

Les fonctions

1. Fonction qui retourne une valeur scalaire

Syntaxe :

Create Function Nom_Fonction (Nom_Param1 Type_Donnée, ...)

Returns type_de_retour

as

Instructions

...

Return Valeur

Exemple :

Créer une fonction nommée F_NbrCommandes qui retourne le nombre de commandes :

```
Create Function F_NbrCommandes() Returns int
as
    begin declare @Nbr int
    Set @Nbr = (Select count(NumCom) from Commande)
Return @Nbr end
```

2. Fonction qui retourne une table

Syntaxe :

Create Function Nom_Fonction (Nom_Param1 Type_Donnée, ...) **Returns** nom_table

Table (champ1 type1, ...)

as

Select

...

Return

Exemple :

Créer une fonction nommée F_ListeArticles qui return la liste des articles d'une commande dont le numéro est donné en paramètre :

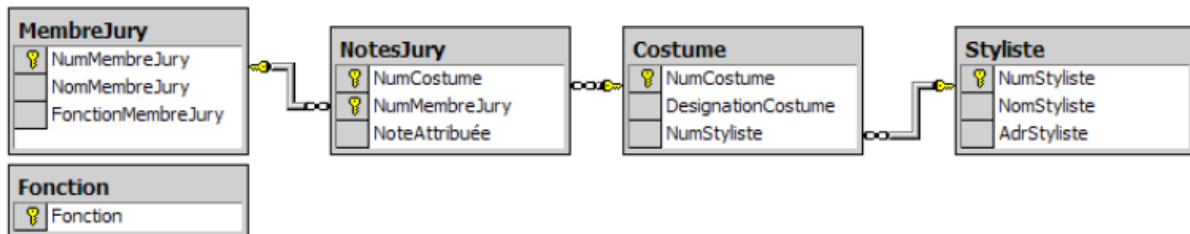
```
Create Function F_ListeArticles (@NumCom int ) Returns Liste-Art
Table ( Num int , nom varchar(29))
as
    Select A.NumArt, NomArt From Article A, LigneCommande LC
    Where LC.NumArt=A.NumArt and LC.NumCom= @NumCom
```

--Exécuter cette fonction pour afficher la liste des articles de la commande numéro

```
Select * from F_ListeArticles (1)
```

Exercice 1 :

"Inter Défilés" est une société d'organisation de défilés de modes. Une de ces 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 base de données a la structure suivante :



Créer les procédures stockées suivantes :

PS 1. 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.

PS 2. Qui reçoit un numéro de costume et qui affiche la désignation, le nom et l'adresse du styliste concerné.

PS 3. 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.

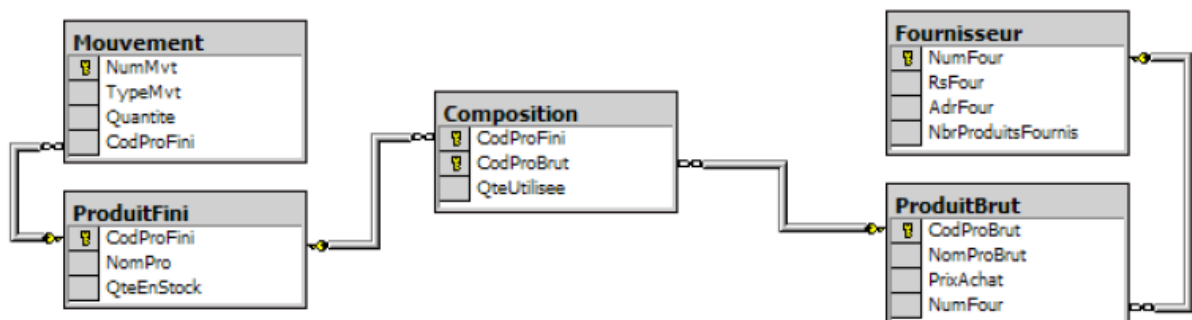
PS 4. Qui **retourne** le nombre total de costumes

PS 5. 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

PS 6. Qui reçoit un numéro de costume et qui **retourne** sa moyenne.

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 base de données a la structure suivante :



On suppose que les tables 'Mouvement', 'Produit Fini' et 'Fournisseur' sont créées. Créer les procédures suivantes :

PS 1. Qui crée les tables ProduitBrut et Composition

PS 2. Qui affiche le nombre de produits bruts par produit Fini

PS 3. Qui **retourne** en sortie le prix d'achat le plus élevé

PS 4. Qui affiche la liste des produits finis utilisant plus de deux produits bruts

PS 5. Qui reçoit le nom d'un produit brut et **retourne** en sortie la raison sociale de son fournisseur

PS 6. Qui reçoit le code d'un produit fini et qui affiche la liste des mouvements de sortie pour ce produit

PS 7. 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

PS 8. 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'

PS 9. Qui reçoit un code produit fini et **retourne** en sortie son prix de reviens

PS 10. Qui affiche pour chaque produit fini :

- Le prix de reviens (utiliser la procédure précédente)
- La liste des produits bruts le composant (nom, Mt, RSFour)
- Le nombre de ces produits