****

**《大型数据库》**

**课内实验五**

指导老师： 盛 津 芳

专 业： 物联网工程

班 级： 物联网1802

学 号： 8208181125

姓 名： 王灏洋

**目录**

[一、 实验目的和要求 3](#_Toc59393078)

[二、 实验操作环境 3](#_Toc59393079)

[三、 实验内容 3](#_Toc59393080)

[四、 实验步骤 4](#_Toc59393081)

[五、 实验中遇到的问题与解决方法 10](#_Toc59393082)

[六、 实验感想 10](#_Toc59393083)

[七、 实验源代码 11](#_Toc59393084)

# 实验目的和要求

1. 本实验为综合实验，要求综合运用用户管理、逻辑备份、访问数据字典等技术解决问题。

# 实验操作环境

1. 硬件：主频2GHz以上服务器（内存2GB以上、硬件空闲2.2GB以上），主频1GHz以上微机，内存1GB以上。
2. 软件：WINDOWS XP/2000/2003/或Win7/Win8/Win10等。如操作系统是 SERVER版， 可安装ORACLE/9i/10g/11g/12C FOR NT/WINDOWS(注意有32位与64位的区别，可选企业版)； 如果Windows非server如XP/win7等，安装时请选择个人版(PERSONAL)，注意安装时要有兼容性设置与用管理员运行。安装过程中需要关注系统预定义的账号SYS与SYSTEM的密码设置。

# 实验内容

1. 创建一个PROFILE文件pTester，设置锁定用户的登录失败次数为3次，会话的总计连接时间60分钟，口令可用天数30天。
2. 创建一个新用户Tester，密码为Tester123，缺省表空间是tabspace\_????。在tabspace\_????表空间中可以使用50M空间，指定环境资源文件为pTester。
3. 将角色RESOURCE指派给用户Tester。
4. 用EXP和IMP工具将之前创建的major\_????表导入到Tester用户下。
5. 利用PL/SQL语言，以major\_????表为例，编写一个存储过程实现针对单张表的逻辑数据导出功能，要求将给定表的数据转换成SQL语言的Insert语句，表的结构转换成SQL语言的Create Table语句，并保存在文件中。该过程以要导出的表名和保存SQL语句的文件名为参数。

# 实验步骤

1. 创建一个PROFILE文件pTester，设置锁定用户的登录失败次数为3次，会话的总计连接时间60分钟，口令可用天数30天。

|  |
| --- |
| **create profile** Ptester **limit FAILED\_LOGIN\_ATTEMPTS** 3 **connect\_time** 60 **password\_lock\_time** 30; |

* 执行结果

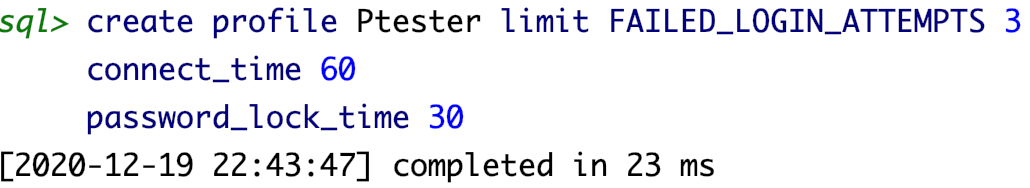


图1 语句1执行结果

1. 创建一个新用户Tester，密码为Tester123，缺省表空间是tabspace\_????。在tabspace\_????表空间中可以使用50M空间，指定环境资源文件为pTester。

|  |
| --- |
| **create user** U\_Tester **identified by** Tester123 **default tablespace** tabSpace\_W125 **quota** 50M **on** tabSpace\_W125 **profile** Ptester; |

* 执行结果

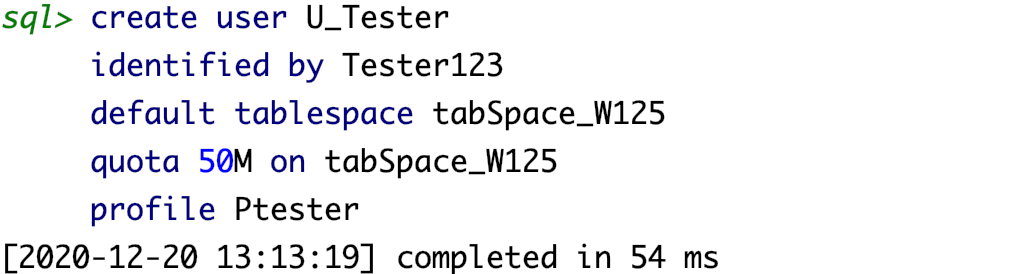


图2 语句2执行结果

1. 将角色RESOURCE指派给用户Tester。

|  |
| --- |
| **grant resource to** U\_Tester;  **grant create session**, **dba to** U\_Tester; |

* 执行结果

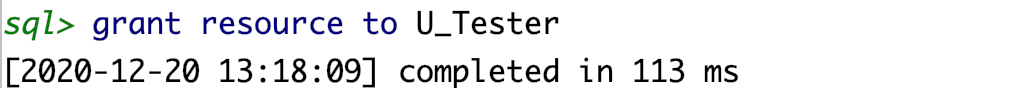


图3 语句3执行结果

1. 用EXP和IMP工具将之前创建的major\_????表导入到Tester用户下。

|  |
| --- |
| exp why/281612 **file**=major.dmp **tables**=T\_major\_W125 imp U\_Tester/Tester123 **file**=major.dmp **full**=y **ignore**=y |

* 执行结果

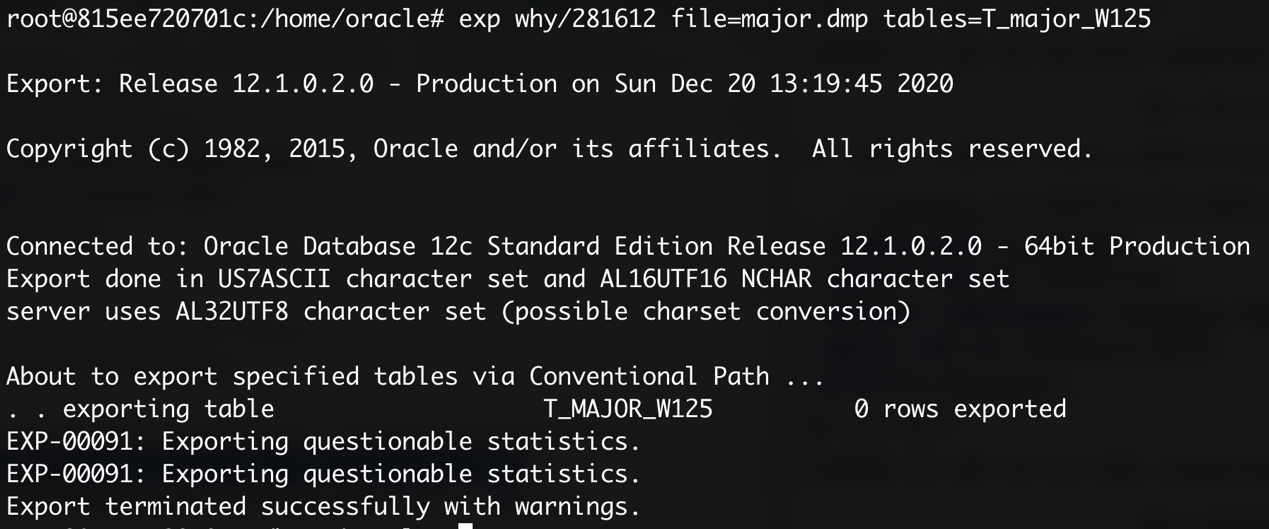


图4 exp语句执行结果

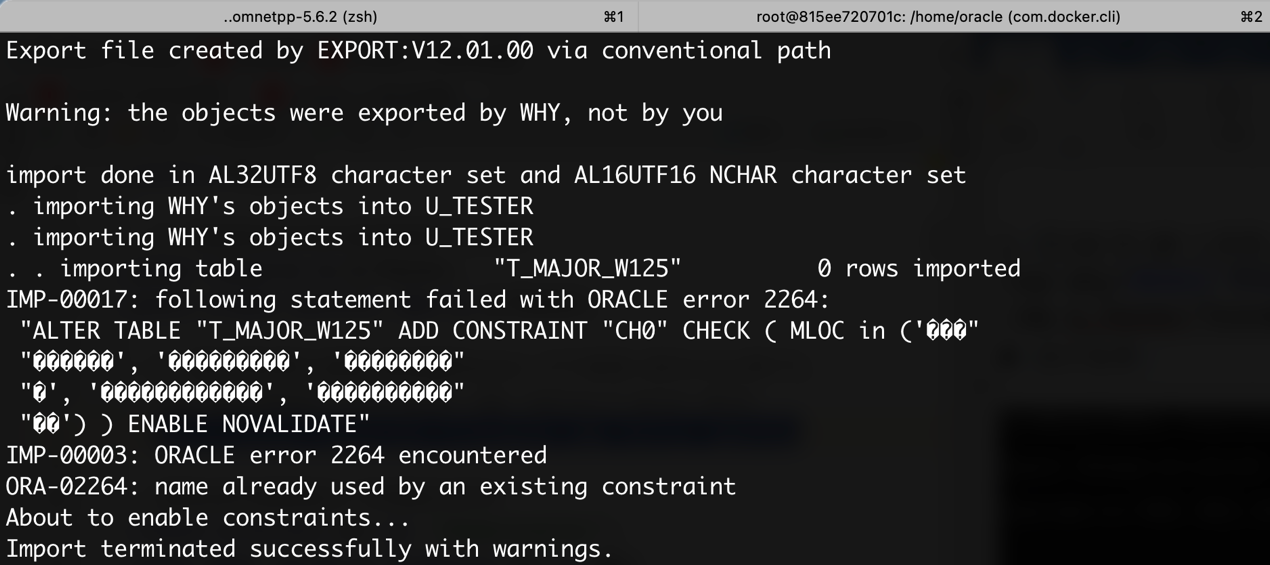


图5 imp语句执行结果

* 注意事项：因为在操作的时候可能没有进行字符集的兼容，所以导致了一些乱码，但是代码是切实可行的。

1. 利用PL/SQL语言，以major\_????表为例，编写一个存储过程实现针对单张表的逻辑数据导出功能，要求将给定表的数据转换成SQL语言的Insert语句，表的结构转换成SQL语言的Create Table语句，并保存在文件中。该过程以要导出的表名和保存SQL语句的文件名为参数。

|  |
| --- |
| *--将导出的内容放到dir下的指定文件中*  **create directory** dir **as '/home/oracle/'**; **grant read**, **write on directory** dir **to** why;  **create or replace procedure** *P\_expLogic* (tabName **in char**, fileName **in char**) **as** *--查询主键* **cursor** primKey **is  select** cu.**COLUMN\_NAME from** user\_cons\_columns cu, user\_constraints au  **where** cu.**constraint\_name** = au.**constraint\_name and** au.**constraint\_type** = **'P' AND** cu.**table\_name** = tabName;  primaryKey **varchar**(256);  cnt **number** := 0;  *--添加check约束* **cursor** ConstraintCur **is  select SEARCH\_CONDITION from** user\_constraints **c  where c**.**table\_name**=tabName **and SEARCH\_CONDITION is not null**;  **constraint varchar**(256);  *--添加外键约束* **cursor** refKeyCur **is  select 'foreign key '**||  **'('** || a1.**column\_name** || **') references '** ||  t2.**table\_name** || **'('** || a2.**column\_name** || **')'  from** user\_constraints t1, user\_constraints t2, user\_cons\_columns a1, user\_cons\_columns a2  **where** t1.**r\_constraint\_name** = t2.**constraint\_name and** t1.**constraint\_name** = a1.**constraint\_name and** t1.**r\_constraint\_name** = a2.**constraint\_name and** t1.**TABLE\_NAME** = tabName;  refCnt **number** := 0;  *--一个模版* refTemp **varchar**(32) := **' constraint rf'**;  refVal **varchar**(512) ;  *--构建create语句所需要的变量* **cursor** attriCur **is  select** *\** **from** USER\_TAB\_COLUMNS **where TABLE\_NAME** = tabName;  attris USER\_TAB\_COLUMNS%**rowtype**;  createSQL **varchar**(1024) := **'create table '** || tabName || **'('**;  *--构建insert语句所需要的变量* insertSQL **varchar**(256) := **'insert into '** || tabName || **'('**;  insertSQLTemp **varchar**(256);  insertVal **varchar2**(256);  valCur **sys\_refcursor**;  val **varchar**(256);  sqlline **varchar**(256);  *--文件操作* fileHandle UTL\_FILE.**file\_type**; **begin** *--找到主键* **open** primKey;  **fetch** primKey **into** primaryKey;  *--文件操作* fileHandle := UTL\_FILE.*FOPEN*(**'DIR'**, fileName, **'a'**);  *--构建create语句* **open** attriCur;  **fetch** attriCur **into** attris;  **while** attriCur%**found loop** *--构造主键* **if** attris.**CHAR\_LENGTH** = 0 **then** createSQL := createSQL || attris.**COLUMN\_NAME** || **' '** || attris.**DATA\_TYPE**;  **else** createSQL := createSQL || attris.**COLUMN\_NAME** || **' '** || attris.**DATA\_TYPE** || **'('** || attris.**CHAR\_LENGTH** || **')'**;  **end if** ;  **if** attris.**COLUMN\_NAME** = primaryKey **then** createSQL := createSQL || **' primary key '**;  **end if**;  *--构造create主体* insertSQL := insertSQL || attris.**COLUMN\_NAME**;  insertVal := insertVal || **'**'''''''' **||'** ||attris.**COLUMN\_NAME** || **'||**''''''''**'**;  **fetch** attriCur **into** attris;  **if** attriCur%**found then** createSQL := createSQL || **', '**;  insertSQL := insertSQL || **', '**;  insertVal := insertVal || **' ||** ''**,**'' **|| '**;  **end if**;  **end loop**;   *--添加外键约束* **open** refKeyCur;  **fetch** refKeyCur **into** refVal;  *--完善性构造* **if** refKeyCur%**found then** createSQL := createSQL || **','**;  **end if**;  **while** refKeyCur%**found loop** createSQL := createSQL || refTemp || refCnt || **' '** || refVal;  refCnt := refCnt + 1;  **fetch** refKeyCur **into** refVal;  **if** refKeyCur%**found then** createSQL := createSQL || **', '**;  **end if**;  **end loop**;   *--添加constrains约束* **open** ConstraintCur;  **fetch** ConstraintCur **into constraint**;  *--完善性构造* **if** ConstraintCur%**found then** createSQL := createSQL || **','**;  **end if**;  **while** ConstraintCur%**found loop** createSQL := createSQL || **' constraint ch'** || cnt || **' check ('** || **constraint** || **')'**;  cnt := cnt + 1;  **fetch** ConstraintCur **into constraint**;  **if** ConstraintCur%**found then** createSQL := createSQL || **', '**;  **end if**;  **end loop**;   *--完善create语句* createSQL := createSQL || **');'**;  insertSQL := insertSQL || **') values ('**;  **close** attriCur;  insertSQLTemp := insertSQL;   *--放入createSQL* UTL\_FILE.*PUT\_LINE*(fileHandle, createSQL);   *--处理插入的语句* sqlline := **'select '** || insertVal || **' from '** || tabName;  **open** valCur **for** sqlline;  **loop  fetch** valCur **into** val;  **exit when** valCur%**notfound**;  insertSQL := insertSQL || val || **');'**;  UTL\_FILE.*PUT\_LINE*(fileHandle, insertSQL);  DBMS\_OUTPUT.*PUT\_LINE*(insertSQL);  insertSQL := insertSQLTemp;  **end loop**;  UTL\_FILE.*FCLOSE*(fileHandle);  **close** valCur; **end**; |

* 执行结果

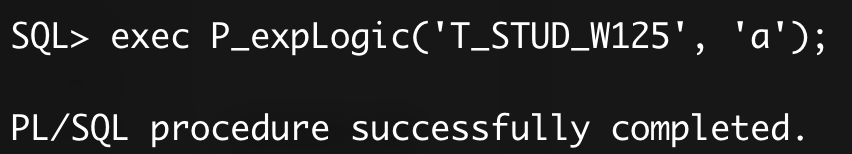
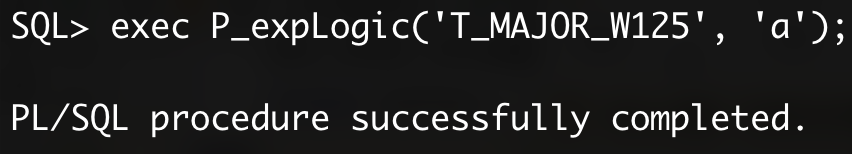


图6 执行两条逻辑导出语句

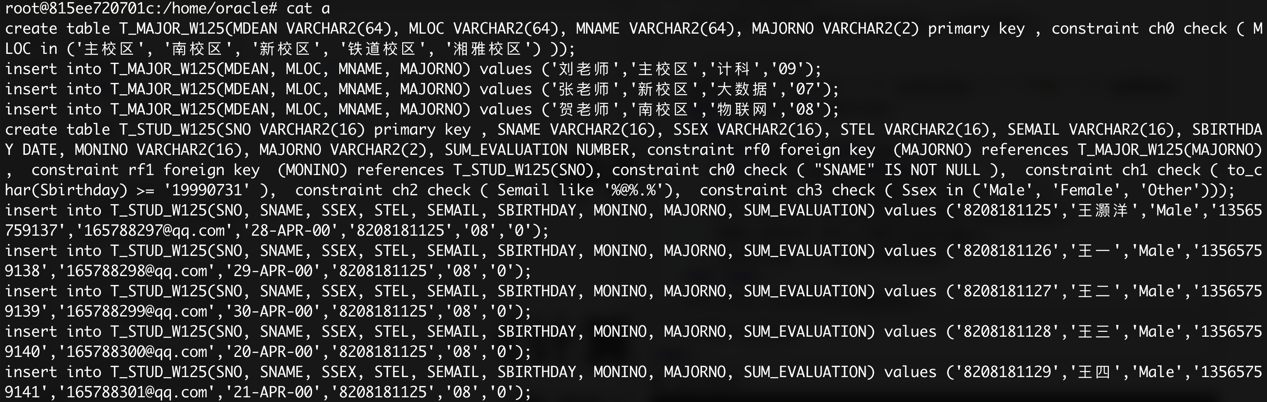


图7 执行结果

* 从执行结果中，我们可以看出，过程成功逻辑导出数据，将所有的数据用insert进行导出，同时，将建表语句使用create语句导出，在导出的过程中，可以实现自定义导出文件名。不仅如此，为了完美恢复语句，我还做出了如下努力

1. 导出了所有的constraints，并将所有除了主键和外键的constraints转化为check约束
2. 导出了所有的主键和外键关系，并完美复原主键外键关系
3. **导出的语句可以执行，完美恢复原表**

# 实验中遇到的问题与解决方法

1. 在使用imp语句为U\_Tester用户导入数据的时候，发现这个用户没有创建session的权限

解决方法：为U\_Tester添加权限。

1. 在sqlplus中可以看到T\_MAJOR\_W125表中有三条数据，但是在使用exp工具进行导出时发现导出的数据行为0，而不是我们在上面看到的3.

解决方法：未解决该问题，还在进行探索，初步观察是因为表空间的问题，但是目前为止还是没有解决这个问题。我也尝试过修改表空间了，看来不是表空间的问题。真的好奇怪，在DataGrip中插入数据后，使用select可以查到数据，但是打开侧边栏中的表的时候就一条数据都没有了。但是又很奇怪的是，使用我写的逻辑导出，就可以完美导出我上面搜索不到的数据，有意思啊。

1. 在使用逻辑导出后，用vi查看我导出的数据，发现汉字都是乱码

解决方法：进行了字符集修改后还是乱码，尝试使用cat得到文件数据，居然可以了，看来修改系统的字符集还是无法改变vi的字符集选择，但是使用cat也够用了。

1. 在使用动态游标的时候，一步一步组的句子居然不能用

解决方法：然后就开始尝试刚开始一步只组一点句子，在执行动态sql的时候，再完成整个句子，就是使用一个新的变量来承载这个select，管用了。

# 实验感想

在本次实验中，让我感触最大的就是在写逻辑导出语句的过程，整个过程花了我好几个小时的时间，刚开始只是实现了所有数据的导出，但是约束条件一个都不见了，本来觉得这样就可以了，但是总感觉少了点什么（有点完美主义了）。

就开始查资料，发现user\_数据字典中存储了关于这些约束的数据，于是就开始各种组表，希望以一个合理的方式组出来这些约束条件，放到create里面。一开始，我是想直接恢复到原来的样子，就是在每一个属性后面直接写约束的样子，但是一想，这样放一个约束就需要遍历一遍我的约束表找到一个属性对应的约束，复杂度也太高了。

于是思考能不能用一个外部约束的方式更改约束的规定方式，这样的话，不会改变表的约束，但是在功能上也是我们想要的样子，于是就开始组表，最终达到了我想要的样子。

整个过程花了些时间，但是看到写出来的百来行的代码还是很有成就感的，不知不觉也就加深了对PL的理解，受益颇多。

# 实验源代码

|  |
| --- |
| *--1．创建一个PROFILE文件pTester，设置锁定用户的登录失败次数为3次，会话的总计连接时间60分钟，口令可用天数30天。 --2．创建一个新用户Tester，密码为Tester123，缺省表空间是tabspace\_????。在tabspace\_????表空间中可以使用50M空间，指定环境资源文件为pTester。 --3．将角色RESOURCE指派给用户Tester。 --4．用EXP和IMP工具将之前创建的major\_????表导入到Tester用户下。 --5．利用PL/SQL语言，以major\_????表为例，编写一个存储过程实现针对单张表的逻辑数据导出功能，要求将给定表的数据转换成SQL语言的Insert语句，表的结构转换成SQL语言的Create Table语句，并保存在文件中。该过程以要导出的表名和保存SQL语句的文件名为参数。  --1．创建一个PROFILE文件pTester，设置锁定用户的登录失败次数为3次， -- 会话的总计连接时间60分钟，口令可用天数30天。* **drop profile** Ptester; **create profile** Ptester **limit FAILED\_LOGIN\_ATTEMPTS** 3 **connect\_time** 60 **password\_lock\_time** 30;  *--2．创建一个新用户Tester，密码为Tester123，缺省表空间是tabspace\_????。 -- 在tabspace\_????表空间中可以使用50M空间，指定环境资源文件为pTester。* **create user** U\_Tester **identified by** Tester123 **default tablespace** tabSpace\_W125 **quota** 50M **on** tabSpace\_W125 **profile** Ptester;  *--3．将角色RESOURCE指派给用户Tester。* **grant resource to** U\_Tester; **grant create session**, **dba to** U\_Tester;  *--4．用EXP和IMP工具将之前创建的major\_????表导入到Tester用户下。 --exp why/281612 file=major.dmp tables=T\_major\_W125 --imp U\_Tester/Tester123 file=major.dmp full=y ignore=y  --将导出的内容放到dir下的指定文件中* **create directory** dir **as '/home/oracle/'**; **grant read**, **write on directory** dir **to** why;  *--5．利用PL/SQL语言，以major\_????表为例， -- 编写一个存储过程实现针对单张表的逻辑数据导出功能， -- 要求将给定表的数据转换成SQL语言的Insert语句， -- 表的结构转换成SQL语言的Create Table语句，并保存在文件中。 -- 该过程以要导出的表名和保存SQL语句的文件名为参数。* **create or replace procedure** *P\_expLogic* (tabName **in char**, fileName **in char**) **as** *--查询主键* **cursor** primKey **is  select** cu.**COLUMN\_NAME from** user\_cons\_columns cu, user\_constraints au  **where** cu.**constraint\_name** = au.**constraint\_name and** au.**constraint\_type** = **'P' AND** cu.**table\_name** = tabName;  primaryKey **varchar**(256);  cnt **number** := 0;  *--添加check约束* **cursor** ConstraintCur **is  select SEARCH\_CONDITION from** user\_constraints **c  where c**.**table\_name**=tabName **and SEARCH\_CONDITION is not null**;  **constraint varchar**(1024);  *--添加外键约束* **cursor** refKeyCur **is  select 'foreign key '**||  **'('** || a1.**column\_name** || **') references '** ||  t2.**table\_name** || **'('** || a2.**column\_name** || **')'  from** user\_constraints t1, user\_constraints t2, user\_cons\_columns a1, user\_cons\_columns a2  **where** t1.**r\_constraint\_name** = t2.**constraint\_name and** t1.**constraint\_name** = a1.**constraint\_name and** t1.**r\_constraint\_name** = a2.**constraint\_name and** t1.**TABLE\_NAME** = tabName;  refCnt **number** := 0;  *--一个模版* refTemp **varchar**(32) := **' constraint rf'**;  refVal **varchar**(512) ;  *--构建create语句所需要的变量* **cursor** attriCur **is  select** *\** **from** USER\_TAB\_COLUMNS **where TABLE\_NAME** = tabName;  attris USER\_TAB\_COLUMNS%**rowtype**;  createSQL **varchar**(2048) := **'create table '** || tabName || **'('**;  *--构建insert语句所需要的变量* insertSQL **varchar**(1024) := **'insert into '** || tabName || **'('**;  insertSQLTemp **varchar**(256);  insertVal **varchar2**(1024);  valCur **sys\_refcursor**;  val **varchar**(256);  sqlline **varchar**(1024);  *--文件操作* fileHandle UTL\_FILE.**file\_type**; **begin** *--找到主键* **open** primKey;  **fetch** primKey **into** primaryKey;  *--文件操作* fileHandle := UTL\_FILE.*FOPEN*(**'DIR'**, fileName, **'a'**);  *--构建create语句* **open** attriCur;  **fetch** attriCur **into** attris;  **while** attriCur%**found loop** *--构造主键* **if** attris.**CHAR\_LENGTH** = 0 **then** createSQL := createSQL || attris.**COLUMN\_NAME** || **' '** || attris.**DATA\_TYPE**;  **else** createSQL := createSQL || attris.**COLUMN\_NAME** || **' '** || attris.**DATA\_TYPE** || **'('** || attris.**CHAR\_LENGTH** || **')'**;  **end if** ;  **if** attris.**COLUMN\_NAME** = primaryKey **then** createSQL := createSQL || **' primary key '**;  **end if**;  *--构造create主体* insertSQL := insertSQL || attris.**COLUMN\_NAME**;  insertVal := insertVal || **'**'''''''' **||'** ||attris.**COLUMN\_NAME** || **'||**''''''''**'**;  **fetch** attriCur **into** attris;  **if** attriCur%**found then** createSQL := createSQL || **', '**;  insertSQL := insertSQL || **', '**;  insertVal := insertVal || **' ||** ''**,**'' **|| '**;  **end if**;  **end loop**;   *--添加外键约束* **open** refKeyCur;  **fetch** refKeyCur **into** refVal;  *--完善性构造* **if** refKeyCur%**found then** createSQL := createSQL || **','**;  **end if**;  **while** refKeyCur%**found loop** createSQL := createSQL || refTemp || refCnt || **' '** || refVal;  refCnt := refCnt + 1;  **fetch** refKeyCur **into** refVal;  **if** refKeyCur%**found then** createSQL := createSQL || **', '**;  **end if**;  **end loop**;   *--添加constrains约束* **open** ConstraintCur;  **fetch** ConstraintCur **into constraint**;  *--完善性构造* **if** ConstraintCur%**found then** createSQL := createSQL || **','**;  **end if**;  **while** ConstraintCur%**found loop** createSQL := createSQL || **' constraint ch'** || cnt || **' check ('** || **constraint** || **')'**;  cnt := cnt + 1;  **fetch** ConstraintCur **into constraint**;  **if** ConstraintCur%**found then** createSQL := createSQL || **', '**;  **end if**;  **end loop**;   *--完善create语句* createSQL := createSQL || **');'**;  insertSQL := insertSQL || **') values ('**;  **close** attriCur;  insertSQLTemp := insertSQL;   *--放入createSQL* UTL\_FILE.*PUT\_LINE*(fileHandle, createSQL);   *--处理插入的语句* sqlline := **'select '** || insertVal || **' from '** || tabName;  **open** valCur **for** sqlline;  **loop  fetch** valCur **into** val;  **exit when** valCur%**notfound**;  insertSQL := insertSQL || val || **');'**;  UTL\_FILE.*PUT\_LINE*(fileHandle, insertSQL);  DBMS\_OUTPUT.*PUT\_LINE*(insertSQL);  insertSQL := insertSQLTemp;  **end loop**;  UTL\_FILE.*FCLOSE*(fileHandle);  **close** valCur; **end**;  *--测试* **declare begin** *P\_expLogic*(**'T\_STUD\_W125'**, **'a'**); **end**;   *--查询表的主键* **select** cu.**COLUMN\_NAME from** user\_cons\_columns cu, user\_constraints au **where** cu.**constraint\_name** = au.**constraint\_name and** au.**constraint\_type** = **'P' AND** cu.**table\_name** = **'T\_MAJOR\_W125'**;  *--查询所有约束，很可惜，外键只有R，但是不知道引用的是谁* **select** *\** **from** user\_cons\_columns cl **where** cl.**constraint\_name in** (**select CONSTRAINT\_NAME from** user\_constraints **c where c**.**table\_name**=**'T\_STUD\_W125' and CONSTRAINT\_TYPE** = **'R'**);  *--查询外键约束，bingo* **select** t2.**table\_name as** "TABLE\_NAME(R)",  a1.**column\_name**,  a2.**column\_name as** "COLUMN\_NAME(R)" **from** user\_constraints t1, user\_constraints t2, user\_cons\_columns a1, user\_cons\_columns a2 **where** t1.**r\_constraint\_name** = t2.**constraint\_name and** t1.**constraint\_name** = a1.**constraint\_name and** t1.**r\_constraint\_name** = a2.**constraint\_name and** t1.**TABLE\_NAME** = **'T\_STUD\_W125'**;  **select  'T\_STUD\_W125('** || a1.**column\_name** || **') references '** ||  t2.**table\_name** || **'('** || a2.**column\_name** || **')'  from** user\_constraints t1, user\_constraints t2, user\_cons\_columns a1, user\_cons\_columns a2  **where** t1.**r\_constraint\_name** = t2.**constraint\_name and** t1.**constraint\_name** = a1.**constraint\_name and** t1.**r\_constraint\_name** = a2.**constraint\_name and** t1.**TABLE\_NAME** = **'T\_STUD\_W125'**;  *--板子* **select** *\** **from** T\_stud\_W125; **drop table** T\_stud\_W125; **create table** T\_stud\_W125 (  **Sno varchar**(16) **primary key** ,  **Sname varchar**(16),  **Ssex varchar**(16),  **Stel varchar**(16),  *--对邮箱格式进行正则匹配* **Semail varchar**(16) ,  **Sbirthday date** ,  *--定义班长的学号是一个外键* **MoniNo varchar**(16) **references** T\_stud\_W125(**Sno**),  *--定义专业号是一个外键* **MajorNo varchar**(2) **references** T\_major\_W125(**MajorNo**),  **sum\_evaluation number**,  **constraint** ch1 **check** ( **Ssex in** (**'Male'**, **'Female'**, **'Other'**)),  **constraint** ch2 **check** ( **Semail like '%@%.%'**),  **constraint** ch3 **check** ( *to\_char*(**Sbirthday**) >= **'19990731'** ),  **constraint** ch4 **check** ( **"SNAME" IS NOT NULL** ) ); **insert into** T\_stud\_W125 **values** (**'8208181125'**, **'王灏洋'**,  **'Male'**, **'13565759137'**, **'165788297@qq.com'**, **'28-4月-2000'**, **'8208181125'**, **'08'**, 0); **insert into** T\_stud\_W125 **values** (**'8208181126'**, **'王一'**,  **'Male'**, **'13565759138'**, **'165788298@qq.com'**, **'29-4月-2000'**, **'8208181125'**, **'08'**, 0); **insert into** T\_stud\_W125 **values** (**'8208181127'**, **'王二'**,  **'Male'**, **'13565759139'**, **'165788299@qq.com'**, **'30-4月-2000'**, **'8208181125'**, **'08'**, 0); **insert into** T\_stud\_W125 **values** (**'8208181128'**, **'王三'**,  **'Male'**, **'13565759140'**, **'165788300@qq.com'**, **'20-4月-2000'**, **'8208181125'**, **'08'**, 0); **insert into** T\_stud\_W125 **values** (**'8208181129'**, **'王四'**,  **'Male'**, **'13565759141'**, **'165788301@qq.com'**, **'21-4月-2000'**, **'8208181125'**, **'08'**, 0);   *--实验报告导出数据* **create table** T\_MAJOR\_W125(**MDEAN VARCHAR2**(64), **MLOC VARCHAR2**(64), **MNAME VARCHAR2**(64), **MAJORNO VARCHAR2**(2) **primary key** , **constraint** ch0 **check** ( **MLOC in** (**'主校区'**, **'南校区'**, **'新校区'**, **'铁道校区'**, **'湘雅校区'**) )); **insert into** T\_MAJOR\_W125(**MDEAN**, **MLOC**, **MNAME**, **MAJORNO**) **values** (**'刘老师'**,**'主校区'**,**'计科'**,**'09'**); **insert into** T\_MAJOR\_W125(**MDEAN**, **MLOC**, **MNAME**, **MAJORNO**) **values** (**'张老师'**,**'新校区'**,**'大数据'**,**'07'**); **insert into** T\_MAJOR\_W125(**MDEAN**, **MLOC**, **MNAME**, **MAJORNO**) **values** (**'贺老师'**,**'南校区'**,**'物联网'**,**'08'**); **create table** T\_STUD\_W125(**SNO VARCHAR2**(16) **primary key** , **SNAME VARCHAR2**(16), **SSEX VARCHAR2**(16), **STEL VARCHAR2**(16), **SEMAIL VARCHAR2**(16), **SBIRTHDAY DATE**, **MONINO VARCHAR2**(16), **MAJORNO VARCHAR2**(2), **SUM\_EVALUATION NUMBER**, **constraint** rf0 **foreign key** (**MAJORNO**) **references** T\_MAJOR\_W125(**MAJORNO**), **constraint** rf1 **foreign key** (**MONINO**) **references** T\_STUD\_W125(**SNO**), **constraint** ch0 **check** ( **"SNAME" IS NOT NULL** ), **constraint** ch1 **check** ( *to\_char*(**Sbirthday**) >= **'19990731'** ), **constraint** ch2 **check** ( **Semail like '%@%.%'**), **constraint** ch3 **check** ( **Ssex in** (**'Male'**, **'Female'**, **'Other'**))); **insert into** T\_STUD\_W125(**SNO**, **SNAME**, **SSEX**, **STEL**, **SEMAIL**, **SBIRTHDAY**, **MONINO**, **MAJORNO**, **SUM\_EVALUATION**) **values** (**'8208181125'**,**'王灏洋'**,**'Male'**,**'13565759137'**,**'165788297@qq.com'**,**'28-APR-00'**,**'8208181125'**,**'08'**,**'0'**); **insert into** T\_STUD\_W125(**SNO**, **SNAME**, **SSEX**, **STEL**, **SEMAIL**, **SBIRTHDAY**, **MONINO**, **MAJORNO**, **SUM\_EVALUATION**) **values** (**'8208181126'**,**'王一'**,**'Male'**,**'13565759138'**,**'165788298@qq.com'**,**'29-APR-00'**,**'8208181125'**,**'08'**,**'0'**); **insert into** T\_STUD\_W125(**SNO**, **SNAME**, **SSEX**, **STEL**, **SEMAIL**, **SBIRTHDAY**, **MONINO**, **MAJORNO**, **SUM\_EVALUATION**) **values** (**'8208181127'**,**'王二'**,**'Male'**,**'13565759139'**,**'165788299@qq.com'**,**'30-APR-00'**,**'8208181125'**,**'08'**,**'0'**); **insert into** T\_STUD\_W125(**SNO**, **SNAME**, **SSEX**, **STEL**, **SEMAIL**, **SBIRTHDAY**, **MONINO**, **MAJORNO**, **SUM\_EVALUATION**) **values** (**'8208181128'**,**'王三'**,**'Male'**,**'13565759140'**,**'165788300@qq.com'**,**'20-APR-00'**,**'8208181125'**,**'08'**,**'0'**); **insert into** T\_STUD\_W125(**SNO**, **SNAME**, **SSEX**, **STEL**, **SEMAIL**, **SBIRTHDAY**, **MONINO**, **MAJORNO**, **SUM\_EVALUATION**) **values** (**'8208181129'**,**'王四'**,**'Male'**,**'13565759141'**,**'165788301@qq.com'**,**'21-APR-00'**,**'8208181125'**,**'08'**,**'0'**); |