

SQL et Programmation – TP1

Partie 1 : construction de la base

Remarque : Avant de commencer le TP, pensez à faire le ménage dans votre base de données
Utilisez l'interface graphique de workbench pour effacer vos tables et vues.

Enoncé :

Nous voulons construire une base de données de gestion de commandes qui inclut les entités Customer (i.e., Client), Product (produit), et Invoice (Facture). Chaque client est identifié par un numéro et est caractérisé par son nom, son prénom, son adresse, son numéro de tél, sa date de naissance et son sexe. Un produit est identifié par un numéro et est caractérisé par sa désignation, son prix unitaire et sa quantité en stock. Une facture est identifiée par un numéro et possède une date d'établissement. Les numéros de client, produit et facture sont des numéros séquentiels. De plus, nous avons les règles de gestion suivantes :

- Une facture comporte un ou plusieurs produits. Pour chaque produit d'une facture, nous avons la quantité achetée de ce produit
- Un produit peut paraître dans plusieurs factures (avec des quantités différentes).
- Une facture appartient à un seul client.
- Un client peut établir plusieurs factures.

Conception et mise en place de la base de données

- 1) Etablir un schéma conceptuel sous forme d'un diagramme de classes. Vous utiliserez plantuml pour votre modélisation et les règles de passage diagrammes de classes—relationnel pour passer au modèle relationnel. Un rappel est disponible sur moodle.
- 2) Créer le modèle relationnel et le script SQL.
- 3) Construire la base

Rappel :

Use mabase

SET default_storage_engine= InnoDB;

SET SQL_SAFE_UPDATES=0;

Partie 2 : Mise en place des procédures de basiques de manipulation de la base

- **Mettre en commun la base avant de commencer cette partie**

Pour des raisons de sécurité (les utilisateurs et les applications n'auront pas accès aux tables en modification) et de simplicité, en particulier pour cacher à l'utilisateur la gestion des séquences (auto_increment) pour les clients, les produits et les factures, nous souhaitons que cette base soit manipulée via des fonctions stockées.

- 1) Ecrire une procédure stockée permettant d'insérer 1 client.

```
PROCEDURE insertCustomer (IN name varchar(25),  
    IN firstName          varchar(25),  
    IN address             varchar(254),  
    IN birthDate           datetime,  
    IN phoneNumber         varchar(10),  
    IN gender              CHAR)
```

- 2) Ecrire de la même manière la procédure `insertProduct`

- 3) Ecrire les instructions permettant de remplir les tables Customer et Product en utilisant les deux procédures et les valeurs suivantes

Customer

Coulomb	Francois	4 rue liberte	19800212	06456780	m
Dupont	Bernard	120, square zola	19720812	01345678	m
Corda	Nathalie	66, bd napoleon III	19770210	06455790	f
Pierre	Alexandra	1, place des martyres	19720908	01645870	f

Product

DVD Matrix	23.5	100
DVD The Island	49.30	100
CD Album Yanni	25.9	100
DVD Divergent	10.00	5
DVD Allegiant	12.00	5
DVD Insurgent	11.5	5

- 4) Lorsqu'un client achète un produit, cet achat est soit rajouté à une facture existante soit on crée une nouvelle facture.

- Ecrire la procédure

```
buyProduct( IN customer int, IN product INT, IN qty int, IN last_invoice boolean)
```

Cette procédure permet de rajouter le fait que le client **customer** a acheté le produit **product** en quantité **qty**. Cet achat sera rajouté à la dernière facture du client si le paramètre **last_invoice** vaut **True**. Si **last_invoice** vaut **True** mais le client n'a pas encore de facture, ou si **last_invoice** vaut **False**, alors une nouvelle facture est créée pour le client avec la date courante. Attention, votre procédure ne doit pas insérer un achat si la quantité en stock du produit acheté est insuffisante. Elle doit aussi mettre à jour la quantité en stock après l'achat.

- Utiliser cette procédure pour noter les achats suivants :
 - o Client 2 achète produit 1 en qte 2, produit 2 en qte 4 sur la même facture
 - o Client 3 achète produit 2 en qte 20, produit 1 en qte 3 sur la même facture
 - o Client 2 achète produit 4 en qte 9, produit 5 en qte 7 sur la même facture
 - o Client 1 achète produit 1 en qte 9, produit 2 en qte 7 sur la même facture
 - o Client 2 achète produit 1 en qte 7, produit 3 en qte 6 et le produit 4 en qte 7 sur la même facture
 - o Client 4 achète produit 4 en qte 1, produit 3 en qte 7 sur la même facture
 - o Client 3 achète produit 4 en qte 6.
 - o Client 4 achète produit 2 en qte 2, produit 1 en qte 3 sur la même facture

- 5) Ecrire la procédure

```
supplyProduct(designation varchar(254), unitPrice float, qty int)
```

Cette procédure permet de prendre en compte les approvisionnements en produits. Le produit doit exister dans la base. La quantité en stock est mise à jour avec la nouvelle quantité acquise. On s'aligne aussi sur le nouveau prix unitaire même s'il est plus faible que celui présent dans la base. Si le produit n'existe, affichez un message.

- 6) Donner les requêtes suivantes :
- Lister les noms des clients

- Quel est le produit le plus cher ?
- Nombre de factures du client 1
- Prix du produit 5
- Liste des produits de la facture 2, on affichera leur description.
- Le nombre de produits par facture
- Quel est le client qui a le moins de factures