**PROJET 6 : Concevoir la solution technique d’un système de gestion de pizzeria**

2 - INTRODUCTION

**Besoin du client**

*Contexte :*

  

Votre enseigne « OC Pizza » est un jeune groupe de pizzeria en plein essor et spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et vous prévoyez d’en ouvrir au moins 3 de plus d’ici la fin de l’année. Vous avez pris contact avec nous afin de mettre en place un système informatique, déployé dans toutes vos pizzerias qui vous permettrait notamment :

* D’être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
* De suivre en temps réel les commandes passées et en préparation ;
* De suivre en temps réel le stock d’ingrédients restants pour savoir quelles  Pizzas sont encore réalisables ;
* De proposer un site Internet pour que les clients puissent :

O Passer leurs commandes, en plus de la prise de commande par téléphone ou sur place,

O Payer en ligne leur commande s’ils le souhaitent – sinon, ils paieront directement à la livraison

O Modifier ou annuler leur commande tant que celle-ci n’a pas été préparée

• De proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza

Vous avez à priori déjà fait une prospection et les logiciels existants n’ont pas répondu à vos attentes Je vais donc essayer de faire au mieux pour répondre à vos besoins de manière efficace.

Ce document est donc une proposition de solution technique de votre système de gestion

**Diagramme de classe :**

****

Nous pouvons donc voir notre diagramme de classe représenté ci-dessus avec 8 différentes classes, je vais donc vous faire une présentation de chaque classe ainsi que leur rôle.

**Classe « EMPLOYE » :**  
Nous avons en premier lieu la classe « EMPLOYE » qui représentera tous les employés des pizzérias, cette classe est liée avec la classe pizzeria avec une relation 1..\* et 1 puisqu’un employé ne peut travailler quand dans une seule pizzeria par contre une pizzeria peut avoir 1 ou plusieurs employés qui y travaille.  
Il y a également une relation entre la classe employé et la classe commande avec une relation \*, 1..\* puisqu’il y peut y avoir plusieurs employés qui travaillent sur la même commande (livreur et pizzaiolo par exemple) et un employé peut avoir travaillé sur plusieurs commandes.

**Classe « PIZZERIA » :**Nous avons ensuite la classe « PIZZERIA » qui représente les différentes pizzerias du groupe. Cette classe est reliée à la classe « EMPLOYE » comme expliqué plus haut puis nous avons d’autre relation.  
Elle est ensuite reliée au stock afin d’avoir un suivi du stock des différentes pizzerias, elle est relié par une relation \*, 1..\* puisque que le même aliment du stocks peut être présent dans plusieurs pizzeria à la fois et une pizzeria a plusieurs aliments du stock.  
Ensuite, nous avons une relation entre la classe « Pizzeria » et la classe « Commande » avec une relation 1, 1..\* puisqu’une commande ne peut appartenir qu’a une seule pizzeria et une pizzeria peut avoir plusieurs commandes.  
Puis également une relation entre la classe « PIZZA » et la classe « PIZZERIA » ce qui correspondrait au menu par pizzeria, nous avons une relation 1..\*, \* puisque qu’une pizza peut appartenir à toutes les pizzerias et qu’une pizzeria peut avoir plusieurs pizza au menu.

**Classe « STOCK » :**Cette classe correspond au stock disponible par pizzeria, elle joue le rôle de suivi du stock des aliments.  
Elle est reliée à la classe « PIZZERIA » comme déjà expliqué précédemment puis à la classe « PIZZA » puisque qu’une pizza est composé de différents ingrédients disponible dans le stock, nous avons une relation de cardinalité \*, 1..\*, effectivement un ingrédient du stock peut être présent sur plusieurs pizzas et une pizza peut avoir différents aliments par sorte de pizza.

**Classe « PIZZA » :**Cette classe correspond aux différentes pizzas disponibles au menu et cela par pizzeria. Elle est liée à la classe « PIZZA » et la classe « STOCK » (déjà mentionnée dans la classe pizza et la classe stock)  
Elle est également liée à la classe « COMMANDE » puisqu’une commande va être composée d’aucune ou plusieurs pizzas d’où la relation 0..\*, 0..\*. Et une pizza peut faire partie d’aucune commande ou bien de plusieurs commandes.

**Classe « COMMANDE » :**La classe « COMMANDE » représente le détail de l’ensemble commandes présentes pour chaque pizzeria, la relation entre la classe « PIZZERIA » et la classe « COMMANDE » a déjà été traitée dans la classe « PIZZERIA ».  
Il y a ensuite une relation entre la classe « COMMANDE » et la classe « ETAT COMMANDE », il s’agit d’une relation 1..\* ,1 puisqu’une commande ne peut avoir qu’une seul statut à la fois mais un statut peut être présent dans différentes commandes.  
Puis, nous avons bien entendu une relation entre la classe « COMMANDE » et la classe « CLIENT » et cette relation est 1, 0..\*, évidemment, une commande ne peut appartenir qu’a un seul client à la fois tandis qu’un client peut avoir aucune commande à plusieurs commandes.

En ayant, réalisé le diagramme de classe, nous pouvons obtenir notre modèle physique de données. C’est que nous allons vous présenter à présent.

**MODELE PHYSIQUE DE DONNEES :**



Table « EMPLOYE » :  
Cette table va contenir les informations sur les employés avec leur fonction et savoir dans quel pizzeria ils travailles.  
Toutes les colonnes seront « NOT NULL » puisqu’il faudra renseigner chaque colonne.  
idEMPLOYE est un SMALLINT AUTO-INCREMENT cette colonne est la clé primaire de la table, cela permettra de reconnaître de manière unique les employés  
Les colonnes nom et prénom seront de type VARCHAR(45) pour 45 caractère maximum  
La colonne fonction pour déterminer le rôle de l’employé sera également de type VARCHAR(45)  
Puis la colonne IdPizzeria qui sera de type SMALLINT NON NULL UNSIGNED et qui sera la clé étrangère vers la table « PIZZERIA » ce qui permettra d’identifier sur quel pizzeria l’employé travaille.  
Table « PIZZERIA » :  
Nous avons idPIZZERIA qui est de type SMALLINT NON NULL UNSIGNED AUTO-INCREMENT puisqu’il sera incrémenté automatiquement lors de l’insertion, ne peut être nul et il s’agit également de clé primaire de la table pour permettre d’identifier de ma manière unique une pizzeria.  
Nous avons la colonne « ADRESSE » de type TEXT car assez long du fait du nom de la rue, le numéro, le code postal et la ville également NON NULL puisque les informations doivent obligatoirement être fournis

Table « STOCK » :  
Cette table va stocker les ingrédients disponibles par pizzeria, classé par catégorie.  
Nous avons donc la colonne IdSTOCK de type SMALLINT UNSIGNED NOTNULL AUTO-INCREMENT, qui sera la clé primaire de la table afin d’identifier de manière unique.  
Puis les colonnes nomAliment et nomCat qui seront de types VARCHAR NOTNULL puisque les colonnes ne peuvent rester vides.

Table « COMPOSITION STOCK » :  
Il s’agit d’une table de composition du fait de la cardinalité de la relation, nous sommes obligés de créer cette table intermédiaire pour représenter la relation des différentes pizzeria avec les différents aliments.  
Nous avons ici une clé primaire composite avec la colonne idPizzeria qui est de type identique à sa colonne de référence de la table « PIZZERIA » puisqu’il y s’agit également d’une clé étrangère qui identifiera la pizzeria concernée par le stock.  
Nous avons également la colonne idStock qui sera également de même type que la colonne qu’elle référence puisqu’encore une fois il s’agit de la clé étrangère relié à l’identifiant de la table « STOCK ».  
Le couple de colonne va donc nous donner les informations sur la pizzeria concernée avec l’aliment correspondant.  
Puis nous avons la colonne quantitéRestante qui sera de type SMALLINT UNSIGNED, cette colonne n’est pas NOT NULL puisqu’elle peut être égal à zéro s’il n’y a plus de stock disponible.

Table « PIZZA » :  
Cette table va stocker les différentes pizzas disponibles dans chaque pizzeria.   
Nous avons encore la clé primaire sur la colonne idPIZZAS de type TINYINT UNSIGNED NOT NULL AUTO-INCREMENT qui permettra d’identifier les pizzas de manières uniques.  
Nous avons la colonne nomPizzas qui correspond au nom de la pizza de type VARCHAR(30), nous n’avons pas besoin de nombreux caractère pour cette colonne puisque les noms de pizzas ne sont jamais très long généralement.  
Puis une colonne prix de type REAL UNSIGNED NOT NULL puisque le prix de la pizza peut ne pas être un nombre entier, doit être renseigné et ne peut pas être nul.