**FORMATION DA JAVA**

OPEN CLASSROOMS – 2019

Étudiant : Éric AUBRUN

Projet 5 : Concevez la solution technique d’un système de gestion de pizzeria



https://github.com/RicoBSJ/Projet5\_OCR\_Rico\_2019.git

**SOMMAIRE**

[INTRODUCTION 3](#_Toc25136845)

[RAPPEL DE LA COMMANDE DU CLIENT 3](#_Toc25136846)

[1. Le contexte 3](#_Toc25136847)

[2. Les besoins du Client 3](#_Toc25136848)

[I. LE DOMAINE FONCTIONNEL 4](#_Toc25136849)

[1. Description 4](#_Toc25136850)

[2. Le diagramme de classe d’OC Pizza 4](#_Toc25136851)

[3. Le descriptif des classes et de leurs interactions 4](#_Toc25136852)

[ La classe « Commande » 5](#_Toc25136853)

[ Les classes « Client » et « Adresse » 5](#_Toc25136854)

[ Les classes « Produit », « Ingedient » et l’énumération « EtatCommande » 6](#_Toc25136855)

[ Les classes « Employe », « Livraison », « ProduitsStock » et l’énumération « Poste » 6](#_Toc25136856)

[ Les classes « Paiement », « Facture » et l’énumération « EtatDuPaiement » 7](#_Toc25136857)

[II. LE MODELE PHYSIQUE DE DONNEES 8](#_Toc25136858)

[1. La relation entre les classes « client » et « commande » 8](#_Toc25136859)

[2. La relation entre les classes « client » et « adresse » 9](#_Toc25136860)

[3. La relation entre les classes « adresse » et « commande » 9](#_Toc25136861)

[4. La relation entre les classes « commande » et « livraison », « commande » et « paiement » et « paiement » et « facture » 9](#_Toc25136862)

[5. Les spécificités des relations entre les classes « commande », « produit » et « ingredient » 10](#_Toc25136863)

[6. La relation entre les classes « etatCommande » et « commande » 11](#_Toc25136864)

[7. La relation entre les classes « commande », « employe » et « poste » 11](#_Toc25136865)

[8. La relation entre les classes « employe » et « livraison » 11](#_Toc25136866)

[9. La classe « produitsStock » 12](#_Toc25136867)

[III. LES COMPOSANTS DU SYSTEME 13](#_Toc25136868)

[1. Le diagramme de composants 13](#_Toc25136869)

[IV. L’ARCHITECTURE DE DEPLOIEMENT 14](#_Toc25136870)

[1. Le diagramme de déploiement 14](#_Toc25136871)

# INTRODUCTION

En tant qu’analyste-programmeur, mon travail vise à définir le domaine fonctionnel et à concevoir l’architecture technique de la solution répondant aux besoins du client, c’est-à-dire :

* Modéliser les objets du domaine fonctionnel ;
* Identifier les différents éléments composant le système à mettre en place et leurs interactions ;
* Décrire le déploiement des différents composants que vous envisagez ;
* Élaborer le schéma de la ou des bases de données que vous comptez créer.

Nous conclurons ce document par les solutions fonctionnelles et techniques retenues.

# RAPPEL DE LA COMMANDE DU CLIENT

## Le contexte

« OC Pizza » est un jeune groupe de pizzeria en plein essor spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et prévoit d’en ouvrir au moins 3 de plus d’ici 6 mois.

## Les besoins du Client

* Être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
* Suivre en temps réel les commandes passées, en préparation et en livraison ;
* Suivre en temps réel le stock d’ingrédients restants pour savoir quelles pizzas peuvent encore être réalisées ;
* Proposer un site Internet pour que les Clientes puissent :
  + Passer leurs commandes, en plus de la prise de commande par téléphone ou sur place ;
  + Payer en ligne leur commande s’ils le souhaitent – sinon, ils paieront directement à la livraison ;
  + Modifier ou annuler leur commande tant que celle-ci n’a pas été préparée.
* Proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza.

# LE DOMAINE FONCTIONNEL

## Description

Le domaine fonctionnel est représenté par le diagramme de classe. Ce dernier vise à relier les classes entre elles, de manière à développer une application dynamique et en capacité de satisfaire la commande client.

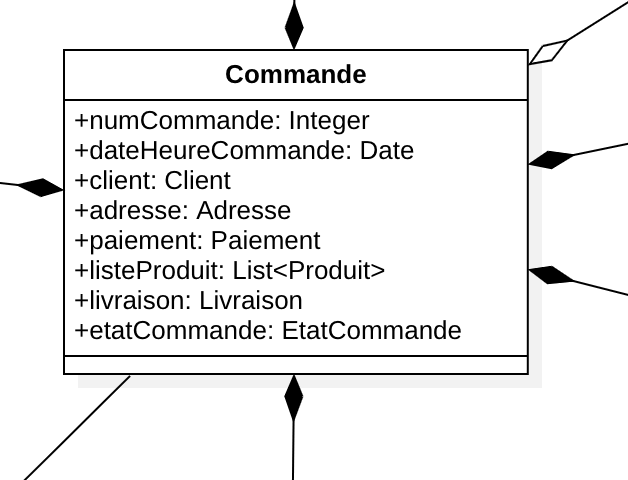
## Le diagramme de classe d’OC Pizza

Une image contenant texte, carte

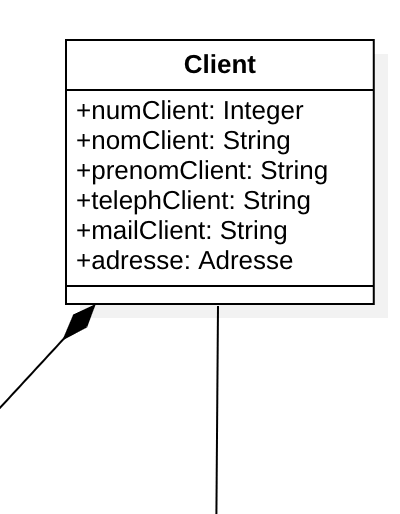
Description générée automatiquement

## Le descriptif des classes et de leurs interactions

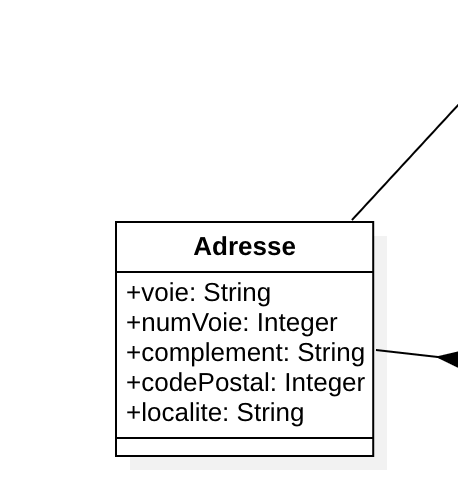
### La classe « Commande »

La classe « Commande » est « hyper sollicitée ». Elle est en fait la pierre angulaire de l’application : Les cinq clés étrangères « client », « adresse », « paiement », « livraison » et « etatCommande » permettent respectivement, de relier la commande au client l’ayant produite, la commande à l’adresse où doivent être livrés les produits, la commande en cours au paiement que celle-ci génère, la commande à sa livraison et enfin, la commande à son état, selon qu’elle est acceptée, en préparation, prête, affectée à un livreur ou bien en cours de livraison.

### Les classes « Client » et « Adresse »

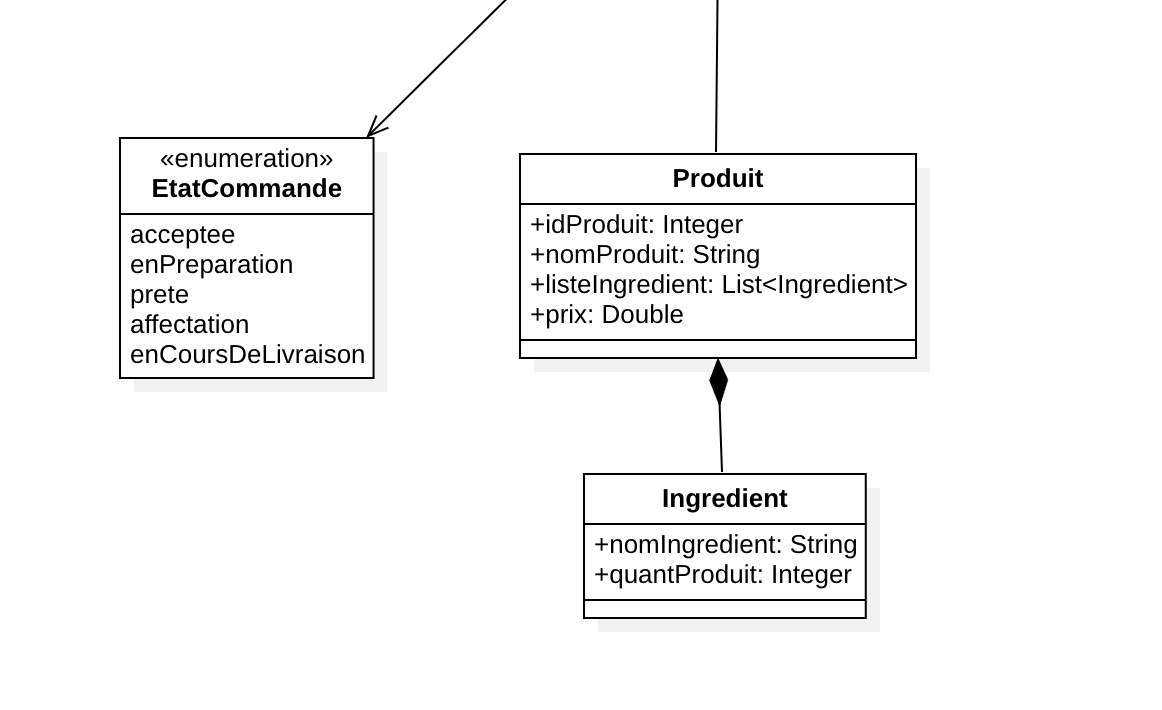


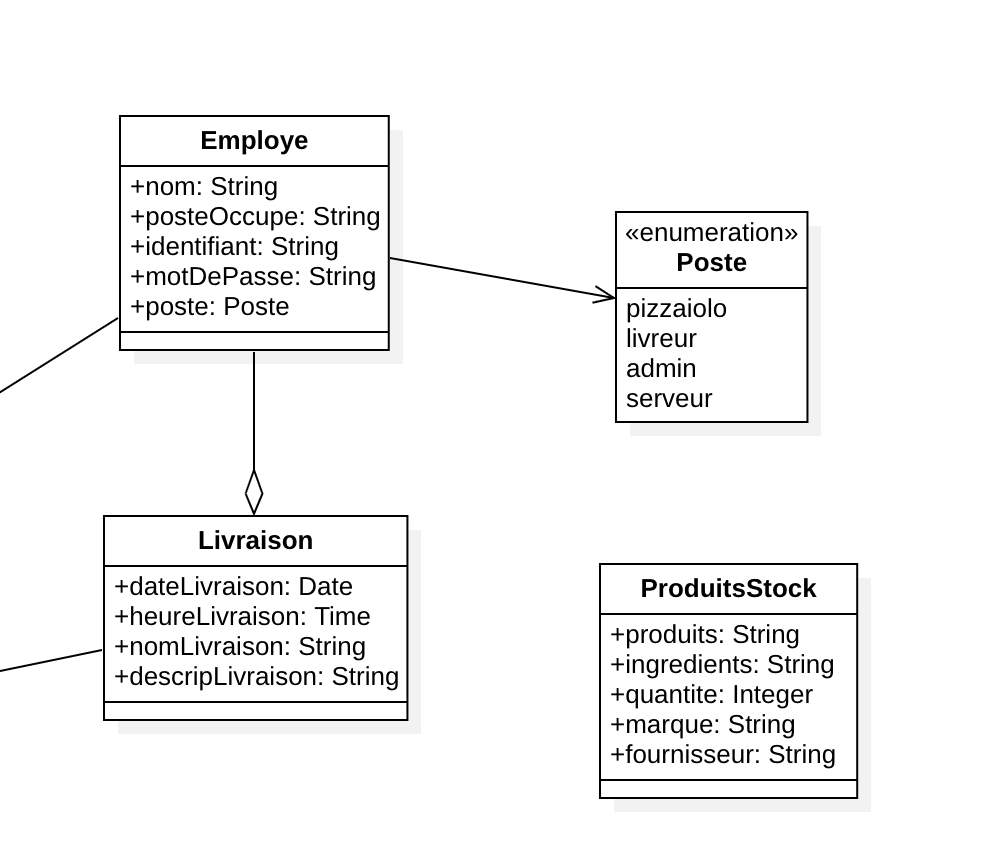
La classe « Client » répertorie : le numéro de client (attribut auto-incrémenté), son nom, son prénom, son téléphone ainsi que son adresse mail. La clé étrangère « adresse » permet de relier la classe « Client » à la classe « Adresse », de sorte que chaque client puisse avoir une adresse de livraison définie.



La classe « Adresse » contient pour chaque client une adresse spécifique, détaillée et unique (« one to one »).

### Les classes « Produit », « Ingedient » et l’énumération « EtatCommande »



Les classes « Produit » et « Ingredient » permettent d’enregistrer les produits et ingrédients que le client choisit. L’énumération « EtatCommande » paramètre les différents changements d’états de la commande client.

### Les classes « Employe », « Livraison », « ProduitsStock » et l’énumération « Poste »

L’énumération « Poste » définit l’employé en charge de la commande et/ou de sa livraison. La classe « Employe » attribue pour chaque employé le nom, le poste occupé, l’identifiant ainsi que le mot de passe : L’ensemble de ces éléments permet, dès la connexion à l’application, une authentification. La classe « Livraison » enregistre la date, l’heure, le nom rattaché à la livraison ainsi que son descriptif. La classe « ProduitsStock » est plus particulièrement utilisée par l’administrateur dans le but d’effectuer une gestion de stock en cohérence avec les besoins et de pouvoir lancer une commande auprès des fournisseurs.

### Les classes « Paiement », « Facture » et l’énumération « EtatDuPaiement »

La classe « Paiement » finalise le paiement client dès lors que les différentes étapes du paiement sont validées. Ce paiement produit une facture, celle-ci établissant le coût de la commande. L’énumération « EtatDuPaiement » recense les différents états du paiement.

# LE MODELE PHYSIQUE DE DONNEES

Le modèle physique de données est par excellence la représentation schématique en mesure de générer du code SQL, notamment par l’intermédiaire du logiciel SQL Power Architect, ce code pouvant construire une base de données PostgreSQL. Le modèle physique de données crée pour chacune des classes une clé primaire, celles-ci permettant d’établir des correspondances entre l’ensemble des classes.

Une image contenant capture d’écran

Description générée automatiquement

## Une image contenant capture d’écran Description générée automatiquementLa relation entre les classes « client » et « commande »

Une commande possède un client (one to one) aussi, la classe « commande » comporte « client\_id » en clé étrangère. Cette clé permet d’associer chaque client à une commande unique.

## La relation entre les classes « client » et « adresse »

Une image contenant capture d’écran

Description générée automatiquement

Chaque client possède une adresse unique. Celle-ci est enregistrée dans la classe « client » via la clé étrangère « adresse\_id ».

## La relation entre les classes « adresse » et « commande »

Une image contenant capture d’écran

Description générée automatiquement

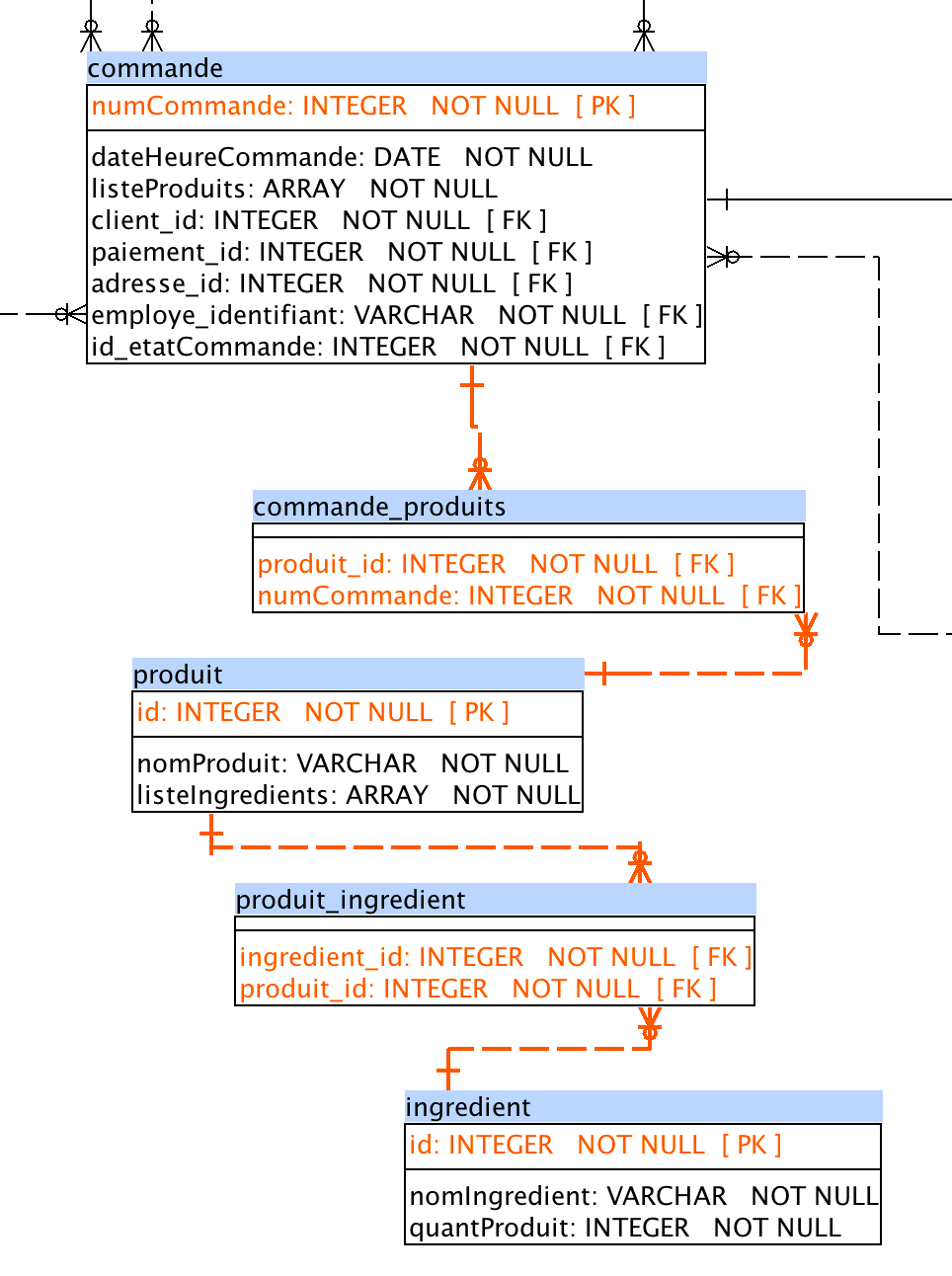
Chaque commande est reliée à une adresse unique. Pour ce faire, la classe « commande » possède la clé étrangère « adresse\_id ».

## La relation entre les classes « commande » et « livraison », « commande » et « paiement » et « paiement » et « facture »

Une image contenant capture d’écran

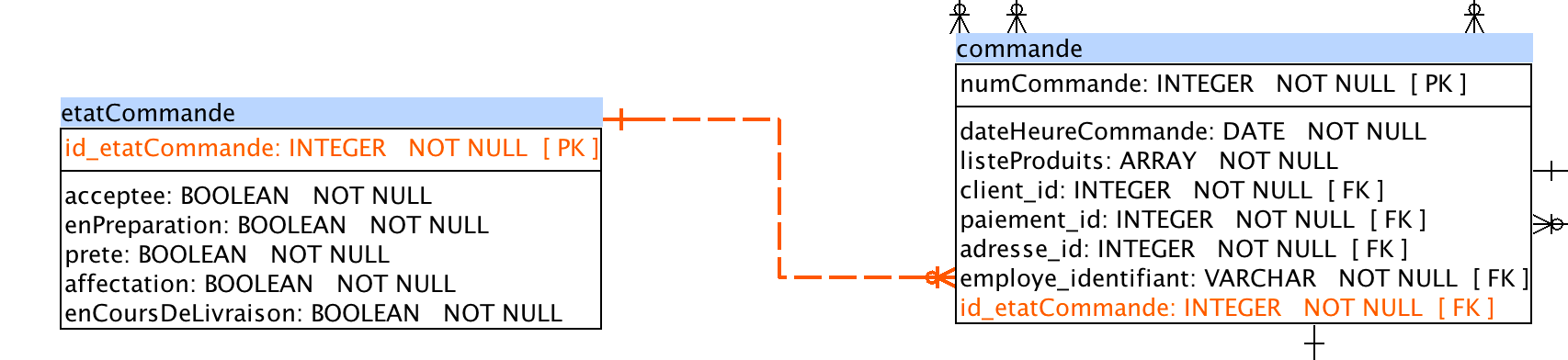
Description générée automatiquementUne commande est en relation avec un paiement en particulier, de même que ce paiement est en référence avec une facture unique.

## Les spécificités des relations entre les classes « commande », « produit » et « ingredient »



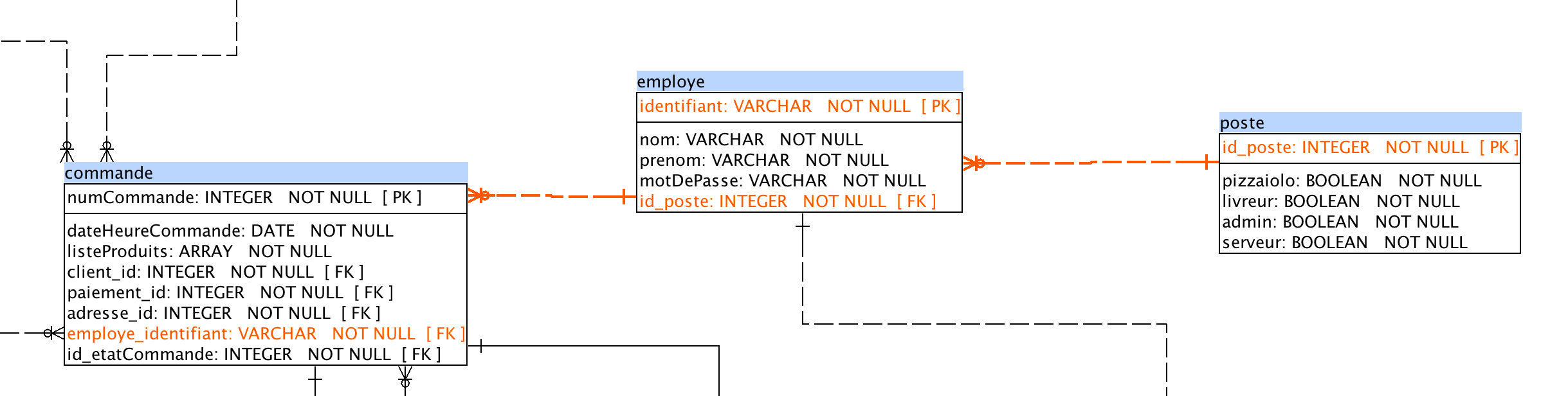
Plusieurs commandes comportent plusieurs produits et plusieurs ingrédients (« many to many »). Afin de mieux structurer les relations entre ces trois tables, il a été nécessaire de créer des tables de liaison : « commande\_produits » et « produit\_ingredient ». La table de liaison « commande\_produits » permet d’enregistrer en clés étrangères les deux clés primaires « numCommande » (clé primaire de la table « commande ») et « produit\_id » (clé primaire de la table « produit »). Cette table de liaison définit une première spécificité de la commande client. La seconde table de liaison « produit\_ingredient » enregistre en clés étrangères « produit\_id » (clé primaire de la table « produit ») et « ingredient\_id » (clé primaire de la table « ingredient »). Cette deuxième table de liaison fournit une seconde spécificité de la commande client.

## La relation entre les classes « etatCommande » et « commande »



La classe « commande » intègre en clé étrangère « id\_etatCommande » afin de spécifier pour chaque commande son état : acceptée ou en préparation ou prête ou affectée à un livreur ou en cours de livraison. Cet état permet au client de connaître la situation de sa commande.

## La relation entre les classes « commande », « employe » et « poste »



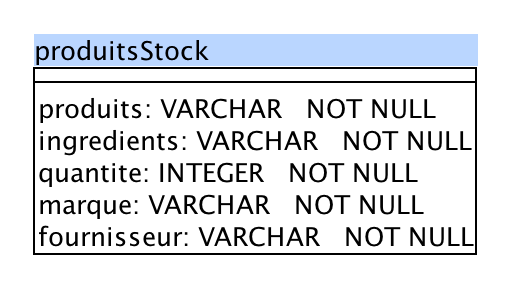
La classe « employe » intègre la clé étrangère « id\_poste ». Cette clé étrangère fait référence à la classe « poste ». Cette classe désigne l’employé en charge de la commande ou de sa livraison. De son côté, la classe « commande » comporte la clé étrangère « employe\_identifiant » de manière à récupérer l’identifiant de l’employé désigné en amont.

## La relation entre les classes « employe » et « livraison »



La classe « livraison » possède la clé étrangère « employe\_identifiant » afin de connaître l’employé en gestion de la commande.

## La classe « produitsStock »



La classe « produitsStock » ne possède aucune clé primaire car cette classe est en relation avec aucune classe. Son unique fonction consiste dans la gestion du stock, de son niveau suffisant et en concordance avec les besoins clients. Par ailleurs, seul l’administrateur peut y accéder et lorsque cela est nécessaire, effectuer une commande auprès des fournisseurs.

# LES COMPOSANTS DU SYSTEME

## Le diagramme de composants

Une image contenant texte

Description générée automatiquement

# L’ARCHITECTURE DE DEPLOIEMENT

## Le diagramme de déploiement

Une image contenant capture d’écran

Description générée automatiquement