Filière : Développement Digital
Option : Web Full Stuck

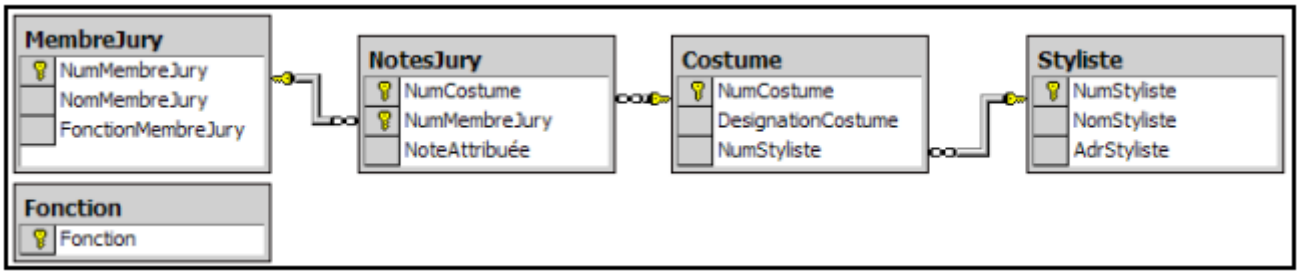
ISTA TINGHIR

Module N° 3 : Gérer les données
Groupe : DDoWFS201

TP N°3

Menu thématique : Instructions de bases- Structures de contrôle- procédures stockées et fonctions- transaction.

EXERCICE N°1 : "***Inter Défilés***" est une société d'organisation de défilés de modes. Une de ses activités les plus réputées : Grand Défilé "***Tradition Marocaine***". Dans ce défilé, des costumes défilent devant un jury professionnel composé de plusieurs membres. Chaque membre va attribuer une note à chaque costume. La BD a la structure suivante :



1. Créer la procédure stockée ***PS1*** qui affiche la liste des costumes avec pour chaque costume le numéro, la désignation, le nom et l'adresse du styliste qui l'a réalisé.

2. Créer une fonction ***FN1*** qui reçoit un numéro de costume et qui affiche la désignation, le nom et l'adresse du styliste concerné.

3. Refaire la question ***N°2*** en créant une procédure ***PS2*** à la place de la fonction ***FN1.***

4. Créer la procédure stockée ***PS3*** qui reçoit un numéro de costume et qui affiche la liste des notes attribuées avec pour chaque note le numéro du membre de jury qui l'a attribué, son nom, sa fonction et la note.

5. Créer la fonction ***FN2*** qui retourne le nombre total de costumes.

6. Refaire la question ***N°5*** en créant une procédure ***PS4*** à la place de la fonction ***FN2***.

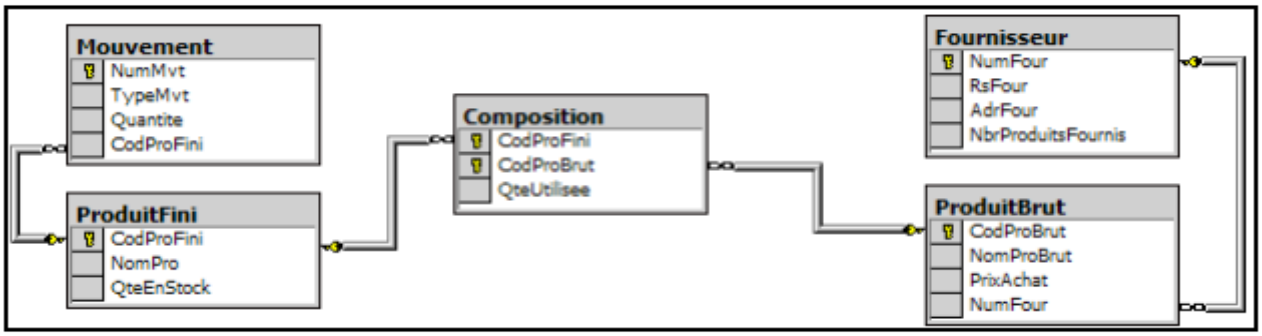
7. Créer la fonction ***FN3*** qui reçoit un numéro de costume et un numéro de membre de jury et qui retourne la note que ce membre a attribué à ce costume.

8. Refaire la question ***N°7*** en créant une procédure ***PS5*** à la place de la fonction ***FN3***.

9. Créer la fonction ***FN4*** qui reçoit un numéro de costume et qui retourne la moyenne des ses notes.

10. Refaire la question ***N°9*** en créant une procédure ***PS6*** à la place de la fonction ***FN4***.

Exercice 2 : Une société achète à ses fournisseurs des produits bruts qu'elle utilise dans la fabrication de produits finis. On souhaite gérer la composition et les mouvements de stock de chaque produit fini. Les Mouvements de stock sont les opérations d'entrée ou de sortie (type=S ou type=E) de produits finis vers ou depuis le magasin. La BD a la structure suivante :



On suppose que les tables **'Mouvement'**, **'ProduitFini'** et **'Fournisseur'** sont créées.

1. Créer la procédure stockée ***PS1*** qui crée les tables **ProduitBrut** et **Composition**.

2. Créer la procédure stockée ***PS2*** qui affiche le nombre de produits bruts par produit Fini.

3. Créer la procédure stockée ***PS3*** qui retourne en sortie le prix d'achat le plus élevé.

4. Refaire la question ***N°3*** et créer la fonction ***FN1*** à la place de la procédure stockée ***PS3***.

5. Créer la procédure stockée ***PS4*** qui affiche la liste des produits finis utilisant plus de deux produits bruts.

6. Créer la procédure stockée ***PS5*** qui reçoit le nom d'un produit brut et retourne en sortie la raison sociale de son fournisseur.

7. Refaire la question ***N°6*** et créer la fonction ***FN2*** à la place de la procédure stockée ***PS5***.

8. Créer la procédure stockée ***PS6*** qui reçoit le code d'un produit fini et qui affiche la liste des mouvements de sortie pour ce produit.

9. Créer la procédure stockée ***PS7*** qui reçoit le code d'un produit fini et le type de mouvement et qui affiche la liste des mouvements de ce type pour ce produit fini.

10. Créer la procédure stockée ***PS8*** qui pour chaque produit fini affiche :

- La quantité en stock pour ce produit ;
- La liste des mouvements concernant ce produit ;
- La quantité totale en sortie et la quantité totale en entrée ;
- La différence sera comparée à la quantité en stock. Si elle correspond afficher **'Stock Ok'** sinon afficher **'Problème de Stock'.**

11. Créer la procédure stockée ***PS9*** qui reçoit un code produit fini et retourne en sortie son prix de reviens

12. Refaire la question ***N°11*** et créer la fonction ***FN3*** à la place de la procédure stockée **PS9**.

13. Créer la procédure stockée ***PS10*** qui affiche pour chaque produit fini :

- Le prix de reviens (utiliser ***PS9)***;
- La liste des produits bruts le composant (**nom**, **Mt**, **RSFour**) ;
- Le nombre de ces produits.

Exercice 3 : Par défaut, le client MySQL est configuré pour utiliser la validation automatique. Pour désactiver le commit automatique, on utilise : **SET autocommit = 0;** Ou bien **SET autocommit = OFF ;**

Si jamais, on souhaite avoir la possibilité d'annuler une instruction **INSERT** par exemple, on doit utiliser une **transaction**. Celle-ci est réalisée moyennant une instruction **BEGIN** ou bien une instruction **START TRANSACTION** suivi du code MySQL à exécuter et terminé par **COMMIT** pour valider et exécuter les instructions SQL ou bien **ROLLBACK** afin d'annuler l'exécution. ***Application*** : Soit la table compte suivante : **compte** (numCompte, nom, prenom, solde, adresse).

1. Créer la table Compte.

2. Créer une procédure stockée **inserer_client** qui permet d'insérer un client saisi en argument.

3. En appelant **inserer_client**, saisir l'enregistrement (**12345, Safraoui, Ahmed, 10000, Tinghir**).

4. Désactiver la validation automatique et réappeler la procédure stockée **inserer_client** pour insérer l'enregistrement (**67890, Adnani, Ibrahim, 20000, Errachidia**). Conclure.

5. Créer une procédure stockée nommée **debiter_solde** qui permet de débiter un compte dont le numéro est saisi en argument.

6. Créer une fonction nommée **creditFunct** qui permet de créditer un compte dont le numéro est saisi en argument.

7. Créer une procédure stockée nommée **crediter_solde** qui refait le même traitement que **creditFunct.**

8. Le meilleur exemple pour comprendre la notion de **TRANSACTION MySQL** est le transfert d'argent entre 2 comptes d'une même banque. Supposons que le débit ait lieu avec succès, mais que le crédit n'ait pas eu lieu. Dans ce cas, la BD serait dans un état incohérent. Idéalement, cette transaction (à la fois de crédit et de débit) se veut se produire, sinon aucune d'entre elles n'aura lieu. C'est-à-dire que dans ce cas, s'il s'agissait d'une transaction, un échec de crédit aurait entraîné une annulation de l'opération de débit et il n'y aurait eu aucun changement dans l'état de la BD. On admet que la table compte contient les deux enregistrements précédents.

   a. Programmer une **TRANSACTION MySQL** qui met en œuvre, avec succès, une transaction débit-crédit d'un montant de **300** DHS du compte de M **Safraoui** à celui de M **Adnani**.

   b. Vérifier la cohérence de la table compte.

   c. Est-il possible de programmer une **TRANSACTION MySQL** qui réalise, avec succès, une transaction débit-crédit d'un montant de **250** DHS du compte de M **Adnani** à celui de M **Safraoui** en utilisant les procédures stockées **crediter_solde** et **debiter_solde** précédentes ?
   Si oui comment ? Et vérifier encore la cohérence de la table compte.