**Projet - Partie 2/3**

**Réalisation d’une application en PL/SQL sur une BD Oracle**

Procédure « OuvrirCompte » :

La procédure OuvrirCompte peut renvoyer 3 erreurs différentes :

-si le montant est inférieur ou égal à 0

-si le nom est null

-si le nom est vide

On vérifie ces erreurs directement dans les variables de la procédure "OuvrirCompte".

Si les paramètres sont satisfaisants on ajoute un compte dans la table Compte avec un INSERT. On utilise les variables de la procédure, la date du jour et on initialise le PMVR à 0. Le numéro de compte est renseigné à l'aide d'une séquence :

CREATE SEQUENCE seqCompte START WITH 101 INCREMENT BY 1;

Le 1er compte créé aura comme numéro de compte "101" et tous les suivants seront incrémenté de 1.

Procédure « MajValeur » :

La procédure MAJValeur peut émettre 2 exceptions lorsque :

- la nouvelle valeur du cours donnée est inférieure à 0

- le code valeur donné ne correspond à aucune valeur existante dans la base

Pour vérifier que le code valeur donné existe, il nous fallait un moyen de savoir si une requête SELECT retourne quelque chose ou rien. Nous avons opté pour une méthode utilisant la fonction COUNT.

SELECT COUNT(\*) INTO code\_count FROM Valeur WHERE CodeValeur = CodeValeurIn;

À la suite de cette requête, si la variable code\_count est égale à 0, c'est que le code valeur n'existe pas. Sinon, c'est qu'il existe. Cette procédure déclenche un et un seul trigger qui va mettre à jour la valeur de PMVL selon la formule donnée dans l'énoncé.

Procédure « Acheter » :

La procédure Acheter peut renvoyer 4 erreurs différentes :

-Le compte "acheteur" n'existe pas.

-Le code valeur à acheter n'existe pas.

-La quantité est inférieur ou égal à 0.

-Le montant n'est pas inférieur ou égal à la solde du compte et le montant est inférieur à 0.

Pour vérifier que le compte et le code valeur existent, on utilise une requête SELECT pour tester si la requête renvoie quelque chose ou non. Pour la quantité et le montant, on teste directement les variables avec des conditions.

Si le portefeuille du compte contient déjà des actions du code valeur entré en paramètre alors la ligne est "update" dans la table "Portefeuille" avec les nouvelles valeurs et le solde de la table "Compte" est mis à jour. Dans le cas contraire on ajoute une ligne dans la table "Portefeuille" et on met à jours le solde dans la table "Compte".

Pour tester si le client acheteur possède déjà des actions on effectue un COUNT dans la table Portefeuille avec le code valeur et le numéro de compte.

Si le client acheteur ne possède pas déjà les actions, on insère une nouvelle ligne dans la table portefeuille avec le numéro de compte, le code valeur, la quantité, la valeur du cours actuel et on initialise le PMVL à 0 car le calcul du PMVL correspond à (valeur du cours actuel-PAM)\*Quantité mais à l'instant T de l'achat, si l'acheteur n'a pas déjà d'actions le PMVL sera toujours égal à 0.

Si le client possède déjà des actions alors on met à jour la ligne dans la table Portefeuille avec la nouvelle quantité, le nouveau calcul du PAM et le nouveau calcul du PMVL.

Que le client possède ou non des actions similaires lors de l'achat, on doit insérer une nouvelle ligne dans la table Opérations avec le numéro de l'opération indique par une séquence, le numéro du compte, le code valeur, la date d'achat, la nature de l'opération, la quantité et le montant.

Un trigger met à jour le solde de l'acheteur après l'achat.

Procédure « Vendre » :

La procédure Vendre peut émettre 2 exceptions lorsque :

- la quantité donnée est inférieure ou égale à 0

- le montant donné est inférieur ou égal à 0

- le vendeur ne possède pas la valeur à vendre en quantité suffisante dans son portefeuille

Si la quantité de valeur à vendre est exactement égale à la quantité de valeur possédée par le vendeur, l'entrée correspondante dans la table portefeuille est effacée car il ne possédera plus du tout la valeur vendue à la suite de la vente. Si la quantité de valeur à vendre est inférieure à la quantité de valeur possédée par le vendeur, l'entrée correspondante dans la table portefeuille sera simplement mise à jour. On y déduira la quantité vendue à la quantité possédée.

Dans le cas d'une vente qui n'entraîne que la mise à jour du portefeuille et pas sa suppression (quantité vendue < quantité possédée), le PMVL est recalculé selon la formule donnée dans l'énoncé.

Un trigger met à jour le PMVR du compte vendeur. Puisque ce trigger met à jour la table Compte, on en profite pour aussi mettre à jour le solde du vendeur. On y ajoute le montant total de la vente.

Procédure « RepartitionPortefeuille » :

La procédure RepartitionPortefeuille peut afficher 3 erreurs différentes :

-Le compte n'a pas de protefeuille

-Le compte n'existe pas

-Le critère n'est pas reconnu

On teste si le compte existe avec un SELECT, on teste si le critère est reconnu avec des conditions et on renvoie une exception si le comtpe n'a pas de portefeuille.

Pour le calcul de pourcentage :

-Si le critère est 'se' on récupère le secteur économique et le pourcentage puis on affiche avec for loop le nom du secteur économique et le pourcentage pour chaque secteur dans le portefeuille

-Si le critère est 'ib' on récupère l'indice de l'action et le pourcentage puis on affiche avec for loop le nom de l'indice et le pourcentage pour chaque indice dans le portefeuille

-Si le critère est 'null' on affiche le secteur et son indice avec le pourcentage pour chaque couple différent dans le portefeuille avec le for loop.

Calcul du pourcentage d'un secteur ou indice: (SUM(Quantite\*PAM) du secteur(indice) avec numCompte / SUM(Quantite\*PAM) de tout les secteurs(indices) du portefeuille avec le numCompte)\*100

Fonction « TotalPortefeuille » :

Pour chaque valeur possédée par le compte donné en entrée, on multiplie la quantité possédée par le cours actuel de la valeur. La somme de tous ces produits nous donne le total du portefeuille. Si le numéro de compte donné n'existe pas, la fonction renvoie 0. On utilise à nouveau la fonction SQL COUNT pour savoir si le compte existe.