ISTA Tinghir

Groupe TDI1A

M18 : SGBD II

Travaux pratiques

Exercice N°1

Pour gérer les membres d’un forum de programmation, on dispose de la table suivante :

Membre (**Pseudo**, E-mail, Mot de passe, Date de naissance, Date d’inscription, Ville)

**Travail à faire :**

Créer la table Membre.

Enregistrer quelques membres dans la table.

1. Déclarer une variable @ville et l’initialiser avec le nom de votre ville.
2. En utilisant la variable @ville dans une requête SELECT, renvoyer la liste des membres originaire de votre ville.
3. Déclarer une variable @nbMembre de type int
4. Affecter à cette variable le nombre total des membres du forum
5. Afficher, si elles existent, les villes contenant plus que 20% du nombre total des membres
6. Déclarer une variable @DateDémarrage de type date
7. Affecter à cette variable la date d’inscription du premier membre (la date d’inscription la plus ancienne)
8. Affecter à une autre variable la date du dixième jour après l’inscription du premier membre, (utiliser DATEADD (day , 10, @DateDémarrage))
9. En utilisant les deux variables, calculer le nombre des membres inscrits, pendant les 10 premiers jours.
10. De la même manière, calculer le nombre des inscrits les 10 jours suivants
11. Quel était le taux d’évolution du nombre d’inscription ?  
    taux d’évolution = (Nombre d’inscrits 2 - Nombre d’inscrits 1) / Nombre d’inscrits 1
12. Calculer dans une nouvelle variable l’âge moyen des membres
13. Afficher les membres dont l’âge est supérieur à l’âge moyen calculé précédemment.

Exercice N°2

**Facture d’eau**

La facture d’eau est calculée selon un barème à tranche.

La consommation mensuelle d’eau est répartie en tranches, les informations concernant les tranches sont données dans le tableau suivant :

|  |  |
| --- | --- |
| Quantité m3 | Prix unitaire Dh/ m3 |
| < 9 | 1,29 |
| Entre 9 et 20  < 20 | 4,29 |
| Entre 20 et 36  < 36 | 6,59 |
| >= 36 | 10,37 |

Ecrire un lot transact-sql qui calcule le montant à payer pour une consommation donnée.

Déclarer une variable @consommation et lui affecter la valeur 26.

En utilisant la structure if … else calculer et afficher le montant de la consommation.

Exécuter le lot

Calculer le montant de la facture pour les consommations suivantes : 6 m3, 15 m3 et 40 m3

Exercice N°3

Pour encourager l’entraide des membres du forum, un système de bonus à point a été instauré.

Un extrait de la base de données est le suivant :

Membre (Pseudo, E-mail, Mot de passe, Date de naissance, Date d’inscription, Ville)

Bonus (PseudoMembre, NbPoint)

**Travail à faire :**

* Créer la table Membre
* Ajouter quelques membres à la table
* Écrire un lot Transact-sql qui permet d’ajouter à la base de données la table Bonus si elle n’existe pas et/ou créer la contrainte FOREIGN KEY si elle n’existe pas.

**Pour tester l’existence d’un objet dans la base de données, vous pouvez utiliser**

**IF NOT EXISTS (SELECT \* FROM dbo.sysobjects WHERE id = OBJECT\_ID('Nom objet'))**

Dans un nouveau lot Transact-sql, on souhaite ajouter un nombre de points au score d’ un membre :

* Déclarer une variable @pseudo et lui affecter la valeur ‘TDI2’
* Déclarer une variable @point et lui affecter la valeur 2
* Si le membre a déjà eu un bonus, augmenter donc ses points par la valeur de la variable @point
* Sinon ajouter une nouvelle entrée dans la table Bonus (@pseudo, @point).

Exercice N°4

Pour la gestion des rendez-vous d’un cabinet médical, un extrait de la base de données vous est fourni :

RDV (**NUM\_RDV**, DATE\_RDV, HEURE\_RDV, PATIENT)

Heure\_RDV (**Heure**)

Le champ Heure est de type time

Déclarer les variables suivantes

|  |  |  |
| --- | --- | --- |
| Nom variable | Type | Valeur initiale |
| @HeureDebut | Time | 8 :00 |
| @HeureFin | Time | 14:00 |
| @durée | Int | 20 |

Supprimer le contenu de la table Heure\_RDV

En utilisant les trois variables, et la structure While, écrire une boucle permettant de remplir la table Heure\_RDV comme suit

|  |
| --- |
| Heure |
| 8 :00 |
| 8 :20 |
| 8 :40 |
|  |
| 13 :40 |

Pour ajouter @n minute à @heure, on écrit :

SET @heure = dateadd(minute, @n, @heure)

Pour la prise d'un RDV, on souhaite trouver la date et l’heure les plus proches :

Déclarer une variable @jour de type date

Affecter à cette variable la date système.

Dans le bloc d’une boucle While, en incrémentant la variable @jour, chercher si elle existe une heure libre (non affectée à un rendez-vous)

Le code ne doit pas proposer des rendez-vous le week-end.

Les heures libres peuvent être obtenues par la requête

SELECT Heure FROM Heure\_RDV

EXCEPT

SELECT HEURE\_RDV FROM RDV WHERE DATE\_rdv = @jour

DATEPART (weekday, @jour) retourne le numéro du jour dans la semaine