Tarena
长沙
2011

Tarena_Cook_Book

Oracle 学习资料

Randy

Oracle 常用命令大全

==========

清屏

clear screen

****

数据字典 ========== desc user_views(关键词)

****

==========

查看当前用户的角色

SQL>select * from user_role_privs;

==========

查看当前用户的系统权限和表级权限

SQL>select * from user_sys_privs;

SQL>select * from user_tab_privs;

==========

查看当前用户的缺省表空间

SQL>select username,default_tablespace from user_users;

==========

换用户

conn as sysdba

sys

tsinghua

sqlplus "sys/tsinghua as sysdba"

conn sys/zl as sysdba

==========

修改表结构

alter table test modify(name not null);

alter table test add(name varchar2(20));

==========

更改用户密码

sql>alter user 管理员 identified by 密码;

创建表空间的数据文件

sql>create tablespace test datafile 'd:\oracle\binbo.dbf' size 10m;

==========

创建用户

sql>create user 用户名 identified by 用户名;

==========

bfile 类型实例

创建目录

create directory tnpdir as 'c:\';

删除目录

drop directory tnpdir

授权

crant read on directory tnpdir to scott;

建表

create table bfiletest(id number(3), fname bfile);

添加数据

insert into bfiletest values(1,bfilename('TMPDIR','tmptest.java'));

==========

查看用户

sql>show user

==========

检查语句是否有错

show error

==========

锁定用户

sql>alter user 用户名 account lock

解除用户

sql>alter user 用户名 account unlock

==========

删除用户

sql>drop user zl;

==========

给用户创建表权限

sql>grant create table to 用户名;

==========

授管理员权限

sql>grant dba to 用户名;

==========

给用户登录权限

sql>grant connect to 用户名

==========

给用户无限表空间权限

sql>grant unlinmited tablespace to 用户名;

==========

收回权限

sql>revoke dba from 用户名;

==========

查看用户下所有的表

```
SQL>select * from user_tables;
```

==========

查看名称包含 log 字符的表

```
SQL>select object_name,object_id from user_objects
    where instr(object_name,'LOG')>0;
```

==========

查看某表的创建时间

```
SQL>select object_name,created from user_objects
    where object_name=upper('&table_name');
```

查看某表的大小

```
SQL>select sum(bytes)/(1024*1024) as "size(M)" from user_segments
    where segment_name=upper('&table_name');
```

==========

查看放在 ORACLE 的内存区里的表

```
SQL>select table_name,cache from user_tables where instr(cache,'Y')>0;
```

==========

再添加一个表空间的数据文件

sql>alter tablespace test add datafile 'd:\oracle\test1.dbf' size 10m;

==========

建表

SQL>create table studen(stuno int,stuname varchar(8) not null,stubirth date default to_date('1987-5-9','YYYY-MM-DD'));

向表结构中加入一列　SQL>alter table studen add(stuphoto varchar(9));

从表结构中删除一列　SQL>alter table studen drop column stuphoto;

修改表一列的长度　　SQL>alter table studen modify(stuno number(4));

隐藏将要删除的一列　SQL>alter table studen set unused column stuphoto;

删除隐藏的列　　SQL>alter table studen drop unused columns;

向表中加入约束　　SQL>alter table studen add constraint pk primary key(stuno);

删除约束　　SQL>alter table studen drop constraint pk;

==========

创建表

sql>create table 表名(name varchar2(20),password varchar(20));

==========

添加字段

sql>alter table test add(column_x char(10) not null);

==========

更改字段

sql>alter table emp modify(column_x char (20));

删除字段

如待删除域属于某个索引，则不允许删除操作，必须将此域先设置为 NULL。

sql>alter table emp modify(column_x null);

sql>update emp set column_x=null;

sql>commit;

sql>alter table emp drop(column_x);

==========

选择表空间

sql>alter user 用户名 default tablespace test;

==========

管理员删除别的用户中的表

sql>drop table 用户名.表名;

==========

退出

sql>exit;

==========

默认进入

sql>sqlplus "/ as sysdba"

==========

查看数据库

sql>show parameter block;

==========

写大量语句用记事本，新建方式。

输入"ed"回车

保存后

输入"/"运行;

==========

查询用户有多少表

sql>select * from tab;

==========

SQLServer 取时间

sql>select getdate

Oracle 取时间

sql>sysdate;

操作表结构数据库定义语言命令(不记录在日志文件中)

create table 建表

sql>create table test(name varchar2(20),age date,sex char(2));

sql>insert into test(name,age,sex) values('aa',sysdate,'男');

sql>insert into test(name,age,sex) values('bb',to_date('1888-8-8',"yyyy-aa-dd hh24:mi:ss"),'男');

sql>select * from test;

==========

查询男和女总数

sql>select sex,count(sex) from test group by sex;

---

test 表中数据输入 test1 表中

SQLSserver---select * into test1 from test;

oracle---create table test1 as select * from test;

更改会话时间

sql>alter session set nls_date_format='yyyy-mm-dd hh24:mi:ss';

---

sql>show parameter block 表和视图

sql>show parameter date 查数据结构

SQLServer 中

--删除表中相同数据

sql>create table test1 as select distinct * from test;

--删除表数据

sql>truncate table test;

--把 test 中数据输入到 test1 中

sql>insert into test(select * from test1);

---

rowid(表中存储地址相当表 id)和 rownum(表序号)称伪列(用法)

sql>select name,age,sex,rowid,rownum from test1;

查出前三行

sql>select * from test where rownum<=3;

查出后三行

sql>select * from (select name n,age a,sex s,rownum r from test) where r>(select count(*) from test)-3;

删除后三行

SQL> delete from test where name not in(select name from test where rownum<=(select count(*) from test)-3);

删除相同行

sql>delete from test where rowid not in(select max(rowid) from test group by name,age,sex);

删除所有表

sql>select 'drop table' ||tname|| ':' from tab;

sql>spool c:\test.sql;

sql>select 'drop table' ||tname|| ':' from tab;

sql>spool off

sql>@c:\test.sql;

---

alter table 修改表

truncate table 节段表(只删除数据)

drop table 删除表

==========

查看表结构

desc 表名;

==========

查出成绩的前三名

sql>select * from (select * from stu order by score desc) where rownum<=3;

==========

更改字符集

SQL>startup mount

SQL>alter system enable restricted session;

SQL>alter system set job_queue_processes=0;

SQL>alter database open;

SQL>alter database character set ZHS16GBK;

SQL>shutdown

SQL>startup

==========

将一张表或几张表中的域重新组合后插入新表。假定原先的两张表为 emp,work，现选择部分数据域合并为 emp_work 建立 emp_work

SQL>insert into emp_new select a.no, sysdate, a.name, b.service_duration from emp a, work b where a.no=b.no;

SQL>commit;

这样的方式仍然要使用回滚段，为加快数据迁移速度，可将 insert 替换成 insert /*+APPEND*/（大小写不论），指示 oracle 以直通方式直接写数据文件，绕过回滚空间。

SQL>insert /*+APPEND*/ into emp_new select a.no, sysdate, a.name,

b.service_duration from emp a, work b where a.no=b.no;

SQL>commit;

==========

DDL 数据定义语言

DML 数据操纵语言

TCL 事务控制语言

DCL 数据控制语言 GRANT　REVOKE

==========

一个表中的某一列输到另一个表中

insert into stu1(name)(select name from stu);

==========

事务

rollback;

insert into stu1(name)(select name from stu);

commit;提交

COMMIT - 提交并结束事务处理

ROLLBACK - 撤销事务中已完成的工作

SAVEPOINT – 标记事务中可以回滚的点

SQL> update order_master set del_date ='30-8 月-05' WHERE orderno <= 'o002';

SQL> savepoint mark1;

SQL> delete FROM order_master WHERE orderno = 'o002';

SQL> savepoint mark2;

SQL> rollback TO SAVEPOINT mark1;

SQL> COMMIT;

==========

换名

set sqlprompt "scott>";

==========

GRANT 授予权限

SQL> GRANT SELECT ON vendor_master TO accounts WITH GRANT OPTION;

REVOKE 撤销已授予的权限

SQL> REVOKE SELECT, UPDATE ON order_master FROM MARTIN;

==========

比较操作符

```
SQL> SELECT vencode,venname,tel_no
        FROM vendor_master
        WHERE venname LIKE 'j___s';
SQL> SELECT orderno FROM order_master
        WHERE del_date IN (‘06-1 月-05’,‘05-2 月-05');
SQL> SELECT itemdesc, re_level
        FROM    itemfile
        WHERE qty_hand < max_level/2;
```

==========

逻辑操作符

```
SQL> SELECT * FROM order_master
        WHERE odate >  ‘10-5 月-05'
        AND del_date <  ‘26-5 月-05’;
```

==========

集合操作符将两个查询的结果组合成一个结果

```
SQL> SELECT orderno FROM order_master
        MINUS
        SELECT orderno FROM order_detail;
```

---

select * from scott.stu

union (all)重复的去掉[intersect 把相同的取出来][minus 显示不相同的数]

select * from stu

显示相同的数据

select name from stu intersect select name from stu1;

==========

连接操作符

连接操作符用于将多个字符串或数据值合并成一个字符串

```
SQL> SELECT (venname|| '  的地址是  '
        ||venadd1||' '||venadd2 ||' '||venadd3) address
        FROM vendor_master WHERE vencode='V001';
```

==========

操作符的优先级

SQL 操作符的优先级从高到低的顺序是：

算术操作符　　--------最高优先级

连接操作符

比较操作符

NOT 逻辑操作符

AND 逻辑操作符

OR　 逻辑操作符　 --------最低优先级

==========

用来转换空值的函数

NVL

NVL2

NULLIF

SELECT itemdesc, NVL(re_level,0) FROM itemfile;

SELECT itemdesc, NVL2(re_level,re_level,max_level) FROM itemfile;

SELECT itemdesc, NULLIF(re_level,max_level) FROM itemfile;

GROUP BY 和 HAVING 子句

GROUP BY 子句

用于将信息划分为更小的组

每一组行返回针对该组的单个结果

HAVING 子句

用于指定 GROUP BY 子句检索行的条件

SELECT p_category, MAX(itemrate) FROM itemfile GROUP BY p_category;

SELECT p_category, MAX(itemrate) FROM itemfile GROUP BY p_category

HAVING p_category NOT IN ('accessories');

==========

ROW_NUMBER (row_number)返回连续的排位，不论值是否相等

RANK(rank) 具有相等值的行排位相同，序数随后跳跃

DENSE_RANK(dense_rank) 具有相等值的行排位相同，序号是连续的

```
SELECT d.dname, e.ename, e.sal, DENSE_RANK()
   OVER (PARTITION BY e.deptno ORDER BY e.sal DESC)
   AS DENRANK
FROM emp e, dept d WHERE e.deptno = d.deptno;
```

==========

日期函数

ADD_MONTHS(当前只加月)

```
alter session set nls_date_format='yyyymmdd hh24miss';
select add_months(sysdate,2) from dual;
```

MONTHS_BETWEEN(前面时间减后面时间=得之间月差)

```
select months_between(sysdate,to_date('2007-6-10','yyyy-mm-dd')) from dual;
```

LAST_DAY(求得当前月的最后一天)

```
select last_day(sysdate) from dual;
```

ROUND(round 年-月-日-->四舍五入)

```
select round(2.3) from dual;
select round(to_date('2007-6-10','yyyy-mm-dd'),'year') from dual;
select round(to_date('2007-6-10','yyyy-mm-dd'),'month') from dual;
select round(to_date('2007-6-10','yyyy-mm-dd'),'day') from dual;
```

NEXT_DAY(下一星期的星期二）

```
select next_day(to_date('2007-6-10','yyyy-mm-dd'),'星期二') from dual;
```

TRUNC(trunc)

EXTRACT(extract)

select extract(year from date '1998-03-07') from dual;

select extract(month from to_date ('1998-03-07','yyyy-mm-dd')) from dual;

2008 年 2 月有多少天

inbo---->select extract(day from last_day(to_date ('2008-02-07','yyyy-mm-dd'))) from dual;

2003-4-3 与 1956-3-1 之间有多少天

inbo---->select round(months_between(to_date('2003-4-3','yyyy-mm-dd'),to_date('1956-3-1','yyyy-mm-dd'))/12) from dual;

==========

把两边的 9 去掉

select trim('9' from '9999ddddddd99999') from dual;

去空格

select trim(' ' from ' 9999ddddddd99999') from dual;

==========

函数	输入	输出
Initcap(char)	Select initcap(‘hello’) from dual;	Hello
Lower(char)	Select lower(‘FUN’) from dual;	fun
Upper(char)	Select upper(‘sun’) from dual;	SUN
Ltrim(char,set)	Select ltrim(‘xyzadams’,’xyz’) from dual;	adams
Rtrim(char,set)	Select rtrim(‘xyzadams’,’ams’) from dual;	xyzad
Translate(char, from, to)	Select translate(‘jack’,’j’ ,’b’) from dual;	back
Replace(char,searchstring,[rep string])	Select replace(‘jack and jue’ ,’j’,’bl’) from dual;	black and blue
Instr (char, m, n)	Select instr (‘worldwide’,’d’) from dual;	5
Substr (char, m, n)	Select substr(‘abcdefg’,3,2) from dual;	cd
Concat (expr1, expr2)	Select concat (‘Hello’,’ world’) from dual;	Hello world

==========

数字函数接受数字输入并返回数值结果

函数	输入	输出
Abs(n)	Select abs(-15) from dual;	15
Ceil(n)	Select ceil(44.778) from dual;	45
Cos(n)	Select cos(180) from dual;	-.5984601
Cosh(n)	Select cosh(0) from dual;	1
Floor(n)	Select floor(100.2) from dual;	100
Power(m,n)	Select power(4,2) from dual;	16
Mod(m,n)	Select mod(10,3) from dual;	1
Round(m,n)	Select round(100.256,2) from dual;	100.26
Trunc(m,n)	Select trunc(100.256,2) from dual;	100.25
Sqrt(n)	Select sqrt(4) from dual;	2
Sign(n)	Select sign(-30) from dual;	-1

==========

字符函数

查看有多少个字符

```
SQL> SELECT LENGTH('frances') FROM dual;
```

```
SQL> SELECT vencode,
        DECODE(venname,'frances','Francis') name
        FROM vendor_master WHERE vencode='v001';
```

查找人是否存在 加字段 decode 主明是否有人

select name,decode(name,'rbb','有人') from stu;

==

排续

select dense_rank() over(partition by sex order by score) from test;

select row_number() over(order by score),name,sex,score from test;

select rank() over(order by score) from test;

select dense_rank() over(order by score) from test;

==

创建同义词

SQL> create public synonym test for rbb.test;

SQL> create synonym test for mytest;

同一类的才可以替换,同义词替换同义词

替换

SQL> create or replace synonym emp_sysn for scott.emp;

****

创建序列

SQL>create sequence xule increment by 1 start with 1 maxvalue 999;

increment by 增长值

start with 起始值

maxvalue 最大值

minvalue 最小值

nocycle 不循环

chare 10 缓存

xule.nextval ===========下一个序列的值

xule.currval ===========可以查询序列当前的值

更改序列 start with 不能改

alter sequence xule maxvalue 100 [sycle nocycle];

序列用法

SQL>create table xl(name varchar2(4));

SQL>insert into test values(xl.nextval);

SQL>select xl.currval from dual;

****

删除序列

drop sequence x;

desc user_sequences

****

创建视图　　视图中可以使用函数和表达式

create or replace view

****

创建视图

SQL> create or replace view 视图名 as select * from rbb union all select * from rbbb union all select * from test;

SQL> create or replace view 视图名 as

  2 select empno as 编号,ename as 姓名 from scott.emp

  3 where deptno=10;

==========

如果在当前用户下没有这个视图就创建此视图

如果有此视图就覆盖此视图

create or replace view view_name as select empno,ename from emp where deptno=10;

****

在创建视图前要为当前用户授权

grant resource to scott;

create or replace view v_sal as select ename,sal from emp order by sal desc;

****

使用视图

select * from v_sal;

****

删除一个视图

drop view view_name;

****

重新编译已有的视图

alter view view_name compile;

****

数据字典 ==========desc user_views

****

union all 连接两个表或者多个表为一个视图

****

锁

一致性

完整性

行级锁和表级锁

行级锁:是一种排他锁,防止其他事务修改此行.

解锁：提交事务(commit),(rollback)

更新表数据:update test set score=80 where name='xiaoli';

自动提交

set autocommit on

set sutocommit off

锁定某行更新语句

select * from scott.test where name='xiaoli' for update;

select * from scott.test where name='xiaoli' for update of score;

select * from scott.test atest,test b where a.name=b.name and b.name='bbb' for update of b.score;

等待 update

select * from scott.test where name='xiaoli' for update wait 2;

select * from scott.test where name='xiaoli' for update nowait;

表级锁：锁定整个表

表级锁语法:lock table 表名 in mode mode;

行共享 row share--行排他 row exclusive--共享 share-共享行排他 share row exclusive-----排他 exclusive

---

行共享(row share):lock table scott.test in (row share) mode;

[其他用户.行共享---其他用户.行排他---其他用户.共享----其他用户.共享行排他----其他用户.不可以(排他)]

行排他(row exclusive):lock table scott.test in (row exclusive) mode;

[其他用户.行共享----其他用户.行排他----其他用户.不可以(共享)---其他用户.不可以(共享行排他)--其他用户.不可以(排他)]

共享(share):lock table scott.test in (share) mode;

[其他用户.行共享---其他用户.不可以(行排他)---其他用户.共享----其他用户.不可以(共享行排他)---其他用户.不可以(排他)]

共享行排他(share row exclusive):lock table scott.test in (share row exclusive) mode;

[其他用户.行共享,其他用户.不可以(行排他),其他用户.不可以(共享),其他用户.不可以(共享行排他),其他用户.不可以(排他)]

排他(exclusive):lock table scott.test in (exclusive) mode;

[其他用户.不可以(行共享),其他用户.不可以(行排他),其他用户.不可以(共享),其他用户.不可以(共享行排他,)其他用户.不可以(排他)]

表分区

---范围分区

```
create table test(name varchar2(20),sex char(2),score number(3))
partition by range(score)
(
partition p1 values less than (50) tablespace users,
partition p2 values less than (80),
partitiom p3 values less than (maxvalue)
)
select * from test partition(p1) union select * from test partitiom(p3);
```

删除分区

```
alter table test drop partition p3;
```

添加分区

```
alter table test add partition p3 values less than (maxvalue);
```

拆分分区

```
alter table test split partition p2 at(60)
into (partition p21,partition p22);
```

合并分区

```
alter table test merge partitions p21,p22 into partition p2;
```

截断分区(删除数据)

```
alter table test truncate partition p3;
```

现有表分区

```
create table str as select * from student;
drop table student;
create table student(
    studentid integer not null,
    studentname varchar2(20),
    score integer
)
```

```
partition by range(score)(
    partition p1 values less than(60),
    partition p2 values less than(75),
    partition p3 values less than(85),
    partition p4 values less than(maxvalue)
)
insert into student(select * from stu);

select * from test scott.emp@tsinghua
```

****

表分区

Oracle 允许用户对表进一步的规化，即对表进一步拆分，将表分成若干个逻辑部分，每个部分称其为表分区

优点：增强可用性，单个分区出现故障，不影响其他分区

均衡的 I/O，不同的分区可以映射到不同的磁盘　　改善性能

****

①范围分区法

```
create table st(
    studentid integer not null,
    studentname varchar2(20),
    score integer
)
partition by range(score)(
    partition p1 values less than(60),
    partition p2 values less than(75),
    partition p3 values less than(85),
    partition p4 values less than(maxvalue)
)
```

==========select * from stu partition(p1)==========

②散列分区

```
create table st(deptno int,deptname varchar(14))
partition by hash(deptno)(
partition p1,partition p2
)
```

组合分区

```
alter table test coalesce partition;
```

****

③复合分区

范围分区和列表分区

```
create table salgrade(
grade number(2),losal number(2),hisal number(2)
)
partition by range(grade)
subpartition by list(losal)
(
partition p1 values less than(10)
   (
     subpartition p1a values('湖北'),
     subpartition p1b values(default)
   ),
partition p2 values less than(20)
   (
     subpartition p1a values('河南'),
     subpartition p1b values(default)
   ),
partition p3 values less than(30)
   (
     subpartition p1a values('上海'),
     subpartition p1b values(default)
   )
)
```

范围分区和散列分区

```
create table salgrade(
grade number(2),losal number(2),hisal number(2)
)
partition by range(grade)
subpartition by hash(losal)
[subpartitions 5]
(
partition p1 values less than(10)(subpartition p1a,subpartition p1b),
partition p2 values less than(20)(subpartition p2a,subpartition p2b),
partition p3 values less than(30)(subpartition p3a,subpartition p3b)
)
```

```
 create table salg(
 grade number(2),losal number(2),hisal number(2)
 )
 partition by range(grade)
 subpartition by hash(losal)
 subpartitions 3
 (
 partition p1 values less than(10),
 partition p2 values less than(20),
 partition p3 values less than(30)
 )
```

****

④列表分区

```
create table test stu(id int,name varchar(20),add varchar(8))
partition by list(add)
(
partition p1 values('中国'),
partition p2 values('英国'),
partition p3 values(default)
)
```

****

移动分区

alter table test move partition p5 tablespace users;

****

修改存档

SQL> shutdown immediate

数据库已经关闭。

已经卸载数据库。

ORACLE 例程已经关闭。

SQL> startup mount

ORACLE 例程已经启动。

Total System Global Area 135338868 bytes

Fixed Size 453492 bytes

Variable Size 109051904 bytes

Database Buffers 25165824 bytes

Redo Buffers 667648 bytes

数据库装载完毕。

SQL> alter database archivelog;

数据库已更改。

SQL> archive log list;

数据库日志模式 存档模式

自动存档 禁用

存档终点 d:\oracle\ora92\RDBMS

最早的概要日志序列 1

下一个存档日志序列 2

当前日志序列 2

SQL> alter system set log_archive_dest=true scope=spfile;

系统已更改。

SQL> alter database open;

数据库已更改。

SQL> spool off
****

PL/SSQL(过程化语言) 声明部分 执行语句部分 异常处理部分
identifier constant datatype not null
[:=|default expr];

```
declare
my number(5);
begin
  select quantity into my
from products where product='wawa'
for update of quantity;
if my>0 then
update products set quantity=quantity+1
where product='wawa';
insert into purchase_record
values('wawawa',sysdate);
end if;
commit;
Exception
where others then
dbms_output.put_line('chucuo'||SQLERRM);
END;
```

```
declare icode varchar2(6)
p_catg varchar2(20);
c_catg constant datatype:=0.10
```

数字类型

number

decrmdl

int/integer

real(实数)

binary_integer(带符号的整数)

pls_integer(同上)

字符类型

character

char 3276

Raw(2000)

long/long Raw(32760)

Rowid/rowid()

varchar2 (string(nchar/nvarchar)/varchar)

日期时间

date

timeStamp(固定日期 dd-mm-yy 秒 6 位)

子 timestamp with time zone

ti timestamp(9)

布尔

boolean

true

false

null

打印出时间

```
declare
test_tz timestamp with time zone;
begin test_tz:=to_timestamp_tz('2006-6-22 09:07:11','yyyy-mm-dd hh24:mi:ss');
dbms_output.put_line(test_tz);
end;
```

lob 类型

BFILE

BLOB

CLOB

NCLOB

属性类型

%type %rowtype

==========

bfile 类型实例

创建目录

create directory tnpdir as 'c:\';

删除目录

drop directory tnpdir

授权

crant read on directory tnpdir to scott;

建表

create table bfiletest(id number(3), fname bfile);

添加数据

insert into bfiletest values(1,bfilename('TMPDIR','tmptest.java'));

==========

向数据库中添加图片

create directory images as 'c:\images';

crant read on directory images to scott;

create table my_diagrams(

chapter_descr varchar2(40);

diagram_no integer,

diagram blob

);

```
declare
   l_bfile bfile;
   l_blob blob;
begin
  insert into my_diagrams(diagram)
  values(emptv_blob())
  return diagram into l_blob;
  l_bfile:=bfilename('images','\nvimage.jpg');
  dbms_lob.open(l_bfile,dbms_lob.file_readonly);
  dbms_lob.loadfromfile(l_blob,l_bfile,dbms_lob,getlength(l_bfile));
  dbms_lob.close(l_bfile);
  commit;
end;
```

==========

%type 实例 查询

```
declare
dtr dept.dname%type;
begin
select dname into str from dept where deptno=30;
dbms_output.put_line(str);
end;
set serverout on
```

==========

%rowtype 实例

```
declare
row dept%rowtype;
begin
select * into row from dept where deptno=30;
dbms_output.put.line(row.dname||' '||row.deptno||' '||row.loc);
```

```
//异常
exception
when no_data_found then
  dbms_output.put_lin('没有数据');
when too_many_rows(others) then
  dbms_output.put_lin('太多拉');
end;
```

==========

格式

```
if 条件 then

elsif 条件 then

else

end if
```

==========

格式

```
begin
  case'&grade'
    when 'a' then dbms_output.put_line('优异');
    when 'b' then dbms_output.put_line('良好');
    else dbms_output.put_line('其它')
  end case;
end;
```

外界变量

```
var vnm varchar2(20);
begin
:v:='aaaaa';
end;
```

打印

```
print v
```

==========

loop 实例

```
begin
loop
exit when 3>4;

end loop;
end;
```

==========

while 实例

```
begin
while (条件)condition loop
语句体;
end loop;
end;
```

==========

循环实例

正

```
begin
for c in 1..10
loop
dbms_output.put_line(c);
end loop
end;
```

倒

```
begin
for c in reverse 1..10
loop
dbms_output.put_line(c);
end loop
end;
```

==========

```
declare
  num number(3):=1;
begin
  while num<10 loop
    dbms_output.put_line(num);
    num:=num+1;
    end loop;
end;
```

```
declare
  num number(3):=1;
begin
  loop
    dbms_output.put_line(num);
    exit when num>10;
    num:=num+1;
    end loop;
end;
```

goto 实例

```
DECLARE
   qtyhand itemfile.qty_hand%type;
   relevel itemfile.re_level%type;
BEGIN
   SELECT qty_hand,re_level INTO qtyhand,relevel
   FROM itemfile WHERE itemcode = 'i201';
   IF qtyhand < relevel THEN
      GOTO updation;
   ELSE
      GOTO quit;
   END IF;
   <<updation>>
   UPDATE itemfile SET qty_hand = qty_hand + re_level
   WHERE itemcode = 'i201';
   <<quit>>
   NULL;
END;
```

==========

打开服务器

net start oracleservicebinbo

打开监听器

lsnrctl start

关闭服务器

net stop oracleservicebinbo

关闭监听器

lsnrctl stop