

Raport de projet

Base de données VeloMax



Organisation de la base de données :

Nous avons principalement respecté le cahier des charges donné en essayant d'interpréter au mieux certaines contraintes implicites. Nous avons créée notre base de données en nous basant sur le modèle E/A que nous avons réalisé (joint avec ce rapport EA_VeloMax.pdf). A partir de ce MCD, nous avons obtenu le MLD suivant :

- Commande (no_commande, adresse_livraison, date_commande);
- Modele (no_modele, nom_modele, grandeur, prix_modele, date_introduction_modele, stock_modele, date_discontinuation_modele, ligne_produit);
- Piece (no_piece, description_piece, no_catalogue, prix_piece, date_introduction_piece, date_discontinuation_piece, delai_approvisionnement, stock_piece);
- Programme (no_programme, description_programme, cout, duree, rabais);
- Fournisseur (siret, nom_fournisseur, contact_fournisseur, adresse_fournisseur, libelle);
- client (id_client, adresse, telephone, courriel, nom);
- Client_particulier (#id_client, prenom_particulier);
- Client_entreprise = (#id_client, contact_entreprise, remise);
- Contient_modele = (#no_commande, quantite_modele, #no_modele);
- Contient_piece = (#no_commande, quantite_piece, #no_piece);
- Adhere (#(#id_client), date_adhesion, #no_programme);
- Fournit (#no_piece, #siret);

- Associé (#no modele, #no piece);
- Effectue (#no commande, #id client);

Dans les tables Modèle et Pièce nous avons également indiqué la quantité en stock chez VeloMax pour chaque pièce et modèle.

De plus, afin de permettre une meilleure gestion des clients et éviter la redondance des attributs, nous avons d'abord créé une table client qui se sépare en deux entités distinctes (clients particuliers et clients d'entreprises) qui héritent de cette table. Cela permet une meilleure gestion des noms (prénom du particulier et nom d'entreprise) et des promotions (remise et programme).

Enfin, la quantité de modèle ou de pièces à commander est enregistrée à chaque commande dans contient modèle et contient pièce tout comme la date d'adhésion au programme dans adhère.

Afin de permettre l'utilisation de notre base de données, nous avons créé des scripts de création de tables et des tuples pour le remplissage (joint avec ce rapport VeloMax.sql).

Puis, nous nous sommes connectés à notre base sur Visual studio afin de pouvoir la visualiser et la modifier (joint avec ce rapport le projet zippé).

Options de codage :

Tout d'abord, lorsque le programme se lance, un menu apparaît, proposant différentes fonctionnalités adaptées aux différents utilisateurs pouvant avoir accès à cette base de données.

```
Bienvenue sur le site de Vélo max.
Tapez 0 si vous voulez quitter
Tapez 1: Gestion des pièces de rechanges et Gestion des vélos
Tapez 2: Gestion des clients particuliers et des clients Entreprise
Tapez 3: Gestion des fournisseurs
Tapez 4: Gestion des commandes
Tapez 5: Gestion des stocks
Tapez 6: Statistiques du magasin
Tapez 7 : Mode Démo
Tapez 8 : Requêtes de création
```

Si l'utilisateur est un client, il peut accéder à la "Gestion clients particuliers et des clients entreprise". Cela lui permet de créer un compte, le supprimer ou le modifier (adresse, téléphone, courriel, programme).

Pour éviter les problèmes de numéro de compte en double, l'id client est généré automatiquement par incrémentation du dernier id existant dans la base.

Pour la date d'adhésion au programme, elle est aussi générée automatiquement grâce à la date actuelle (datetime.now).

Le calcul du prix se fait automatiquement. Le client n'a donc pas à se préoccuper de la saisie de ces informations.

Le client peut également accéder à la "Gestion des commandes" afin de passer commande pour des pièces de rechange ou vélos.

Pareil que pour l'id client, le numéro de suivi de la commande est généré automatiquement et donné à la fin.

Il peut aussi supprimer une commande qu'il a passé (on demande le numéro et on affiche le contenu de la commande pour que le client soit sûr) ou modifier certaines informations (l'adresse de livraison, le contenu de la commande). Pour la date de la commande, elle aussi générée automatiquement.

Si l'utilisateur est un employé / administrateur, il a accès au "module statistique" qui permet d'afficher des informations concernant les Pièces, Clients, Fournisseurs ou Commandes (quantités vendues, meilleurs clients, programme d'adhésion des clients...).

Il peut aussi accéder à la "Gestion des pièces de rechange et gestion des vélos" afin de rajouter/supprimer des pièces/modèles au catalogue ou au stock, ou de modifier ces derniers (prix). C'est aussi à lui de faire la "gestion des fournisseurs" (pour plus de sécurité, ce n'est pas le fournisseur lui-même qui s'enregistre).

Pour l'évaluateur, le "mode démo" demandé permet d'afficher des informations concernant Pièces, Clients, Fournisseurs ou Commandes (nombres de clients, stocks...) et exporter des données concernant les stocks ou les clients au format xml et json. De plus, il est possible d'afficher des "requêtes de création".

Gestion des tables :

Nous n'avons pas rencontré de problème pour l'affichage (en mode console) et la création des commandes, pièces etc ou pour l'affichage du stock.

Nous n'avons pas eu de problèmes non plus pour les mises à jour car nous avons essayé de veiller à ce qu'elles impliqueraient. Par exemple, si l'on crée une nouvelle commande cela aura pour effet de modifier la quantité de produit en conséquence dans le stock. Un employé ne peut que modifier le prix d'une pièce ou vélo pour ne pas mettre une pièce sur un autre vélo en changeant son identifiant numéro de pièce.

Au niveau de la gestion des dépendances c'est là où nous nous sommes aperçus de l'importance des triggers (non utilisé). En effet, la suppression d'une pièce sur un modèle de vélo est possible, nous n'avons pas vérifié qu'elle était déjà prise. D'autre part, la suppression d'un vélo ne transforme pas les pièces en pièces de rechange disponible dans le stock. Enfin, la suppression d'un client entraîne la suppression de l'historique de ses commandes, et les produits associés ne sont pas remis dans les stocks de VeloMax.

Modules statistiques :

L'export en XML donne les informations sur les pièces avec un stock faible (choisi inférieur ou égal à 5) avec les informations sur les fournisseurs qui vendent cette pièce ordonnées sur le prix de la pièce, le délai de livraison et le libellé par ordre croissant. Cela permet de suggérer à Mr Legrand une commande de pièces auprès du meilleur fournisseur possible mais de lui laisser le choix en fonction du qualificatif des fournisseurs car tous les fournisseurs sont visibles pour les comparer.

L'export en JSON donne les informations sur les clients dont le programme de fidélité Fidélité arrive à expiration dans moins de 2 mois avec historique des abonnements afin de les relancer. On peut aussi avoir une vue sur les clients sans programme (null) et ceux qui ont eu leur programme expiré afin de relancer aussi ces derniers.