Groupe : DSI 2.2 (GRP 1)	**EXAM TP**	**Environnement :** livesql.oracle.com
Matière : SGBD (PL/SQL)	**PL/SQL-Oracle**	**Documents & Internet : NON Autorisés**
Date : 22/05/2023 & **Durée : 1h30**		**Enseignant :** Adel DAHMANE

NOM & PRENOM : ……

Supposons que vous travaillez sur un projet de gestion de stock pour une entreprise de vente au détail en ligne. La base de données contient trois tables :

- PROD (**PRODUCT_ID** [PK], PRODUCT_NAME, UNIT_PRICE, **STOCK**)
  *--Elle contient des informations sur les produits en vente, y compris leur **quantité en stock**.*
- ORD (**ORDER_ID** [PK], ORDER_DATETIME, ORDER_STATUS)
  *--Elle contient des informations sur les commandes passées en ligne (**COMPLETE** & **CANCELLED**)*
- ORD_ITEMS (**ORDER_ID** [PK, FK], **PRODUCT_ID** [PK, FK], UNIT_PRICE, **QUANTITY**)
  *--Elle contient des informations sur les produits commandés. **Quantity** représente la quantité commandée.*

Voici le script de création de la base de données (sans contraintes) :

```
create table prod as
select PRODUCT_ID, PRODUCT_NAME, UNIT_PRICE from co.products;
alter table prod add stock number default 100;

create table ord as
select ORDER_ID, ORDER_DATETIME, ORDER_STATUS from co.orders;

create table ord_items as
select ORDER_ID, PRODUCT_ID, UNIT_PRICE, QUANTITY from co.order_items;
```

QUESTION BONUS [2 points] :

Ecrire une requête SQL qui permet d'afficher le code source du trigger **TRIG_MAJ_STOCK** (VOIR Q3).

……

……

……

……

TRAVAIL DEMANDE :

1) [4 points] Développer une fonction **GET_STOCK** qui permet de retourner la quantité en stock d'un produit passé en paramètre (à travers son identifiant).

……

……

……

……

……

……

……

……

……

……

……

2) [3 points] Ecrire une procédure **MAJ_STOCK** permettant de modifier la quantité en stock d'un produit donné. L'identifiant du produit ainsi que la quantité à ajouter (ou éventuellement à soustraire) doivent être tous les deux passés en paramètres.

NB : Appliquer les mêmes types de données utilisés dans la base.

……

……

……

……

……

……

……

……

3) [6 points] Ecrire un trigger **TRIG_MAJ_STOCK** qui sera déclenché après chaque annulation d'une commande (ORDER_STATUS = 'CANCELLED'). Le traitement de ce trigger consiste à actualiser la quantité en stock pour chaque produit inclus dans la dernière commande annulée en utilisant la procédure **MAJ_STOCK**. La nouvelle quantité en stock = quantité actuelle + quantité commandée (annulée). Penser à utiliser un curseur pour accéder aux produits de la commande en question.

NB : Ce trigger ne sera déclenché que si une commande a été **annulée** !

CREATE ……

……

……

……

DECLARE

……

……

……

……

BEGIN

……

……

……

……

……

……

END ;

4) [7 points] Ecrire un trigger **TRIG_ORD_BLOCKED** qui sera déclenché avant toute opération de mise à jour (modification et suppression) effectuée dans la table ORD_ITEMS. Le traitement de trigger consiste à lever une exception personnalisée (en utilisant la commande **RAISE**) dont le message est : **«** Opération impossible ! la commande N° ??? est verrouillée. **»**, et ceci seulement dans le cas où l'état (**ORDER_STATUS**) de la commande en question est déjà confirmée (**'COMPLETE'**).

NB : Pour éviter tout problème éventuel, penser aussi à lever une exception d'ordre général en envoyant le message du système.

CREATE ……

……

……

……

DECLARE

……

……

BEGIN

……

……

……

……

……

……

……

……

……

EXCEPTION

……

……

……

……

……

END ;

BON TRAVAIL !