



Programmation Orientée

Objet avancé

Surquin Nathan, Lambert Antoine



1.Enoncé	3
1.1 Présentation du domaine de l'application	3
2. Base de donnée	4
2.1 Schéma entités-associations	4
2.2 Schéma logique de tables	5
2.3 Documentation des tables	6
2.3.1 Table "Client"	6
2.3.2 Table "HouseLocation"	6
2.3.3 Table "Locality"	6
2.3.4 Table "Order"	6
2.3.5 Table "Component"	6
2.3.6 Table "Beer"	6
2.4 Fonctionnalités de l'application	7
2.4.1 CRUD	7
Les fonctionnalités de création, listing, modification et suppression seront appliquées sur la table "Order".	7
2.4.2 Recherches	7
Recherche numéro : 1	7
Recherche numéro : 2	7
Recherche numéro : 3	8
2.4.3 Tâche métier	9

1. Enoncé

1.1 Présentation du domaine de l'application

L'application développée dans le cadre de ce projet est destinée à une brasserie Namuroise.

Cette application sera utilisée en interne. Le but principal de l'implémentation du logiciel est de permettre une meilleure gestion des stocks et une organisation plus efficace des commandes.

Une commande s'effectue par le biais d'un appel à la brasserie ou directement au comptoir de celle-ci. La commande est alors encodée et envoyée au gestionnaire des stocks.

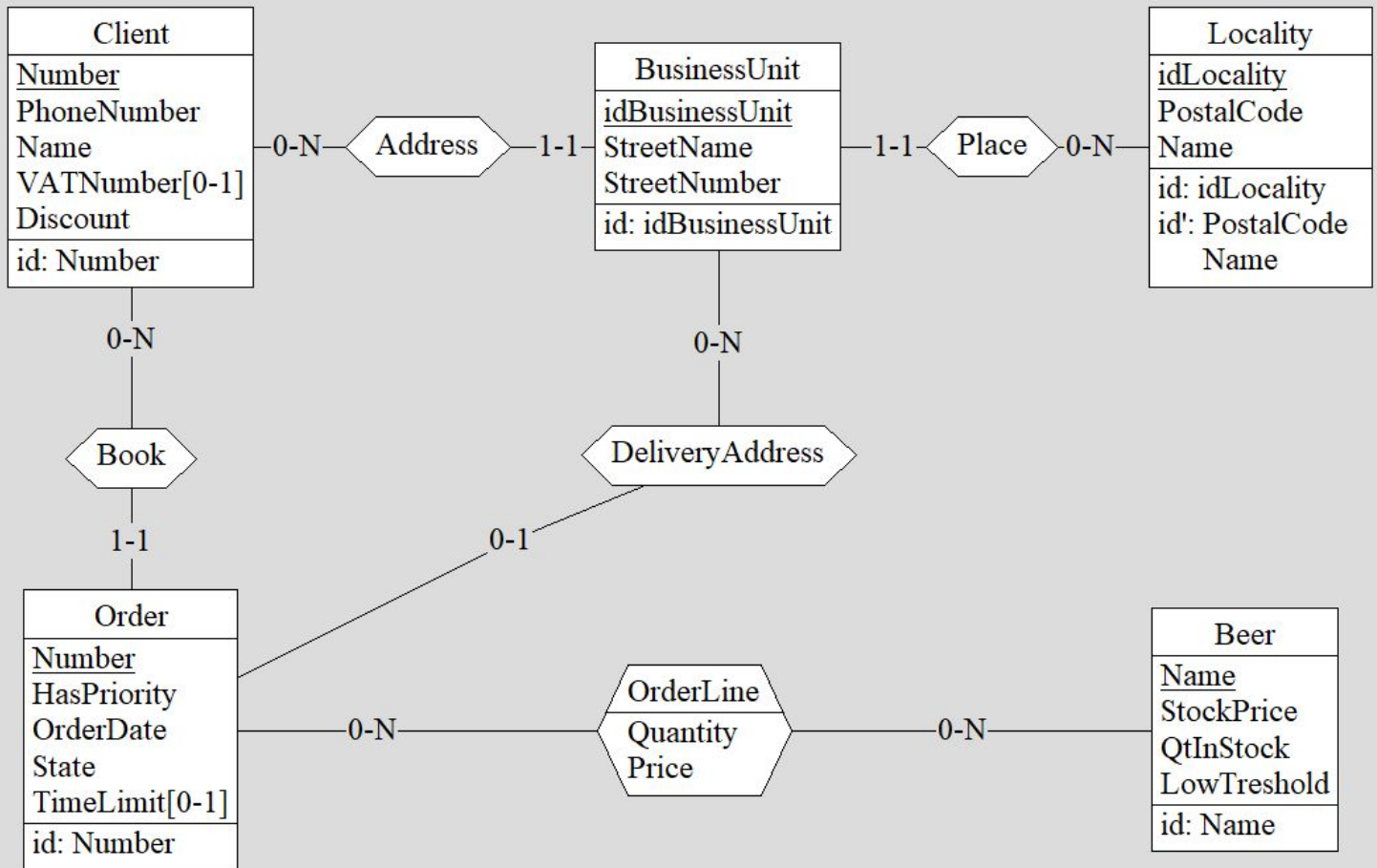
Le gestionnaire des stocks est en charge de la préparation des commandes et des commandes aux fournisseurs.

Une fois que la commande est préparée, le livreur prépare sa livraison en vérifiant si il peut coupler plusieurs livraisons simultanément.

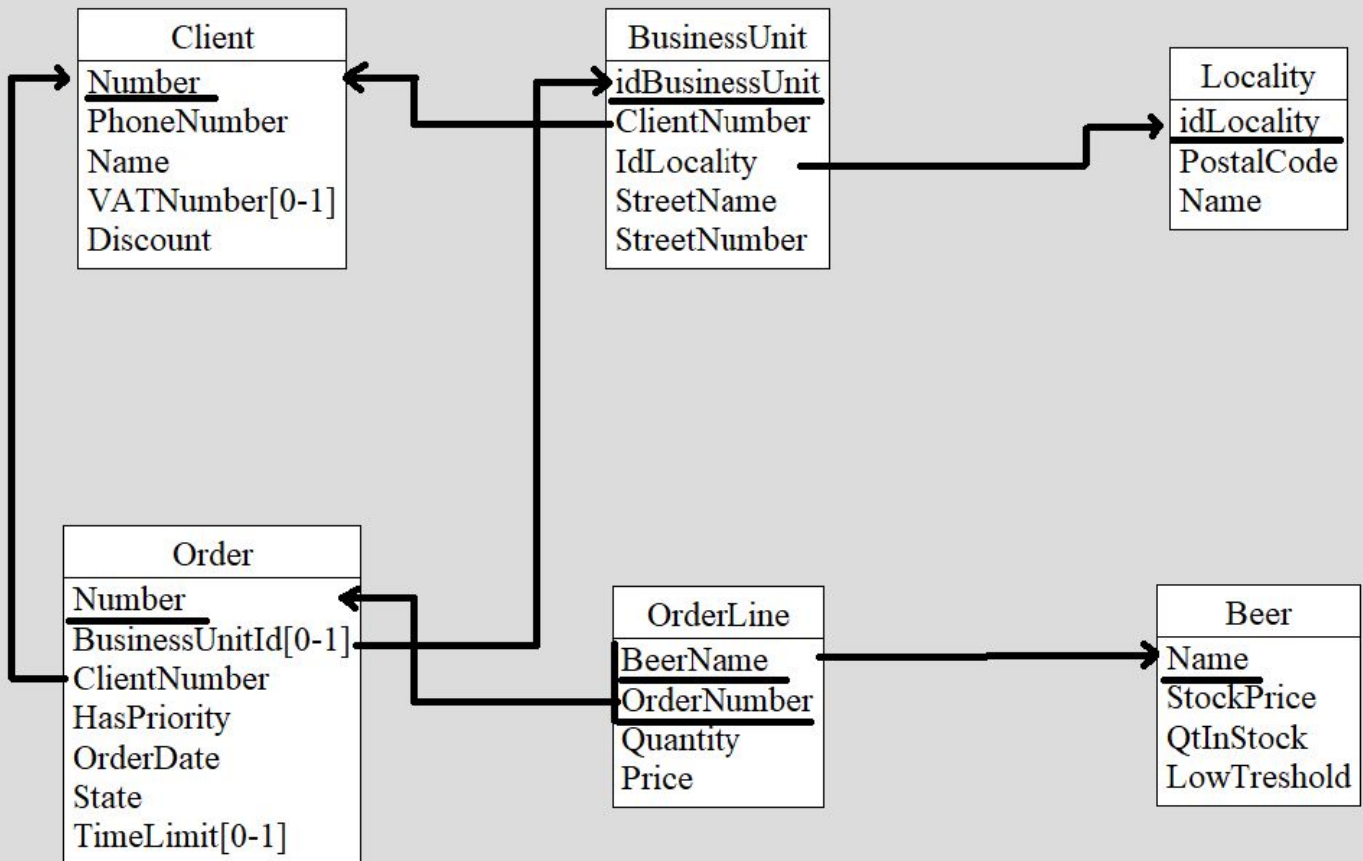
Une fois livrée, la commande donne lieu à un paiement dont la validation sera effectuée par la partie comptable de l'entreprise.

2. Base de donnée

2.1 Schéma entités-associations



2.2 Schéma logique de tables



2.3 Documentation des tables

2.3.1 Table "Client"

Chaque client de la brasserie est identifié par un numéro unique au sein de la base de donnée, un numéro de téléphone, un nom, un numéro de TVA (facultatif), un entier représentant un pourcentage de réduction auquel il a droit sur ses achats, et facultativement, un nom et numéro de rue (un client peut passer commande sans donner son adresse s'il ne désire pas de livraison).

2.3.2 Table "HouseLocation"

Chaque localisation de maison est identifié par 3 clés étrangères. Un numéro de client, un code postal et un nom de localité.

2.3.3 Table "Locality"

Chaque localité est identifiée par un code postal et un nom de localité.

2.3.4 Table "Order"

Chaque commande est décrite par un numéro unique au sein de la base de donnée, un numéro de client qui référence le client concerné par la commande, un booléen qui indique si la commande est à livrer et enfin la date à laquelle la commande a été prise.

2.3.5 Table "Component"

Chaque composant comporte 2 clés étrangères. La première est le numéro de la commande et la deuxième est le nom de la bière concernée. Une troisième colonne indique le nombre de bière de chaque type présent dans la commande.

2.3.6 Table "Beer"

Chaque type de bière est identifié par un nom, possède un prix, a une quantité en stock connue, et a un seuil de quantité minimale qui, quand il est atteint, provoque un avertissement auprès du gestionnaire de stock pour qu'il puisse passer une commande chez le fournisseur.

2.4 Fonctionnalités de l'application

2.4.1 CRUD

Les fonctionnalités de création, listing, modification et suppression seront appliquées sur la table "Order".

2.4.2 Recherches

Recherche numéro : 1

Titre de la recherche : Lister des commandes par date ou état

Objectif de la recherche : Lister des commandes et leurs composants pour deux dates données ou pour un état donné (commandé, préparé, livré ou payé).

Jointure entre les tables suivantes :

Table 1 : Order Table 2 : OrderLine Table 3 : Beer

Entrées :

- Deux dates : JSpinner
- Etat : JRadioButton

Sorties :

- Numéro de commande : Table "Order"
- Date de commande : Table "Order"
- Quantité : Table "OrderLine"
- Prix réel : Table "OrderLine"
- Nom bière : Table "Beer"
- Prix catalogue : Table "Beer"

Recherche numéro : 2

Titre de la recherche : Recherche d'une livraison à effectuer

Objectif de la recherche : Lister commandes à livrer une localité donnée

Jointure entre les tables suivantes :

Table 1: Order, Table 2 : Client, Table 3 : BusinessUnit, Table 4 : Locality

Entrées :

- Code postal : Jcombobox proposant les localités associées à ce code postal
- Localité : Jcombobox proposant les code postaux associés à cette localité

Sorties :

- Nom client : Table "Client"
- Numéro de la commande : Table "Order"
- Etat de la commande: Table "Order"
- Commande prioritaire : Table "Order"
- Numéro rue : Table "BusinessUnit"
- Nom rue : Table "BusinessUnit"

→ Trié par état

Recherche numéro : 3

Titre de la recherche : Commandes d'un client

Objectif de la recherche : Trouver toutes les commandes d'un client

Jointure entre les tables suivantes :

Table 1: Order, Table 2 : Client, Table 3 : OrderLine, Table 4 : Beer

Entrées :

- Nom (Numéro) de client : Jcombobox Proposant des clients et leurs code d'identification

Sorties :

- Adresse Livraison (N°, Nom rue, code postal localité) : Table "BusinessUnit"
- Nom bière : Table "Beer"
- Quantité : Table "OrderLine"

→ Trié par commande et par adresses

→ Jointure externe

2.4.3 Tâche métier

La tâche métier aura pour but de créer une interface pour le préparateur de commande et pour le livreur. Cette interface aidera à la préparation des commandes et aidera le livreur à préparer sa livraison (par exemple : trier les livraisons par localité).