

Province de Namur  
Haute École de Namur-Liège-Luxembourg  
Campus IESN  
Rue Joseph Calozet 19  
5000 Namur

Programmation orientée-objet  
avancée (Java)

Dossier sur Libio-pizza

Présenté par :

Gheerts Alexandre

Lempereur Pierre-Henri

Inscrit dans la section :   
Informatique de gestion  
Bloc 2 – Groupe A & B2

Année scolaire 2018-2019

Table des matières

[Introduction 3](#_Toc17645730)

[Domaine d’application 4](#_Toc17645731)

[Schéma conceptuel (entités-associations) de la base de données 5](#_Toc17645732)

[Schéma logique des tables 6](#_Toc17645733)

[Fonctionnalités du programme 10](#_Toc17645734)

[Thread 12](#_Toc17645735)

[Tâche métier 12](#_Toc17645736)

[Tâche métier supplémentaire : 13](#_Toc17645737)

# Introduction

Ce dossier contient l’énoncé nécessaire pour notre projet en programmation orientée objet avancée.

Notre domaine d’application se porte sur une pizzeria qui effectue des livraisons.

Ce dossier contient :

* Une présentation du domaine d’application
* Un schéma conceptuel (entités-associations) de la base de données
* Un schéma logique des tables
* Une documentation de ces tables
* Une description des fonctionnalités du programme final
* Un bref résumé des tâches métier à implémenter

# Domaine d’application

Libio-pizza est un service de livraison de pizzas. Celui-ci permet la commande de pizzas en tant que client inscrit ou non-inscrit.

Le public ciblé est, tout particulier qui désire se faire livrer des pizzas.

Après analyse, nous avons noté le fonctionnement de Libio-pizza ;

* La création d’un bon de commande
* La suppression d’un bon de commande
* La modification d’un bon de commande
* La recherche d’un ou plusieurs bon(s) de commande
* Le listing de tous les bons de commande
* La recherche d’information à propos d’un client
* La recherche d’information à propos d’un menu de pizza

La commande de pizza permet de commander une pizza déjà existante dans la base de données via un formulaire.

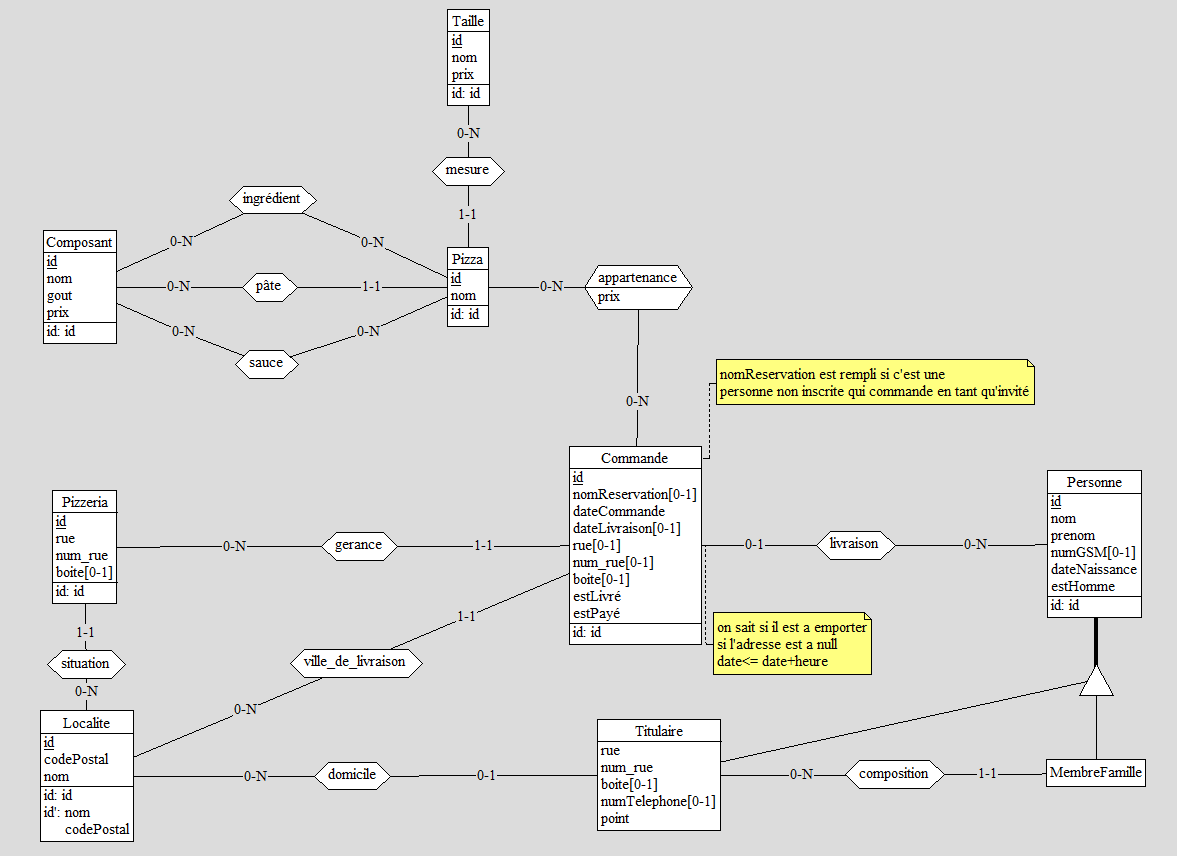
Il en va de même pour la modification où l’utilisateur peut choisir un client qui a déjà passé commande pour pouvoir modifier cette commande.

La recherche d’un ou plusieurs bon(s) de commande se fera en fonction de la date et de s’il l’on souhaite afficher les bons déjà payés ou non.

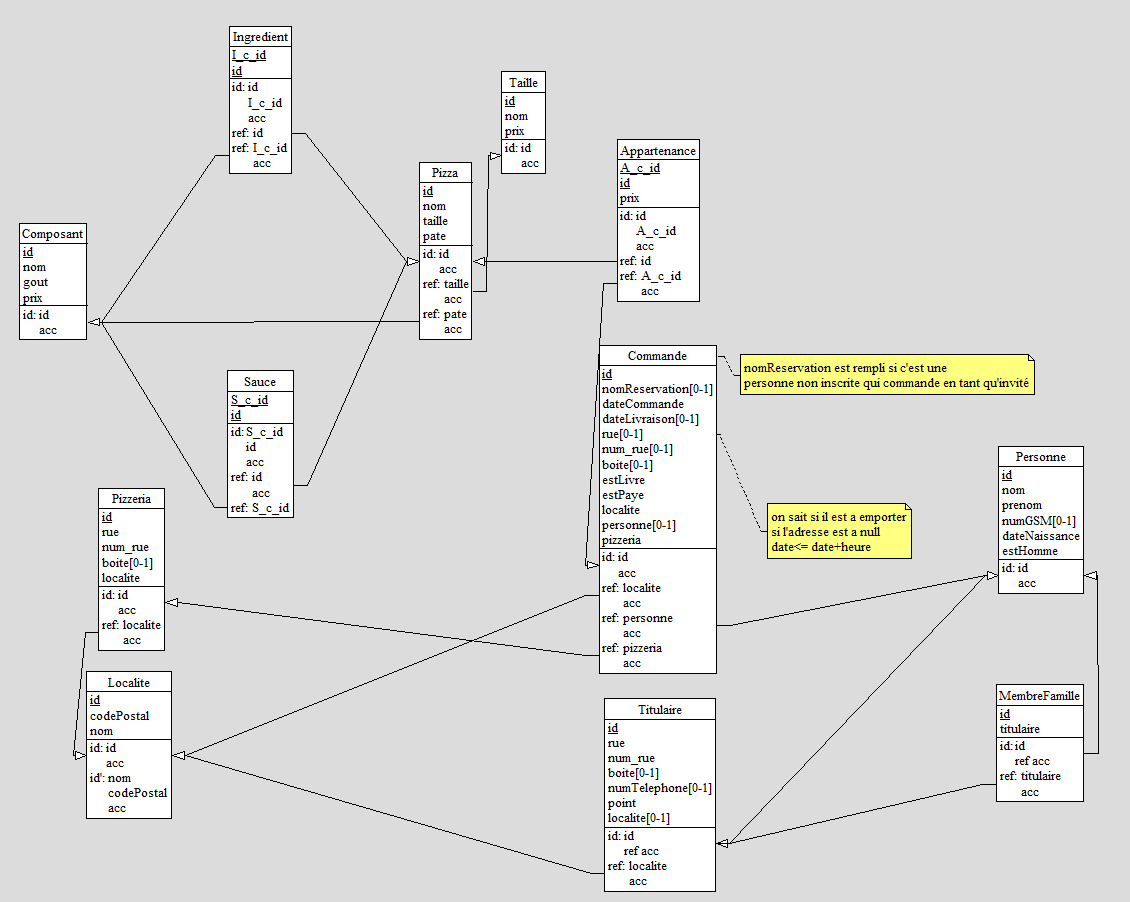
La recherche d’un client ne demandera que de sélectionner le client à qui l’on souhaite afficher les informations.

La recherche du menu d’une pizza permettra d’obtenir divers informations à propos d’une pizza (son prix / ses composants / …) en fonction de trois critères d’entrées : Le prix, les composants que contient cette pizza et sa taille.

# Schéma conceptuel (entités-associations) de la base de données



# Schéma logique des tables



Documentation des tables

Table « composant »

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la colonne | Type de la colonne | Description | Contraintes |
| id | int | Identifiant du composant | Clé primaire auto-incrément |
| nom | varchar | Nom du composant |  |
| gout | int | Enumération sur les gouts disponibles (salé / sucré / amer/ …) |  |
| prix | double | Prix à l’unité du composant |  |

Un composant peut-être de 3 types : un ingrédient, une pâte ou une sauce.

Table « taille »

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la colonne | Type de la colonne | Description | Contraintes |
| id | int | Identifiant du composant | Clé primaire auto-incrément |
| nom | varchar | Taille de la pizza (petit/moyen/grand/…) |  |
| prix | double | Prix fixe sur la taille |  |

Table « pizza »

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la colonne | Type de la colonne | Description | Contraintes |
| id | int | Identifiant de la pizza | Clé primaire auto-incrément |
| nom | varchar | Nom de la pizza |  |
| pâte | int | Clé étrangère correspondant à la pâte de la pizza | Clé étrangère |
| taille | int | Clé étrangère correspondant à la taille de la pizza | Clé étrangère |

Table « pizzeria »

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la colonne | Type de la colonne | Description | Contraintes |
| id | int | Identifiant de la pizzeria | Clé primaire auto-incrément |
| rue | varchar | Nom de la rue |  |
| num\_rue | varchar | Numéro de la rue |  |
| boite | varchar | Numéro de la boite | Optionnel |
| localite | int | Clé étrangère correspondant à la localité de la pizzeria | Clé étrangère |

Table « localite »

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la colonne | Type de la colonne | Description | Contraintes |
| id | int | Identifiant de la localité | Clé primaire auto-incrément |
| code\_postal | int | Code postal |  |
| nom | varchar | Nom de la localité |  |

Table « commande »

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la colonne | Type de la colonne | Description | Contraintes |
| id | int | Identifiant de la commande | Clé primaire auto-incrément |
| nom\_reservation | varchar | Nom de la réservation | Optionnel |
| date\_commande | date | Date où la commande a été enregistrée |  |
| date\_livraison | date | Date où la commande doit être livrée | Optionnel |
| rue | varchar | Rue de la livraison | Optionnel |
| num\_rue | varchar | Numéro de la rue de la livraison | Optionnel |
| boite | varchar | Boite de la livraison | Optionnel |
| est\_livré | boolean | Indique si la commande a été livrée |  |
| est\_payé | boolean | Indique si la commande a été payée |  |
| pizzeria | int | Clé étrangère représentant la pizzeria à qui le bon est destiné | Clé étrangère |
| localite | int | Clé étrangère correspondant à la localité indiquée sur le bon de commande | Clé étrangère |
| personne |  | Clé étrangère optionnelle qui pointe vers la personne qui a commandé | Clé étrangère et optionnelle |

« nom\_reservation » est rempli si c’est une personne non inscrite qui commande (En tant qu’invité donc).

Si l’adresse est à « null » on sait que la commande n’est pas à emporter.

Les dates contiennent la date et l’heure.

Table « personne »

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la colonne | Type de la colonne | Description | Contraintes |
| id | int | Identifiant de la personne | Clé primaire auto-increment |
| nom | varchar | Nom de la personne |  |
| prenom | varchar | Prénom de la personne |  |
| num\_gsm | int | Numéro de GSM de la personne | Optionnel |
| date\_naissance | date | Date de naissance de la personne |  |
| est\_homme | boolean | Indique si la personne est un homme ou non |  |

La table « personne » est une table « mère » d’héritage

Table « titulaire »

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la colonne | Type de la colonne | Description | Contraintes |
| id | int | Identifiant du titulaire (le même que celui de la table personne) | Clé primaire |
| rue | varchar | Rue du domicile du titulaire |  |
| num\_rue | varchar | Numéro du domicile du titulaire |  |
| boite | varchar | Boite du domicile du titulaire | Optionnel |
| num\_telephone | int | Numéro de téléphone fixe du titulaire | Optionnel |
| point | int | Point(s) actuel(s) du titulaire |  |
| localite | int | Clé étrangère représentant la localité du titulaire | Clé étrangère |

L’id du titulaire est l’id de son parent (personne).

Table « membre\_famille »

|  |  |  |  |
| --- | --- | --- | --- |
| Nom de la colonne | Type de la colonne | Description | Contraintes |
| id | int | Identifiant du membre de la famille (qui est le même que dans sa table parent « personne ») | Clé primaire |
| titulaire | int | Clé étrangère représentant le titulaire du membre | Clé étrangère |

L’id du membreFamille est l’id de son parent (personne).

# Fonctionnalités du programme

Le programme contiendra les options suivantes :

* Insertion :
  + Insertion d’un bon de commande
* Modification :
  + Pour la modification, nous utiliserons en entrée l’id et/ou nom et/ou date et/ou livré
* Suppression :
  + Pour la suppression, nous utiliserons en entrée l’id et/ou nom et/ou date et/ou livré
* Listing.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Titre de la recherche : présenter un menu des pizzas**  **Objectif de la recherche : afficher un menu composé de pizzas ainsi que leurs ingrédients, à partir d'un prix minimum et/ou d'un composant et/ou d'une taille**  **Jointure entre les tables suivantes :**   * Table 1 : Composant * Table 2 : Pizza * Table 3 : Taille   **Entrées :**   |  |  | | --- | --- | | **Critère de recherche** | **Format** | | Prix | JSpinner | | Composant | JComboBox | | Taille | JComboBox |   **Sorties :**   |  |  |  | | --- | --- | --- | | NomPizza |  | Pizza | | Composants | Provenant de la table | Composant | | Tailles |  | Taille | | prix | Calculé à partir de | Pizza | |  |  |  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Titre de la recherche : Affichage des bons de commande**  **Objectif de la recherche : Afficher et sortir tous les bons de commande, établis entre deux dates données**  **Jointure entre les tables suivantes :**   * Table 1: Commande * Table 2: Pizza * Table 3: Taille * Table 4: Composant * Table 5: Localité * Table 6: Pizzeria * Table 7: Personne * Table 8: Titulaire   **Entrées :**   |  |  | | --- | --- | | **Critère de recherche** | **Format** | | Traités, non-traité, tous | jRadioButton | | Jour/mois/année début | Jspinner /JComboBox / JSpinner | | Jour/mois/année fin | Jspinner /JComboBox / JSpinner |   **Sorties :**   |  |  |  | | --- | --- | --- | | Liste de pizza |  | pizza | | Nom de réservation, dateCommande, dateLivraison, etatLivraison, etatPayement |  | commande | | adresse | Provenant de la table | (Commande ou titulaire) et localité | | Taille de pizza |  | taille | | Pizzeria d’origine |  | pizzeria | |  |  |  | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Titre de la recherche : Afficher les infos relatives à un client**  **Objectif de la recherche : Afficher et sortir tous les informations du client que l’on a sélectionné.**  **Jointure entre les tables suivantes :**   * Table 1: Personne (2x) * Table 2: Localite (2x)   **Entrées :**   |  |  | | --- | --- | | **Critère de recherche** | **Format** | | Client | JComboBox |   **Sorties :**   |  |  |  | | --- | --- | --- | | Nom, Prenom, est un homme ?, GSM, Telephone, Rue, N de rue, Boite, Code Postal, Ville, Nb Point, typePersonne | Provenant de la table | Personne | |  | Particularité : Si une personne est un membre de famille, on va chercher les infos manquantes dans le titulaire |  | |

# Thread

Le thread est un logo qui rebondi sur la page d’accueil

# Tâche métier

Afficher le ticket de caisse d’un bon de commande et valider l’état de payement

* Infos sur les pizzas
  + Nom
  + Prix / unité
* Les points ajoutés (en fonction de la commande)
* TVA
* prix HTVA
* Prix TVAC
* Les informations relatives à la personne qui commande
  + Nom
  + Prénom
  + Points actuels (si inscrit dans la BD)
  + Adresse de livraison
* Informations relative à la pizzeria
  + Adresse complète
    - Numéro
    - Rue
    - Code postal
    - Ville

# Tâche métier supplémentaire :

Afficher le nom, prénom et la date de commande des clients ayant commandé une pizza donnée entre 2 dates données.

* Entrées :
  + Pizza
  + Date de début
  + Date de fin
* Sortie :

Liste des clients avec leur nom, prénom et date de commande

Afficher pour une pizzéria choisie le prix moyen des commandes effectuées entre 2 dates données.

* Entrées :
  + Pizzeria
  + Date de début
  + Date de fin
* Sortie :

Prix moyen des commandes