

TP : ARBD

PL SQL

Nom : ACHACHI

Prénom : El hadj Ali

Mat : 181832008538

- 1) Afficher les noms des sportifs conseillés par rang ensuite afficher le nombre des conseillés existant :

```
SQL> SET SERVEROUTPUT ON;
SQL>
SQL> declare
2  cursor cr is
3  select * from sportifs where idsportif in (select IDSPORTIFCONSEILLER from sportifs);
4  a integer;
5  begin
6  a := 0;
7  for item in cr
8  loop
9  a := a+1;
10 dbms_output.put_line('Nom: ' || ' ' || item.Nom || ' ' || ' ' Prenom: ' || ' ' || item.PRENOM || ' est un conseiller');
11 end loop;
12 dbms_output.put_line('Nombre des conseiller est: ' || a );
13 exception
14 when NO_DATA_FOUND then dbms_output.put_line('la base ne contient aucun demande ');
15 end;
16 /
Nom: BOUTAHAR Prenom: Abderahim est un conseiller
Nom: BOUROUBI Prenom: Anis est un conseiller
Nom: BOUZIDI Prenom: Amel est un conseiller
Nom: LACHEMI Prenom: Bouzid est un conseiller
Nom: AAKOUB Prenom: Linda est un conseiller
Nom: ABBAS Prenom: Sophia est un conseiller
Nom: HADJ Prenom: Zouhir est un conseiller
Nom: HAMADI Prenom: Hani est un conseiller
Nom: ABDELMOUMEN Prenom: Nadia est un conseiller
Nom: ABDELOUAHAB Prenom: Lamia est un conseiller
Nom: BABACI Prenom: Mourad est un conseiller
Nom: ADIBOU Prenom: Ibrahim est un conseiller
Nom: RIADI Prenom: Walid est un conseiller
Nombre des conseiller est:13
```

- 2) Un code PLSQL qui permet d'afficher pour chaque sport le nombre total des gymnases qui organisent des séances pour ce sport :

```
SQL> SET SERVEROUTPUT ON;
SQL> declare
2  cursor cr is
3  select s.LIBELLE, count(g.IDGYMNASES) as nb_gymnases
4  from sports s , seances c, gymnases g
5  where c.IDGYMNASE = g.IDGYMNASES
6  and s.IDSPORT = c.IDSPORT
7  group by s.LIBELLE;
8  begin
9  for item in cr
10 loop
11 dbms_output.put_line('Le sport' || ' ' || item.LIBELLE || ' ' || 'est organiser par ' || ' ' || item.nb_gymnases || ' ' || 'gymnases');
12 end loop;
13 exception
14 when NO_DATA_FOUND then dbms_output.put_line('Le sport pas organiser ');
15 when others then dbms_output.put_line('erreur ' || sqlcode || sqlerrm);
16 end;
17 /
Le sport Basket ball est organiser par 6 gymnases
Le sport Hockey est organiser par 20 gymnases
Le sport Hand ball est organiser par 18 gymnases
Le sport Volley ball est organiser par 11 gymnases
```

- 3) Ajouter la contrainte suivante : l'âge d'un sportif doit être entre 20 ans et 45 ans et Ecrire une procédure qui augmente l'âge de chaque sportif a 5 ans :

```
SQL> ALTER TABLE SPORTIFSS
  2 ADD CONSTRAINT test CHECK(AGE BETWEEN 20 AND 45);

Table modifiée.

SQL> ALTER TABLE SPORTIFSS
  2 DISABLE CONSTRAINT test;

Table modifiée.

SQL> select constraint_name, constraint_type, status from user_constraints where table_name = upper('sportifss');

CONSTRAINT_NAME          C STATUS
-----
DK_INDISPONIBLE          C ENABLED
AK_SPORTIFS              P ENABLED
SK_SPORTIFS              R ENABLED
TEST                     C DISABLED

SQL> CREATE OR REPLACE PROCEDURE AUGMENT_AGE IS
  2 CURSOR cr IS
  3 SELECT idsportif,nom,prenom,age from sportifss S;
  4 BEGIN
  5 FOR item IN cr
  6 LOOP
  7 UPDATE SPORTIFSS set AGE = AGE + 5 WHERE idsportif = item.idsportif;
  8 dbms_output.put_line('Le : '||' '||item.nom||' '||item.prenom||' ' son age passe de '|| (item.age -5 )||' '|| 'a '||item.age );
  9 end loop;
 10 end;
 11 /

Procédure créée.

SQL> SET SERVEROUTPUT ON;
SQL> execute AUGMENT_AGE;
Le : BOUZIDI Amel son age passe de 40 a 45
Le : LACHEMI Bouzid son age passe de 47 a 52
Le : BOUTAHAR Abderahim son age passe de 45 a 50
Le : BOUROUBI Anis son age passe de 43 a 48
Le : AAKOUB Linda son age passe de 37 a 42
Le : ABBAS Sophia son age passe de 45 a 50
Le : HADJ Zouhir son age passe de 40 a 45
Le : HAMADI Hani son age passe de 45 a 50
Le : ABDELMOUEN Nadia son age passe de 38 a 43
Le : ABAD Abdelhamid son age passe de 38 a 43
Le : ABAYAHIA Amine son age passe de 39 a 44
Le : ABBACI Riad son age passe de 39 a 44
Le : ABBACI Mohamed son age passe de 37 a 42
Le : ABDELOUAHAB Lamia son age passe de 39 a 44
Le : ABDEMEZIANE Majid son age passe de 40 a 45
Le : BENOUDAHA Lamine son age passe de 39 a 44
Le : ACHAIBOU Rachid son age passe de 37 a 42
Le : HOSNI Leila son age passe de 40 a 45
```

- 4) Ecrire une procédure qui effectue cet affichage pour chaque gymnase donner par jour d'ouverture les horaires des premières et dernières séances :

```
SQL> CREATE OR REPLACE PROCEDURE affichage(gym gymnases.NOMGYMNASES%TYPE) IS
  2 premier float:=0;
  3 dernier float:=24;
  4
  5 CURSOR cr IS
  6 select distinct(jour) , horaire from seances s , gymnases g
  7 where s.idgymnase = g.idgymnases and NOMGYMNASES = gym
  8 order by jour;
  9
  10 cursor crr is
  11 select distinct(jour) from seances s , gymnases g
  12 where s.idgymnase = g.idgymnases and NOMGYMNASES = gym
  13 order by jour;
  14
  15 BEGIN
  16 FOR item IN crr
  17 LOOP
  18 dbms_output.put_line('Pour le jour '|| item.jour ||': ');
  19 for items in cr
  20 loop
  21 IF ( items.jour = item.jour and items.horaire>premier) THEN
  22 premier := items.horaire;
  23 end if;
  24
  25 IF ( items.jour = item.jour and items.horaire<dernier) THEN
  26 dernier := items.horaire;
  27 end if;
  28 end loop;
  29 dbms_output.put_line(' debut a : '|| dernier||'.');
  30 dbms_output.put_line(' fin a : '|| premier||'.');
  31
  32 premier:=0;
  33 dernier:=24;
  34 END LOOP;
  35 END affichage;
  36 /
```

Procédure créée.

```
SQL> SET SERVEROUTPUT ON;
SQL> execute affichage('Five Gym Club');
Pour le jour Dimanche:
debut a : 17,3.
fin a : 19,3.
Pour le jour Jeudi:
debut a : 20.
fin a : 20.
Pour le jour Lundi:
debut a : 9.
fin a : 20.
Pour le jour Mardi:
debut a : 17,3.
fin a : 17,3.
Pour le jour Mercredi:
debut a : 17.
fin a : 17,3.
Pour le jour Samedi:
debut a : 9.
fin a : 17,3.
```

Procédure PL/SQL terminée avec succès.

- 5) Ecrire une fonction qui retourne, pour chaque **sportif** donné, le **nombre des sports entraînés**:

```
SQL> create or replace function nb_Sport (ref_sportif entrainer.IDSPORTIFENTRAINEUR%type) return int IS
  2 total_sport integer;
  3 BEGIN
  4 select count(idsport) into total_sport from entrainer where IDSPORTIFENTRAINEUR= ref_sportif;
  5 return total_sport;
  6 end nb_Sport;
  7 /
```

Fonction créée.

```
SQL> declare
  2 cursor cr is select distinct s.idsportif , s.nom, s.prenom from sportifs s,entrainer e
  3 where s.idsportif= e.IDSPORTIFENTRAINEUR
  4 group by s.idsportif , s.nom, s.prenom ;
  5 begin
  6 for item in cr
  7 loop
  8 dbms_output.put_line('Le sportif ' ||item.nom ||' ' || item.prenom ||' a entrainer ' || nb_Sport(item.idsportif) ||' sports ');
  9 end loop;
 10 EXCEPTION
 11 WHEN no_data_found THEN dbms_output.put_line('La base de données ne contient aucun entraineur');
 12 when others then dbms_output.put_line('Erreur: ' || sqlcode||' ' ||sqlerrm);
 13 close cr;
 14 end;
 15 /
Le sportif BOUROUBI Anis a entrainer 8 sports
Le sportif BOUZIDI Amel a entrainer 4 sports
Le sportif ABBAS Sophia a entrainer 3 sports
Le sportif DJELOUDANE Zinedine a entrainer 1 sports
Le sportif BELHAOUA Besma a entrainer 1 sports
Le sportif COTERI Daouad a entrainer 1 sports
Le sportif KALI Yasser a entrainer 1 sports
Le sportif HENDI Mouad a entrainer 2 sports
Le sportif TERIKI Yacine a entrainer 2 sports
Le sportif GUITENI Adam a entrainer 2 sports
Le sportif ROUSSELI Lamice a entrainer 1 sports
Le sportif MESSOUNI Ismail a entrainer 1 sports
Le sportif TIZEGHAT Badis a entrainer 3 sports
Le sportif HADJ Zouhir a entrainer 4 sports
Le sportif SETIHA Moustapha a entrainer 1 sports
Le sportif BOUACHA Aziz a entrainer 2 sports
Le sportif KACI Samia a entrainer 2 sports
Le sportif HEDDI Zohra a entrainer 1 sports
Le sportif LAZARRI Jamel a entrainer 2 sports
Le sportif BOUTAHAR Abderahim a entrainer 5 sports
Le sportif LACHEMI Bouzid a entrainer 3 sports
Le sportif CHIKHI Nidal a entrainer 1 sports
Le sportif FATAHI Majid a entrainer 2 sports
Le sportif GUERRAOUI Zohra a entrainer 1 sports
```

Procédure PL/SQL terminée avec succès.

- 6) Créer une procédure qui permet d'ajouter **une séance** à partir de tous les attributs nécessaires avec la vérification d'unicité de la clé et l'existence de clé étrangère vers **les tables référencées** :

```
SQL> create or replace procedure Ajout_Seance( IDGYMNASES_ in seances.IDGYMNASE%type, IDSPORT_ in seances.IDSPORT%type,
2 IDSPORTIFENTRAINEUR_ in seances.IDSPORTIFENTRAINEUR%type, JOUR_ in seances.JOUR%type, HORAIRE_ in seances.HORAIRE%type, DUREE_ in seances.DUREE%type) is
3     erreur boolean:= false;
4     id_gymnases seances.IDGYMNASE%type;
5     id_sport seances.IDSPORT%type;
6     ID_SPORTIFENTRAINEUR seances.IDSPORTIFENTRAINEUR%type;
7
8
9 BEGIN
10
11     DECLARE
12         idgymnases_null EXCEPTION;
13     BEGIN
14         if(IDGYMNASES_ is NULL) then RAISE idgymnases_null;
15     else
16         select IDGYMNASES into id_gymnases from gymnases where IDGYMNASES=IDGYMNASES_;
17     end if;
18 EXCEPTION
19     WHEN idgymnases_null then DBMS_OUTPUT.PUT_LINE('error le numéro de gymnase est obligatoire'); erreur:=true;
20     WHEN NO_DATA_FOUND then DBMS_OUTPUT.PUT_LINE('error le numéro de gymnase existe pas'); erreur:=true;
21 END;
22
23
24     DECLARE
25         idsport_null EXCEPTION;
26     BEGIN
27         if(idsport_ is NULL) then RAISE idsport_null;
28     else
29         select IDSPORT into id_sport from sports where idsport=idsport_;
30     end if;
31 EXCEPTION
32     WHEN idsport_null then DBMS_OUTPUT.PUT_LINE('error : le id sport est obligatoire'); erreur:=true;
33     WHEN NO_DATA_FOUND then DBMS_OUTPUT.PUT_LINE('erreur : le id sport existe pas'); erreur:=true;
34 END;
35
36
37     DECLARE
38         IDSPORTIFENTRAINEUR_null EXCEPTION;
39     BEGIN
40         if(IDSPORTIFENTRAINEUR_ is NULL) then RAISE IDSPORTIFENTRAINEUR_null;
41     else
42         select IDSPORTIFENTRAINEUR into ID_SPORTIFENTRAINEUR from entraineur where IDSPORTIFENTRAINEUR=IDSPORTIFENTRAINEUR_;
43     end if;
44 EXCEPTION
45     WHEN IDSPORTIFENTRAINEUR_null then DBMS_OUTPUT.PUT_LINE('error le numéro IDSPORTIFENTRAINEUR est obligatoire'); erreur:=true;
46     WHEN NO_DATA_FOUND then DBMS_OUTPUT.PUT_LINE('erreur le numéro IDSPORTIFENTRAINEUR existe pas'); erreur:=true;
47 END;
48
49
50     DECLARE
51         jour_null EXCEPTION;
52     BEGIN
53         if(jour_ is NULL) then RAISE jour_null;
54     end if;
55 EXCEPTION
56     WHEN jour_null then DBMS_OUTPUT.PUT_LINE('error le jour est obligatoire'); erreur:=true;
57 END;
58
59
60     DECLARE
61         horaire_null EXCEPTION;
62     BEGIN
63         if(horaire_ is NULL) then RAISE horaire_null;
64     end if;
65 EXCEPTION
66     WHEN horaire_null then DBMS_OUTPUT.PUT_LINE('error horaire est obligatoire'); erreur:=true;
67 END;
68
69
70     DECLARE
71         duree_null EXCEPTION;
72     BEGIN
73         if(duree_ is NULL) then RAISE duree_null;
74     end if;
75 EXCEPTION
76     WHEN duree_null then DBMS_OUTPUT.PUT_LINE('error la duree est obligatoire'); erreur:=true;
77 END;
78
79
```

```
80 if (erreur=false) then
81     insert into seances values (IDGYMNASES_ , idsport_ , IDSPORTIFENTRAINEUR_ , jour_ , horaire_ , duree_);
82     DBMS_OUTPUT.PUT_LINE('la seance est bien ajoutée');
83 end if;
84 EXCEPTION
85     WHEN OTHERS then DBMS_OUTPUT.PUT_LINE('error : '||sqlcode||' '||sqlerrm);
86 END Ajout_Seance;
87 /
```

Procédure créée.

```
SQL> SET SERVEROUTPUT ON;
SQL> execute Ajout_Seance(1,1,1,'Mardi',10,15);
error : -1422 ORA-01422: l'extraction exacte ramène plus que le nombre de lignes
demandé
```

Procédure PL/SQL terminée avec succès.