

Projet BDD : veloMax

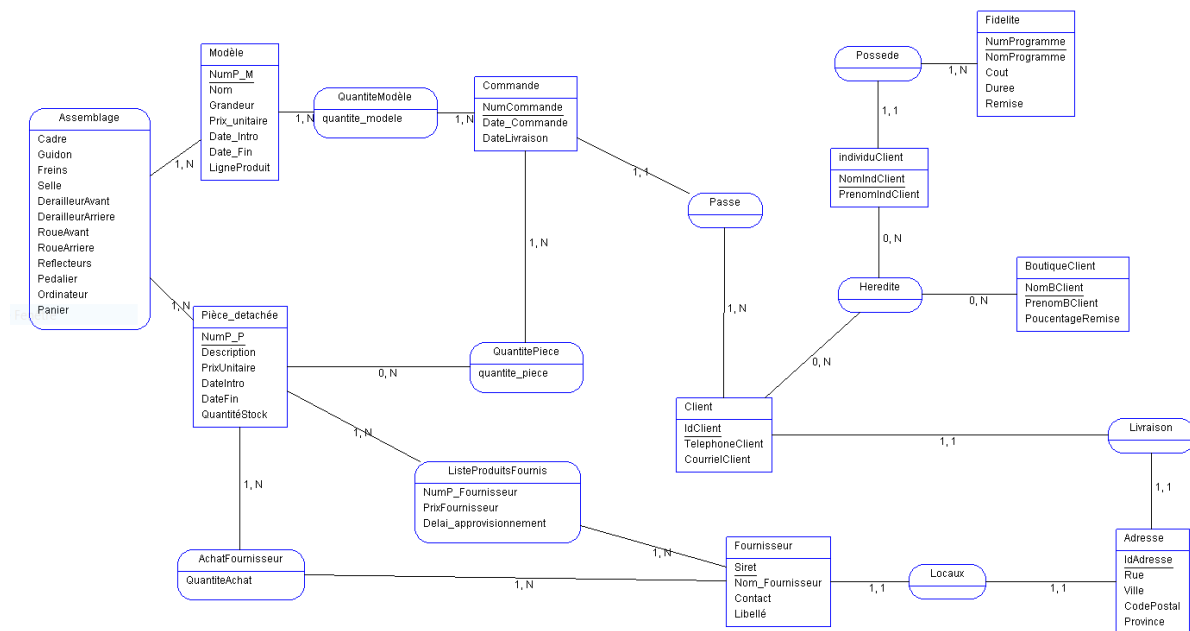
Afin de mettre en pratique nos compétences apprises lors du module BDD & Interopérabilité, nous avons réalisé un projet réaliste simulant la gestion de commandes d'une compagnie qui vend des bicyclettes et des pièces de rechange pour les bicyclettes.

I. Un schéma E/A complexe

Notre schéma E/A est assez complexe de par la formulation du problème : les pièces détachées ont une ID pour la boutique et une ID pour les fournisseurs, les modèles ne sont que des groupes de pièces détachés. Nous avons aussi une gestion des commandes avec ajout de pièces ou de vélos. Chaque commande est reliée à un client qui peut-être un individu ou une boutique et qui possède une adresse.

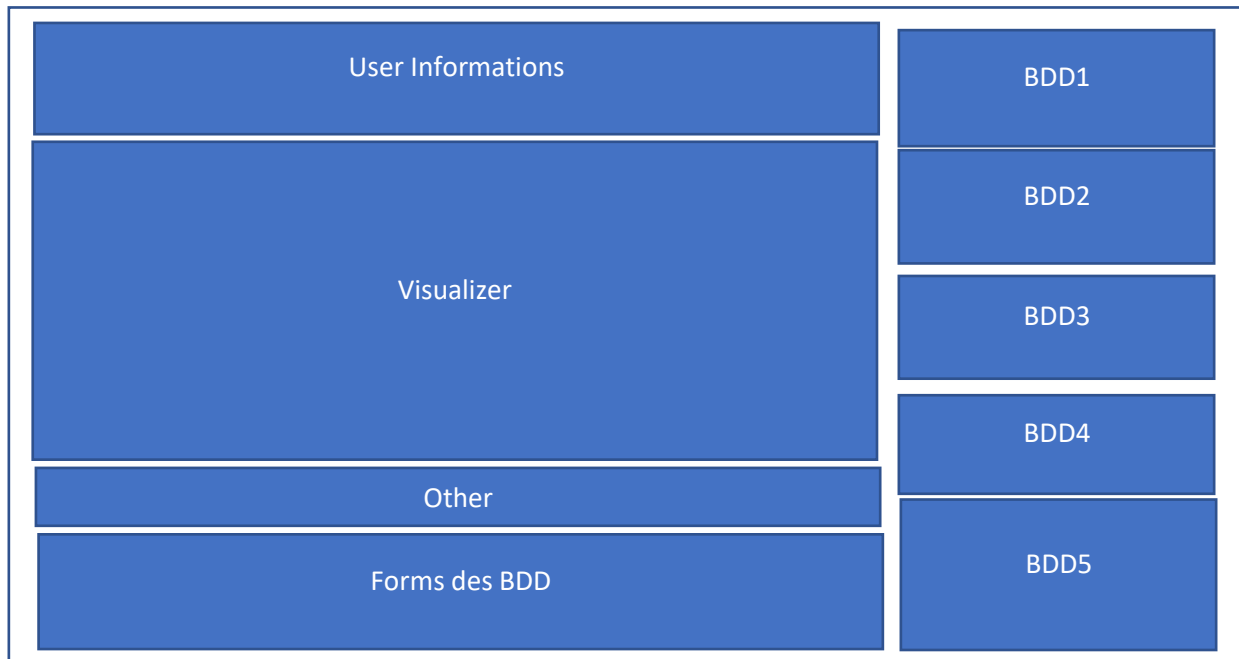
Notre boutique devait également être capable de se réapprovisionner en faisant des achats de pièces auprès de différents fournisseurs.

Nous avons aussi connecté les modèles de vélos avec les pièces qui les composent grâce à une table assemblage.



II. Application WPF

a. Réflexion UI



b. DataGrid

Un visualizer de données assez pratique qui permet de visualiser une table facilement. Nous avons créé une fonction qui prend en entrée la commande sql et qui rentre le tableau de mysql dans le datagrid.

Le datagrid permet aussi de trier interactivement sur n'importe quelle colonne.

c. ComboBox : Jouer sur la personnalisation encadrée

L'objet ComboBox est un menu déroulant. L'avantage de ce dernier est que les possibilités de valeur sont direct proposées, l'erreur utilisateur est réduite. On peut accéder

III. REX : Un travail chronophage pour des tâches pas si optimisées

a. C# / SQL (pas si rentable dans notre cas)

Le C# est un langage haut niveau que nous maîtrisons depuis notre première année. Pour autant, son implémentation dans un projet SQL, réécrire les select from sur C# au lieu de le faire sur le WorkBench est finalement assez « bas niveau »

IV. Tableau des tâches

<u>Tâche</u>	<u>Réalisée</u>
Diagramme E/A	Oui, à l'aide d'Analyse SI
Création de toutes les tables	Oui, avec visualisation, insertion, modification et suppression
WPF	Oui avec datagrid (c'est vraiment très joli)
Gestion du stock	Oui, gestion du stock des pièces avec incrémentation lors d'achat auprès de fournisseur et décrémentation lors de commande client
Affichage des clients avec leur programme de fidélité	Oui
Meilleur client	Non, facilement ajoutable mais nous n'avons pas eu le temps
Requête avec auto-jointure	Oui pour les affichages de clients ou de fournisseurs avec leur adresse ou les clients, leurs commandes, les pièces et les modèles, ect
Exportation en XML	Oui, une exportation de n'importe quelle table affichée est disponible à l'aide d'un bouton facilement utilisable
Mode démo	Non (pas utile avec un WPF) Ps : c'est nous la démo
Liste des pièces dont le stock est inférieur ou égal à 2	Oui
Pièces fournis par les fournisseurs	Oui

V. Exemples de requêtes SQL

Ci-dessous quelque requête SQL brute (même si peu d'intérêt car non représentatif de l'intégration de ces dernières dans C# avec les entrées nécessaires) :

Exemple création de table :

```
• drop table if exists Piece_detachee;  
• create table if not exists Piece_detachee(  
    NumP_P varchar(20) primary key,  
    Description varchar(50),  
    PrixUnitaire float,  
    QuantiteStock int,  
    DateIntro datetime,  
    DateFin datetime  
);
```

Exemple d'insertion dans une table :

```
insert into velomax.piece_detachee (NumP_P,Description,PrixUnitaire,QuantiteStock,DateIntro,DateFin) values  
("R44","Roue Avant",37,4,"2022-05-05","2032-05-05");
```

Exemple de selection des vélos commandé pour une commande en particulier :

```
select * from QuantiteModeleC natural join commande where NumCommande=1;
```

Exemple de suppression d'un élément :

```
delete from Client where IdClient = 1;
```

Exemple de modification d'un élément :

```
update Fournisseur set Nom_Fournisseur = "DeVinci", Contact = "test" where Siret = "552";
```