

## DEVOIR DE CONTROLE

ECUE : SGBD			Classe(s) : L2DSI*	
Enseignantes : Mmes L.Gatri, J.Dhiab, A.Ben Salem & N.Abidi			Documents autorisés : <input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non	
Date : 20/05/2021	Durée : 1heure	Nombre de pages : 2	Calculatrice autorisée : <input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non	

La Fédération Tunisienne d'Athlétisme veut développer un système informatique pour gérer les différentes compétitions.

Ce système doit reposer sur le schéma de base de données suivant :

LIEU\_COMPETITION (Lieu, Ville, Pays)  
 COMPETITION (Code\_Comp, Nom\_Comp, lieu#, Date\_Comp)  
 JURY (No\_Jury, Nom\_Jury, Adresse, Grade)  
 PARTICIPANT (No\_part, Nom\_Part, Date\_Naissance, Adresse, Email)  
 SCORE (No\_Part#, Code\_Comp#, No\_Jury#, Note)

On vous demande d'écrire en PL/SQL :

- 1) Une fonction **Calcul\_Score** (id\_part in varchar2(6), id\_comp in varchar2(6)) return **number** qui calcule le score total d'un participant donné dans une compétition donnée.  
 Le score total d'un participant dans une compétition sera égal à la somme de toutes ses notes obtenues lors de la compétition.  
 Nb : N'oubliez pas de gérer le cas où le participant ou la compétition n'existe pas.
- 2) Un bloc anonyme qui remplit la table **Résultat** puis affiche les numéros, les noms des participants avec leurs scores totaux pour chaque compétition.  
 La table **Résultat** étant créée et vide, dispose de la structure suivante :

**RESULTAT** (No\_par#, code\_comp#, score\_total)

L'affichage du résultat se fera comme suit :

La compétition **CMP001** :

Le participant **P00120** a obtenu le score 65

Le participant **P00010** a obtenu le score 82

...

La compétition **CMP002** :

...

3) Une procédure **Resultat\_Comp** (**id\_comp** in varchar2(6)) qui permet d'afficher le rang et la médaille de chaque participant dans la compétition donnée en paramètres, sachant que la valeur de la médaille est :

- « OR » pour le premier
- « ARGENT » pour le deuxième
- « BRONZE » pour le troisième
- « AUCUN » pour les autres

NB : utilisez la table Resultat.

4) Un déclencheur **TRG\_MAJ\_SCORE** qui modifie automatiquement le score total dans la table Resultat suite à l'ajout ou la suppression d'une ligne de la table SCORE.

ok

<b>Institut Supérieur des Études Technologiques de Radès</b> <b>Département Technologies de l'informatique</b>		
	<b>Épreuve : SGBD</b>  Date : Avril 2016 Enseignante : J. DHIAB	
	Documents : Non autorisés Durée : 1 heure	
<b>Classe : L2-DSI</b>	<b>DEVOIR SURVEILLE</b>	<b>Nombre de pages : 2</b>

Soit le schéma relationnel suivant :

**Individu** (Num\_Individu, Nom\_Individu, Prenom\_Individu)

**Acteur** (Num\_Film, Num\_Individu)

**Genre\_Film** (Num\_Film, Code\_Genre)

**Genre** (Code\_Genre, Libelle\_Genre)

**Film** (Num\_Film, Titre, Realisateur)

**Exemplaire** (Num\_Exemplaire, Num\_Film, Code\_Support, Probleme)

**Client** (Login\_Client, Motpasse\_Client, Nom\_Client, Prenom\_Client, Adr\_Client, Ville\_Client)

**Location** (Login\_Client, Num\_Exemplaire, Date\_Location, Date\_Retour\_Prevu, Date\_Retour\_Effective)

Ce schéma décrit un extrait de la base de données d'une gestion de location de film dans une vidéothèque.

Un Film lui est relatif un ensemble d'exemplaires. Si un exemplaire est défectueux, ceci est mentionné dans le champ Problème, sinon le champ reste vide.

### Question 1 (3pts) :

Ecrire une procédure **info\_film** qui affiche pour un Num\_Film donné, son titre, le nombre total de ses exemplaires, le nombre de ses exemplaires à problème ainsi que le nombre de fois qu'il a été loué.

Traiter le cas où le Num\_Film n'existe pas.



**Question 2 (3pts) :**

Ecrire une procédure qui affiche le nombre ainsi que la liste des acteurs (prénom et nom) ayant joué dans un film dont le titre est donné en paramètre.

Traiter les cas où :

- Le titre n'existe pas.
- Pas d'acteurs mentionnés pour ce film.

**Question 3 (4pts) :**

On désire accorder une remise à chaque client en fonction du nombre de **locations effectuées en 2015**. Pour cela, on désire peupler la table REMISE\_2016 ayant la structure (Login\_Client, remise, nbrExLoues) sachant que le champ remise est de type réel.

Ecrire la procédure **Remise** permettant de peupler la table existante REMISE\_2016 sachant que le champ remise est calculé comme suit :

- si  $0 < \text{nbrExLoues} < 10$  alors accorder une remise de 5%.
- si  $10 \leq \text{nbrExLoues} < 20$  alors accorder une remise de 10%.
- si  $\text{nbrExLoues} \geq 20$  alors accorder une remise de 20%.
- si  $\text{nbrExLoues} = 0$  alors accorder une remise de 0.

**Question 4 (2+2+3+2+1pts) :**

Afin d'améliorer le service marketing et la gestion de son stock, on souhaite avoir à sa disposition à tout moment une série de données statistiques.

1- Créer un package statistiques qui rassemble les fonctions permettant de savoir :

- Le nombre de films pour lesquels on ne possède aucun exemplaire.
- Le pourcentage des exemplaires à problèmes.
- Les titres des 3 films les plus loués.

2- Ecrire la procédure **Log\_Statistiques** qui permet d'ajouter une entrée dans la table LOG\_STAT (Date\_Stat, Nb1, Nb2, Top1, Top2, Top3).

Où :

Date\_Stat désigne la date à laquelle la statistique a été lancée..

Nb1 désigne le nombre de film ne possédant aucun exemplaire à cette date.

Nb2 désigne le pourcentage des exemplaires à problèmes.

Top1 désigne le titre du film le plus loué.

Top2 désigne le 2<sup>ème</sup> titre de film le plus loué.

Top désigne le 3<sup>ème</sup> titre de film le plus loué.