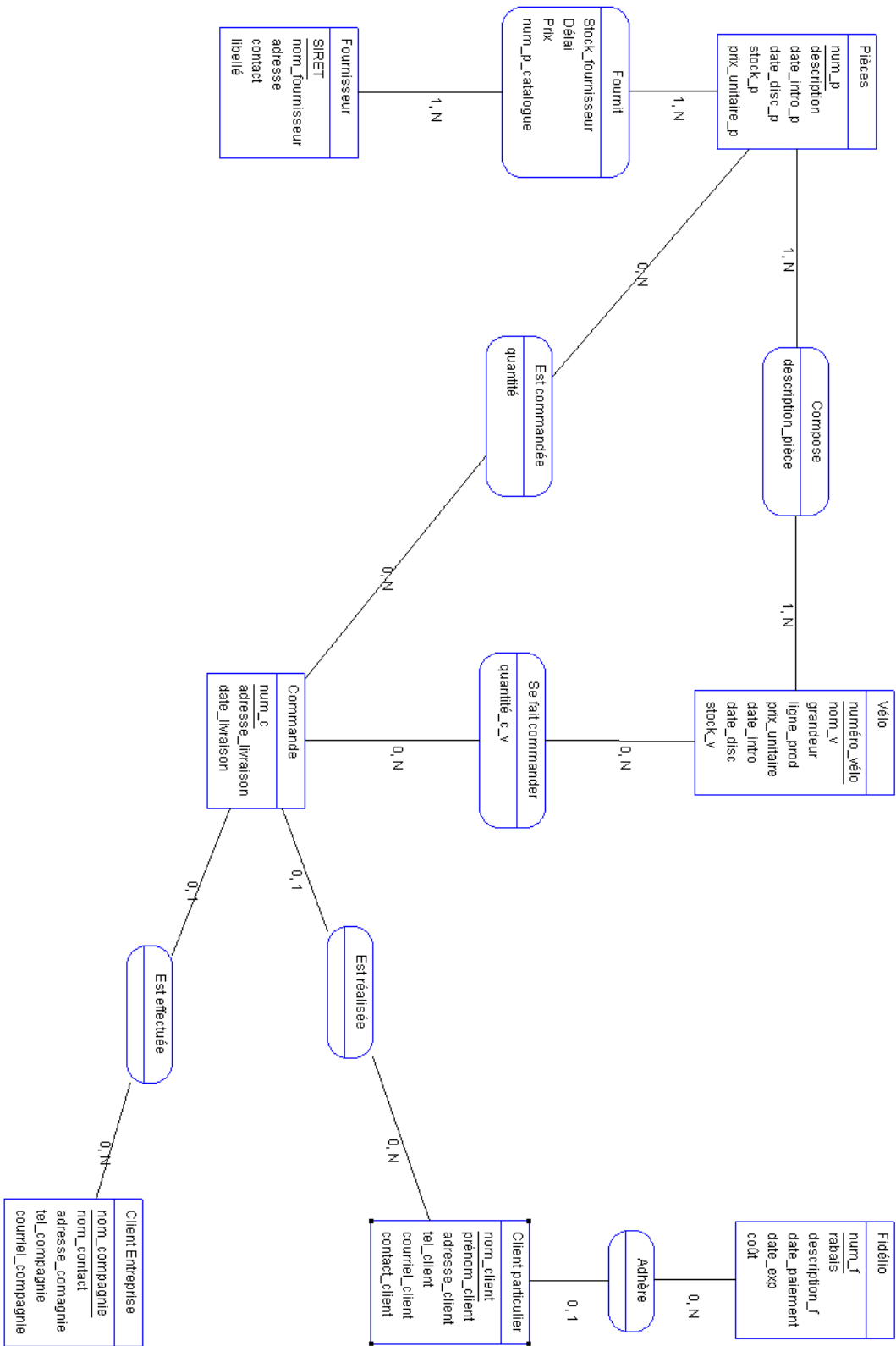


Rapport base de données VeloMax

Schéma Entite/Association



Le projet C# a été conçu pour exécuter des requêtes dans SQL, à l'aide d'une classe Connexion qui fait le lien entre Visual Studio et MySQL (root/root et bozo/bozo). Le programme principal contient toutes les fonctions nécessaires pour faire des opérations d'ajout, de suppression ou de mise à jour de la base de données. Les classes créées sont les suivantes : Piece, Velo, Fournisseur, Fidelio et Jsonification et sont utilisées afin d'exporter les données demandées en Json/XML.

Export en Json :

Chacune des classes Piece, Velo, Fournisseur et Fidelio a pour variables de classe les attributs associés à leurs tables respectives. Par exemple, pour exporter le stock faible des pièces, une liste de string contenant les pièces ayant un stock inférieur à 2 et un objet Jsonification est créé, prenant en paramètre le nom du fichier qui va être créé. Pour chaque pièce (pour chaque itération de la liste de pièce), un objet Piece est créé et la fonction write_object est appelée.

Détails de la classe Jsonification et ses deux fonctions : les variables de classe de la classe sont le nom du fichier qui va être créé et une liste de tous les objets qui vont y être ajoutés.

Write_object prend en paramètre un objet o qu'elle va ajouter à la variable de classe liste. Si le fichier n'existe pas, elle crée le fichier et sérialise l'objet en question. Sinon, elle écrase le fichier déjà existant et sérialise la liste d'objet. Cela permet d'ajouter un à un chaque objet (chaque pièce dans notre exemple) à chaque appel de la fonction write_object de l'objet jsonification voulu, dans un unique fichier.

Pour exporter un fichier en XML, le programme exporte dans un premier temps le fichier en json puis fait la conversion du Json au xml avec la fonction de classe xmlification_machiavélique de la classe Jsonification. La simplicité de l'export du Json au XML nous a fait opter pour cette solution (readalltext du fichier json créé, puis utilisation de JsonConvert.DeserializeObject et writealltext).

Format d'input : On demande à l'utilisateur d'entrer certaines données, notamment lors de la création d'un client particulier/entreprise, d'un fournisseur, d'une pièce... Les formats obligatoires concernent les stocks (qui doivent forcément être des int) et les date qui doivent suivre un format (YYYY-MM-DD) afin de pouvoir être convertis en DateTime (par exemple pour comparer la date d'expiration et le jour présent). Autrement, les autres inputs obligatoires concernent les menus, qui attendent des entrées particulières selon ce que l'utilisateur veut faire.

Détails de la fonction assemblification : Cette fonction demande un numéro de vélo pour notifier son assemblage et mettre à jour le stock du vélo ainsi que les stocks de toutes les pièces le composant tout en prenant soin de vérifier que le stock de pièces est suffisant.

Détails de la classe Connexion : la classe contient l'intégralité de informations de connexion à la base MySQL et ses méthodes permettent de :

-Faire une requête et obtenir en retour un tableau de string avec la fonction requête(string)

-Faire une requête et avoir un affichage automatique, surtout utilisé dans le module statistique avec la fonction requeteA(void)

-Modifier les tables en normalisant la syntaxe à l'aide de fonction spécifiques à chaque type d'opération. La fonction mod prend donc en argument un type (ajout, suppression ou mise à jour), le tableau sur lequel effectuer la modification ainsi que deux arguments : val pour les valeurs si l'on veut update ou add et cond pour imposer des conditions sur les updates et delete. En fonction de la valeur de la string type un formatage spécifique est effectué puis la requête est envoyée.

Base de données SQL :

La base de données contenant peu de clients déjà enregistrés, nous vous conseillons d'entrer quelques commandes de clients particuliers et de vélo en entrant les input suivants :

"Voulez vous modifier des données" => oui

Entrer le type d'élément à modifier => commande

Sélectionner type d'opération => Add

Commandes de pièces => oui

Numéro de pièce => S88

Quantité voulue => 20

Commander une autre pièce => non

Commande de vélos? => oui

Numéro du modèle => 102

Quantité voulue => 2

Commander un autre vélo => non

ENTREE POUR CONTINUER

Module statistiques :

Entrer le vélo demandé => tout

ENTREE POUR CONTINUER

Entrer la pièce demandée => tout

Continuer à faire entrer pour aller jusqu'à la fin