

Cycle Master en Sciences et Techniques Filière : Intelligence Artificielle et Sciences de Données

Pr. Mostafa Ezziyyani

Devoir à rendre (évaluation sera comptabilisée)
Bases de Données (T-SQL)
SQL Dynamique & Procédures/Fonctions Stockées & Curseurs & indexation
Par Binôme

Soit la base de données suivante :

Passagers (Code_Passager, Nom_Passager, Pre_Passager, Num_Passport, Categorie, Num_Tel)

Reservations (Num_Reservation, Date_Reservation, Date_Validation, Etat_Reservation, Code_Agence, Code_Passager, Prix_Total)

Ligne_Reservation(Num_Ligne, Num_Order, Num_Vol, Num_Reservation)

Billets (Num_Billet, Num_Reservation)

Vols (Num_Vol, Date_Depart, Heure_Depart, Ville_Depart, Ville_Arrivee, Code_Avion, Code_Pilote, Prix_Vol)

Voyages(Code_Passager, Num_Billet, Num_Vol, Num_place)

Avions (Num_Avion, Poids_Max, Nom_Compagnie, Nbr_Place)

Pilotes (Num_Pilote, Nom_Pilote, Prenom_Pilote)

On suppose les règles de gestion suivantes :

1. Les passagers peuvent effectuer plusieurs réservations.
2. Chaque réservation concerne un seul passager et un seul billet.
3. Chaque billet comprend plusieurs vols ordonnés.
4. Chaque vol est assuré par un seul pilote et par un seul avion.

Remarque : Pour tous les exercices, générer les exceptions si nécessaires (implicites et explicites).

Questions

1. Ecrire une procédure stockée qui permet de trouver tous les entiers inférieurs à **un nombre donné** dont la somme de ces chiffres égale à **6**. En stockant les informations suivantes dans une table temporaire:
 - a. le nombre de chiffres paires
 - b. le nombre de chiffres impaires

NB :

- Utiliser une table TMP pour stocker les résultats puis la supprimer, en utilisant SQL Dynamique (Voir le tableau).

(select * from TMP)

NBR	Nbr paire	Nbr Impaire
6	1	0
15	0	2
24	2	0
33	0	2
42	2	0
51	0	2
60	2	0

2. Ecrire une fonction stockée qui permet de calculer le code binaire d'un entier (la valeur de retour de la fonction est une chaîne de caractères).
3. Ecrire une fonction stockée qui permet de tester si une chaîne caractères est un palindrome (Exemple : TOTOT et TOUSUOT sont deux palindromes).
4. Ecrire une fonction stockée qui permet de compter le nombre des mots dans une chaîne de caractère passer en paramètre.
5. Ecrire une fonction stockée qui permet de compter le nombre d'occurrences d'une chaîne dans une autre chaîne caractères.
6. Ecrire une fonction stockée qui permet de trouver le plus long mots dans une chaîne de caractères.

Soient les fonctions natives pour le traitement des chaînes de caractères suivantes :

- LENGTH(chaîne) : longueur d'une chaîne en caractères
- SUBSTR(chaîne, début [, 'nombre']) : Extraction d'une sous-chaîne à partir d'une chaîne.
- INSTR(chaîne, sous-chaîne [, début [,nombre occurrences]]) : Recherche de la position d'une sous-chaîne dans une chaîne.
- REPLACE(chaîne, chaîne source, chaîne cible) : Remplace des caractères d'une chaîne par une sous-chaîne.
- || : la concaténation

7. Ecrire une procédure stockée qui permet d'afficher un nombre des minutes X sous la forme : AA Années MM Mois JJJ Jours HH Heures MM Minutes, sans utiliser les fonctions natives prédéfinies .
8. Ecrire une procédure stockée qui permet la création de la table Vols (Avec la prise en considération toutes les contraintes : d'intégrités structurelles, référentielles, unicité de la clés ...)

9. Ecrire une procédure stockée qui permet d'afficher toutes les réservations non validées à une date donnée.
10. Ecrire une procédure stockée qui permet d'afficher toutes les informations d'un vol donnée.
11. Ecrire une procédure stockée qui permet d'afficher toutes les informations d'un vol donnée. (Pilotes, Ville/heure départ, ville/Heure arrivée et villes/durées des escales, Etc ...)
12. Ecrire une procédure stockée qui permet d'afficher toutes les informations d'une réservation validée (Billet).
13. Ecrire une procédure stockée qui permet d'afficher le nombre des voyage de chaque avion dans l'ordre décroissant.
14. Ecrire une fonction stockée qui permet de calculer le nombre des voyages d'un passager donnée.
15. Ecrire une Fonction stockée qui permet de calculer le prix de revient d'un vol donné.
16. Ecrire une procédure stockée qui permet de supprimer toutes les réservations non validées.
17. Ecrire une procédure stockée qui permet d'insérer un enregistrement dans la table **Voyages** sous les contraintes suivantes :
 - Teste de l'unicité des enregistrements dans la table **Voyages**
 - **Contrôle** si le numéro du billet est correspond bien le passager et le vol.
 - Teste de l'unicité du numéro de la place accordée au passager.
18. Ecrire une procédure stockée qui permet d'insérer un enregistrement dans la table **Ligne_Reservation** sous les contraintes suivantes :
 - Teste de l'unicité de la clés de la table **Ligne_Reservation**
 - **contrôle** si le numéro d'ordre est sérial pour la nouvelle réservation.
 - **contrôle** si la ville de départ du vol de la nouvelle réservation coïncide avec la ville d'arrivé du vol de réservation précédente (à l'exception du premier vol).
 - **contrôle** s'il y a encore une place dans l'avion.
19. Ecrire une procédure stockée qui permet d'ajouter deux colonnes **Nbr_Res**, et **Nbr_Att** dans la table **Vols** pour stocker respectivement le nombre de places réservées et le nombre de places attribuées pour chaque vol, et initialiser les deux colonnes par 0 (Utiliser SQL dynamique avec la commande EXECUTE IMMEDIATE).
20. Ecrire une procédure stockée qui permet de mettre à jours les deux colonnes **Nbr_Res**, et **Nbr_Att** respectivement par le nombre de places réservées et le nombre de places attribuées pour un vol donné.
21. On veut classer les passagers de l'année en cours selon le taux de voyage via les avions, pour cet objectif on utilise la colonne de la table Passagers « **Categorie** » pour stocker l'information.
Ecrire une procédure stockée qui permet de calculer le champ « **Categorie** » d'un passager donné selon les cas suivants :
 - '**Très Actif**' : Pour les passagers qui ont effectués plus de 20 voyages avec un montant de paiement dépasse 200 000 durant l'année encours.
 - '**Actif**' : Pour les passagers qui ont effectués plus de 20 voyages durant l'année encours.
 - '**Moyen**' : Pour les autres passagers
22. Ecrire une procédure stockée qui permet de calculer le nombre des voyages de chaque passager.

23. Ecrire une procédure stockée qui permet de calculer le coût de revient de tous les vols.
24. Même question (14), pour tous les passagers.
25. Ecrire une procédure stockée qui permet d'afficher les pilotes qui ont piloté plus d'un pourcentage donné des avions de la compagnie (par exemple, plus de 20% des avions).
26. Ecrire une procédure stockée qui permet d'ajouter les colonnes **NbrAvions**, **NbrVoyages** et **Statut** dans la table Pilotes et l'initialiser selon les cas suivants :

NbrVoyages : est le nombre des voyages réalisés par un pilote donné.

NbrAvions : Nombres des avions pilotés par un pilote donné.

Statut vaut :

 - **Expert** : Si le pilote a piloté plus de 50% des avions de la compagnie.
 - **Qualifié** : Si le pilote a piloté entre 50% et 5% des avions de la compagnie.
 - **Débiteur** : Si le pilote a piloté moins de 5% des avions de la compagnie.
27. Ecrire une procédure stockée qui permet de proposer tous les billets possible (Classés par ordre décroissant des prix) pour une ville de départ et une ville d'arrivée données pour un nombre d'escale donné.
28. Ecrire un trigger qui permet de contrôler la disponibilité d'une place dans un avion pour un voyage et un passager donnés (utiliser les deux fonctions, **Complet** qui test si le voyage est complet et **Occuper** qui test si le numéro de la place est occupée). Si le voyage n'est pas complet et la place est occupée le trigger propose le numéro d'une place disponible automatiquement.
29. Ecrire un trigger qui permet d'assurer l'insertion des noms et les prénoms des passagers en majuscule et de contrôler l'unicité de la clé.
30. Ecrire un trigger qui permet de contrôler l'insertion d'un voyage pour un passager et son billet réservé avec le contrôle de pré-enregistrement du vol correspondant.
31. Ecrire un trigger qui permet la mise à jour le **Statut** et le **NbrAvions** et **NbrVoyages** correspondant à un pilote automatiquement dès l'insertion d'une ligne dans la table **Voyages**.
32. Ecrire un trigger qui provoque une erreur à l'insertion d'un tuple dans la table Voyage si le nombre de places accordées dépasse la capacité de l'avion.
33. Ecrire un trigger qui mémorise dans une table quel utilisateur a fait, à quelle heure, une modification (insert, delete ou update) et quelles nouvelles valeurs d'attributs il a insérées dans la table **Reservations**.
34. Créer un déclencheur qui permet la suppression en cascade d'un passager donnée. (Avec la mise à jour des tables en correspondances).
35. Ecrire un trigger qui permet la suppression en cascade d'un passager donnée. (Avec la mise à jour des tables en correspondances).
36. Ecrire un trigger qui consiste à corriger à la volée des saisies ou des modifications incorrectes. Tous les caractères de séparation de numéro de tel du passager que soit le tiret ou l'espace d'un numéro de téléphone devra être convertis en point et le caractère 'O' par celui de '0' (Zéro).
37. Ecrire un trigger qui permet de contrôler la validité de saisie de format de la date de départ et date d'arrivée. La date est valide sauf si elle ne possède que les chiffres, le caractère '/' et au maximum 10 caractères. Si la date contient l'un des caractères 'O' ou 'Q' le trigger les remplace par le chiffre 0.

38. Ecrire un trigger qui permet d'archiver toutes les opérations de suppression sur la table **Voyages** (Créer la table nécessaire pour l'archive).
39. Ecrire un trigger qui permet d'archiver la suppression des réservations selon la nature de traitement de la réservation soit annulée ou validée. Une réservation validée ne peut être supprimée qu'après 10 jours de la date de voyage et une réservation annuler ne peut être supprimer qu'après la date de validation (Créer la table nécessaire pour l'archive)
40. Créer la vue **ReservationValidees** à partir de la table réservations pour avoir dans cette vue uniquement les réservations validées de l'agence 001. Les attributs de la vue seront : idzone, type, caract, dist. Cette vue est-elle modifiable ? Si vous pensez qu'elle n'est pas modifiable, comment la rendre modifiable ?
41. L'ajout dans un arbre B+ : Schématisé et décrire l'insertion étape par étape dans un arbre B+ de 3 clés la liste des valeurs suivantes :
3-6-44-5-2-77-1-7-9-91-33-43-0-27-88-55-54-56-57-52-44-24-25-26-27-98-99-4-6-8
42. Suppression de l'arbre B+ : Schématisé et décrire la suppression étape par étape de l'arbre B+ (Question 44) des valeurs suivantes :
3-24-25-98-6-44-0-27-88-55-7-56-57-77-9-91-33-43-5-52-44-26-27-99-54-4-6-8-2-1.