mon document plsql

Yongcheng MA

17 janvier 2022

Table des matières

1 introduction SQL | ORACLE

1.0.1 SQL

a.gammoudi@caplogy.com **instaler oracle dans ubuntu : http ://www2.hawaii.edu/ lipyeow/ics321/2015spr/installoracle11g.html**

— dml,ddl,dcl :
dml ->data manipulation language
ddl ->data definition language
dcl ->data control language
— UML diagramme de classe

Etudiant
idEtudiant
nom
prénom
adresse

Matière
idMatière
libMat
nombreMat
coffecien

Enseignement
idEnseigne
nom
prénom
etudiant

1.1 démarrer oracle

```
    sudo apt-get install putty
lsnrctl start
lsnrctl stop
sqlplus / as sysdba
startup
clear screen
select username,password from dab_users;
alert user system identified by 123456
net stop 'nom de sid de base de donnee'
net start 'nom de sid de base de donnee'
net start OracleServiceOrcl
alter session set nls_date_format = 'dd-mm-yyyy hh24:mi:ss' ;
alter session set nls_date_format = 'dd/mm/yyyy hh24:mi:ss';
}

```

— connecter service d'oracle *sqlplus sys/system@orcl as sysdba*

1.2 dérouiller un compte et créer un compte

```
        sqlplus / as sysdba
        alter user 'nom de User' account unlock;
```

```
        alert user 'nom de User' identified by 'password de User';
        alert user 'nom de User' account lock;

```

— Quand on créer un user dans oralce :

```
      create user mathieu identified by azerty \\
      default tablespace users quota 10M on users;

```

il a une erreur suivant :

```
    ORA-65096: nom utilisateur ou de role commun non valide
    on utilise cette instruction suivante: alter session set
    "_ORACLE_SCRIPT"=true;

```

```
      grant connect,resource to mathieu with admin option;
      grant debug any procedure,debug connect session to
    mathieu;
      grant create any view to mathieu;
      grant administer database trigger to mathieu;

```

1.3 l'instruction d'oracle

— *show user ;*
conn scott/tiger ;
desc 'nom de table' ;
quit or exit ;
disconnect ;
*select * from all_ users ; –lister tous les Users ;*
clear screen ;
*select * from tab ; – liseter tous les tables d'user actuel*
@'nom de scripte de sql' ; –@test.sql

1.4 tablespace

— example de tablespace
  — donner des doit

```
          grant crete tablespace to mathieu;
          grant alter tablespace to mathieu;

          create tablespace mytablespace
          datafile
          'D:\myts1.dbf' siez 10M
          'D:\myts2.dbf' siez 20M

```

  — consulte tablespace de l'utilisateur

```
            select * from user_tablespaces;

```

— utiliser tablespace

```


```

1.5 CRUD dans oracle

1.5.1 add

— créer une novelle table :

```
    create table Test(id number(5) primary key,name varcher2
    (50) not null)
    inster into Test(id,name) values(2,'alen')
    inster into Test values(2,'alen')
    inster into Test values(2,'alen',to_date('07-01-2022','dd
    -mm-yyyy'))
    insert into Test (
    with
    v1 as (select 50,'alen' from dual),
    v2 as (select 60,'fabien' from dual)
    select * from v1
    union
    select * from v2
    )

```

— créer une table à autre tables :
create table Test as select ename nom,sal salire from emp ;
create table Test1 as (select emp.name,dept.dname from emp,dept where emp.deptno = dept.deptno)

— ajouter des données dans la table insert all :

```
      create table myemp as select * from emp;
      create table Test(id number(5) primary key,name
    varcher2(50) not null)
      truncate table myemp;
    insert all
    into myemp(empno,ename,deptno) values(1,'alen',10),
    into myemp(empno,ename,sal,comm) values(2,'july',2000,0),
    into myemp(empno,ename,sal,comm) values(3,'Leo',1000,200)
    ,
    into myemp(empno,ename,sal,comm) values(2,'fabien'
    ,3000,100),
    into Test(id,name) values(1,'mathieu'),
    into Test values(2,'Amine'),
    into Test values(3,'idress'),
    select 1 from dual;

```

— insert all when :
insert all
when deptno > 20 then
into e
when deptno > 30 then
into e_20
else
into e_30
select e.name,e.sal,e.deptno,d.name from emp e,dept d where e.deptno = d.deptno ;

1.5.2 delte

— truncate, effacer tous les données dans la table :
truncate table 'nom de table'

1.5.3 update

— mettre à jour une donneée *update emp set sal = 902 where empno = 7369 ;*

— update plusieurs des données

```
1       update emp set sal = 902 where empno in (7369,8888);
2       update emp set comm =100;
3
```

1.5.4 select

— lister tous les données :
*select * from emp ;*
— supprimer des redondances de données :
select distinct job from emp ;
— not null :
select empno,ename,sal,comm where sal > 1500 and comm is not null ;

— between and :
select empno,ename,sal from emp where (sal >=1000 and sal <= 2000) or (sal >=2500 and sal <= 3000)
select empno,ename,sal from emp where (sal between 1000 and 2000) or (sal between 2500 and 3000)
— <>
*select * from emp where empno <>7369 ;*

— not
select empno,ename,sal,comm from emp where not (sal >1500 and sal <= 2000) ;
select empno,initcap(ename),sal,comm from emp where not (sal >1500 and sal <= 2000) ;
— in
select empno,ename from emp where empno in (7369,7654,7900) ;

select empno,'nom de employée' || ename nom from emp where empno in (7369,7654,7900) ;
— like
select empno,enmae from emp where ename like 'SM%' ;
select empno,enmae from emp where ename like '_M%' ;
— order by
select empno,ename,sal from enp where sal >=1500 order by sal asc ;
select empno,ename,sal from enp where sal >=1500 order by sal desc ;
select empno,ename,sal from enp where sal >=1500 order by 1 desc ;– order by par 1 colonne,donc par empno
select empno,ename,sal,hiredate from enp where sal >=1500 order by sal desc,hiredate asc ;

— union supprimer des redondances données,union all ne supprimer pas des redondances des données
select empno,sal from emp
union
select empno,sal from emp ;
select a,b from table1
union
select c,d from table2 ;
type a = type c, type b =type d
select a,b,e from table1
union
select c,d,3.14 from table2 ;
type a = type c, type b =type d, type e = type (3.14)number
select a,b from table1
union all
select c,d from table2 ;
type a = type c, type b =type d
— dual
select 1 from dual ;
select 'bonjour' as b from dual ;

— function de date : *type date + numéro = nouvelle date*
type date - type date = numérique
select hiredate,hiredate+2 from emp ; +2 = 2 jour
select hiredate,hiredate+ (2/24) from emp ; +(2/24) = 2 heurs

*select hiredate,hiredate+ (2/(24*60)) from emp ; +(2/(24*60)) = 2 minutes*
select sysdate from dual ;
select to_ char(hiredate,'dd%mm%yyyy'),ename from emp

— obtenir le nombres de weekends :
select ename,(sysdate-hiredate)/7 weekend from emp whre deptno ;
select ename,round((sysdate-hiredate)/7) weekend from emp whre deptno ;
select ename,round(months_ between(sysdata,hiredate)) from emp ;

— to_char format l'argent :
select to_ char(sal,'999,999,999,999.99') from emp ;
select to_ char(sal,'$999,999,999,999.99') from emp ;
select to_ char(sal,'€999,999,999,999.99') from emp ;
select to_ char(sal,'L999,999,999,999.99') from emp ;
select to_ char(deptno,'000000'),ename from emp ;

— calculer salaire d'année :
*select ename,job,sal,comm,nvl(comm,0) + sal*12 from emp ;*
*select ename,job,sal,comm,nvl2(comm,comm,0) + sal*12 from emp ;*
nvl2 si comm n'est pas null,on utilise sa valeur sinon on prend 0

— decode :
select ename,decode(deptno,10,'Vendeur',20,'Développe',30,'Cadre') from emp ; select ename,decode(deptno,10,'Vendeur',20,'Développe','Autre') from emp ;

— case..when :
select case when mod(&n,2)=1 then 'f' else 'h' end from dual ;
select hiredate,
case
when to_ char(hriedate,'mm') <='06' then
'S'
else
'X' end ti
from emp ;

— count,min,max,avg,sum :
select count(*),min(sal),max(sal),avg(sal),sum(sal) from emp ;
select count(2),sum(2) from emp ;
select count(comm),sum(comm) from emp ;
select sum(sal,nvl(comm,0)) from emp ;
select sum(sal) + sum(comm) from emp ;

— group by, having :
select count(*) from emp where deptno = 10 ;
select count(*),deptno from emp group by deptno ;
select deptno,avg(sal) from emp group by deptno having avg(sal) >2500 ;
select deptno,avg(sal) from emp where deptno in (20 ,30) groupe by deptno having avg(sal) > 2000 order by avg(sal) ;

— left join.. on,right join ..on,join ..on,natural join,cross join, full join .. on :

select emp.ename,dept.dname from emp,dept where dept.deptno = emp.deptno ;
select emp.ename,dept.dname from emp,dept where dept.deptno(+) = emp.deptno ;—>table emp est table principale
select emp.ename,dept.dname from emp,dept where dept.deptno = emp.deptno(+) ;—->table dept est table principale
select emp.ename,dept.dname from emp natural join dept
select emp.ename,dept.dname from emp join dept on dept.deptno = emp.deptno ;
select e1.empno,e1.ename,e2.empno,e2.ename from emp e1 left join emp e2 on e1.mgr = e2.empno ;
select emp.ename,dept.dname from emp right join dept on dept.deptno = emp.deptno ;
select emp.ename,dept.dname from emp full join dept on dept.deptno = emp.deptno ;

— any ,all :

```
  select ename,deptno,job from emp where (deptno,job) =
                          any(select deptno,job from emp
                      where ename in ('SMITH','JONES'))
                    and ename not in ('SMITH','JONES'));

select deptno,min(sal) from emp group by depno;

  select ename,sal,deptno from emp where sal > any(
      select deptno,min(sal) from emp group by depno)
      order by sal asc;

  select ename,sal,deptno from emp where sal < any(
      select deptno,min(sal) from emp group by depno)
      order by sal asc;

  select * from emp where sal > all (
  select min(sal) from emp group by deptno
  ) order by sal;

  select e.ename,e.sal,e.deptno from emp e where sal > any(
   select sal from emp where deptno = e.deptnno
  );

```

— with :

```
with
  myemp as (select empno,ename,deptno from emp where sal > 1000)
  select * from myemp where deptno = 10;

with
  e as (select * from emp where deptno = 20),
  d as (select * from dept)
  select e.name,d.dname from e,d where e.deptno = d.deptno;

with
```

```
    a as (
    select 11,'xyz' from dual
    union\\
    select 456,'ggg' from dual),
    b as (select sal,ename from emp)
    select b.sal,b.ename from a ,b;

```

— existe , not existe :

```
    select 1 from dual where  1 > 2;
    select 1 from  dual where exists(select ename,sal from
    emp)
    select 1 from  dual where exists(select 2 from were 1 <>
    1 )
    select 1 from  dual where not exists(select 2 from were 1
     <> 1 )

```

— select récursif

```
        select e.empno,e.ename,m.empno,m.ename from emp e,
    emp m
        from e.mgr =m.empno;

        select empno,ename,mgr,connect_by_isleaf emp connect
    by prior mgr = empno;

        select empno,ename,mgr,connect_by_isleaf emp
      start with mgr is nout null
        connect by prior mgr = empno;
        /**chercher des responstables de 7369**/
        select empno,ename,mgr,connect_by_isleaf emp
      start with empno = 7369
        connect by prior mgr = empno;

        select empno,ename,mgr,connect_by_isleaf emp connect
    by mgr = prior empno;

        /*chercher des subordonnes de 7788*/
        select empno,
               ename,
               mgr,
               connect_by_isleaf emp,
               level,
               connect_by_root ename 'nom'
         start with empno = 7788
           connect by mgr = prior empno;

        select empno,
             ename,
             mgr,
             connect_by_isleaf emp,
             level,connect_by_root ename 'nom',
           sys_connect_by_path(mgr,'-')
          start with mgr in not null
            connect by prior mgr= empno;

        select empno,
```

```
            ename,
            connect_by_root ename root,
            mgr,
            sys_connect_by_path(empno,'-->'),
            level couche,
            connect_by_isleaf is_est_feuille
            from emp
            start with mgr is null
            connect by mgr = prior empno;

```

1.5.5 view

```
    create or replace view v_emp as select deptno,
        count(*) nbemp from emp
        group by deptno;
        select * from v_emp;
        update v_emp set nbemp where deptno = 30;

```

1.6 function

1.6.1 les function d'oracle

— upper,lower,initcap,concat,||,substr,length, replace,translate,round,ascii,chr,lpad,rpad,trim,least,mod,add_months months_between ,next_day, last_day, trunc, dbms_random ,to_char, to_number , to_date , nvl2 ,nullif ,deconde ,sum,max,min,count, count(distinct colonne), avg

```
      select upper('bonjour') as
           b,lower('BONJOUR') as c
            from dual;

      select initcap('bONJOUR tout le monde') as b
            from dual;

      select concat('bon','jour')
             from dual;

      select concat(concat('bon','jour'),
             'tous le monde')
            from dual;

      select 'bon'||'jour'||'tous le monde'
          from dual;

      select substr(ename,0,2)
            from emp;

      select substr(ename,-2)
          from emp;

```

```
24        select length(ename)
25              from emp;
26
27        select replace('192.168.1.63','.','/')
28            from dual;
29
30        select replace('192  168  1  63',' ','')
31              from dual;
32
33        select translate('i love you','oi','*#')
34            from dual;
35
36        select translate('i love you','oiu','*#')
37              from dual;
38
39        select 'from de numerique' as numerique
40            from dual  where translate('12a8305214','
    #0123456789','*') is null;
41
42        select round(5.9)
43              from dual;
44
45        select round(978.987245,4)
46              from dual;
47
48        select ascii(a) from dual;
49
50        select 'is numerique'
51           from dual where ascii('4') between 48 and 57;
52
53        select chr(97) from dual;
54
55        select lpad('alen',6,'CH') from dual;
56
57        select rpad('alen',6,'fr') from dual;
58
59        select trim(' 123 46 789  ') from dual;
60
61        select trim(leading from ' 123 46 789  ')
62           from dual;
63
64        select trim(trailing from ' 123 46 789  ')
65           from dual;
66
67        select least(17,0,32,90,6,-17) from dual;
68
69        select mod(3,10) from dual;
70
71        select 'impaire'
72           from dual where mode(1,3) <> 0;
73
74        select add_months(sysdate,2) from dual;
75
76        select add_months(sysdate,-2) from dual;
77
78        select months_between(sysdata,sysdate) from dual;
79
```

```
80        select next_day(sysdate,'jeudi') from dual;
81
82        select last_day(sysdate) from dual;
83
84        select trunc(sysdate,'yyyy') from dual;
85
86        select trunc(sysdate,'mm') from dual;
87
88        select trunc(sysdate,'q'),
89             trunc(sysdate,'dy')+1
90           from dual;
91
92        select trunc(878.99,1) from dual;
93
94        select trunc(878.99097,4) from dual;
95
96        select dbms_random.value() from dual;
97
98        select dbms_random.value(10,21)
99           from dual;#generer une numerique aleatoire entre 10
      et 20
100
101       select trunc(dbms_random.value(10,21))
102          from dual;#generer une numerique aleatoire entre 10
      et 20
103
104       select trunc(dbms_random.value(10,21))
105          from dual connect by level < 6;#generer 5
     numerique aleatoire entre 10 et 20
106
107       select trunc(dbms_random.value(10,21))
108          from dual connect by level <= 15;#generer 15
     numerique aleatoire entre 10 et 20
109       select to_char(sysdate,'dd\%mm\%yyyy') from dual;
110
111       select to_char(sysdate,'yyyy') from dual;
112
113       select to_char(sysdate,'dd') from dual;
114
115       select to_number('9999') from dual;
116
117       select to_date('2008-08-08','yyyy-mm-dd') from dual;
118
119       select nvl(null,3,2) from dual;
120
121       select nullif('a','b') from dual;
122
123       select nullif('a','a') from dual;
124
125       select check,
126          decode(sex,'M','homme','f','femme',NULL,'None sex
     ')
127          from table;
128
```

— chercher ename,sal,sal de moyenn de chaque deptno

```
1       select ename,sal deptno from emp;
```

```
2
3       select ave(sal),deptno from emp group by deptno;
4
5       select t1.*,t2.* from
6         (select avg(sal) moyeen, deptno
7           from emp groupe by deptno ) t1,
8         (select ename,sal,deptno from emp ) t2
9       where t2.sal > t1.moyeen
10          and t1.deptno = t2.deptno;
11
12
13      select ename,sal,deptno
14         from emp t1
15         where sal > (select avg(sal)
16                from emp
17                where t1.deptno = deptno
18                group by deptno);
19
20
21      select ename,
22           sal,
23           avg(sal) over(partition by deptno)
24        from emp;
25
26      with
27        emp_moyen as (select ename,
28           sal,
29           deptno
30           avg(sal) over(partition by deptno) moyeen
31        from emp
32      )
33      select e1.sal,e1.moyeen,e.ename
34        from emp_moyen e1,emp e
35        where e.deptno = e1.deptno
36        and e.sal > e1.moyeen;
37
38      select ename,
39           sal,
40           sum(sal) over(partition by deptno order by sal)
    somme
41      from emp;
42
43
44      select ename,
45           sal,
46           sum(sal) over(partition by deptno
47                  order by sal rows
48                  between unbounded preceding
49                  and unbounded following ) somme
50      from emp;
51
52
53
54      select ename,
55           sal,
56           sum(sal) over(order by sal
57                  rows between unbounded preceding
```

```
                  and unbounded following ) somme
    from emp;


    select empno,
         enmae,
         sal,
         rank() over(order by sal desc) ranks,
         dense_rank() over(order by sal desc) desrank,
         row_number() over(order by sal des) rows
    from emp;


    select min(ename) keep(dense_rank first order by sal
   desc) ename1,
         min(ename) keep(dense_rank last order by sal desc)
    ename2
     from emp;


```

1.7 pl-sql

1.7.1 pl-sql simple

— **déclarer une variable en plsql**

```
DECLARE
v_job emp.job%Type;
v_prime NUMBER(5,2) := 500.50
v_prime_min v_prime%Type :=v_prime *2
emp_record emp%ROWTYPE;
BEGIN
emp_record.empnp := 2462;
emp_record.sal := v_sal + 100; END;
```

—
```
DECLARE
TYPE brevets_typetab is TABLE of varcher2(6) INDEX BY BINARY_INTEGER
BEGIN
emp_record.empnp := 2462
emp_record.sal := v_sal + 100;
END;
```

—
```
SET SERVEROUTPUT ON ACCEPT s_nbr PROMPT 'Saisir un entier';
DECLARE
v_doub NUMBER;
BEGIN
v_doub := &s_nbr;
dbms_output.put_line('le résultat est ' || v_doub);
END;
```

— goto

```
    Begin
```

```
        <<loc>>
        for i in 1..5 LOOP
            if  i > 2 then
               goto loc;
             end if;
             dbms_output.put_line(i);
         end loop;
       end;
```

1.7.2 cursor

— found

```
      decalre
       cursor cur_emp is select empno,ename,e.deptno deptno,
   dname
       from emp e ,dept d
       where e.deptno = d.deptno;
       v_empno emp.empno%type;
       v_ename emp.ename%type;
       v_deptno emp.deptno%type;
       v_dname dept.dname%type;
      begin
        open cur_emp;
        fetch cur_emp into v_empno,v_ename,v_deptno,v_dname;
        while cur_emp%found loop
         dbms_output.put('ligne ' || cur_emp%rowcount ||
   v_empno || v_ename
         || v_deptno || v_dname);
      end loop;
      close cur_emp;
      end;
```

— sql dynamique execute immediate

```
         decalre
       cursor cur_emp is select * from emp;
       v_add varchar2;
      begin
        open cur_emp;
        for v_emp in cur_emp loop
        if v_emp.sal <= 1000 then
         v_add := '+20%'
        elsif v_emp.sal > 1000 and v_emp.sal >= 2000 then
         v_add := '+15%'
        else
         v_add := '+5%'
        end if;
         dbms_output.put('le sal de ' || v_emp.name ||'
   est ' || v_emp.sal);
         dbms_output.put('modification apees le sal de '
         || v_emp.name ||' est ' || v_emp.sal);
         execute immediate 'insert in to  A(id,name)
   values(:1,:2)' using 2,'alen';
      end loop;
```

```
      close cur_emp;
      end;

```

```
        decalre
          v_emp emp%rowtype;
        begin
          execute immediate 'select * from emp where sal
    >:sal and deptno =:deptno'
          into v_emp
          using 2000,10;
          dbms_output.put('le sal de ' || v_emp.name ||'
    est ' || v_emp.sal);
        end;

```

— cursor dynamique

```
        declare
        type refcur is ref cousor; --[ref couosr ] cursor
    dynamique
      v_cur refcur;
      v_emp emp%rowtype;
        begin
          open v_cur for 'select * from emp where sal <
    3000';
          loop
          fetch v_cur into v_emp;
          exit when v_cur%notfound;
          dbms_output.put('le sal de ' || v_emp.name ||'
    est ' || v_emp.sal);
          end loop;
          close v_cur;
        end;

```

1.7.3 exception

— exception

```
        declare
          v_x int;
        begin
          dbms_output.put_line('commencer');
          select to_number('abc') into v_x from dual;
          dbms_output.put_line('finit');
        exception
          when no_data_found then
            dbms_output.put_line('il y a pas de donnee');
          when others then
            dbms_output.put_line('des autres exceptions');
        end;

```

1.7.4 procédure

— procédure

— consulter toutes les function sous l'utilisateur courant

```
    select * from user_procedures;

```

— examplse simple

```
    select * from user_procedures;
      create or replace procedure myproc
      is
      being
        delete dept where deptno in (80,81);
        insert into depte(deptno,dname) values(80,'
education');
        insert into depte(deptno,dname) values(81,'
makting');
        commit
      exception
        when others then
          rollback;
      end;
    excuter procedure methode 1
    call myproc();
    excuter procedure methode 2
    decalre
      begin
        myproc();
      end;
    excuter procedure methode 3
    excute myproc();

```

— avec paramètre in out

A in

```
      create or replace procedure myproc1(v_empno in
number)
      is
      v_emp emp%rowtype;
      being
        select * into v_emp from emp where empno=
v_empno;
        dbms_output.put_line('empno ' ||v_emp.empno || '
 nom ' || v_emp.ename);
      exception
        when others then
          dbms_output.put_line('il y a pas de cette
numero');
      end;

    call myproc1(7369)

```

B out

```
      create or replace procedure myproc2(v_empno in
number,v_name out varchar2(30))
      is
      being
```

```
        select ename into v_name from emp where empno=
    v_empno;
        dbms_output.put_line('empno ' || v_empno || '
    nom ' || v_name);
      exception
        when others then
          dbms_output.put_line('il y a pas de cette
    numero');
      end;
      ##excute le procedure###
      decalre
        --v_name varchar2(30);
        v_name emp.ename%type;
      begin
         myproc2(7369,v_name);
         dbms_output.put_line('name ' || v_name);
      end;

```

C inout

```
     create or replace procedure myproc3(v_num1 in out
    number,v_num2 in out number)
      is
      v_temp number;
      being
         v_temp := v_num1;
         v_num1 := v_num2;
         v_num2 := v_temp;
      exception
        when others then
          dbms_output.put_line('il y a pas de cette
    numero');
      end;
      ##excute le procedure###
      decalre
        v_n1 number :=10;
        v_n2 number :=20;
      begin
         myproc3(v_n1,v_n2);
         dbms_output.put_line('v_n1 ' || to_char(v_n1)
    || ' v_n2' || to_char(v_n2));
      end;

```

D le paramètre est cursor dynamique

```
     create or replace procedure myproc4(v_emp out
    sys_refcursor,v_deptno  number)
      is
      v_e emp%rowtype;
      being
        open v_emp  for 'select * from emp where deptno
    =:deptno' using(v_deptno);
        loop
          fetch v_emp into v_e;
          exit when v_emp%notfound;
          dbms_output.put_line('ename ' || v_e.ename);
        end loop;
```

```
          close v_emp;
        exception
          when others then
            dbms_output.put_line('il y a pas de cette
    numero');
        end;
        ##excute le procedure###
        decalre
          type cur_emp is ref cursor;
          v_emp cur_emp;
        begin
           myproc4(v_emp,10);
        end;

```

1.7.5 function

— consulter toutes les function sous l'utilisateur courant

```
    select * from user_objects uo where uo.object_type = '
    FUNCTION';

```

— function simple

— retourner un chaîne caractère

```
        create or replace function fun_out
        return varchar2
        is
        begin
          return 'bonjour';
        end;
      ##excute la function ###
        decalre
          v_ch varchar2(100);
        begin
          v_ch := fun_out();
          dbms_output.put_line('le valeur est ' || v_ch);
        end;

        ##excute la function 2 ###
        select fun_out() from dual;

```

```
        create or replace function max_emp_sal
        return number
        is
        v_sal emp.sal%type;
        begin
          select max(sal) into v_sal from emp;
          return v_sal;
        exception
          when others then
            dbms_output.put_line('il exsite pas');
        end;
      ##excute la function ###
        decalre
```

```
14          v_num emp.sal%type;
15        begin
16          v_num := max_emp_sal();
17          dbms_output.put_line('salaire ' || v_num);
18        end;
19
20        ##excute la function 2 ###
21        select max_emp_sal() from dual;
22
```

— avec paramètre

```
1         create or replace function double_number(v_x number)
2         return number
3         is
4         begin
5           return v_x * 2;
6         end;
7         ##excute la function 1###
8         select sal,double_number(sal) from emp;
9
```

2 avec paramètre

```
1         create or replace function math_sum(v_x number,v_y
    number)
2         return number
3         is
4         begin
5           return v_x  + v_y;
6         end;
7         ##excute la function 1###
8         select sal,math_sum(sal,comm) from emp;
9
```

1.7.6 trigger

— trigger exemple

— rigger simple , for each row

```
1       /*
2         tant qu'on modifier deptno de table dept,
3         on doit aussi modifie la deptno de table emp
4       */
5         create or replace trigger emp_update_trigger
6         after update on dept
7         for each row
8         declare
9         begin
10          update emp set deptno =:new.deptno where deptno
    = :old.deptno;
11        end;
12        ## consulter trigger est resusire a creer 1###
13        select * from user_triggers;
14        update dept set deptno = 80 where deptno = 20;
15        select * from emp;
16
```

— when

```
      /*
        tant qu'on modifier deptno de table dept,
        on doit aussi modifie la deptno de table emp
      */
        create or replace trigger emp_update_trigger
        after update on dept
        for each row
        when(new.deptno=80)
        declare
        begin
          update emp set deptno =:new.deptno where deptno
    = :old.deptno;
        end;
        ## consulter trigger est resusire a creer 1###
        select * from user_triggers;
        update dept set deptno = 80 where deptno = 20;
        select * from emp;

```

— of

```
          create or replace trigger emp_update_trigger
      after update or delete or inster of deptno,dname on
    dept
      for each row
      when(new.deptno=80)
      declare
      begin
        update emp set deptno =:new.deptno where deptno = :
    old.deptno;
      end;
      ## consulter trigger est resusire a creer 1###
      select * from user_triggers;
      update dept set deptno = 80 where deptno = 20;
      select * from emp;

```

— copie de sécurité

```
      create table dept_bak as select * from dept where 1
    <> 1;
      alter table dept_bak add (
      bak_type varchar2(100),
      bak_time Timestamn
      );
        create or replace trigger back_dept_tri
      before update or delete or inster of deptno,dname on
    mathieur.dept
      for each row
      declare
      begin
        insert into dept_bak
        values(:old.deptno,:old.dname,:old.loc,'update',
    sysdate);
      end;
      ## consulter trigger est resusire a creer 1###
      select * from user_triggers;
```

```
        update dept set deptno = 80 where deptno = 20;
        select * from emp;

```

— sans for each row

```
          create or replace trigger upd_emp_tri
        after update on mathieur.dept
        declare
        begin
          update emp set sal = 3000 where deptno = 10;
        end;
        ## consulter trigger est resusire a creer 1###
        select * from user_triggers;
        update dept set dname = 'makting' where deptno = 20;
        select * from emp;

```

— instead of

```
        create or replace view v_emp as select empno,
        ename,e.deptno,dname from emp  e,dept d
        where e.deptno = d.deptno ;
        update v_emp set deptno =80 where deptno = 20;

          create or replace trigger upd_emp_tri
          instead of v_emp
          for each row
        declare
        begin
          update emp set deptno = :new.deptno where deptno =
    :old.deptno;
          update dept set deptno = :new.deptno where deptno =
     :old.deptno;
        end;
        ## consulter trigger est resusire a creer 1###
        ;

```

— system trigger

```
        create table database_log(demaari_date,timestamp);


          create or replace trigger database_startup
          after startup
          on database
        begin
          insert into database_log valeus(sysdate);
        end;
        ## consulter trigger est resusire a creer 1###
        shutdown immediate
        startup
        ;

```

— ddl trigger (trigger d'événement d'utilisateur)

```
        create table login_database(who varchar2(20),log
    timestamp);
```

```

        create or replace trigger login_database_trigger
        after logon
        on scheama
      begin
        insert into login_database valeus(user,sysdate);
      end;
      ## consulter trigger est resusire a creer 1###
      ;

```

— recompiler trigger [compile]

```
      alter trigger login_database_trigger compile

```

— éteindre trigger [disable]

```
      alter trigger mathieu.login_database_trigger disable

```

— allumer trigger [enable]

```
      alter trigger mathieu.login_database_trigger enable

```

— supprier trigger

```
      drop trigger mathieu.login_database_trigger

```

1.7.7 package

— exemple

```
      create or replace package  pak_test
      is
      v_x number;
      procedure prc_cal(v_ename out varchar2,v_empno number);
      function fun_1(x number, y number,v_oper varchar2) return number;
      type cur_dyna is ref cursor;
      end;
      create  or replace package body pak_test
      is
      procedure prc_cal(v_ename out varchar2,v_empno number)
      is
        begin
          select enmae into v_ename from emp where empno = v_empno;
        end;
      function fun_1(x number, y number,v_oper varchar2) return number
      is
      begin
        if v_oper = '+' then
          return  x+y;
```

```
20          elsif v_oper = '-' then
21             return x - y;
22          elsif v_oper = '/' then
23            return x / y
24          elsif v_oper = '*' then
25             return x * y;
26          else
27            return = -1
28           end if;
29        end;
30
31
32      declare
33        v_refur pak_test.cur_dyna;
34        v_name emp.ename%type;
35      begin
36        pak_test.v_x := 100;
37        dbms_output.put_line(pak_test.v_x);
38        open v_refur for 'select ename from emp';
39        loop
40          fetch v_refur into v_name;
41          exit when v_refur%notfount;
42          dbms_output.put_line(v_name);
43        end loop;
44        close v_refur;
45      end;
46
```

1.7.8 job

— exemple

```
1         create table A (content varchar2(20));
2         create or replace procedure pro1
3         is
4         declare
5         begin
6           insert into A values(to_char(sysdate,''hh:mi''))
7         end;
8
9         declare
10          jobid number;
11        beign
12          dbms_job.submit(jobid,
13          'declare
14            begin
15              insert into A values(to_char(sysdate,''hh:mi
    ''))
16            end;
17          ',--ou on peut mettre 'pro1;' a ici
18          to_date('16:00','hh24:mi'),
19          'sysdate +1/(24*60)');
20          commit;
21        end;
22
23        /********consulter tous les jobs*********/
24        select job,log_user,priv_user from user_jobs;
```

```
25          begin
26            dbms_job.run(job);
27          end;
28          /*remove job**/
29          begin
30            dbms_job.remove(job);
31            commit;
32          end;
33
```