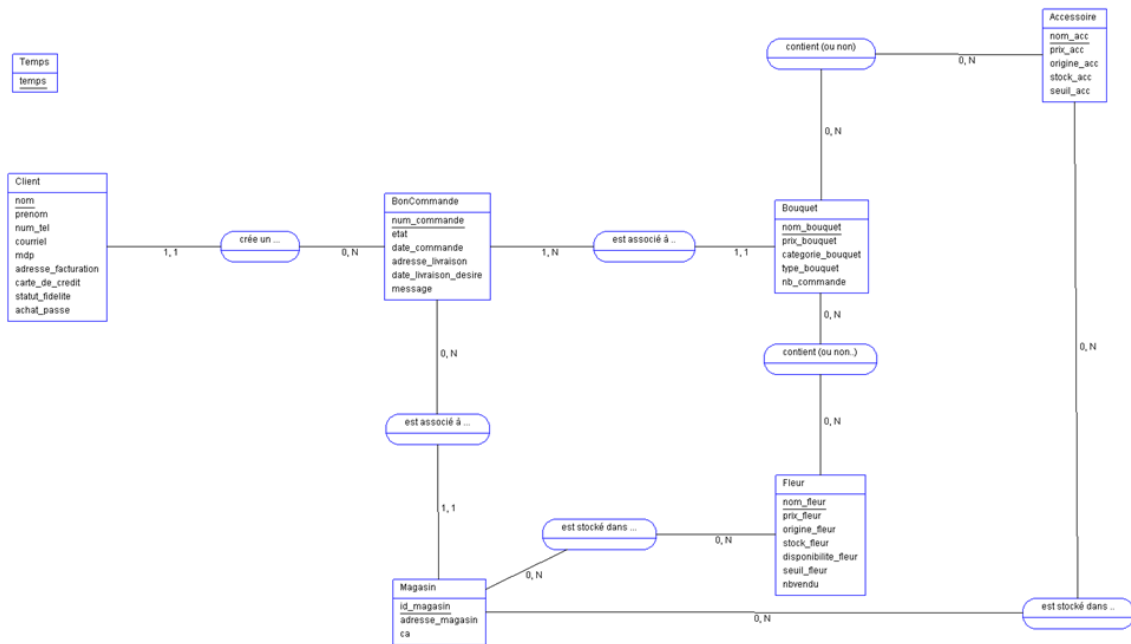


Rapport Base de données – Le pouvoir des fleurs.

Organisation de la base de données.



Pour réaliser le projet de Base de Données, nous sommes partis sur une base de 7 tables. Il semblait évident dans un premier temps de créer les tables suivantes :

- La table client, pour référencer l'ensemble des clients existants chez M. Bellefleur ;
- La Table BonCommande, pour pouvoir référencer les commandes émises par un client, on suppose qu'un client ne peut faire qu'un bon de commande à la fois ;
- La table bouquet, qui indique quels types de bouquets le client veut commander ;
- Ce bouquet est composé soit d'accessoires, soit de fleurs, chacun identifiés par deux tables distinctes.
- La table Temps, quant à elle, sert à l'approvisionnement des stocks qui doit être fait tous les 3 jours.

On suppose qu'un client peut faire entre 0 et N bons de commandes et on a un bon de commande pour chaque client.

Un bon de commande est toujours associé à un bouquet, où on lui associe 1 à N bons de commandes, dans la mesure où différents clients peuvent commander le même type de bouquets (en particulier les bouquets standards).

Chaque bouquet peut être composé entre 0 et N fleurs, ainsi que 0 à N accessoires, réciproquement, une fleur peut être présente 0 à N fois dans un bouquet, tout comme un accessoire.

De même qu'une fleur peut être présente à la fois dans 0 (si rupture de stock), ou N magasins (ici au nombre de 3), et un magasin (logiquement), peut comporter entre 0 et N fleurs. Le mécanisme est analogue pour les accessoires.

Choix de code.

Après avoir implanté le diagramme E/A en langage SQL et avoir créé les fichiers de peuplement, l'idée était de créer une interface graphique, sous la forme d'une mini-application, dans l'objectif d'accéder plus simplement aux informations demandées. Nous sommes partis sur une méthode globale en C# appelée « Application », qui contient toutes les méthodes nécessaires à l'affichage de l'application.

Ensuite, les méthodes sont défilées à la suite, en faisant toujours l'interopérabilité entre le logiciel MySQL Workbench et le logiciel Visual Studio.

Dès l'exécution du code, le programme affiche un message de bienvenue. Si l'utilisateur se considère comme client, il se connectera en tant que « bozo », si l'utilisateur veut se connecter en tant qu'administrateur, il se connectera en tant que « root », cela n'est pas spécifié pour des raisons logiques (un client ne peut pas avoir les accès administrateur). De même, pour éviter toute modification malvenue, un client ne peut pas effectuer de requête SQL sur notre application, seul l'administrateur le peut, il a donc tout les privilèges lors de sa requête contrairement au connexion client qui sont de toute manière limité sur SQL à une simple lecture (select...) des données.

Quand on se connecte en tant qu'administrateur, on peut avoir accès à plusieurs informations, tels que l'accès à la base de donnée, si on veut regarder le contenu des stocks, si on veut les réapprovisionner (hors alerte indiqué), la visibilité de l'état des commandes, voir les données statistiques (tels que la moyenne du bouquet acheté, meilleur client du mois, meilleur client de l'année, le bouquet standard qui a généré le plus de ventes, le magasin qui a généré le plus de chiffres d'affaires ainsi que d'autres informations comme les clients qui n'ont jamais commandé ou encore la valeur des stocks), l'administrateur a également la possibilité s'il le souhaite d'exporter les informations de la base de donnée qui l'intéresse dans un fichier XML ou JSON.

Si la valeur des stocks est trop basse, l'administrateur peut à tout moment renflouer les stocks en ajoutant 20 éléments afin d'avoir une certaine marge sur la valeur du seuil de réapprovisionnement. En dehors de cela, les stocks sont renfloués automatiquement tous les débuts de mois ou bien tous les trois jours. A noter que le réapprovisionnement des fleurs dépend de la disponibilité des fleurs (période de floraison) au mois en question. Si ce n'est pas la période, les fleurs en question ne seront pas réapprovisionnées.

Quand on se connecte en tant que client, le système propose deux choix :

- Soit l'individu est déjà un client, et alors il peut effectuer une commande ;
- Soit l'individu n'est pas encore client, alors l'interface fait en sorte qu'il se crée en demandant toutes les informations nécessaires pour la création du compte client, lui permettant ainsi de réaliser une commande par la suite.

Quand un individu effectue une commande dans un magasin choisi, il a deux possibilités :

- Si le client choisit de réaliser un bouquet standard, alors le programme nous permet de commander ce bouquet en fonction des stocks disponibles. Si certains items manquent au bouquet, nous avons fait le choix de conserver le bouquet sans les items manquants afin de tout de même honorer la commande dans les délais. Mais on avertit le client de ce manque.
- Si le client décide de choisir un bouquet personnalisé, alors la liste de toutes les fleurs s'affichent, et le client peut choisir les types de fleurs qu'il veut, et en fonction de son budget, l'algorithme sélectionne pour lui la quantité maximale de fleurs des types choisis. A noter qu'en cas de rupture de stock dans le magasin de la commande, un employé contacte le client pour lui proposer une alternative de disponibilité dans un autre magasin si possible, qui est facturé 2 euros supplémentaires pour les frais de port. Le même procédé se réalise pour les accessoires.

Dans les deux cas, le programme indique si le client bénéficie (ou non) d'une réduction, en fonction de sa fidélité (or ou bronze), ou bien s'il est un nouveau client (il reçoit alors une réduction de 5% en gage de cadeau de bienvenue).

De plus, nous avons décidé de fixer un seuil d'alerte pour chaque article, ce qui nous permet de déclencher une alerte de stock en fonction de l'article concerné. Les articles moins fréquemment commandés ont des seuils d'alertes plus bas que ceux très demandés.

Pour les commandes passées plus de trois jours à l'avance, les articles ne sont parfois pas en stock. Cependant, nous avons pris la décision de commander et de nous approvisionner en conséquence pour pouvoir honorer la commande dans les temps sans avoir à déplacer les stocks existants.

Conclusion :

Après recul et analyse de notre code, nous avons constaté que pour améliorer la lisibilité et la compréhension du code, il serait plus judicieux d'utiliser des classes en répartissant les différentes fonctionnalités dans plusieurs fichiers. Cette approche faciliterait également la résolution d'erreurs.

En outre, nous envisageons plusieurs voies d'améliorations pour notre système. Tout d'abord, il serait intéressant de créer une interface graphique WPS pour offrir une expérience utilisateur plus intuitive.

De plus, nous pourrions améliorer la gestion de l'état de la commande au fil du temps, en faisant en sorte que l'état se mette à jour automatiquement plusieurs jours après la commande (3 jours par exemple). Cela permettrait aux clients de suivre l'évolution de leur commande de manière plus réaliste.

Nous pourrions également envisager une meilleure gestion du budget du client, en lui permettant de savoir combien d'argent il lui reste à la fin de son achat. Enfin, nous pourrions améliorer la gestion des stocks en stockant la même fleur dans plusieurs magasins, comme cela est déjà prévu dans notre fichier de peuplement, ce qui rendrait nos données plus concrètes.