

**Client : OC Pizza**

**Projet : OC Pizza**

Description du modèle physique de données et du jeu de données

Version 1

**Auteur**

Guillaume Toussaint

*Développeur*

## TABLE DES MATIERES

<b>1 - Versions .....</b>	<b>4</b>
<b>2 - Introduction .....</b>	<b>5</b>
2.1 - Objet du document .....	5
2.2 - Références.....	5
<b>3 - Les tables.....</b>	<b>6</b>
3.1 - Table version.....	6
3.2 - Table point_de_vente .....	6
3.2.1 - Jeu de données.....	6
3.3 - Table rôle.....	6
3.3.1 - Valeurs initiales.....	6
3.3.2 - Jeu de données.....	7
3.4 - Table employe.....	7
3.4.1 - Relations .....	7
3.4.2 - Jeu de données.....	8
3.4.2.1 - Données brutes .....	8
3.4.2.2 - Liste par point de vente et rôle.....	8
3.4.2.3 - Requête .....	8
3.5 - Table client_type .....	9
3.5.1 - Valeurs initiales.....	9
3.5.2 - Jeu de données.....	9
3.6 - Table client .....	10
3.6.1 - Champs obligatoires en fonction du type de client.....	10
3.6.2 - Relations .....	10
3.6.3 - Jeu de données.....	11
3.6.3.1 - Données brutes.....	11
3.6.3.2 - Liste des clients par point de vente et type de client .....	12
3.6.3.3 - Requête .....	12
3.7 - Table tva .....	12
3.7.1 - Valeurs initiales.....	12
3.7.2 - Jeu de données.....	13
3.8 - Table unite.....	13
3.8.1 - Valeurs initiales.....	13
3.8.2 - .....	13
3.8.3 - Jeu de données.....	14
3.9 - Table recette .....	14
3.9.1 - Jeu de données.....	14
3.10 - Table produit.....	15
3.10.1 - Relations .....	15
3.10.2 - Jeu de données.....	15
3.11 - Table recette_produit.....	17

3.11.1 - Jeu de données.....	18
3.11.1.1 - Données brutes.....	18
3.11.1.2 - Liste des ingrédients par pizza .....	19
3.11.1.3 - Requête.....	19
3.12 - Table pizza.....	20
3.12.1 - Relations .....	20
3.12.2 - Jeu de données.....	20
3.13 - Table panier .....	21
3.13.1 - Relation.....	21
3.13.2 - Jeu de données.....	21
3.13.2.1 - Détail des paniers pizza et produit et quantité.....	22
3.13.2.2 - Requête.....	23
3.14 - Table panier_pizza .....	23
3.14.1 - Relations .....	23
3.14.2 - Jeu de données.....	24
3.15 - Table panier_produit.....	25
3.15.1 - Relations .....	25
3.15.2 - Jeu de données.....	25
3.16 - Table produit_point_de_vente .....	26
3.16.1 - Relation.....	26
3.16.2 - Jeu de données.....	26
3.16.2.1 - Données brutes.....	26
3.16.2.2 - Etat du stock par produit, point de vente, quantité et unité .....	28
3.16.2.3 - Requête.....	29
3.17 - Table status_commande .....	29
3.17.1 - Valeurs initiales.....	29
3.18 - Table commande .....	29
3.18.1 - Relation.....	30
3.18.2 - Jeu de données.....	30
3.18.3 - Requête .....	31
<b>4 - Opérations sur la base de données.....</b>	<b>32</b>
4.1 - Mise en place de la base de données ocpizza.....	32
4.1.1 - Suppression du schema ocpizza.....	32
4.1.2 - Création du schema ocpizza .....	32
4.1.3 - Suppression des tables .....	32
4.1.4 - Création des tables.....	32
4.1.5 - Ajout du jeu de données .....	32

# 1 - VERSIONS

Auteur	Date	Description	Version
Guillaume Toussaint	16/08/2021	Création du document	1

## 2 - INTRODUCTION

### 2.1 - Objet du document

Ce document décrit le modèle physique de données et le jeu de données livré avec.

Il énumère les tables de la base de données, leur rôle, leurs relations et leurs particularités.

Il commente le jeu de données.

### 2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **Modèle physique de données**
2. **Script de création de la structure de la base de données**
3. **Script de création du jeu de données**

## 3 - LES TABLES

### 3.1 - Table version

Cette table permet de versionner la base de données à des fins de mise à jour.

La base de données sera créée avec le champ numéro = 1.

version		
PK	numero	INTEGER

### 3.2 - Table point\_de\_vente

Représente les points de ventes

point_de_vente		
PK	id	SERIAL
	nom	TEXT
N	adresse	TEXT
N	telephone	TEXT
N	email	TEXT

#### 3.2.1 - Jeu de données

	id [PK] integer	nom text	adresse text	telephone text	email text
1	1	Tours Grammont	31 Av. de Grammont 37000 Tours	02.47.01.02.03	grammont.tours@ocpizza.co...
2	2	Joué	70, Bd Jean Jaurès 37300 Joué-lès-Tours	02.47.01.02.04	jaurès.joue@ocpizza.com
3	3	Blois	50, Rue Michel Ange 41000 Blois	02.54.01.02.04	ange.blois@ocpizza.com
4	4	Au Petit Paris	109 Av. du Maréchal Maunoury 41000 Blois	02.54.01.02.04	paris.blois@ocpizza.com
5	5	Pizza Bilange	22 place de la Bilange 49400 Saumur	02.41.01.02.04	bilange.saumur@ocpizza.com

### 3.3 - Table rôle

La table rôle définit les différents rôles des employés.

role		
PK	id	INTEGER
	libelle	TEXT

#### 3.3.1 - Valeurs initiales.

Les valeurs initiales ci-dessous sont enregistrées dans la base pour gérer les prévus.

Id	Libelle
1	LIVREUR
2	PIZZAIOLO
3	MANAGER
4	DIRIGEANT

### 3.3.2 - Jeu de données

	id [PK] integer	libelle text
1	1	LIVREUR
2	2	PIZZAIOLO
3	3	MANAGER
4	4	DIRIGEANT

## 3.4 - Table employe

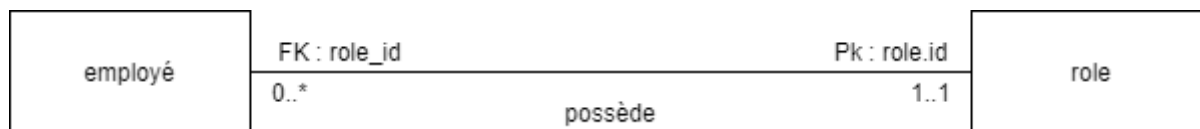
Cette table contient la liste des employés.

employe		
PK	id	SERIAL
	nom	TEXT
FK	role_id	INTEGER
N	prenom	TEXT
	email	TEXT
	password	TEXT
FK,N	point_de_vente_id	INTEGER

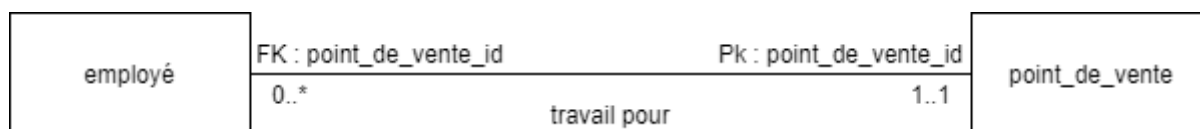
- Le champ « **role** » permet de spécifier quel rôle endosse l'employé, livreur, pizzaiolo, manager, dirigeant.
- Le champ « **point\_de\_vente\_id** » permet d'indiquer pour quel point de vente travail un employé. Remarque : Dans les cas des dirigeant ce champ restera NULL car un dirigeant travail pour l'ensemble du groupe et doit avoir accès a tous les points de vente.
- Le champ « **prenom** » n'est pas obligatoire.

### 3.4.1 - Relations

- Un employé possède un rôle



- Un employé travail pour un point de vente



### 3.4.2 - Jeu de données

#### 3.4.2.1 - Données brutes

	id [PK] integer	nom text	role_id integer	prenom text	email text	password text	point_de_vente_id integer
1	1	Lola	4	[null]	lola@ocpizza.com	03ac674216f3...	[null]
2	2	Franck	4	[null]	franck@ocpizza.com	f8638b979b2f...	[null]
3	3	Habram	1	[null]	habram@ocpizza.com	80a2a51ee74...	1
4	4	Marion	2	[null]	marion@ocpizza.com	8bfeff79372b...	1
5	5	Carol	3	[null]	carol@ocpizza.com	f8bfa2426633...	1
6	6	Alexandre	1	[null]	alexandre@ocpizza.com	fef59c07049d...	2
7	7	Luciano	2	[null]	luciano@ocpizza.com	0931f448c62a...	2
8	8	Simon	3	[null]	simon@ocpizza.com	62b24c1e36bf...	2

#### 3.4.2.2 - Liste par point de vente et rôle

	point de vente text	employé text	rôle text
1	Joué	Alexandre	LIVREUR
2	Joué	Luciano	PIZZAIOLO
3	Joué	Simon	MANAGER
4	Tours Grammont	Carol	MANAGER
5	Tours Grammont	Habram	LIVREUR
6	Tours Grammont	Marion	PIZZAIOLO
7	[null]	Franck	DIRIGEANT
8	[null]	Lola	DIRIGEANT

#### 3.4.2.3 - Requête

```
-- Liste des employés et de leur rôle par point de vente
```



```
select point_de_vente.nom, employe.nom, role.libelle
from employe
left join point_de_vente on employe.point_de_vente_id = point_de_vente.id
inner join role on employe.role_id = role.id
order by point_de_vente.nom, employe.nom;
```

### 3.5 - Table client\_type

La table client\_type définit les différents types de client.

client_type		
PK	id libelle	SERIAL TEXT

#### 3.5.1 - Valeurs initiales

Les valeurs initiales ci-dessous sont enregistrées dans la base pour gérer les cas client internet, client aux téléphone et client aux comptoir.

Id	Libelle
1	CLIENT_INTERNET
2	CLIENT_TELEPHONE
3	CLIENT_COMPTOIR

#### 3.5.2 - Jeu de données

	id [PK] integer		libelle text	
1		1	CLIENT_INTERNET	
2		2	CLIENT_TELEPHONE	
3		3	CLIENT_COMPTOIRE	

### 3.6 - Table client

Cette table représente un client.

client		
PK	id	SERIAL
N	email	TEXT
N	password	TEXT
N	nom	TEXT
N	prenom	TEXT
N	telephone	TEXT
N	adresse	TEXT
FK	client_type_id	INTEGER
FK	point_de_vente_id	INTEGER

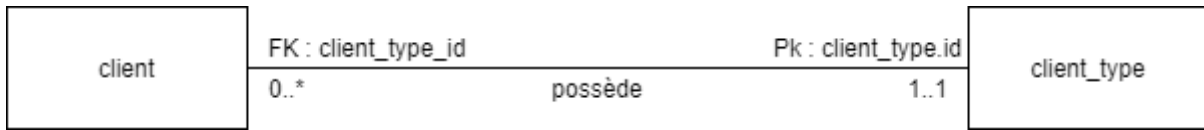
#### 3.6.1 - Champs obligatoires en fonction du type de client

Selon le type de client, internet, téléphone, comptoir certain champs ne sont pas nécessaires, c'est pour cela que les champs email, password, nom, prenom, telephone, adresse sont nullable. Dans le cas d'un client de type « comptoir » seul l'id est nécessaire.

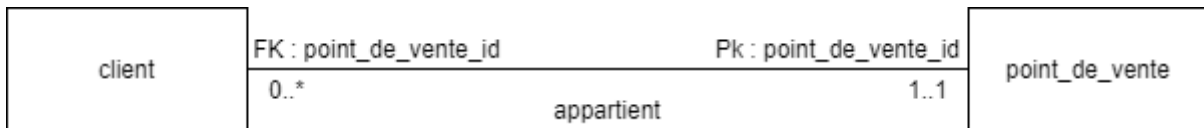
Client_type	Champs obligatoires
<b>CLIENT_INTERNET</b>	id, email, password, nom, prenom, telephone adresse
<b>CLIENT_TELEPHONE</b>	id, nom, prenom, adresse
<b>CLIENT_COMPTOIRE</b>	Id

#### 3.6.2 - Relations

- Un client possède un client\_type



- Un client appartient à un point de vente



### 3.6.3 - Jeu de données

#### 3.6.3.1 - Données brutes

	id [PK] integer	email text	password text	nom text	prenom text	telephone text	adresse text	client_type_id integer	point_de_vente_id integer
1	1	jean.dupont@gmail.com	03ac674216f3...	Dupont	Jean	0647124578	2 rue du Plat d'Etai...	1	1
2	2	alain.bernard@gmail.com	f8638b979b2f...	Bernard	Alain	0621568757	154 rue du Colonel ...	1	2
3	3	gabriel.thomas@orange.fr	80a2a51ee74...	Gabriel	Thomas	0621788754	5 rue Forest 37300 ...	1	2
4	4	leo.petit@hotmail.com	f8638b979b2f...	Léo	Petit	0254156875	70 rue du Père Brot...	1	3
5	5	alice.robert@orange.fr	80a2a51ee74...	Alice	Robert	0687788765	202 rue du Bourg S...	1	3
6	6	[null]	[null]	Chloé	Richard	0247558874	5 rue de la Préfectu...	2	1
7	7	[null]	[null]	Ethan	Durant	0247558874	8 bis rue de Luce 3...	2	1
8	8	[null]	[null]	Claire	Dubois	0247788874	70 rue Ampère 373...	2	2
9	9	[null]	[null]	Julia	Spears	0247625184	25 rue Augustin Thi...	2	3
10	10	[null]	[null]	[null]	[null]	[null]	[null]	3	1
11	11	[null]	[null]	[null]	[null]	[null]	[null]	3	1
12	12	[null]	[null]	[null]	[null]	[null]	[null]	3	2
13	13	[null]	[null]	[null]	[null]	[null]	[null]	3	2
14	14	[null]	[null]	[null]	[null]	[null]	[null]	3	3

### 3.6.3.2 - Liste des clients par point de vente et type de client

	client.id integer	nom text	libelle text	nom text	telephone text	adresse text	password text
1	1	Tours Grammont	CLIENT_INTERNET	Dupont	0647124578	2 rue du Plat d'Etain ...	03ac674216f3...
2	7	Tours Grammont	CLIENT_TELEPHONE	Ethan	0247558874	8 bis rue de Luce 370...	[null]
3	6	Tours Grammont	CLIENT_TELEPHONE	Chloé	0247558874	5 rue de la Préfecture...	[null]
4	11	Tours Grammont	CLIENT_COMPTOIRE	[null]	[null]	[null]	[null]
5	10	Tours Grammont	CLIENT_COMPTOIRE	[null]	[null]	[null]	[null]
6	3	Joué	CLIENT_INTERNET	Gabriel	0621788754	5 rue Forest 37300 J...	80a2a51ee74...
7	2	Joué	CLIENT_INTERNET	Bernard	0621568757	154 rue du Colonel Pi...	f8638b979b2f...
8	8	Joué	CLIENT_TELEPHONE	Claire	0247788874	70 rue Ampère 3730...	[null]
9	12	Joué	CLIENT_COMPTOIRE	[null]	[null]	[null]	[null]
10	13	Joué	CLIENT_COMPTOIRE	[null]	[null]	[null]	[null]
11	5	Blois	CLIENT_INTERNET	Alice	0687788765	202 rue du Bourg Sai...	80a2a51ee74...
12	4	Blois	CLIENT_INTERNET	Léo	0254156875	70 rue du Père Brotti...	f8638b979b2f...
13	9	Blois	CLIENT_TELEPHONE	Julia	0247625184	25 rue Augustin Thier...	[null]
14	14	Blois	CLIENT_COMPTOIRE	[null]	[null]	[null]	[null]

### 3.6.3.3 - Requête

```
-- Liste des clients avec leur type et leur point de vente
SELECT client.id as "client.id", point_de_vente.nom, client_type.libelle, client.nom,
client.telephone, client.adresse, client.password
FROM public.client
inner join point_de_vente on client.point_de_vente_id = point_de_vente.id
inner join client_type on client.client_type_id=client_type.id
order by point_de_vente.id, client_type.id;
```

## 3.7 - Table tva

Cette table définit les TVA possibles

tva		
PK	id	SERIAL
	libelle	TEXT
	taux	NUMERIC

### 3.7.1 - Valeurs initiales

Des valeurs initiales ci-dessous sont enregistrées dans la base pour gérer les cas prévus.

id	libelle	taux
1	Taux normal	20,00
2	Taux intermédiaire	10,00
3	Taux réduit	5,50

### 3.7.2 - Jeu de données

	id [PK] integer	libelle text	taux numeric
1	1	Taux normal	20
2	2	Taux intermédiaire	10
3	3	Taux réduit	5.50

## 3.8 - Table unite

Cette table définit les unités.

unite		
PK	id libelle symbole	SERIAL TEXT TEXT




### 3.8.1 - Valeurs initiales

Des valeurs initiales ci-dessous sont enregistrées dans la base pour gérer les cas prévus.

Id	Libelle	Symbole
1	Gramme	G
2	Kilogramme	Kg
3	Centilitre	cl
4	Décilitre	dl
5	Litre	L
6	Unité	U

### 3.8.2 -

### 3.8.3 - Jeu de données





	 id [PK] integer	 libelle text	 symbole text
1	1	Grammes	g
2	2	Kilogrammes	Kg
3	3	Centilitre	cl
4	4	Décilitre	dl
5	5	Litre	l
6	6	Unité	U

## 3.9 - Table recette

Cette table enregistre les recettes des pizzas. Le texte de la recette est enregistré dans le champ mode\_preparatoire.

recette		
PK	id	SERIAL
	nom	TEXT
N	mode_preparatoire	TEXT
N	photos	TEXT

### 3.9.1 - Jeu de données

	 id [PK] integer	 nom text	 mode_preparatoire text	 photos text
1	1	Margherita	Etaler la pâte. Etaler la sauce toma...	[null]
2	2	Régina	Etaler la pâte. Replier légèrement l...	[null]
3	3	Végétarienne	Sauce tomate, mozzarella, mélang...	[null]

### 3.10 - Table produit

La table produit enregistre les produits.

produit		
PK	id	SERIAL
	nom	TEXT
N	photos	TEXT
N	description	TEXT
N	prix_vente	MONEY
FK,N	tva_id	INTEGER
FK,N	unite_id	INTEGER

Deux types de produits existent

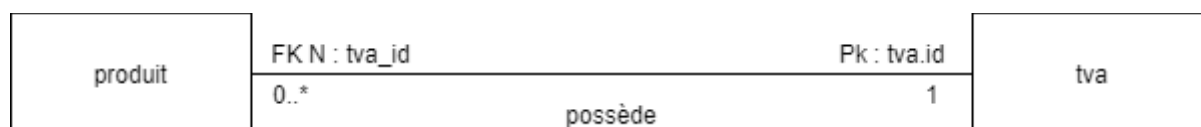
- Les ingrédients nécessaires à la réalisation des pâtes à pizza et des pizzas. Ces produits n'ont pas de prix de vente et pas de tva.
- Les produits destinés à la vente vins, sodas, glaces. Ces produits doivent avoir un prix de vente et une tva.

*Remarque* : Il n'y a pas de liaison directe entre la table produit et la table point\_de\_vente. Un produit est disponible pour l'ensemble du groupe. Les quantités de produit par point de vente sont gérés par la table produit\_point\_de\_vente.

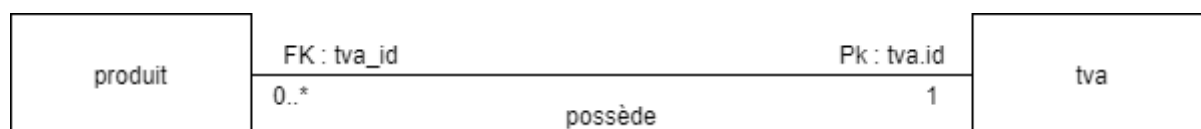
#### 3.10.1 - Relations

- Un produit peut posséder une tva.

Dans la table produit le champ tva\_id est nullable.



- Un produit possède une unité



#### 3.10.2 - Jeu de données

Id	Nom	Description	Prix_vente	Tva_id	Unite_id
1	Farine	Ingrédient de base pour fabriquer la pâte à pizza			2
2	Levure de boulanger	Ingrédient de base pour fabriquer la			1

		pâte à pizza			
3	Sel	Ingrédient de base pour fabriquer la pâte à pizza			2
4	Huile d'olive	Ingrédient de base pour fabriquer la pâte à pizza			5
5	Sucre				1
6	Pâte à pizza				1
7	Sauce tomate				1
8	Bouquet garni	Mélange thym, romarin et laurier			1
9	Origan	Origan en poudre			1
10	Poivre	Poivre en poudre			1
11	Mozzarella	Boule de mozzarella			1
12	Basilic	Feuille de Basilic			1
13	Crème fraîche				1
14	Jambon blanc	Jambon blanc en tranches			1
15	Roquette	Salade de roquette			1
16	Emmental râpé	Fromage râpé			1
17	Champignons de Paris	Champignons émincé			1
18	Olives noires	Olives noires dénoyautées			1
19	Olives verts	Olives vertes dénoyautées			1
20	Ail haché	Ail pré-haché prêt à utiliser			1
21	Piment de Cayenne	Piment en poudre			1
22	Jambon sec	Jambon sec en tranches			1
23	Chorizo	Chorizo tranché			1
24	Poivron	Poivron découpé			1
25	Tomate	Tomate découpé			1
26	Courgette	Courgette découpé			1
27	Oignon	Oignon découpé			1
28	Huile pimentée	Dosette 3ml			6
100	Lambrusco rosso	Vin rosé	7	1	5
101	Chianti Romio	Vin rouge	8,5	1	5
102	Montepulciano d'Abruzzo	Vin rouge	9,5	1	5
103	Le grand clos 2019	Bourgueil, domaine Ménard. Vin rouge	11,9	1	5
104	Coca cola	33cl	3	1	5
105	Bièrre	33cl	4,5	1	5
200	Glace vanille	Petit pot 80g	3	2	1
201	Glace chocolat	Petit pot 80g	3	2	1
202	Glace caramel	Petit pot 80g	3	2	1



### 3.11 - Table recette\_produit

Cette table enregistre les quantités d'ingrédients (produit) par recette.

recette_produit		
PK,FK	recette_id	INTEGER
PK,FK	produit_id	INTEGER
	quantite	NUMERIC

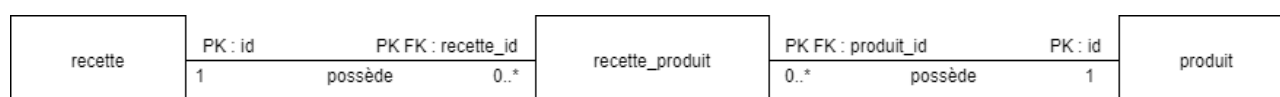
Le couple (recette\_id, produit\_id) permet de créer une relation entre les tables recette et produit.

Les champs recette\_id et produit\_id sont des clés primaires. Il est obligatoire d'avoir ces deux valeurs pour créer un enregistrement. Les doublons ne sont pas possibles.

Le champ quantité indique la quantité de produit nécessaire à la réalisation d'une recette.




Relation

- Une recette peut avoir plusieurs produits
- Un produit peut être présent dans différente recette



### 3.11.1 - Jeu de données

#### 3.11.1.1 - Données brutes

	 recette_id [PK] integer	 produit_id [PK] integer	 quantite numeric
1	1	7	50
2	1	11	100
3	1	12	10
4	2	7	50
5	2	9	10
6	2	11	50
7	2	14	50
8	2	15	10
9	2	16	50
10	2	17	50
11	2	18	10
12	2	20	10
13	2	27	50

### 3.11.1.2 - Liste des ingrédients par pizza

	nom text		nom text		quantite numeric	
1	Margherita		Basilic		10	
2	Margherita		Mozzarella		100	
3	Margherita		Sauce tomate		50	
4	Régina		Ail haché		10	
5	Régina		Champignons d...		50	
6	Régina		Emmental râpé		50	
7	Régina		Jambon blanc		50	
8	Régina		Mozzarella		50	
9	Régina		Oignon		50	
10	Régina		Olives noires		10	
11	Régina		Origan		10	
12	Régina		Roquette		10	
13	Régina		Sauce tomate		50	
14	Végétarienne		Courgette		25	
15	Végétarienne		Mozzarella		50	
16	Végétarienne		Oignon		50	
17	Végétarienne		Poivron		50	
18	Végétarienne		Sauce tomate		50	
19	Végétarienne		Tomate		50	

### 3.11.1.3 - Requête

```
-- ingrédients des recettes
select recette.nom, produit.nom, recette_produit.quantite
from recette_produit
inner join recette on recette_produit.recette_id = recette.id
inner join produit on recette_produit.produit_id = produit.id
order by recette.nom, produit.nom;
```

## 3.12 - Table pizza

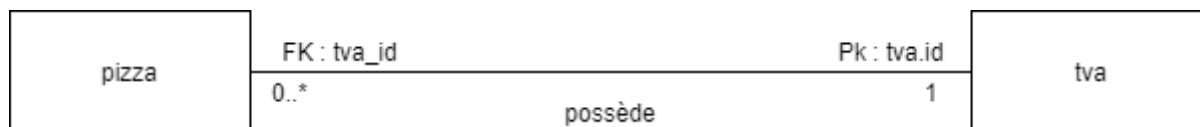
Liste des pizzas au menu

pizza		
PK	id	SERIAL
	nom	TEXT
N	photos	TEXT
N	description	TEXT
N	prix_vente	MONEY
FK,N	tva_id	INTEGER
FK,N	recette_id	INTEGER

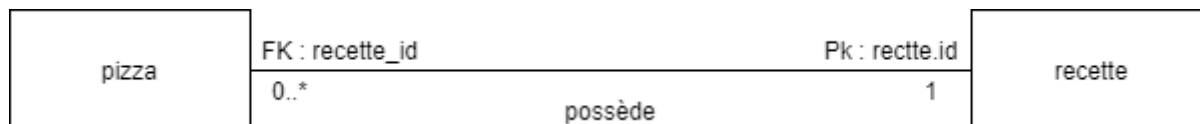
**Remarque** : Il n'y a pas de liaison directe entre la table pizza et la table point\_de\_vente. Une pizza est disponible pour l'ensemble du groupe.

### 3.12.1 - Relations

- Une pizza possède une tva.



- Une pizza possède une recette.



### 3.12.2 - Jeu de données

	id [PK] integer	nom text	photos text	description text	prix_vente money	tva_id integer	recette_id integer
1	1	Margherita	[null]	Pizza napoletana garnie ...	10,00 €	1	1
2	2	Régina	[null]	Un grand classique des p...	12,00 €	1	2
3	3	Végétarienne	[null]	Sauce tomate, mozzarell...	12,50 €	1	3

### 3.13 - Table panier

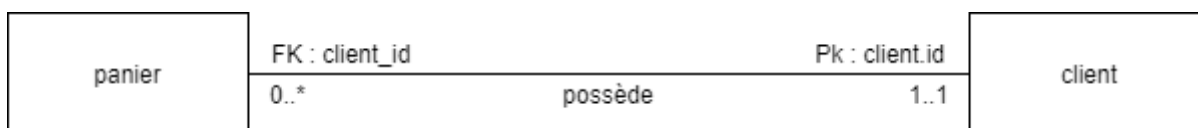
Cette table permet de créer un panier pour un client.

panier		
PK	id	SERIAL
FK	client_id	INTEGER
N	date_creation	DATETIME
N	total_ttc	NUMERIC

Le contenu des paniers est enregistré dans les tables panier\_pizza et panier\_produit.

#### 3.13.1 - Relation

- Un client possède plusieurs paniers
- Un panier appartient à un client



#### 3.13.2 - Jeu de données

	id [PK] integer	client_id integer	date_creation timestamp with time zone	total_ttc numeric
1	1	1	[null]	[null]
2	2	2	[null]	[null]
3	3	5	[null]	[null]
4	4	6	[null]	[null]
5	5	7	[null]	[null]
6	6	8	[null]	[null]
7	7	9	[null]	[null]
8	8	10	[null]	[null]
9	9	11	[null]	[null]
10	10	12	[null]	[null]
11	11	13	[null]	[null]
12	12	14	[null]	[null]

### 3.13.2.1 - Détail des paniers pizza et produit et quantité

	 id integer	 nom text	 quantite integer
1	1	Lambrusco rosso	1
2	1	Margherita	2
3	1	Régina	5
4	2	Régina	2
5	3	Régina	1
6	4	Chianti Romio	1
7	4	Végétarienne	4
8	5	Végétarienne	1
9	5	Régina	1
10	6	Margherita	1
11	7	Coca cola	2
12	7	Glace chocolat	2
13	7	Végétarienne	1
14	7	Margherita	1
15	7	Régina	1
16	8	Margherita	1
17	8	Glace caramel	1
18	8	Régina	1
19	8	Glace vanille	1
20	9	Biére	4
21	9	Margherita	1
22	10	Margherita	1
23	11	Végétarienne	1
24	11	Montepulciano d'Abruzzo	4
25	11	Margherita	4
26	11	Régina	3
27	12	Margherita	1

### 3.13.2.2 - Requête

```
-- Détail des paniers pizzas, produits et quantite
select panier.id, pizza.nom as nom, panier_pizza.quantite as quantite
from panier
inner join panier_pizza on panier.id = panier_pizza.panier_id
left join pizza on panier_pizza.pizza_id = pizza.id

union

select panier.id, produit.nom as nom, panier_produit.quantite as quantite
from panier
inner join panier_produit on panier.id = panier_produit.panier_id
left join produit on panier_produit.produit_id = produit.id

order by id
```

## 3.14 - Table panier\_pizza

Cette table enregistre les quantités de pizzas dans chaque panier.

panier_pizza		
PK,FK	panier_id	INTEGER
PK,FK	pizza_id	INTEGER
	quantite	INTEGER

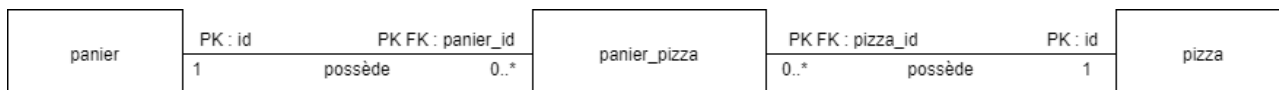
Le couple (panier\_id, pizza\_id) permet de créer une relation entre les tables panier et pizza.

Les champs panier\_id et pizza\_id sont des clés primaires. Il est obligatoire d'avoir ces deux valeurs pour créer un enregistrement. Les doublons ne sont pas possibles.




Le champ quantité indique la quantité de pizza ajoutée dans chaque panier.

### 3.14.1 - Relations

- Un panier peut avoir plusieurs pizzas
- Une pizza peut être présente dans différents paniers



### 3.14.2 - Jeu de données

	 panier_id [PK] integer	 pizza_id [PK] integer	 quantite integer
1	1	1	2
2	1	2	5
3	2	2	2
4	3	2	1
5	4	3	4
6	5	2	1
7	5	3	1
8	6	1	1
9	7	1	1
10	7	2	1
11	7	3	1
12	8	1	1
13	8	2	1
14	9	1	1
15	10	1	1
16	11	1	4
17	11	2	3
18	11	3	1
19	12	1	1



### 3.15 - Table panier\_produit

Cette table enregistre les quantités de produit dans chaque panier.

panier_produit		
PK,FK	panier_id	INTEGER
PK,FK	produit_id	INTEGER
	quantite	INTEGER

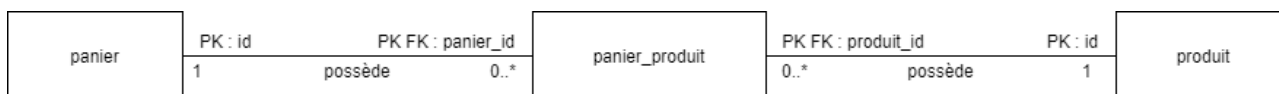
Le couple (panier\_id, produit\_id) permet de créer une relation entre les tables panier et produit.

Les champs panier\_id et produit\_id sont des clés primaires. Il est obligatoire d'avoir ces deux valeurs pour créer un enregistrement. Les doublons ne sont pas possibles.

Le champ quantité indique la quantité de produit ajoutée dans chaque panier.

#### 3.15.1 - Relations

- Un panier peut avoir plusieurs produits
- Un produit peut être présent dans différentes recettes



#### 3.15.2 - Jeu de données

	panier_id [PK] integer	produit_id [PK] integer	quantite integer
1	1	100	1
2	4	101	1
3	7	104	2
4	7	201	2
5	8	200	1
6	8	202	1
7	9	105	4
8	11	102	4

### 3.16 - Table produit\_point\_de\_vente

Cette table enregistre les quantités de produit par point de vente

produit_point_de_vente		
PK,FK	produit_id	INTEGER
PK,FK	point_de_vente_id	INTEGER
	quantite	NUMERIC

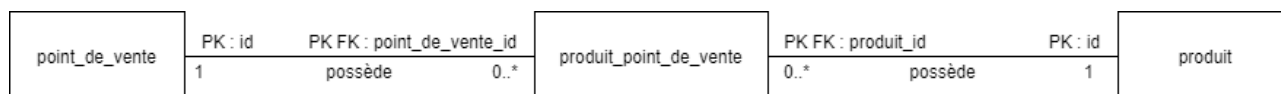
Le couple (produit\_id, point\_de\_vente\_id) permet de créer une relation entre les tables point\_de\_vente et produit.

Les champs produit\_id et point\_de\_vente\_id sont des clés primaires. Il est obligatoire d'avoir ces deux valeurs pour créer un enregistrement. Les doublons ne sont pas possibles.

Le champ quantité indique la quantité de produit disponible dans un point de vente.

#### 3.16.1 - Relation

- Un point de vente peut avoir plusieurs produits
- Un produit peut être présent dans différent point de vente



#### 3.16.2 - Jeu de données

##### 3.16.2.1 - Données brutes

produit_id	point_de_vente_id	quantite
1	1	50
2	1	500
3	1	4
4	1	7
5	1	5000
6	1	8000
7	1	5000
8	1	250
9	1	300
10	1	200
11	1	5000
12	1	0
13	1	2000
14	1	4500
15	1	500

16	1	2500
17	1	1400
18	1	600
19	1	800
20	1	1100
21	1	400
22	1	3000
23	1	3300
24	1	1500
25	1	1500
26	1	0
27	1	5000
28	1	120
100	1	10
101	1	12
102	1	5
103	1	6
104	1	20
105	1	25
200	1	15
201	1	11
202	1	14
1	2	50
2	2	500
3	2	4
4	2	7
5	2	5000
6	2	8000
7	2	5000
8	2	250
9	2	300
10	2	200
11	2	5000
12	2	0
13	2	2000
14	2	4500
15	2	500
16	2	2500
17	2	1400
18	2	600
19	2	800
20	2	1100



### IT Consulting & Development

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Xxxx – SIREN 999 999 999 – Code APE : 6202A

21	2	400
22	2	3000
23	2	3300
24	2	1500
25	2	1500
26	2	0
27	2	5000
28	2	120
100	2	10
101	2	12
102	2	5
103	2	6
104	2	20
105	2	25
200	2	15
201	2	11
202	2	14

### 3.16.2.2 - Etat du stock par produit, point de vente, quantité et unité

Limité à 15 enregistrements

	 nom text	 nom text	 quantite numeric	 symbole text
1	Ail haché	Joué	1100	g
2	Ail haché	Tours Grammont	1100	g
3	Basilic	Joué	0	g
4	Basilic	Tours Grammont	0	g
5	Biére	Joué	25	l
6	Biére	Tours Grammont	25	l
7	Bouquet garni	Joué	250	g
8	Bouquet garni	Tours Grammont	250	g
9	Champignons de Paris	Joué	1400	g
10	Champignons de Paris	Tours Grammont	1400	g
11	Chianti Romio	Joué	12	l
12	Chianti Romio	Tours Grammont	12	l
13	Chorizo	Joué	3300	g
14	Chorizo	Tours Grammont	3300	g
15	Coca cola	Joué	20	l

### 3.16.2.3 - Requête

```
-- Etat du stock par point de vente
SELECT produit.nom, point_de_vente.nom, quantite, unite.symbole
FROM public.produit_point_de_vente
inner join produit on produit.id = produit_point_de_vente.produit_id
inner join unite on unite.id = produit.unite_id
inner join point_de_vente on point_de_vente.id = produit_point_de_vente.point_de_vente_id
order by produit.nom, point_de_vente.nom
```

## 3.17 - Table status\_commande

Cette table définit les statuts possibles d'une commande

status_commande		
PK	id libelle	INTEGER TEXT

### 3.17.1 - Valeurs initiales

Des valeurs initiales ci-dessous sont enregistrées dans la base pour gérer les cas prévus.

Id	Libelle
1	A_VERIFIER
2	IMPOSSIBLE
3	A_PREPARER
4	EN_PREPARATION
5	PREPARER
6	EN_ATTENTE_DE_LIVRAISON
7	EN_COURS_DE_LIVRAISON
9	ANNULEE

Jeu de données

## 3.18 - Table commande

Cette table enregistre les commandes des clients et leur avancement

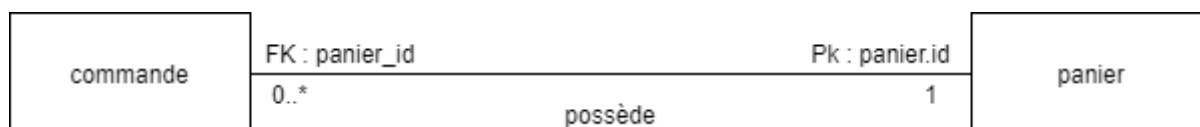
commande		
PK	id	SERIAL
FK	panier_id	INTEGER
N	date_creation	DATETIME
	payement_effectue	BOOLEAN
	livraison_effectue	BOOLEAN
N	date_payement	DATETIME
FK,N	livreur_id	INTEGER
FK	statut_commande_id	INTEGER
	livraison_domicile	BOOLEAN
	payement_livraison	BOOLEAN
N	mt_ttc	NUMERIC
N	mt_tva	NUMERIC

Dans cette table on trouve :

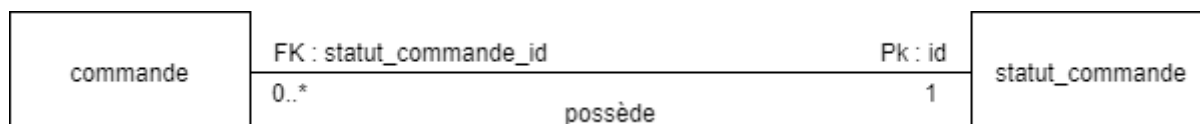
- **La liaison avec le client** qui se fait par la table panier qui contient le contenu de la commande pizzas et/ou produits, *panier\_id*.
- **L'état de la livraison**, *livraison\_domicile*, *livraison\_effectue*, *livreur\_id*.
- **L'état du paiement**, *payement\_livraison*, *payement\_effectue*, *date\_payement*, *mt\_ttc*, *mt\_tva*.
- **L'état d'avancement**, *statut\_commande\_id*.

### 3.18.1 - Relation

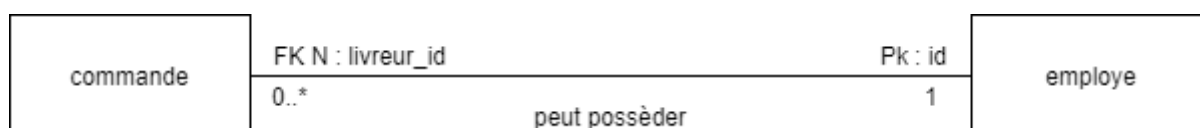
- Une commande possède un panier.








- Une commande possède un statut



- Une commande peut avoir un livreur



### 3.18.2 - Jeu de données

	 id integer	 point text	 client text	 statut text	 mt_ttc numeric
1	1	Tours Grammont	Dupont	A_PREPARER	111.00
2	2	Joué	Bernard	A_PREPARER	12.00
3	3	Blois	Alice	EN_COURS_DE_LIVRAISON	12.00
4	4	Tours Grammont	Chloé	EN_ATTENTE_DE_LIVRAISON	58.50
5	5	Tours Grammont	Ethan	EN_COURS_DE_LIVRAISON	24.50
6	6	Joué	Claire	EN_COURS_DE_LIVRAISON	12.00
7	7	Blois	Julia	EN_ATTENTE_DE_LIVRAISON	46.50
8	8	Tours Grammont	[null]	A_PREPARER	28.00
9	9	Tours Grammont	[null]	EN_PREPARATION	28.00
10	10	Joué	[null]	EN_PREPARATION	10.00
11	11	Joué	[null]	EN_PREPARATION	126.50
12	12	Blois	[null]	EN_PREPARATION	10.00

### 3.18.3 - Requête

```
-- Etat des commandes avec statut, nom du client et nom du point de vente
SELECT commande.id, point_de_vente.nom as point, client.nom as client,
statut_commande.libelle as statut, mt_ttc
FROM public.commande
inner join statut_commande on statut_commande.id = statut_commande_id
inner join panier on panier.id = commande.panier_id
inner join client on client.id = panier.client_id
inner join point_de_vente on point_de_vente.id = client.point_de_vente_id
```

## 4 - OPERATIONS SUR LA BASE DE DONNEES

### 4.1 - Mise en place de la base de données ocpizza

- Les opérations sur le schéma, création et suppression se font avec les commandes *createdb* *dropdb*.
- Les opérations sur la structure et les données se font avec la commande *psql*.

#### 4.1.1 - Suppression du schema ocpizza

*dropdb* --username=postgres ocpizza

#### 4.1.2 - Création du schema ocpizza

*createdb* --username=postgres ocpizza

#### 4.1.3 - Suppression des tables

*psql* --username=postgres ocpizza <  
"c:\OpenClassRooms\tp\P8\SQL\DDL\mydata\_table\_drop.sql"

#### 4.1.4 - Création des tables

*psql* --username=postgres ocpizza <  
"c:\OpenClassRooms\tp\P8\SQL\DDL\mydata\_table\_create.sql"

#### 4.1.5 - Ajout du jeu de données

*psql* --username=postgres ocpizza < "c:\OpenClassRooms\tp\P8\SQL\Insert SQL\insert.sql"