BLOG：http://blog.itpub.net/26736162

本文地址：http://blog.itpub.net/26736162/viewspace-1657828/

本文pdf版：http://yunpan.cn/QCwUAI9bn7g7w 提取码：af2d

QQ：642808185 若加 QQ 请注明你所正在读的文章标题

创作时间地点：2015-05-15 10:00~ 2015-05-16 19:00 于唐镇金唐公寓宿舍

.....