TP2 : PL/SQL (appropriation suite)

1. Schéma de base de données

Le TP2 est la suite du TP1 et nous reprenons le même schéma relationnel (portant sur les employés et les départements).

2. Trigger, procédure et SQL dynamique

Vous exploiterez certaines fonctionnalités (EXECUTE IMMEDIATE) du langage SQL dit dynamique pour supprimer tous les triggers définis jusqu'alors, à l'aide d'une procédure adaptée.

```
CREATE OR REPLACE PROCEDURE DELETE_ALL_TRIGGERS
as
cursor curseur is select trigger_name from user_triggers;
begin
   FOR rec in curseur
   LOOP
      EXECUTE IMMEDIATE 'DROP TRIGGER '||rec.trigger_name;
    END LOOP;
end;
/

exec DELETE_ALL_TRIGGERS
```

Listing 1 – Une correction possible

3. Curseurs, procédures et fonctions

3.1 Traitements métiers

3.1.1 Procédure EmployesDuDepartement

Vous définirez une procédure qui prend en entrée un numéro de département et qui en sortie, retourne une chaîne de caractères listant les différents employés de ce département. Vous proposerez une gestion des exceptions pouvant survenir et vous construirez un programme principal qui tire parti de cette procédure.

```
set serveroutput on
-- chr(10) carriage return
set linesize 200
```

```
create or replace procedure EmployesDuDepartement (numDep in integer, lesEmployes out
    varchar)
is
cursor C is select num, nom, salaire, fonction from emp where n_dept=numDep;
vide exception;
begin
lesEmployes := 'nada';
for c_t in C
loop
lesEmployes := lesEmployes || c_t.num||' '||c_t.nom||' '||c_t.salaire||'
    '||c_t.fonction||chr(10);
end loop;
if lesEmployes = 'nada' then raise vide;
end if;
exception
when vide then dbms_output.put_line('Pas de num correspondant');
when others then dbms_output.put_line('survenue du pb suivant '||SQLERRM);
end;
/

declare
employes varchar(1000);
begin
EmployesDuDepartement(10,employes);
dbms_output.put_line(employes);
end;
/


variable employes varchar(1000);
variable n_dept integer ;
execute :n_dept := 10;
exec EmployesDuDepartement(:n_dept, employees)



declare
employes varchar(1000);
begin
EmployesDuDepartement(2,employes);
dbms_output.put_line(employes);
end;
/
```

Listing 2 – Une correction possible

3.1.2 Fonction CoutSalarialDuDepartement

Vous définirez une fonction qui prend en entrée un numéro de département et qui en sortie, retourne le coût salarial total de ce département. Vous en exploiterez les effets d'abord au travers d'une requête, puis ensuite au travers d'un programme principal.

```
create or replace function CoutSalarialDuDepartement (numDep in integer)
return float is
somme float;
erreur exception;
begin
```

```
select sum(salaire) into somme from emp where n_dept= numdep;
if somme is null then raise erreur; end if ;
return somme;
exception
when erreur then return -1;
end;
/

select CoutSalarialDuDepartement(10) as CoutEuros from dual;

select CoutSalarialDuDepartement(2) as CoutEuros from dual;


declare
budget float;
begin
budget := CoutSalarialDuDepartement(10);
dbms_output.put_line('cout de fonctionnement '||budget||' euros');
end;
/
```

Listing 3 – Une correction possible

3.2 Supervision utilisateurs de la base

Vous construirez un paquetage nommé Supervision qui contient (au moins) les quatre éléments suivants :

1. A partir des vues du dictionnaire de données nommées v$session et dba_users, vous définirez une fonction renvoyant le taux d'utilisation de la base master ((utilisateurs connectés / utilisateurs référencés) * 100)
2. A partir des vues v$session et dba_tables, vous construirez une procédure qui affiche pour chaque utilisateur connecté, le nombre de tables de son schéma ainsi que le nombre total de tuples contenus dans ces tables. Vous pouvez également renvoyer le nombre total de colonnes formant ces tables et de contraintes s'appliquant à ces tables.
3. A partir des vues v$session et v$process, vous construirez une procédure qui affiche différentes informations sur les usagers connectés : sessions en cours, sur quel machine, à partir de quel client et depuis quand.
4. A partir des vues concernant les privilèges et rôles système, vous construirez une procedure qui affiche pour un utilisateur donné, les différents rôles et/ou privilèges dont il dispose.

Vous pouvez aussi ajouter de nouvelles fonctionnalités au paquetage de supervision au gré de votre imagination.

```
-- exploiter une vue dans une fonction demande ce droit (directement et non pas au
    travers d'un role)
GRANT SELECT ANY DICTIONARY TO PUBLIC ;

create or replace package supervise
as
function pourcentageConnexions return number ;
procedure lesConnectes ;
procedure lesTablesDesConnectes ;
```

```
end ;
/


create or replace package body supervise
as

function pourcentageConnexions return number
as
nbre1 integer;
nbre2 integer;
begin
select count(distinct username) into nbre1 from dba_users;
select count(username) into nbre2 from v$session where type='USER';
return nbre2/nbre1 * 100;
exception when others then dbms_output.put_line(sqlcode||' '||sqlerrm);
end;

procedure lesConnectes
is
cursor c is select rpad(s.username,14) as usager, s.module as logiciel,
    rpad(s.osuser,40) as osUsager, p.program,
to_char(s.logon_time,'DD-MM-YY---HH-MM-SS') as dateConnexion, s.machine as machina,
    p.pid as oracleProcess, p.spid as osProcess
from v$session s, v$process p where s.paddr = p.addr and s.type = 'USER';
begin
for t in c
loop
dbms_output.put_line(t.usager||' '||t.osUsager||' Processus Syst exploitation
    '||t.osProcess||' le '||rpad(t.dateConnexion,16));
end loop ;
end ;

procedure lesTablesDesConnectes
is
cursor c1 is select count(table_name) as nbTables, sum(num_rows) as nbTuples, owner from
    dba_tables d, v$session v where d.owner=v.username and v.type ='USER' group by owner
    order by owner;
begin
for t1 in c1
loop
dbms_output.put_line(rpad(t1.owner,14)||' pour '||t1.nbTables||' tables et
    '||t1.nbTuples||' tuples ');
dbms_output.put_line('============================');
for t2 in (select table_name as tbl, owner from dba_tables where owner = t1.owner)
loop
dbms_output.put_line(rpad(t2.tbl,36)||' '||rpad(t2.owner,14));
end loop;
end loop;
end ;

end ;
/


select supervise.pourcentageConnexions from dual;

exec supervise.lesConnectes
```

```
exec supervise.lesTablesDesConnectes
```

Listing 4 – Une correction possible avec paquetage