

Département : Informatique	Année : 2019 – 2020 Semestre : 1
Module d'enseignement : Programmation des Bases de Données	<input checked="" type="checkbox"/> Session principale <input type="checkbox"/> Session de rattrapage
Enseignant : Enis Belhassen	<input checked="" type="checkbox"/> Devoir surveillé <input type="checkbox"/> Examen
Classe : 2 ^{ème} année Filière : ING_INF	Durée : 1h30
Barème de notation : Ex 1 → 11 pts / Ex 2 → 6 pts / Ex 3 → 3 pts	<input type="checkbox"/> Documents autorisés <input checked="" type="checkbox"/> Documents Non autorisés <input type="checkbox"/> Calculatrice autorisée

Exercice 1 (11 pts)

Supposons que nous disposons d'une table ETUDIANT créée comme suit :

```
CREATE TABLE ETUDIANT
( ID_ETUDIANT VARCHAR2(10),
  NOM_PRENOM  VARCHAR2(20),
  CLASSE       VARCHAR2(15),
  MOYENNE      NUMBER(5,2),
  CONSTRAINT PK_ETUDIANT PRIMARY KEY (ID_ETUDIANT)) ;
```

Cette table contient l'ensemble des étudiants d'une université.

Ecrire un bloc PL/SQL qui permet de :

- afficher, en utilisant la boucle WHILE, les noms et les classes des étudiants qui ont une moyenne supérieure ou égale à 15.
NB : L'affichage doit être ordonné selon les identifiants des étudiants (4 pts)
- afficher le nombre total de tous les étudiants inscrits dans la table ETUDIANT (1.5 pts)
- afficher le nombre des étudiants qui ont une moyenne inférieure à 10 (1.5 pts)
- afficher le pourcentage des étudiants qui ont une moyenne inférieure à 10 par rapport au nombre total des étudiants. Le pourcentage doit être arrondi à deux chiffres après la virgule avant son affichage (1.5 pts)
NB : la fonction prédéfinie ROUND (variable, n) permet d'arrondir une variable à n chiffres après la virgule
- afficher (2.5 pts) :
 - « Promotion brillante » si ce pourcentage est compris entre 0 et 35.99
 - « Promotion moyenne » si ce pourcentage est compris entre 36 et 69.99
 - « Promotion !!!! » sinon

```

declare
  n1 number;
  n2 number;
  pourcentage number;
  cursor cur is
    select NOM_PRENOM, CLASSE
    from ETUDIANT
    where MOYENNE >= 15
    order by ID_ETUDIANT;
  vetudiant cur%rowtype;
begin
  open cur;
  fetch cur into vetudiant;
  while (cur%found) loop
    DBMS_OUTPUT.PUT_LINE(vetudiant.NOM_PRENOM||' - Classe : '||
                          vetudiant.CLASSE);

    fetch cur into vetudiant;
  end loop;
  close cur;

  select count(*) into n1 from ETUDIANT;
  DBMS_OUTPUT.PUT_LINE('Nombre total tous les étudiants = '|| n1);

  select count(*) into n2 from ETUDIANT where MOYENNE < 10;
  DBMS_OUTPUT.PUT_LINE('Nombre des étudiants qui ont une moyenne
                        inférieure à 10 = '||n2);

  pourcentage := round((n2/n1) * 100,2);
  DBMS_OUTPUT.PUT_LINE('Pourcentage des étudiants qui ont une
                        moyenne inférieure à 10 = '||pourcentage);

  if (pourcentage between 0 and 35.99)
  then DBMS_OUTPUT.PUT_LINE('Promotion brillante-Pourcentage='||
                            pourcentage);
  elsif (pourcentage between 36 and 69.99)
  then DBMS_OUTPUT.PUT_LINE('Promotion moyenne-Pourcentage='||
                            pourcentage);
  else DBMS_OUTPUT.PUT_LINE('Promotion !!!!-Pourcentage='||
                            pourcentage);
  end if;
end;
/

```

Exercice 2 (6 pts)

Ecrire un programme PL/SQL qui permet de :

- créer un tableau TAB, en utilisant une boucle FOR, dont :
 - les index sont numérotés de 1 à 26
 - le i^{ème} élément de ce tableau contient le i^{ème} caractère de l'alphabet français en majuscule. (4 pts)

NB : la fonction prédéfinie CHR (n) retourne le caractère dont le code ASCII est égal à n (CHR(65) → 'A', CHR(66) → 'B', ..., CHR(90) → 'Z')

- afficher les éléments du tableau TAB en utilisant une boucle LOOP (2 pts)

```
declare
  TYPE alphabet IS TABLE OF varchar2(1) INDEX BY BINARY_INTEGER;
  TAB alphabet;
  i number;
begin
  for i in 1..26 loop
    TAB(i) := CHR(64+i);
  end loop;

  DBMS_OUTPUT.PUT_LINE ('Alphabet français : ');
  i := 1;
  loop
    DBMS_OUTPUT.PUT_LINE (TAB(i));
    i := i + 1;
    exit when (i > 26);
  end loop;
end;
/
```

Exercice 3 (3 pts)

1. Quelle est la commande SQL*Plus qui permet d'activer l'affichage des messages de la procédure DBMS_OUTPUT.PUT_LINE ? (0.75 pts)

```
set serveroutput on
```

2. Quelle est la commande SQL*Plus qui permet d'afficher l'utilisateur connecté ? (0.75 pts)

```
show user
```

3. Quelle est la commande SQL qui à partir d'une vue du dictionnaire de données de Oracle permet d'afficher l'utilisateur connecté ? (0.75 pts)

```
select username from user_users;
```

4. Quelle est la commande SQL qui à partir d'une pseudo-colonne permet d'afficher l'utilisateur connecté ? (0.75 pts)

```
select user from dual;
```