

## **5. Procédures et fonctions**

- 5.1 Créer une procédure qui remplace la fonction de récupération de la date en l'affichant directement à l'écran. L'appel de cette procédure permet donc d'un seul coup de récupérer la date et l'heure du système, sans passer par « getDate() » ou « CURRENT\_TIMESTAMP »
- 5.2 Créer une procédure qui affichera la phrase « Nous sommes le [Date\_du\_jour] et il est actuellement [heure\_du\_moment] ».
- 5.3 Créer une procédure qui insère dans une table temporaire les données vous concernant. Ces données sont fournies en paramètres.
- 5.4 Créer une procédure qui insère des données dans deux tables temporaires distinctes #Anciens et #Jeunes. Si l'employé est né avant 1980 et après 1970, et qu'il a été engagé dans l'entreprise avant 2008, il rentre dans la première table #Anciens. S'il est né après 1990, et qu'il a été engagé après 2010 il se retrouve dans la deuxième table #Jeunes
- 5.5 Créer une fonction qui renvoie le nombre de lignes contenues dans la table HumanResources.Employee.
- 5.6 Créer une fonction qui renvoie le nom du produit (Name de Production.Product) dont le prix (StandardCost de Production.ProductCostHistory) a été modifié le plus grand nombre de fois. S'il y a ex-aequo, il faudra renvoyer une phrase concaténée de l'ensemble des noms !
- 5.7 Créer une procédure ayant un paramètre en mode OUTPUT qui permet de modifier les lignes de la table HumanResources.Employee et de mettre la date « ModifiedDate » à la date du jour, si cette date n'est pas égale au 31 juillet 2008. La procédure renverra le nombre de lignes réellement mises à jour, via son paramètre OUTPUT. Faites l'exercice avec une simple requête d'update. Faites une seconde version qui stock au préalable les valeurs à mettre à jour dans un curseur déclaré au sein de la procédure.
- 5.8 Créer une procédure qui récupère dans un curseur l'ensemble des produits (Name de Production.Product) et leur prix (StandardCost de Production.ProductCostHistory) dont la valeur a été modifiée le plus récemment uniquement.  
Si le prix est de moins de 15, alors il faut inscrire ce produit dans une table non-temporaire que la procédure créera SI ELLE N'EXISTE PAS (référez-vous aux exemples du cours pour trouver comment faire !)  
Si le prix est plus grand que 15, alors il faudra insérer dans une table temporaire le nom du produit ainsi qu'une phrase associée qui sera soit « prix bien trop élevé » si le prix est de plus de 50, soit « prix raisonnable » si le prix est compris entre 15 et 50.  
La procédure renvoi le nombre de valeurs insérées dans la table non-temporaire dans un paramètre passé en mode OUTPUT et afficher le contenu de la table temporaire.

- 5.9 Créer une procédure qui permet d'afficher la phrase « X employés travaillent au poste de [JobTitle de HumanResources.Employees] »  
Ce nombre X sera renvoyé par une fonction que vous aurez préalablement créée et qui demande en paramètre de quel Job il s'agit, paramètre passé par la procédure appelante.  
Tester la procédure pour plusieurs JobTitle différents au sein de la procédure.
- 5.10 Créer une procédure qui affiche le nom des produits (Name de Production.Product) et le prix (ListPrice de Production.Product) des produits appartenant à la même sous-catégorie que le nom de catégorie (Name de Production.ProductSubcategory) passé en paramètres à la procédure et ayant un prix (ListPrice de Production.Product) inférieur à un prix également passé en paramètres.  
Tester la procédure dans le cas de tous les articles ayant un rapport avec « %Bikes% » et un prix inférieur à 700.  
La procédure renverra également une valeur en mode OUTPUT qui informera du nombre total de lignes non-affichées par la procédure.
- 5.11 Créer une table employes pouvant récupérer les ID, noms, prénoms, dates d'embauche et dates de naissance des employés de la base de données AdventureWorks.  
Créer une procédure capable de remplir votre table d'un seul coup en utilisant les données passées en paramètres. Les données passées seront contenues dans une variable de type table qui contiendra les ID, noms, prénoms et dates de naissance des employés. La date d'embauche sera la date du jour que vous complèterez vous-même à partir de la procédure.
- 5.12 Récupérer les données de la table Person.PersonPhone dans une variable de type table. Passer la variable fournie à une fonction qui vous permettra de rajouter les numéros de téléphones correspondants aux employés que vous avez entrés au point 5.11. Par contre, on ne retiendra dans notre table personnelle les numéros que s'ils sont de type 3 (PhoneNumberTypeID).  
La fonction renvoie le nombre de lignes mises à jour.  
Il faudra bien entendu ajouter la colonne « TelNum » à la table créée au point 5.11
- 5.13 Récolter les informations StateProvinceID et Name de Person.StateProvince que vous recouperez avec les informations de Person.Address afin de mettre en relation le nom de la province où réside chaque employé dans une variable de type table.  
Créer une procédure qui rajoutera à la table créée au point 5.12, pour chaque employé dont le numéro de téléphone est fourni, le nom de la province dans laquelle il réside.  
La colonne supplémentaire devra être rajoutée, bien entendu.
- 5.14 Supprimer tout ce que vous avez créé (procédures, types, fonctions).