



PIZZA

OC

PARTIE II : SPECIFICATION TECHNIQUE

1. Introduction	4
La mission	4
Rappel des objectifs fixés :	5
Les enjeux	5
2. Le domaine fonctionnel	6
Définition	6
Le Diagramme de classes.	6
Présentation des classes fondamentales	6
Classe Utilisateur	6
Classe Client lié à la Classe Adresse	7
Classe liée Commande avec la classe Produit	9
Classe liée Point de vente avec la classe ingrédient	10
3. Présentation du modèle physique de donnée	11
Definiton	11
Table Utilisateur	11
Table Client	12
Table Collaborateur	12
Table Adresse	12
Table PointDeVente	13
Table Commande	13
Table Commande_Produit	14
Table Produit	14
Table Produit_ingredient	15
Table Stock	15
4. Diagramme de déploiement	15
Definition	16

Annexe	17
Résultat requête de sélection	17
Affiche le nom des employés et leur fonction	17
Affiche le numéro de la commande et le prix total trié	18
Affiche le classement des points de vente par rapport a leur nombre de vente	18
Affiche le stock d'un point de vente	19
Affiche la préparation d'un produit	20

1. Introduction

La mission

Il s'agit ici de la seconde partie de notre mission qui débouche au dossier de spécification technique de la solution proposée.

Nous allons aborder 5 thématiques qui vont vous permettre de vous familiariser avec votre future système de gestion.

Il s'agit donc de mettre en exergue plusieurs points :

- Le modèle fonctionnel du système
- Le modèle physique de données.
- L'architecture de déploiement.

Le modèle fonctionnel va nous permettre de revoir les besoins qui avait été fixé au préalable lors du dossier d'analyse des besoins(spécification fonctionnelle) . Nous allons par ce biais vous présenter les éléments permettant d'obtenir les résultats prévus.

Puis, nous vous présenterons le modele physique de donnée, le coeur du système de donnée. Celui ci est réalisé en Mysql selon les exigences que nous avons évoqué lors de nos différents échanges.

Enfin, nous identifierons les éléments logiciels et matériels dans le cadre du diagramme de déploiement.

Pour Conclure, nous vous remercions de votre confiance, vous pouvez bien entendu consulter les spécifications fonctionnels établis précédemment.

Rappel des objectifs fixés :

Reprise du cahier des charges OC Pizza :

« 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 prévoit d'en ouvrir au moins 3 de plus d'ici la fin de l'année. Mettre en place un système informatique, déployé dans toutes ses pizzerias et qui lui 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 :
 - 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
- de proposer un aide mémoire aux préparateur indiquant la recette de chaque pizza

Les enjeux

Cette étude s'inscrit dans la suite logique de la précédente partie.

Afin de mettre en place les solutions préconisés, des entretiens collaboratifs sont prévue et permettront de mettre d'optimiser les spécifications techniques de la solution.

2. Le domaine fonctionnel

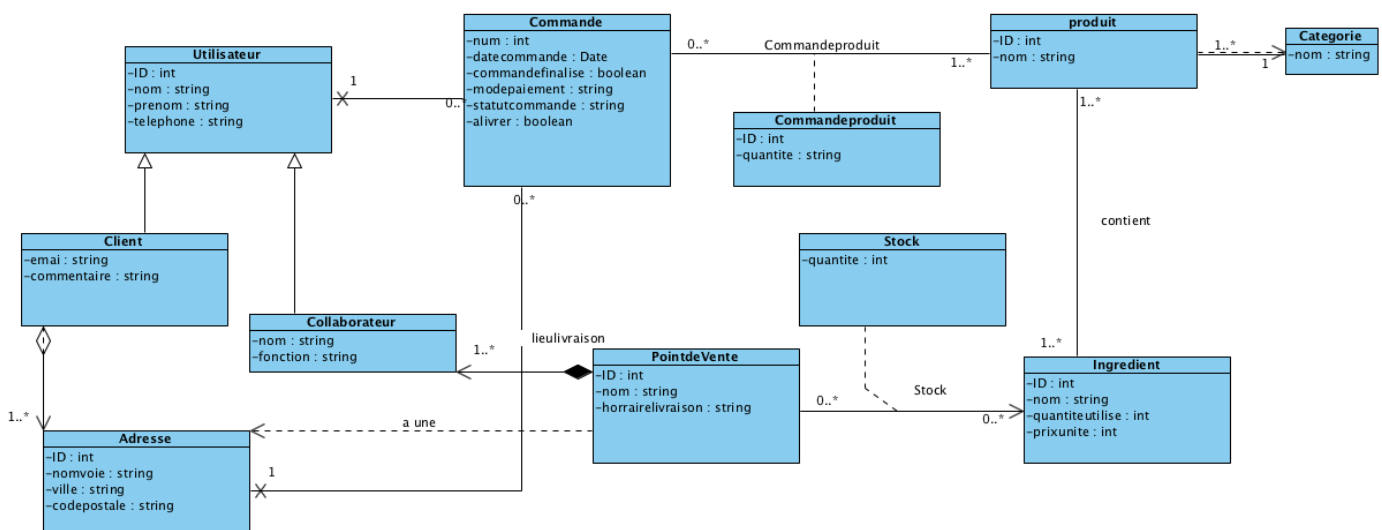
Définition

Il s'agit de mettre en relation des classes d'objets dans le langage de programmation java et d'articuler le Modèle Physique de Données.

Nous allons de plus présenter les interactions de chaque objets et de ce qu'il contiendra.

Le Diagramme de classes.

Nous représentons l'architecture du système d'information grâce aux différentes classes qui sont liés entre-elles.



Présentation des classes fondamentales

La classe utilisateur est a l'image du précédent dossier.

Nos utilisateur sont regroupés en deux catégories.

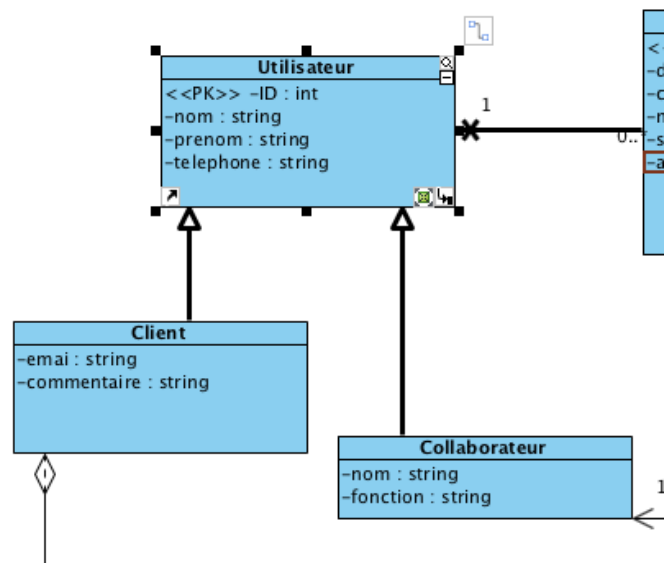
Nous parlons ici de relation mère/fille

L'utilisateur du système peut être soit un employé qui souhaite procéder a la commande ou le client qui directement effectuera l'opération.

La relation sur la droite représente le lien avec la commande.

L'utilisateur effectue une commande.

Il peut faire aucune ou plusieurs commande (d'ou la relation 0 to many)



Les attributs	Roles
Nom	Correspond au nom de l'utilisateur
prenom	Prénom de l'utilisateur
Telephone	Le telephone de l'utilisateur

Classe Client lié à la Classe Adresse

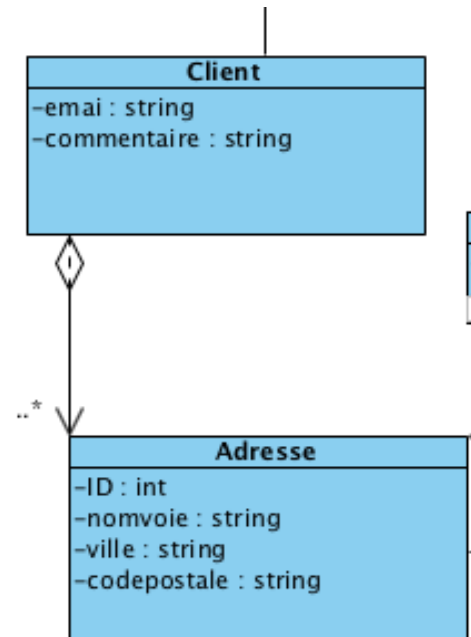
Correspond au donnée du client lors de sa saisie
A noter une relation avec l'adresse.

Nous pouvons donc noter que la relation entre les deux classes est une association.

La classe adresse est lié quant a elle au point de vente.

Chaque point a vente à une adresse.

De plus nous avons un lien supplémentaire avec commande qui permet d'indiquer l'adresse de livraison.

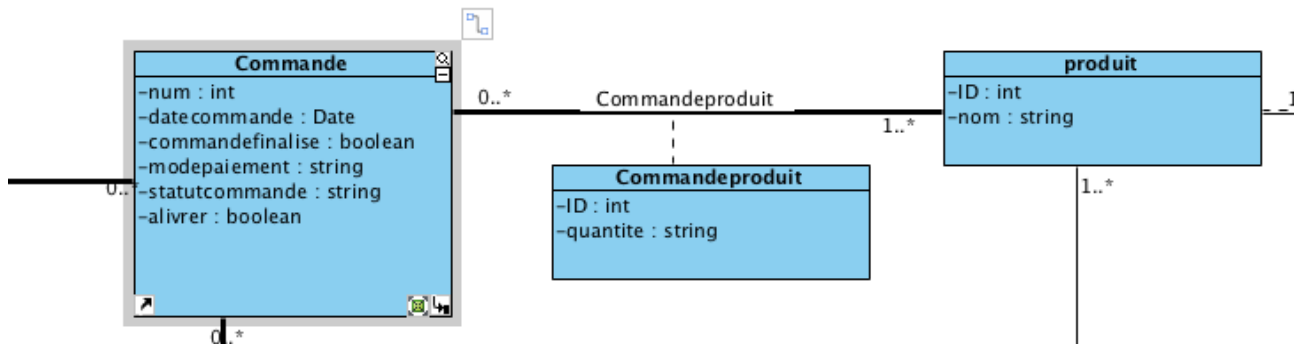


Les attributs	Roles
Email	Correspond au mail du client
Commentaire	Zone d'information laisse par la client
Datecreation	Correspond a la date d'ouverture du compte el client
_____TABLE ASSOCIE ADRESSE	
ID	numéro de l'enregistrement
Nomvoie	Nom de la rue
Ville	Nom de la localité
codepostale	Le code postal

Classe liée Commande avec la classe Produit

Il s'agit ici d'une articulation des plus importantes de l'architecture du logiciel.

C'est elle qui détermine la relation entre les produits et la commande.



Une commande est constituée d'un ou plusieurs produits et un produit peut être dans une ou plusieurs commande.

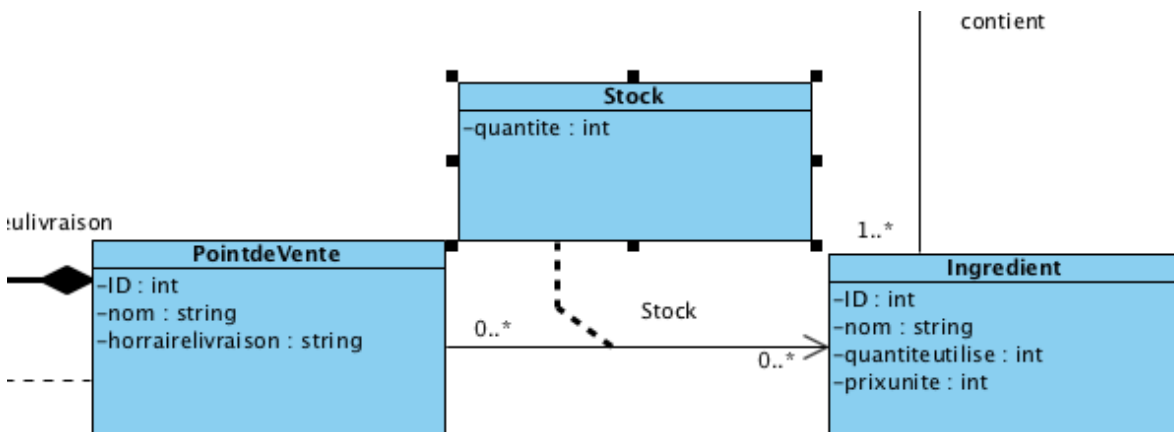
Nous avons donc ici une relation dite (many to many).

Ainsi, la table Commandeproduit fait la jonction entre ces deux tables.

Les Attributs	Roles
Num	Numero de la commande
Datecommande	Date de la commande effectuée
Commandefinalise	Confirme la fin de la commande
Modedepaiement	Choix du mode de paiement
alivrer	Confirme que la commande est a livrer
Statutcommande	il s'agit ici d'indiquer l'état de la commande
_____TABLE liée produit	
ID	numéro de l'enregistrement
Nom	Nom du produit
_____TABLE liée Commandeproduit	
Id	Numéro de l'enregistrement
Quantité	La quantité correspondant au produit commandée

Classe liée Point de vente avec la classe ingrédient

Il s'agit d'une relation comme précédemment dite many to many :
 Un point de vente peut avoir plusieurs ingrédients
 Et un ingrédient peut être détenu dans plusieurs point de vente.
 Nous avons donc une relation dite d'association par le biais de la classe stock.



Les Attributs	Roles
ID	Numero du point de vente
Nom	Nom du point de vente
Horaire de livraison	Detail des horaires du point de vente
_____TABLE liée stock	
ID	numéro de l'enregistrement
Nom	Nom du produit
_____TABLE liée ingrédient	
Id	Numéro de l'enregistrement
Nom	Nom de l'ingrédient
Quantité	Quantité dose pour un produit
Prixunite	prix de l'ingrédient par unité

Quelques préconisations

Dans le cas d'une vente en face a face pour lequel le vendeur effectue une saisie, l'utilisateur saisira le prénom et du client uniquement du client.

Table Client

Il s'agit ici de regrouper les éléments de saisie permettant de qualifier notre web client.

Email devra obligatoirement être saisi (NOT NULL)

Nous utilisons datecreation qui sera une date incrémentée qui renseignera de l'ouverture du compte client.

Cela nous donnera un élément exploitable sur la fidélisation du client.

La colonne id_utilisateur est une clé étrangère liée directement à utilisateur qui se réfère à l'ID de celle-ci.

Table Collaborateur

Nous retrouvons ici les informations des employés de la pizzeria

Il n'y a pas de champ nul, l'employeur doit insérer l'ensemble des données.

Le nom est ici le nom du collaborateur qui est de type varchar.

La fonction est la fonction occupée au sein de l'entreprise. Nous pourrions insérer un type enum des choix possibles (vendeur, livreur, direction, préparateur).

La colonne id_utilisateur comme pour table client est une clé étrangère se référant directement à la table utilisateur.

Table Adresse

Il s'agit dans ce cas présent de regrouper les adresses que l'on retrouve dans les autres tables liées.

Pour la définir nous avons mis en place :

numero_voie : type int pour le numéro de la voie

voie : type varchar avec 50 caractères possible en cas de saisie un peu plus longue
code_postal : type varchar avec une taille de 5.
Localité : type Varchar avec une taille de 20.

La colonne client_id est une clé étrangère se référant à la table client dont id_utilisateur est utilisé comme clé primaire.

Table PointDeVente

La colonne nom de type varchar avec une taille de 55 doit être renseignée (NOT NULL)

Nous retrouvons id qui est la clé primaire avec un auto-incrément de type int.

Présence d'une clé étrangère apparaissant dans la table adresse.

Table Commande

Nom du champ	Type	Description
id	Int avec auto-incrément	Il s'agit de notre numéro d'enregistrement précisons par ailleurs qu'il s'agit d'une clé primaire.
date_livraison	Type date	
statut_commande	Type varchar de taille 20	Un masque de saisie sur le logiciel permettra de choisir entre en cours de préparation en livraison ou terminé
fin_commande	Type booléen	Lorsque la commande est finalisée la saisie sera de true
Commande_a_livrer	Type booléen	Si true est positionné, une requête permettra de se référencer à l'adresse de livraison
Commentaire	Type varchar de taille 70	Indication concernant la commande spécification par exemple sur l'adresse ou si il s'agit d'une commande spéciale
paiement_valide	Type booléen	Si true, le paiement est fait

Nom du champ	Type	Description
moyen_paiement	Type varchar 40	Fera la jonction avec l'établissement bancaire si carte bleue

Présence de plusieurs clés étrangère

Nous retrouvons ici

La colonne utilisateur_id en lien avec la table utilisateur

La colonne pointdevente_id en lien avec le point de vente de la commande

La colonne adresse_id en lien avec la table adresse de notre commande.

préconisation

Si il s'agit d'une commande sur place, l'utilisateur renseigne l'adresse du point de vente.

Table Commande_Produit

Il s'agit de la table constitué par l'association de la classe produit et commande.

En effet, la mission de cette table est de gérer la relation many-to many des deux classes.

La colonne quantite de type int doit être renseigné (NOT NULL) il s'agit de la quantité de chaque produit commandé.

Nous trouvons deux clés étrangères :

commande_id : correspond au num de la classe commande relier donc chaque ligne a une commande

produit_id : correspond au num de la classe produit afin de relier chaque ligne de la commande produit.

Table Produit

Dans cette table se retrouve les différents produits qui peuvent être fabriqué.

nom_commercial de type varchar d'une taille de 20 caractère obligatoirement saisie (NOT NULL)

Prix de type int d'une taille de 11 (NOT NULL) il s'agit du prix d'un produit.

Description de type varchar de taille 50 il s'agit de détailler un produit (pas d'allergène, pas de viande , présence d'anchois etc...)

La clé primaire est id toujours avec une auto-incrémentation de type int.

Table Produit_ingredient

Il s'agit ici d'une table d'association, vous remarquerez effectivement que celle ci n'apparaît pas dans le diagramme de classe.

Elle est liée au fait de la relation many-to-many des deux tables produits et ingrédients. En effet, un produit peut être composé de plusieurs ingrédients et un ingrédient peut apparaître dans plusieurs produits.

Nous avons donc deux colonnes ayant chacune une clé étrangère.

La colonne produit_id rattaché a la table produit chaque ingrédient peut être dans plusieurs produits de

La colonne ingredient_id rattaché a la table ingrédient ainsi chaque produit peut avoir un ou plusieurs ingrédients.

Table Stock

Il s'agit ici aussi d'une table née d'une relation many-to-many entre le point de vente et la table ingrédient.

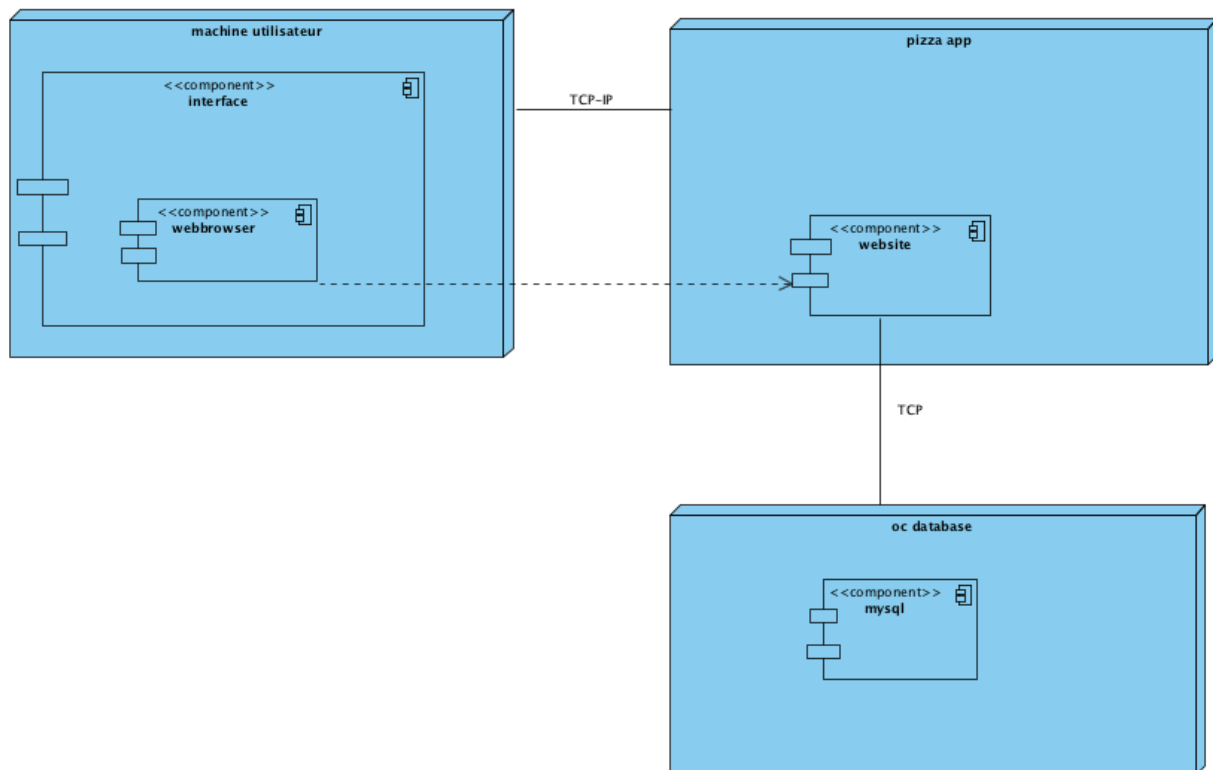
La colonne quantité de type int ne peut être nulle. Il s'agit de la donnée permettant au point de vente de renseigner la quantité détenu d'un ingrédient.

présence de deux clés étrangère :

La colonne pointdevente_id qui se réfère ici a la table point de vente par le biais de son id.

La colonne ingredient_id qui se réfère a la table ingrédient par son id.

4. Diagramme de déploiement



Definition

Nous avons donc vu précédemment les éléments permettant l'architecture de notre système de gestion.

Le diagramme de déploiement a pour objectif de déterminer les éléments qui composent le nouveau système.

Comme on peut le voir dans le diagramme, il y a 3 nœuds représentant les éléments hardware ou software faisant partie du système.

- *Machine utilisateur* indique l'ordinateur avec lequel l'utilisateur se connecte. *Client Interface* représente l'interface utilisateur, à travers laquelle il peut accéder au site web d'OC Pizza (via le navigateur web *Web Browser*).
- OC Pizza App indique l'ensemble de pages web qui composent le site web.

Le transfert de données entre ce nœud et le nœud *User Machine* se fait grâce au protocole TCP/IP.

La dépendance entre les composants *Client Interface* et *Website* est marquée par la flèche en pointillés.

- *OC Pizza Server* indique le serveur chargé de gérer les requêtes des pages du site web. Il repère les informations grâce à la base de données dédiée *Oc Pizza database* et les affiche à l'utilisateur. Les interactions avec la base de données sont assurées par le SGBDR MySQL.

L'échange des données entre les nœuds *OC Pizza App* et *OC Pizza Server* se fait toujours avec TCP/IP.

Annexe

Résultat requête de sélection

Affiche le nom des employés et leur fonction

Code sql

```
SELECT `PointdeVente`.`id`, `PointdeVente`.`nompontvente`, `Collaborateur`.`nom`,  
`Collaborateur`.`fonction`  
FROM `PointdeVente`  
LEFT JOIN `Collaborateur` ON `Collaborateur`.`PointdeVenteid` = `PointdeVente`.`id`  
where PointdeVente.id=1
```

Resultat issue de phpMyadmin

id	nompontvente	nom	fonction
1	OC PIZZA NIMES CENTRE	alfonso	preparateur
1	OC PIZZA NIMES CENTRE	da silva	livreur
1	OC PIZZA NIMES CENTRE	meurice	vendeur

Affiche le numéro de la commande et le prix total trié

Code sql

```
SELECT `Commande`.`ID`, SUM(`Produit`.`prix`*`Commande_Produit`.quantite) FROM `Produit` LEFT JOIN `Commande_Produit` ON `Commande_Produit`.`produitID` = `Produit`.`ID` LEFT JOIN `Commande` ON `Commande_Produit`.`commandeID` = `Commande`.`ID` GROUP BY `Commande`.`ID` ORDER BY SUM(`Produit`.`prix`*`Commande_Produit`.quantite) DESC
```

Resultat issue de phpMyadmin

ID	SUM(`Produit`.`prix`*`Commande_Produit`.quantite)
1	66
5	48
4	30
2	16
3	15

Affiche le classement des points de vente par rapport a leur nombre de vente

Code sql

```
SELECT `PointdeVente`.`id`,`PointdeVente`.nompontvente, SUM(`Commande_Produit`.quantite) FROM `Produit` LEFT JOIN `Commande_Produit` ON `Commande_Produit`.`produitID` = `Produit`.`ID` LEFT JOIN `Commande` ON `Commande_Produit`.`commandeID` = `Commande`.`ID` LEFT JOIN `PointdeVente` ON `Commande`.`pointdeventeid` = `PointdeVente`.`id` GROUP BY `PointdeVente`.id ORDER BY SUM(`Commande_Produit`.quantite)DESC
```

Resultat issue de phpMyadmin

id	nompontvente	SUM(`Commande_Produit`.quantite)
1	OC PIZZA NIMES CENTRE	15
2	OC PIZZA GAMBETTA	5
3	OC PIZZA JEAN JAURES	1

Affiche les commande d'un client

Code sql

```
SELECT `Commande`.`utilisateurID`, `Commande`.ID, `Commande`.date_livraison, `Utilisateur`.`nom`,  
`Commande_Produit`.`quantite`, `Produit`.`nom_commercial` FROM `Produit` LEFT JOIN  
`Commande_Produit` ON `Commande_Produit`.`produitID` = `Produit`.`ID` LEFT JOIN `Commande`  
ON `Commande_Produit`.`commandeID` = `Commande`.`ID` LEFT JOIN `Utilisateur` ON  
`Commande`.`utilisateurID` = `Utilisateur`.`ID` WHERE `Commande`.utilisateurID=1
```

Resultat issue de phpMyadmin

utilisateurID	ID	date_livraison	nom	quantite	nom_commercial
1	1	2018-01-01	Robert	3	pizza regina
1	1	2018-01-01	Robert	2	pizza reine
1	5	2018-06-20	Robert	4	pizza forestiere

Affiche le stock d'un point de vente

Code sql

```
SELECT `Ingredient`.`nom_ingredient`, `Stock`.`quantite`, `PointdeVente`.`nompontvente`  
FROM `Ingredient`  
LEFT JOIN `Stock` ON `Stock`.`ingredientID` = `Ingredient`.`ID`  
LEFT JOIN `PointdeVente` ON `Stock`.`pointdeventeid` = `PointdeVente`.`id`  
WHERE `PointdeVente`.id=1
```

Resultat issue de phpMyadmin

nom_ingredient	quantite	nompontvente
mozarella	43	OC PIZZA NIMES CENTRE
gruyere	43	OC PIZZA NIMES CENTRE
oeufs	43	OC PIZZA NIMES CENTRE
pate fine	55	OC PIZZA NIMES CENTRE
creme fraiche	12	OC PIZZA NIMES CENTRE
farine	23	OC PIZZA NIMES CENTRE
orange	23	OC PIZZA NIMES CENTRE
agrumes	32	OC PIZZA NIMES CENTRE
soda	23	OC PIZZA NIMES CENTRE
eau	32	OC PIZZA NIMES CENTRE
chorizo	55	OC PIZZA NIMES CENTRE
cafe	882	OC PIZZA NIMES CENTRE
chocolat en poudre	33	OC PIZZA NIMES CENTRE
menthe	23	OC PIZZA NIMES CENTRE
glacon	21	OC PIZZA NIMES CENTRE
mascarpone	21	OC PIZZA NIMES CENTRE

Affiche la préparation d'un produit

Code sql

```
SELECT `Produit`.`nom_commercial`, `Produit_Ingredient`.*, `Ingredient`.`nom_ingredient`,
`Ingredient`.`description`, `Ingredient`.`unit` FROM `Ingredient` LEFT JOIN `Produit_Ingredient` ON
`Produit_Ingredient`.`ingredientID` = `Ingredient`.`ID` LEFT JOIN `Produit` ON
`Produit_Ingredient`.`produitID` = `Produit`.`ID` where Produit.ID=1
```

nom_commercial	produitID	ingredientID	nom_ingredient	description	unit
pizza regina	1	1	mozarella	bonne qualite	3
pizza regina	1	2	gruyere	distri 1	5