



Gestion d'une compétition sportive

On souhaite afficher le classement des concurrents après une épreuve sportive (course à pied, descente à ski, etc.). On dispose des données suivantes :

- liste des concurrents,
- résultat (temps réalisé) obtenu par chaque concurrent.

On souhaite mémoriser le rang (1^{er}, 2^{ème}, 3^{ème}...) et le nom de chaque concurrent, ainsi que l'écart de son temps avec le meilleur temps.

1. À l'aide de SQL*Plus, créer une table CONCURRENT (NUMERO, NOM) qui contiendra la liste des noms des concurrents et une table RESULTAT (NUMERO_CONC, TEMPS) associant à chaque concurrent le temps qu'il a réalisé. Une valeur NULL pour le temps signifie un abandon du concurrent.

2. Peupler les tables CONCURRENT et RESULTAT comme ci-dessous.

CONCURRENT		RESULTAT	
NUMERO	NOM	NUMERO_CONC	TEMPS
1	A	1	3.25
2	B	2	3.15
3	C	3	NULL
4	D	4	3.16
5	E	5	3.14
6	F	6	3.15
7	G	7	3.09
8	H	8	3.10
9	I	9	3.10
0	J	0	3.19

3. À l'aide de SQL*Plus, créer une table CLASSEMENT (RANG, NOM, ECART) qui contiendra le classement des concurrents, du premier au dernier. Les concurrents ayant abandonné ne doivent pas y figurer.

4. Écrire un programme PL/SQL permettant de peupler la table classement. Gérer, de préférence, les ex æquo, qui doivent être de même rang. Le résultat attendu est présenté ci-dessous.

RANG	NOM	ECART
1	G	0
2	H	,01
2	I	,01
3	E	,05
4	B	,06
4	F	,06
5	D	,07
6	J	,1
7	A	,16

Indications :

- Calculer en premier lieu le meilleur temps et le stocker dans une variable. Il permettra par la suite de calculer les écarts.
- Utiliser un curseur incluant la clause ORDER BY pour classer les concurrents.

Question subsidiaire : Modifier le programme PL/SQL afin d'améliorer la présentation des résultats (se baser sur le modèle ci-dessous).

RANG	NOM	ECART
1	G	0
2	H	,01
	I	,01
3	E	,05
4	B	,06
	F	,06
5	D	,07
6	J	,1
7	A	,16

Correction

```
-- Creation de la base de donnees source

create table concurrent (numero number(1) primary key, nom char(1));

insert into concurrent values (1, 'A');
insert into concurrent values (2, 'B');
insert into concurrent values (3, 'C');
insert into concurrent values (4, 'D');
insert into concurrent values (5, 'E');
insert into concurrent values (6, 'F');
insert into concurrent values (7, 'G');
insert into concurrent values (8, 'H');
insert into concurrent values (9, 'I');
insert into concurrent values (0, 'J');

create table resultat (numero_conc number(1) primary key, temps number(4,2));

insert into resultat values (1, 3.25);
insert into resultat values (2, 3.15);
insert into resultat values (3, NULL);
insert into resultat values (4, 3.16);
insert into resultat values (5, 3.14);
insert into resultat values (6, 3.15);
insert into resultat values (7, 3.09);
insert into resultat values (8, 3.10);
insert into resultat values (9, 3.10);
insert into resultat values (0, 3.19);

-- Remplissage de la table classement (bloc PL/SQL)

DECLARE

    -- Meilleur temps
    best resultat.temps%TYPE;

    -- Liste des concurrents classes par temps croissant (du meilleur au moins bon)
    -- On ignore les concurrents ayant abandonne
    cursor liste is
        select nom, temps from concurrent, resultat
        where numero=numero_conc
        and temps is not null
        order by temps;
    c liste%ROWTYPE;

    -- Ecart au 1er
    ecart resultat.temps%TYPE;

    -- Indice de rang
    rg integer:=0;

    -- Temps du concurrent precedent
    temps_prec resultat.temps%TYPE:=-1;
```

BEGIN

```
-- Nettoyage de la table classement
delete from classement;

-- Sauvegarde du meilleur temps
select min(temps) into best from resultat;

-- Parcours de la liste des concurrents et remplissage de la table classement
for c in liste loop
    -- Calcul de l'ecart avec le meilleur temps
    ecart:=c.temps-best;
    -- Si le temps du concurrent courant est different
    -- du temps du concurrent precedent
    if c.temps>temps_prec then
        -- Concurrent courant moins bien classe, le rang augmente de 1
        rg:=rg+1;
        insert into classement values (rg, c.nom, ecart);
    else
        -- Meme classement que le concurrent precedent, le rang n'augmente pas
        insert into classement values (NULL, c.nom, ecart);
    end if;
    -- Mise a jour du temps du concurrent precedent (= temps courant)
    temps_prec:=c.temps;
end loop;

END;
;
```