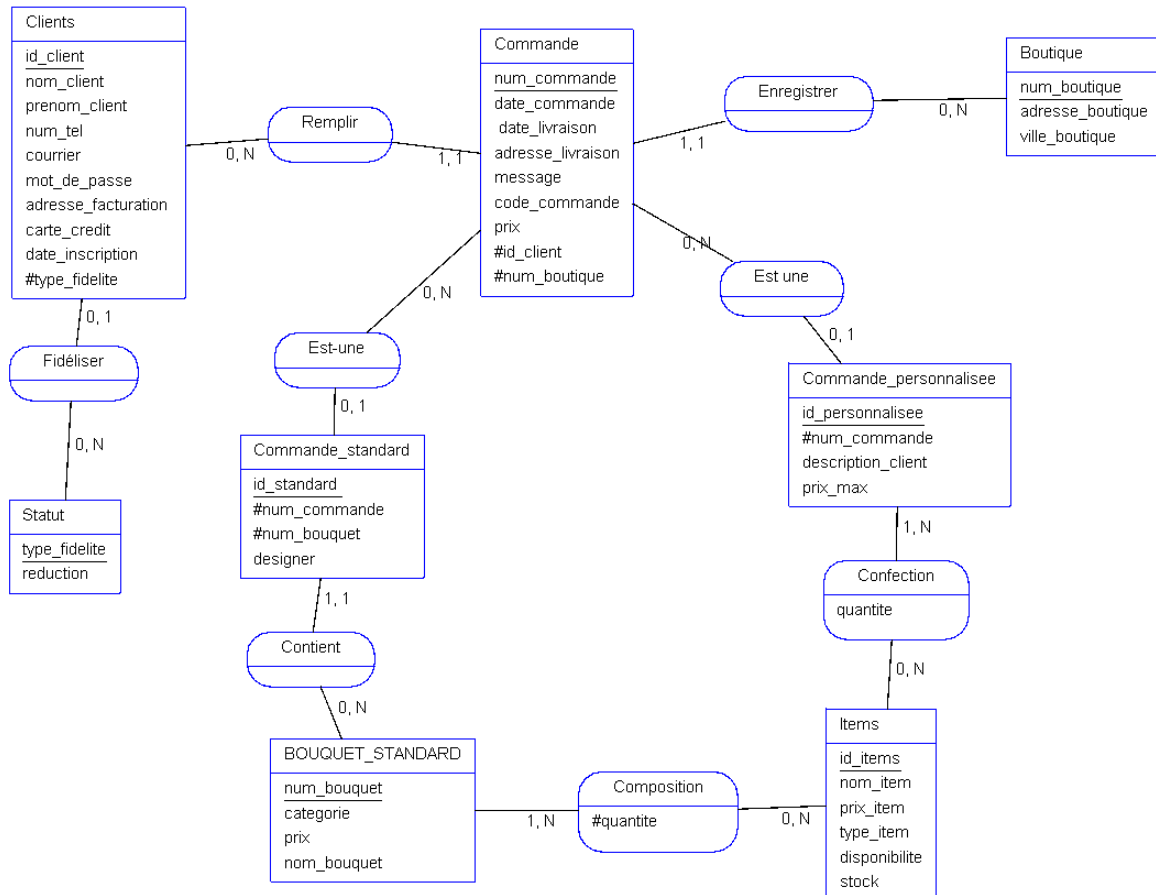


## Note technique explicative Projet BDD

### I - Organisation de la BDD

#### a. Schéma E/A



#### b. Schéma relationnel

**CLIENT**(id\_client INT PK, nom\_client VARCHAR(50), prenom\_client VARCHAR(50), num\_tel VARCHAR(15), courrier VARCHAR(50), mot\_de\_passe VARCHAR(50), adresse\_facturation VARCHAR(100), carte\_credit VARCHAR(50), date\_inscription DATE, #type\_fidelite VARCHAR(10) FK);

**STATUT**(type\_fidelite VARCHAR(10) PK, reduction INT);

**COMMANDE**(num\_commande VARCHAR(10) PK, date\_commande DATE, date\_livraison DATE, adresse\_livraison VARCHAR(100), message VARCHAR(250), code\_commande VARCHAR(5), prix INT, #id\_client INT FK, #num\_boutique INT FK);

**BOUTIQUE**(num\_boutique INT PK, adresse\_boutique VARCHAR(50), ville\_boutique VARCHAR(25));

**BOUQUET\_STANDARD**(num\_bouquet INT PK, categorie VARCHAR(50), prix INT, nom\_bouquet VARCHAR(20));

**COMMANDE\_PERSONALISEE**(id\_personnalisee INT PK, #num\_commande VARCHAR(10) FK, description\_client VARCHAR(250), prix\_max INT);

**COMMANDE\_STANDARD**(id\_standard INT PK, #num\_commande VARCHAR(10) FK, #num\_bouquet INT FK, designer VARCHAR(20));

**ITEMS**(id\_items INT PK, nom\_item VARCHAR(25), prix\_item DECIMAL, type\_item VARCHAR(25), disponibilite BOOLEAN, stock INT);

**COMPOSITION**(#num\_bouquet INT FK, #id\_items INT FK, quantite INT);

**CONFECTION**(#id\_personnalisee INT FK, #id\_items INT FK, quantite INT);

### c. Explications des choix

Nous avons choisi ces schémas E/A et relationnels en fonction des données de l'énoncé. En effet, il était notamment stipulé les points suivants, qui ont appuyé nos choix pour les schémas:

- Le client doit pouvoir passer deux types de commandes, nous séparons donc l'entité commande en deux sous-entités permettant de différencier ces deux types.
- Pas de contrainte sur les stocks : les boutiques sont donc reliées à un seul et même stock
- Les commandes sont reliées au même stocks, les entités commande\_standard et commande\_personnalisées se rejoignent donc à nouveau sur l'entité Items

Ceci n'est qu'un excerpt de nos choix, cependant nous voulions vous représenter les réflexions qui nous ont dirigés vers le schéma E/A final (cf. ci-dessus).

## II - Options de codage

### a. Fonctionnalités du code

Notre code sql contient toutes les commandes que nous avons utilisées dans notre rendu final, notamment : 1 requête synchronisée ; 1 requête avec auto-jointure ; 1 requête avec une union (ligne 540 du code SQL).

Il permet de se connecter en tant que client pour passer commande par exemple ou en tant qu'admin pour gérer les stocks. De plus, l'admin peut consulter les statistiques de différents modules, ou exporter une requête en XML.

Le code c# couplé à la base de données créée par le code SQL nous ouvre une interface graphique qui permet de naviguer sur les différentes pages. Nous avons pour cela utilisé des techniques externes au cours que nous allons développer dans le paragraphe suivant.

### b. SQL, C# & Windows Forms

Nous avons pu créer la base de données sur MySQL Workbench avec l'aide du schéma relationnel. L'insertion des valeurs était cependant ardue, compte tenu de la configuration de la base de données au niveau des types de commandes. Nous avons donc rempli notre base de données à la main, puis nous l'avons complétée une fois que le code c# fut opérationnel. En effet, cette démarche était bien plus pratique et confortable.

Nous avons décidé d'utiliser un WebApplication Framework pour l'interface graphique du système d'information de Mr.Bellefleur et pour l'interface des clients. Nous utilisons donc des WebForms de ASP.NET pour donner la structure globale de chaque page et pour intégrer le code C# derrière.

Le propriétaire Mr.Bellefleur se connecte avec des identifiants que notre équipe lui fournis pour accéder à son système d'information (Id: bellefleur, MDP: root). Mr.Bellefleur aura accès à plusieurs modules pour assurer la gestion des commandes, des clients et des produits en stocks.

Les modules se présentent sous la forme de boutons sur le côté gauche de la page qui sert comme un navigateur. Une fois que Mr.BelleFleur clique sur un module, il est envoyé vers une autre page et il peut visionner les différents tableaux avec des filtres. Il y a aussi la possibilité de revenir en arrière pour voir les autres modules. Voici les différents modules et leurs caractéristiques:

- **Client** : avec ses caractéristiques, son statut de fidélité, ses achats passés
- **Produit** : avec ses caractéristiques et la possibilité de mettre à jour le stock si elle est en dessous d'une limite (Nous avons mis la limite à trois).
- **Commandes** : Mr.BelleFleur a une vue sur toutes les commandes. On peut aussi les filtrer en fonction de plusieurs caractéristiques (Etats de commande, Commande par boutique, Commande par client, Type de commande, etc.).
- **Statistiques** : pour le module statistique, Nous avons décidé de faire des statistiques sur tous les modules : clients, commandes, boutiques, produits.
- **Export** : permet l'export en XML

Un client peut se connecter avec ses identifiants (courriel et mot de passe) sur la page ClientLogin et s'il n'a pas de compte il pourra créer un compte sur la page ClientRegister. Une fois connecté, il a accès à un catalogue numérique où il pourra voir ses informations (identifiants, statut,...), voir les commandes passées et leurs états, et passer une commande standard ou personnalisés.

Pour la commande standard, le client a l'option entre les différents bouquets standards de la boutique qu'il choisit. Pour la commande personnalisée, le client peut choisir différents items (fleurs et accessoires) avec leur quantité pour créer son propre bouquet.

### III - Développements complémentaires

#### **a. ASP.NET, HTML et CSS**

Avec ASP.NET, nous avons utilisé une Master Page pour donner l'aspect visuel global de chaque page. Pour chaque module, nous avons un WebForm qui permet de faciliter la navigation entre les pages et d'avoir un visuel clair pour le propriétaire du magasin.

Nous avons fait la même chose pour le client : nous avons deux WebForms pour l'authentification et l'enregistrement d'un client et un WebForm pour la boutique en ligne où le client peut faire ces commandes.