ORACLE sql

**USE ‘execute immediate’ to apply a query result to a variable:**

-------------------------------------  
Declare     
str   varchar2(100);   
record     number;   
begin   
    str:= 'select count(\*)  from  bookinfo';   
    execute   immediate str into record;   
end ;

-------------------------------------

**Define, set and Bind variable:**

-----------------------------------------

Declare

var01 varchar2(100);

Begin

var01:= ‘test’;

Select VALUE INTO var01 FROM tb\_name where cl\_name = ‘vcin';

execute immediate ‘Select VALUE FROM tb\_name where cl\_name = ‘vcin'’ INTO var01;

execute immediate 'update tb\_name set cl\_name=:1'

using var01;

execute immediate 'update tb\_name set cl\_name= ‘||var01 ||’';

execute immediate 'update tb\_name set cl\_name = ‘’’||’string01 ‘||’’’';

End;

-----------------------------------------

**Define whether the table existed, if yes, drop it:**

-----------------------------------------

Declare

vCount number;

Begin

select count(\*) into vCount from user\_tables where table\_name=’xxx';

if vCount >0 then

execute immediate 'drop table xxx';`

end if;

-----------------------------------------

**Output variable:**

------------------------------------------

dbms\_output.put\_line(var\_name);

see the above command output with following setting:

sql develop: View ->Dbms Output-> click ‘plus’ icon->choose db.

-----------------------------------------

**Query current user tables:**

---------------------------------------------------

select count(\*)from user\_tables where table\_name LIKE '%TMP%';

SELECT \* from user\_tables where table\_name LIKE '%TMP%';

--------------------------------------------------------------------------

**Query all user tables with system user:**

---------------------------------------------------

select \* from dba\_tab\_columns;

----------------------------------------------------

**Grant/revoke other user the right to query table ‘dba\_tab\_columns’:**

------------------------------------------------------------------------------------

grant select on dba\_tab\_columns to otherUser; (login with dba)

revoke select on dba\_tab\_columns to otherUser;

**------------------------------------------------------------------------------------**

**CpuMhz=cpuhz/1000/1000 (cast string to numeric: cast(‘string’ as numeric)**

-----------------------------------------------------------------------------------

Select cast(cast(**cpuhz** as numeric)/1000/1000 as numeric) as **CpuMhz** from XXX;

------------------------------------------------------------------------------------

**Substr(), length(): (600 ->6.0.0)**

----------------------------------------------

VCenterServerVerID :=’600’;

verIDLen := length(VCenterServerVerID);

i :=1;

formatedVERID := substr(VCenterServerVerID,i,1);

Loop

i :=i+1;

EXIT WHEN i>verIDLen;

formatedVERID := formatedVERID||'.'||SUBStr(VCenterServerVerID,i,1);

end Loop;

-----------------------------------------------

**Sql parameters:**

-------------------------------------

&1 the first parameter

&2 the second parameter

--------------------------------------

**Sql exp/imp Commands:**

--------------------------------------

exp system/bdnacn file=/home/oracle/xxxTables.dmp statistics=NONE tables='tb\_name1, tb\_name2'

imp: grant sysdba to username; grant imp\_full\_database to username; Then run ‘imp’ to imp dmp file.

---------------------------------------

**Exp table to csv files: (test.sql)**

---------------------------------------------

set serveroutput on size 100000

set long 2000000000 (for long text content for one column)

set trimspool on set trimout on

set term off

set echo off

set verify off

set termout on

set feedback off

set newpage none

set pages 0

set heading off

set linesize 7500 (in case one line was cut)

set term off;

*spool*  &1/LMS\_VCENTERVMS\_&2\.csv

select '"VMNAME","VMGUESTOS","VMGUESTHOSTNAME","VMIPADDRESS"' from *dual*;

select '"' ||VMNAME|| '","' ||VMGUESTOS|| '","' ||VMGUESTHOSTNAME|| '","' ||VMIPADDRESS|| '"' from LMS\_VCENTERVMS;

*spool off*

*exit*

------------------------------------------------

Eg: sqlplus bdna245/bdna245 @test.sql /tmp/LMS 20160517

*[bdna@VMDC8245 OracleLMS]$ ll /tmp/LMS\**

*-rw-rw-r--. 1 bdna bdna 3755 May 17 15:36 LMS\_VCENTERVMS\_20160517.csv*

-------------------------------------------------

**Cursor**

----------------------------------------------------------------------------------------------------------

***Declare* *cursor*** cursor\_name is select distinct owner from tbname WHERE clnname='xxxx';

Var01 varchar (30);

Var02 varchar (100);

***Begin***

Var02 :=’xxxxx’;

Open cursor\_name;

*Loop*

fetch cursor\_name into Var01;

exit when cursor\_name %notfound;

……………….

.………………

*End Loop*;

***End***

--------------------------------------------------------------------------------------------------------------

1. Create a new empty table: CREATE table tb\_name (cln1 cln1Type, cln2 cln2Type)
2. Create a new tb\_name with copying data from other\_tb:

Create table tb\_name as select \* from other\_tb;

1. Create a new empty tb\_name with other\_tb structure:

Create table tb\_name as (select \* from other\_tb where 1==2);

1. Empty all data of a tb\_name: truncate table tb\_name;
2. Drop a tb\_name: drop table tb\_name;
3. Rename a table name: rename tb\_name to tbn;
4. Rename a column name: alter table tb\_name rename column old\_cln to new\_cln;
5. **Copy data from other\_tb to existed tbn:**

INSERT INTO tbn (a,b,c,d)

SELECT

other\_tb.a AS a,

other\_tb.b AS b,

other\_tb.c AS c,

'0' AS d

FROM other\_table;

1. Add column: alter table tbn add (cln3 cln3Type);
2. Modify column attribute: alter table tbn modify (cln3 newType);
3. Update column: update tbn set cln3=’XXX’;
4. Rownum: select rownum, cln1 from tbn where rownum <=5; (print 1-5 row data)
5. Order by (desc/asc);
6. Count(), min(), max(), avg(), Substr(), length();
7. cast(‘string’ as numeric)
8. exp table:

exp system/bdnacn file=/home/oracle/lmsVcTables.dmp statistics=NONE tables='LMS\_VCENTERVMS,LMS\_VCENTERINV'

1. sss

**About cursor:**

192.168.8.152 datastore1

192.168.8.152 datastore2

192.168.9.100 datastore1

192.168.9.100 datastore2

192.168.8.152 datastore1; datastore2

192.168.9.100 datastore1; datastore2

---------------------------------------

declare

cursor hostSt is select distinct HOST\_NAME from TMP\_HostStore;

stores varchar(1024);

stHostName varchar2(1024);

stCount int;

Begin

open hostSt;

Loop

fetch hostSt into stHostName;

EXIT WHEN hostSt%notfound;

declare cursor STORE is select distinct DS\_NAME from TMP\_HostStore where HOST\_NAME= stHostName;

stores varchar(1024);

stcount NUMBER(20);

stname varchar(300);

begin

stcount:=0;

open STORE;

loop

fetch STORE INTO stname;

exit when STORE%notfound;

stcount:=stcount+1;

if stcount=1 then

stores :=stname;

else

stores := stores ||';' || stname;

end if;

end loop;

close STORE;

dbms\_output.put\_line(stores);

insert into TMP\_HostStores values(stHostName,stores);

end;

end loop;

close hostSt;

end;

/

set colsep' ';　　　 //-域输出分隔符

set newp none //设置查询出来的数据分多少页显示，如果需要连续的数据，中间不要出现空行就把newp设

置为none，这样输出的数据行都是连续的，中间没有空行之类的

set echo off;　　　 //显示start启动的脚本中的每个sql命令，缺省为on

set echo on //设置运行命令是是否显示语句

set feedback on; //设置显示“已选择XX行”

set feedback off;　 //回显本次sql命令处理的记录条数，缺省为on即去掉最后的 "已经选择10000行"

set heading off;　　//输出域标题，缺省为on 设置为off就去掉了select结果的字段名，只显示数据

set pagesize 0;　　 //输出每页行数，缺省为24,为了避免分页，可设定为0。

set linesize 80;　　 //输出一行字符个数，缺省为80

set numwidth 12;　 //输出number类型域长度，缺省为10

set termout off;　 //显示脚本中的命令的执行结果，缺省为on

set trimout on;　　//去除标准输出每行的拖尾空格，缺省为off

set trimspool on;　//去除重定向（spool）输出每行的拖尾空格，缺省为off

set serveroutput on; //设置允许显示输出类似dbms\_output

set timing on; //设置显示“已用时间：XXXX”

set autotrace on-; //设置允许对执行的sql进行分析

set verify off //可以关闭和打开提示确认信息old 1和new 1的显示.