

**Exercice 1 :**

On considère la table suivante :

Produit (**NumProduit**, libelle, PU, stock)

1. Ecrire une PS qui affiche tous les produits ;
2. Ecrire une procédure stockée qui affiche les libellés des produits dont le stock est inférieur à 10 ;
3. Ecrire une PS qui admet en paramètre un numéro de produit et affiche un message contenant le libellé, le prix et la quantité en stock équivalents, si l'utilisateur passe une valeur lors de l'exécution de la procédure ;
4. Ecrire une PS qui permet de supprimer un produit en passant son numéro comme paramètre ;

**Exercice 2 :**

Ecrire une **PS** qui permet de mettre à jour le stock après une opération de vente de produits, la **PS** admet en paramètre le numéro d'article à vendre et la quantité à vendre puis retourne un message suivant les cas :

- a. Opération impossible : si la quantité est supérieure au stock de l'article ;
- b. Besoin de réapprovisionnement si stock-quantité < 10
- c. Opération effectuée avec succès, la nouvelle valeur du stock est (afficher la nouvelle valeur) ;

**Exercice 3 :**

Ecrire une PS qui retourne le prix moyen des produits (utiliser un paramètre OUTPUT) ; Exécuter la PS ;

**Exercice 4 :**

Créer une procédure stockée qui accepte comme paramètre un entier et retourne le factoriel de ce nombre.

**Exercice 5 :**

1. Créer une procédure stockée qui accepte les paramètres suivants :
  - a. 2 paramètres de type entier
  - b. 1 paramètre de type caractère.
  - c. 1 paramètre output de type entier

La procédure doit enregistrer le résultat de calcul entre les deux nombres selon l'opérateur passé dans le troisième paramètre (+,-,%,/,\*).

**Exercice 6 :**

Soit la base de données suivante :

Recettes (**NumRec**, NomRec, MethodePreparation, TempsPreparation)

Ingrédients (**NumIng**, NomIng, PUIng, UniteMesureIng, NumFou)

Composition\_Recette (**NumRec**, **NumIng**, QteUtilisee)

Fournisseur (**NumFou**, RSFou, AdrFou)

### Créer les procédures stockées suivantes :

**PS1 :** Qui affiche la liste des ingrédients avec pour chaque ingrédient le numéro, le nom et la raison sociale du fournisseur.

**PS2 :** Qui affiche pour chaque recette le nombre d'ingrédients et le montant cette recette

**PS3 :** Qui affiche la liste des recettes qui se composent de plus de 10 ingrédients avec pour chaque recette le numéro et le nom

**PS4 :** Qui reçoit un numéro de recette et qui retourne son nom

**PS5 :** Qui reçoit un numéro de recette. Si cette recette a au moins un ingrédient, la procédure retourne son meilleur ingrédient (celui qui a le montant le plus bas) sinon elle ne retourne "Aucun ingrédient associé"

**PS6 :** Qui reçoit un numéro de recette et qui affiche la liste des ingrédients correspondant à cette recette avec pour chaque ingrédient le nom, la quantité utilisée et le montant

**PS7 :** Qui reçoit un numéro de recette et qui affiche :

- ✓ Son nom (Procédure PS\_4)
- ✓ La liste des ingrédients (procédure PS\_6)
- ✓ Son meilleur ingrédient (PS\_5)

**PS8 :** Qui reçoit un numéro de fournisseur vérifie si ce fournisseur existe. Si ce n'est pas le cas afficher le message 'Aucun fournisseur ne porte ce numéro' Sinon vérifier, s'il existe des ingrédients fournis par ce fournisseur si c'est le cas afficher la liste des ingrédients associés (numéro et nom) Sinon afficher un message 'Ce fournisseur n'a aucun ingrédient associé. Il sera supprimé' et supprimer ce fournisseur

**PS9 :** Qui affiche pour chaque recette :

- ✓ Un message sous la forme : "Recette : (Nom de la recette), temps de préparation : (Temps)
- ✓ La liste des ingrédients avec pour chaque ingrédient le nom et la quantité
- ✓ Un message sous la forme : Sa méthode de préparation est : (Méthode)
- ✓ Si le prix de revient pour la recette est inférieur à 50 DH afficher le message

T-SQL

*PROCEDURES STOCKEES*

'Prix intéressant'