Une image contenant texte, capture d’écran, conception

Description générée automatiquement

**Direction Régionale TTA**

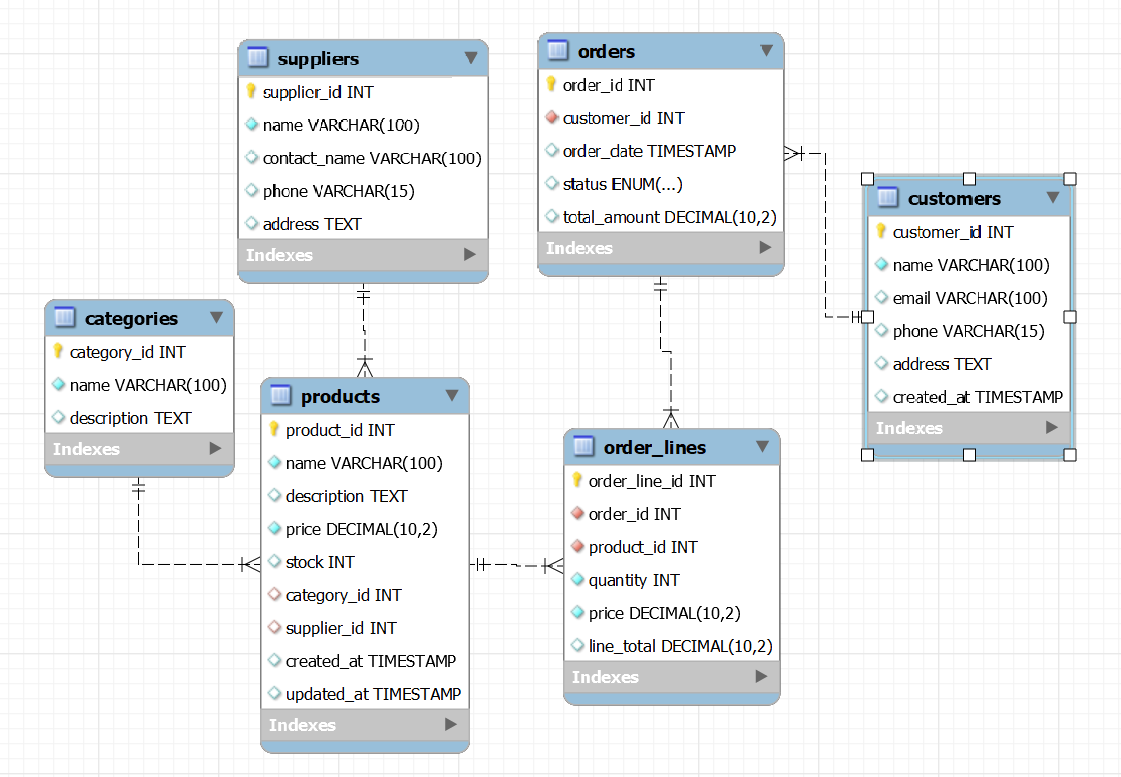
**Année de Formation : 2024-2025**

**Contrôle continu 2**

**M203 : Gestion des données**

|  |  |
| --- | --- |
| **Filière :** **Développement Digital - Option Web Full Stack** | **Epreuve : Pratique** |
| **Année : 2ème année** | **Variante : 1** |
| **Niveau : Technicien spécialisé** | **Barème : 40 Pts** |
| **Date : 31/10/2024** | **Durée : 2h** |

**Utilisez le script fournis par le formateur pour créer la base de données sur votre PC .**

**La structure de la base de données est la suivante : **

**Copiez les questions suivantes sur votre fichier de script, mettez-les en commentaire et insérez vos réponses après chaque question.**

**Enregistrer votre script avec le nom : BDCC2\_V1\_............. en remplaçant les pointillés par votre nom et prénom.**

**Important : N’oubliez pas d’enregistrer régulièrement votre fichier aucune augmentation de temps ne sera attribuée en cas de perte du travail.**

**1. Fonctions**

1. Créez une fonction **get\_total\_order\_amount** qui prend en paramètre **order\_id** (ID d'une commande) et retourne le montant total de cette commande (la somme des produits dans la commande, en prenant en compte les quantités et les prix). **(2 points)**
2. Créez une fonction **get\_product\_stock** qui prend en paramètre product\_id (ID d'un produit) et retourne le stock actuel de ce produit. **(2 points)**

**2. Procédures Stockées**

1. Créez une procédure stockée **update\_stock** qui prend en paramètres product\_id et quantity et met à jour le stock du produit spécifié en soustrayant la quantité passée en paramètre. Attention : la procédure doit vérifier que le stock restant est suffisant avant d’effectuer la soustraction. Si ce n'est pas le cas, elle doit lever une erreur. **(2 points)**
2. Créez une procédure **add\_order\_line** qui prend en paramètres order\_id, product\_id, quantity, et price, puis ajoute une ligne de commande dans la table order\_lines. Remarque : la procédure doit vérifier que le produit spécifié a assez de stock disponible. **(2 points)**

**3. Triggers**

1. Créez un trigger **after\_insert\_order** qui se déclenche après l'insertion d'une commande dans la table orders. Ce trigger doit mettre à jour le statut de la commande en new\_order si aucune ligne de commande n’a encore été ajoutée pour cette commande. **(2 points)**
2. Créez un trigger **after\_insert\_line\_order** qui se déclenche après l'insertion d'une ligne de commande dans la table line\_orders. Ce trigger doit mettre à jour le statut de la commande en pending. **(2 points)**

**4. Transactions**

Écrivez une procédure stockée **new\_order** qui crée une nouvelle commande pour un client donné, cette PS accepte comme paramètres le numéro du client, deux numéros de produits et deux quantités. Ajoute en transaction deux lignes de commande, et met à jour le stock des produits concernés.   
Avant l’insertion de la ligne de commande il faut chercher le prix du produit concerné pour l’insérer dans la ligne.  
La transaction doit être annulée si le stock d’un des produits est insuffisant pour satisfaire une des lignes de commande. **(3 points)**

**5. Curseurs**

Créez un curseur dans la procédure stockées **get\_customer\_infos** pour afficher le nom, l'adresse, et l'email de chaque client ayant passé une commande au cours des 30 derniers jours. Utilisez le curseur pour afficher chaque client avec ses informations de contact. **(3 points)**

**6. Gestion des Utilisateurs**

Créez un nouvel utilisateur SQL nommé **consultant\_order** avec un mot le mot de passe ‘123456’ et accordez-lui uniquement les droits de lecture (**SELECT**) sur la table **orders**. **(2 points)**