|  |
| --- |
|  |

**Direction Régionale Tensift Atlantique**

**Etablissement : Ista Ntic Syba Marrakech**

**Examen de fin module :    Gestion du Temps**

**2011/2012**

**Filière:TDI                                                   Groupe(s):TDI2GE**

**Niveau : 2ème  année                                   Formateur : OUATOUCH Abdeljalil**

**Durée : 2h                                                Barème:    /20**

Au niveau national, la natation est un sport géré par la Fédération Marocaine de Natation, puis par des clubs au niveau des différentes villes du Royaume.

La fédération organise des entraînements de natation communs aux différents athlètes dans le but d’harmoniser les pratiques et de déceler les futurs talents. Ces entraînements communs nécessitent de disposer de créneaux horaires dans trois piscines différentes.

La fédération  souhaite mettre en place une gestion informatisée afin de contrôler que chaque athlète suit bien son plan d'entraînement personnalisé. Pour chaque athlète, le plan d'entraînement proposé définit la distance (exprimée en mètres) à parcourir pour chaque entraînement.

Pour assurer cette gestion, le schéma relationnel suivant a été établi :

**ATHLETE(NumLicence, NomAthlete, PrenomAthlete, CategorieAthlete)**

**ENTRAINEMENT(NumEntrainement, DateEntrainement, HeureDebut, HeureFin, NumPiscine#)**

**PLAN\_ENTRAINEMENT(NumEntrainement#, NumLicence#, DistanceAParcourir, DistanceParcourue)**

**PISCINE(NumPiscine, NomPiscine, AdressePiscine)**

*Les champs soulignés correspondent aux clefs primaires, les champs suivis du caractère # sont des clefs étrangères.*

**TRAVAIL À FAIRE**

**I.**                   **Création de la  base de données**

1.       Créer la base de données sous SQL SERVER        **2 pts**

2.       Créer trois enregistrements par table

**II.**                **Contraintes**

1.      Les valeurs permises pour le champs **CategorieAthlete** sont (Catégorie1,Catégorie2,catégorie3) **0,5pts**

2.      La distance parcourue doit être positives et inférieure ou égale à la distance à parcourir **1 pts**

**III.**             **Requêtes**

1.      Afficher  la liste des  athlètes triés par ordre décroissant des catégories et ordre croissant de leur numéro de licence. **1pts**

2.      Afficher la liste de piscines triées par ordre croissant des noms, les noms doivent avoir le premier caractère en majuscule et les adresses des piscines en minuscule **1,5 pts**.

3.      Afficher les athlètes qui ont participés au plan d’entrainement numéro 20,  (nom,prénom,distanceparcourue,observation), le champs observation permettant d’afficher  le mot débutant si la distance parcouru est inférieure à 2000 m sinon on affiche le mot expert. **2 pts**

4.      Afficher les piscines (numéro, nom,adresse) qui seront disponible pour le mois janvier de l’année 2012. **1,5 pts**

5.      Créer une **vue vue1** affichant le nombre d’athlète par catégorie. **1,5 pts**

6.      Créer une **vue vue2** affichant le total des distances parcourue au niveau des différents entrainements pour chaque athlètes (numéro athlète, total distance parcourue) **1,5 pts**

7.      En utilisant la question N°6 afficher les athlètes dont la distance parcourue dans les différents entrainements est supérieur à 2000 m **1,5 pts**

8.      Afficher les entrainements dont leur distance à parcourir est la valeur maximale. **1,5 pts**

9.      Créer une **vue vue3** permettant d’afficher la listes des entrainements suivis(numéro,date,heure début,heure fin,Nom piscine,Distance à parcourir, distance parcourue)pour chaque athlète. **1,5 pts**

10.  Afficher  les athlètes qui ont participé à au moins 4 entrainements. **1,5 pts**

11.  Créer une **vue  vue4** affichant le nom de piscine le plus utilisé par les athlètes de la catégorie 1 (**1,5 pts)**

**solution : Copier le script dans votre sql server pour qu'il s'affiche correctement bonne chance**  
  
 **create database EFM\_SGBDI**  
 **use EFM\_SGBDI**  
 **--I/1**  
 **create table ATHLETE(**  
 **NumLicence int primary key,**  
 **NomAthlete varchar(255),**  
 **PrenomAthlete  varchar(255),**  
 **CategorieAthlete  varchar(255)**  
 **)**  
 **create table ENTRAINEMENT(**  
 **NumEntrainement  int primary key,**  
 **DateEntrainement date,**  
 **HeureDebut int,**  
 **HeureFin int,**  
 **NumPiscine int foreign key references PISCINE(NumPiscine)**  
 **)**  
 **create table PLAN\_ENTRAINEMENT(**  
 **NumEntrainement int foreign key references ENTRAINEMENT(NumEntrainement),**  
 **NumLicence int foreign key references ATHLETE(NumLicence),**  
 **DistanceAParcourir float,**  
 **DistanceParcourue float,**  
 **constraint PK\_PLAN\_ENTRAINEMENT primary key(NumEntrainement,NumLicence)**  
 **)**  
 **create table PISCINE(**  
 **NumPiscine  int primary key,**  
 **NomPiscine varchar(255),**  
 **AdressePiscine varchar(255)**  
 **)**  
 **set language french**  
 **--I/2**  
 **insert into ATHLETE values (1,'Hillal','Abdessamad','Catégorie2')**  
 **insert into ATHLETE values (2,'Sahri','Anas','Catégorie1')**  
 **insert into ATHLETE values (3,'badir','Zaid','Catégorie3')**  
 **insert into ATHLETE values (4,'Dahmane','Brahim','Catégorie1')**  
 **insert into ATHLETE values (5,'Azrig','Abdelhaq','Catégorie2')**  
 **insert into PISCINE values (1,'Albaladi','Syba')**  
 **insert into PISCINE values (2,'Koutobiya','Jam3 Alfana')**  
 **insert into PISCINE values (3,'Alfarah','Dawdiyat')**  
 **insert into PISCINE values (4,'Alnour','Massira')**  
 **insert into ENTRAINEMENT values (1,'19/01/2010',1,6,1)**  
 **insert into ENTRAINEMENT values (2,'25/01/2010',6,8,2)**  
 **insert into ENTRAINEMENT values (3,'26/02/2010',7,10,1)**  
 **insert into ENTRAINEMENT values (4,'28/02/2010',12,16,3)**  
 **insert into ENTRAINEMENT values (6,'05/03/2010',2,5,2)**  
 **insert into ENTRAINEMENT values (20,'10/03/2010',8,12,2)**  
 **insert into ENTRAINEMENT values (7,'1/1/2011',14,16,3)**  
 **insert into ENTRAINEMENT values (8,'5/1/2012',15,17,1)**  
 **insert into ENTRAINEMENT values (8,'5/1/2012',15,17,1)**  
 **insert into PLAN\_ENTRAINEMENT values (1,1,500,250)**  
 **insert into PLAN\_ENTRAINEMENT values (2,1,300,200)**  
 **insert into PLAN\_ENTRAINEMENT values (3,2,600,150) `**  
 **insert into PLAN\_ENTRAINEMENT values (1,3,100,50)**  
 **insert into PLAN\_ENTRAINEMENT values (4,1,400,250)**  
 **insert into PLAN\_ENTRAINEMENT values (2,2,600,400)**  
 **insert into PLAN\_ENTRAINEMENT values (20,4,250,150)**  
 **insert into PLAN\_ENTRAINEMENT values (20,1,2400,2200)**  
 **insert into PLAN\_ENTRAINEMENT values (7,4,3000,2200)**  
 **insert into PLAN\_ENTRAINEMENT values (1,2,700,150)**  
 **insert into PLAN\_ENTRAINEMENT values (7,2,900,500)**  
 **insert into PLAN\_ENTRAINEMENT values (4,2,1500,1200)**  
 **--II**  
 **--1. Les valeurs permises pour le champs CategorieAthlete sont (Catégorie1,Catégorie2,catégorie3) 0,5pts**  
 **alter table  ATHLETE**  
 **add constraint CheckCat check(CategorieAthlete in ('Catégorie1','Catégorie2','Catégorie3'))**  
 **--2. La distance parcourue doit être positives et inférieure ou égale à la distance à parcourir 1 pts**  
 **alter table PLAN\_ENTRAINEMENT**  
 **add constraint ChekDs check(DistanceParcourue > 0 and DistanceParcourue<= DistanceAParcourir)**  
 **--II. Requêtes**  
 **--1. Afficher  la liste des  athlètes triés par ordre décroissant des catégories et ordre croissant de leur numéro de licence. 1pts**  
 **select \* from ATHLETE  order by CategorieAthlete desc,NumLicence asc**  
 **--2. Afficher la liste de piscines triées par ordre croissant des noms, les noms doivent avoir le premier caractère en majuscule et les adresses des piscines en minuscule 1,5 pts.**  
 **select NumPiscine,UPPER(LEFT(NomPiscine,1))+lower(RIGHT(NomPiscine,LEN(NomPiscine)-1)),lower(AdressePiscine) from PISCINE order by NomPiscine asc**  
 **--3**  
 **select a.NomAthlete,a.PrenomAthlete,p.DistanceParcourue,case**  
 **when  p.DistanceParcourue  < 2000 then 'débutant'**  
 **else 'expert'**  
 **end as 'observation'**  
 **from ATHLETE a, PLAN\_ENTRAINEMENT p**  
 **where a.NumLicence = p.NumLicence and p.NumEntrainement = 20**  
 **--4. Afficher les piscines (numéro, nom,adresse) qui seront disponible pour le mois janvier de l’année 2012. 1,5 pts**  
 **select \* from PISCINE where NumPiscine in (select NumPiscine from ENTRAINEMENT where MONTH(DateEntrainement) = 1 and YEAR(DateEntrainement) = '2012')**  
 **--5. Créer une vue vue1 affichant le nombre d’athlète par catégorie. 1,5 pts**  
 **create view  vue1 as (select CategorieAthlete,COUNT(\*) as [Nombre ATHLETE] from ATHLETE group by CategorieAthlete)**  
 **select \* from vue1**  
 **--6. Créer une vue vue2 affichant le total des distances parcourue au niveau des différents entrainements pour chaque athlètes (numéro athlète, total distance parcourue) 1,5 pts**  
 **create view vue2 as (select a.NumLicence,SUM(p.DistanceParcourue) as [total\_distances] from PLAN\_ENTRAINEMENT p, ATHLETE a where p.NumLicence = a.NumLicence group by a.NumLicence)**  
 **select \* from vue2**  
 **--7. En utilisant la question N°6 afficher les athlètes dont la distance parcourue dans les différents entrainements est supérieur à 2000 m 1,5 pts**  
 **select \* from ATHLETE a, vue2 v where a.NumLicence = v.NumLicence and v.total\_distances > 2000**  
 **-- 8. Afficher les entrainements dont leur distance à parcourir est la valeur maximale. 1,5 pts**  
 **select e.NumEntrainement,e.DateEntrainement,e.HeureDebut,e.HeureFin,p.NomPiscine**  
 **from ENTRAINEMENT e, PISCINE p**  
 **where e.NumPiscine = p.NumPiscine and NumEntrainement in ( select NumEntrainement  from PLAN\_ENTRAINEMENT  where DistanceAParcourir >= all (select DistanceAParcourir from PLAN\_ENTRAINEMENT))**  
 **--9. Créer une vue vue3 permettant d’afficher la listes des entrainements suivis(numéro,date,heure début,heure fin,Nom piscine,Distance à parcourir, distance parcourue)pour chaque athlète. 1,5 pts**  
 **select a.NomAthlete ,e.NumEntrainement,e.DateEntrainement,e.HeureDebut,e.HeureFin,ps.NomPiscine,SUM(p.DistanceAParcourir) as [DistanceAParcourir],SUM(p.DistanceParcourue) as [DistanceParcourue]**  
 **from ATHLETE a, PLAN\_ENTRAINEMENT p, ENTRAINEMENT e, PISCINE ps**  
 **where a.NumLicence = p.NumLicence and e.NumEntrainement = p.NumEntrainement and ps.NumPiscine = e.NumPiscine**  
 **group by  a.NomAthlete ,e.NumEntrainement,e.DateEntrainement,e.HeureDebut,e.HeureFin,ps.NomPiscine**  
 **--10**  
 **select \* from ATHLETE where NumLicence in ( select NumLicence from  PLAN\_ENTRAINEMENT group by NumLicence having COUNT(\*) >= 4)**  
 **--11. Créer une vue  vue4 affichant le nom de piscine le plus utilisé par les athlètes de la catégorie 1 (1,5 pts)**  
 **select ps.NomPiscine,COUNT(\*)**  
 **from ATHLETE a, PLAN\_ENTRAINEMENT p,ENTRAINEMENT e, PISCINE ps**  
 **where a.NumLicence = p.NumLicence and p.NumEntrainement = e.NumEntrainement and e.NumPiscine = ps.NumPiscine and a.CategorieAthlete = 'Catégorie1'**  
 **group by ps.NomPiscine having COUNT(\*) >= all**  
 **(select COUNT(\*) from ATHLETE a, PLAN\_ENTRAINEMENT p,ENTRAINEMENT e, PISCINE ps**  
 **where a.NumLicence = p.NumLicence and p.NumEntrainement = e.NumEntrainement and e.NumPiscine = ps.NumPiscine and a.CategorieAthlete = 'Catégorie1'**  
 **group by ps.NomPiscine )**