

TP 1 PLSQL

Soit le schéma de base de données suivant :

Client : `(id, prenom, nom, email, ville,);`

Commande : `(id, client_id, date_achat, reference);`

ligne_commande: `(id, commande_id, produit_id, quantite, prix_total);`

Produit : `(id, nom_produit, prix_unitaire)`

- 1- Mettre à jour la colonne Prix_total de la relation ligne_commande en implémentant la requête vue en TD.
- 2- Implémenter les requêtes du TD et exécuter les.
- 3- Ecrire une requête qui retourne le montant total dépensé par chaque client de la table client ; le résultat rangé par ordre décroissant de montant_total dépensé.

Attendu :

ID	NOM	PRENOM	MONTANT_TOTAL
17	Saunier	Patrick	2557,84
6	Riou	Olivier	2480,15
2	Vespasien	Valentin	2465,66
3	Collin	Gustave	2443,85

- 4- Afficher les noms des clients ayant dépenser plus de 2000 euros.
- 5- On souhaite calculer le chiffre d'affaires par mois et par année. Ecrire une requête qui affiche le chiffre d'affaires par mois. Le résultat doit être rangé du mois de Janvier à Décembre.

Vous pouvez utiliser la fonction TO_CHAR vu en TD + structure de contrôle

Résultat attendu :

ANNEE	Mois	CA
2019	JANVIER	11819,81
2019	FEVRIER	10679,03
2019	MARS	303

- 6- Créer une séquence seq_tp ayant pour première valeur 121. Afficher sa valeur actuelle et la valeur suivante.
- 7- Implémenter une procédure stockée `sp_insert_ligne_commande(p_commande_id, p_produit_id, p_quantite)` qui permet d'insérer une ligne de commande dans la table ligne_commande. Utiliser la séquence seq_tp créée précédemment pour l'id de la ligne de commande. Le prix total doit être calculer dans la procédure.

Tester votre procédure avec un id de commande ou un id de produit qui n'existe pas.

Modifier la procédure précédente afin de gérer le cas où p_commande_id ou p_produit_id n'existe pas.

NB : Une ligne de commande ne peut être insérée uniquement si la commande et le produit existent déjà.

- 8- Implémenter une procédure stockée **sp_insert_commande(p_client_id, p_reference)** qui permet d'insérer une commande dans la table commande. La date de la commande est la date d'aujourd'hui. Vous pouvez utiliser la fonction **sysdate**. Gérer le cas où le client n'existe pas.
- 9- On souhaite calculer les frais de livraisons des commandes. Implémenter une procédure **sp_frais_livraison (p_commande_id INT, p_frais_livraison OUT NUMBER)** qui calcule les frais de livraison d'une commande. p_frais_livraison est un paramètre de sortie.

Les frais de livraisons valent :

- 0 si commande \geq 100 euros
- 4 si commande \geq 50 euros et $<$ 100 euros
- 10 si commande $<$ 50 euros.

Cette question sera traitée la semaine prochaine avec une fonction.