**SLaboratoire 04**

Définition, manipulation et requête de données

ATTENTION ! Avant de commencer :

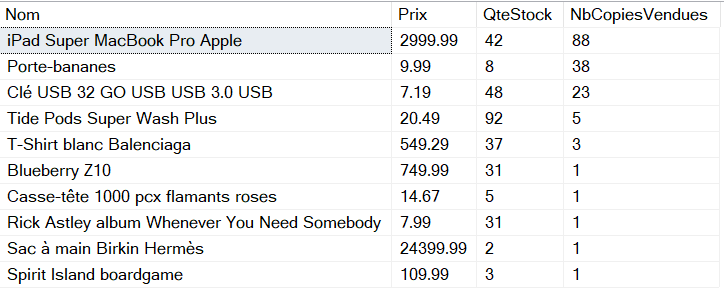
* Exécutez les scripts **Labo04\_create** et **Labo04\_insert** dans SSMS.
* Si en testant une vue ou une procédure vous générez les mauvaises données, utilisez le script **Labo04\_truncate** et refaites les insertions.

**Étape 1 - On aura tout vu 👁**

Dans cette étape, vous aurez à remplir le fichier **4204D5\_Labo04\_vues.sql** pour créer et tester deux vues. Les instructions sont disponibles directement dans le fichier, mais elles sont dupliquées ci-dessous si vous êtes plus visuel.



* Les gestionnaires du site veulent fréquemment vérifier quels articles sont très en demande pour optimiser les profils et s’assurer que les quantités en stock tiennent le coup :
  + **Vue #1** : Le top 10 des articles (nom, prix, quantité en stock) en ordre décroissant du nombre de copies de l’article vendues dans le dernier mois.
    - On ne veut pas voir ni les articles avec une quantité en stock supérieure à 100, ni les articles vendus 0 fois dans le dernier mois.



* L’équipe de distribution doit constamment vérifier les commandes à préparer.
  + **Vue #2** : Les commandes non traitées des utilisateurs en ordre croissant de date, avec toutes les données de l’adresse ainsi que l’id, le prénom, le nom, le courriel et le numéro de téléphone de l’utilisateur. On veut également le nombre de produits différents dans la commande. (Pas besoin de voir la liste d’articles de la commande, une procédure stockée s’en occupera)

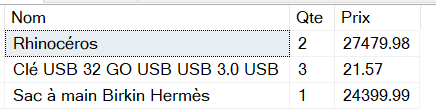
Une image contenant texte, Police, nombre, capture d’écran

Description générée automatiquement

**Étape 2 - Suivre la procédure 📃📌**

Cette fois, vous aurez à remplir le fichier **4204D5\_Labo04\_procedures.sql** pour créer et tester deux procédures stockées. Nous utilisons la même BD qu’à l’étape 1.

* L’équipe de distribution doit constamment vérifier les articles et quantités d’une commande précise en se servant de l’id de la commande.
  + **Procédure #1** : Avec **@CommandeID** en paramètre, donner la liste des articles (nom, prix et la quantité achetée) de la commande. Le prix affiché doit être le prix unitaire de l’article multiplié par la quantité achetée.



(Avec @CommandeID = 5)

* Lorsque le client choisit un article, on l’ajoute à sa commande non traitée la plus récente (si l’adresse est bonne) ou bien s’il n’y en a pas, on crée une nouvelle commande automatiquement dans laquelle on met l’article choisi.
  + **Procédure #2** : Avec le numéro d’article **@ArticleID**, la quantité **@Qte**, le numéro d’utilisateur **@UtilisateurID** et l’adresse **@AdresseID** insérez une rangée dans la table **ArticleCommande**. Pour déterminer l’id de la Commande, utilisez la règle suivante :
    - S’il existe une commande non traitée pour cet utilisateur avec l’adresse fournie, on utilise l’id de la commande la plus récente qui respecte ces conditions pour la rangée qu’on insère dans **ArticleCommande**.
    - Sinon, créez (insérez) une nouvelle rangée dans la table **Commande**. La **DateCommande** est **GETDATE()** et **EstTraite** vaut **0**. Utilisez ensuite l’id de cette toute nouvelle commande pour insérer la nouvelle rangée dans **ArticleCommande**.

Voir script sql pour l’exemple.