



# **CRÉATION D'UN SITE E-COMMERCE**

**POUR UNE ARTISTE QUI VEUX VENDRE SES OEUVRES  
ET DES PRODUITS DÉRIVÉS**



Hortalia

# AVANT-PROPOS

Titre du projet : création d'un site e-commerce pour une artiste qui veut vendre ses œuvres et des produits dérivés

L'objectif est de réaliser un site e-commerce pour une artiste qui souhaite vendre ses œuvres et des produits dérivés de ses œuvres en ligne. Pour cela nous allons rédiger dans ce dossier un cahier des charges fonctionnelles ainsi qu'un cahier de conception technique de la base de données.

Ce projet est réalisé dans le cadre d'un devoir pour la mise en application du cours de bases de données au sein du conservatoire national des arts et métiers (2ème année).

1



Hortalia

# CAHIER DES CHARGES

FONCTIONNELLES



# Sommaire

<b>1. Contexte du projet.....</b>	<b>5</b>
<b>2. Problématiques.....</b>	<b>5</b>
<b>3. Les besoins exprimées.....</b>	<b>6</b>
<b>4. Les livrables attendus.....</b>	<b>7</b>
<b>5. Les solutions.....</b>	<b>8</b>
4.1. Solutions fonctionnelles.....	8
4.2. Solutions techniques.....	8
4.3. Livrables.....	9



## 1. Contexte du projet et objectifs

### Quel est le contexte du projet et les objectifs à atteindre ?

Le projet consiste à réaliser un site e-commerce pour l'artiste [@hortalia](https://www.instagram.com/hortalia) qui poste ses œuvres depuis 2014 sur la plateforme Instagram ( <https://www.instagram.com/hortalia> ) où elle compte plus de 1000 abonnées. Au départ inspirée par l'univers japonais des mangas, elle crée son propre style qui se rapproche aujourd'hui plus ou moins à celui des comics. Elle dessine principalement des personnages (portraits ou corps entier) en travaillant sur les expressions et le sentiment général que l'on ressent face à l'illustration.

Après sept ans de présence sur les réseaux sociaux et quelques périodes de « disparition » pour se consacrer à ses études, elle souhaite aujourd'hui à l'âge de 21 ans générer des revenus grâce à sa passion. À cette période de sa vie où elle cherche à trouver sa voix, ce site e-commerce lui permettra de donner un grand élan vers la création de sa propre entreprise. L'objectif principal de la réalisation de ce site étant l'automatisation des ventes et la gestion de ses produits et de son chiffre d'affaires. Les cibles principales de ce projet sont les amateurs d'art et de ses œuvres en particulier, ainsi que ceux qui sont à la recherche d'un cadeau à offrir à un proche ou à eux-mêmes.

## 2. Problématique

### Quelles sont les problématiques rencontrées ?

Jusqu'à présent elle faisait des ventes de ses illustrations digitales sans produit associés avec des clients de manière directe ou en passant par des plateformes dédiées (ex: Fiverr, Society6, Redbubble). Ces plateformes représentent une réelle concurrence pour l'artiste. En effet avant de valider sa commande un client peut avoir plusieurs suggestions d'autres artistes pouvant réaliser la même demande. Avoir son propre espace lui permettra d'avoir le monopole et de bâtir une meilleure relation de confiance avec ses clients qu'elle connaîtra car elle pourra enfin avoir un aperçu de leur nombre et de leurs caractéristiques (ex : sexe, âge, localisation ext ...) pour pouvoir mieux répondre à leurs attentes.



Une autre problématique de ces sites est bien entendu le pourcentage qu'ils prennent sur les ventes qui est très conséquent. Par exemple sur des plateformes comme Redbubble ou Society6 l'artiste ne récupère que 10 à 20% des ventes réalisées. Même si cela lui évite de prendre en considération l'achat des produits vierges, l'impression et de la livraison au client elle pense pouvoir se payer mieux en gérant tout par elle-même. L'artiste souhaiterait avoir une meilleure maîtrise des tarifs qu'elle affiche sans avoir à se soucier des marges conséquentes que ces plateformes récupèrent.

Les plateformes sur lesquels elle passait ne lui apportaient aucun véritables outils de suivi. Au-delà de la création d'un site de vente qui de par la diversité de ses produits doit lui être beaucoup plus rentable elle souhaite avoir un véritable système d'information qui lui permettra de gérer son entreprise. Elle souhaiterait ainsi répondre aux problématiques suivantes :

- Comment gérer efficacement les stocks et les commandes pour ne jamais être à court de produits sans avoir des stocks d'inventaire importants ?
- Quels sont les produits les plus vendus selon la période de l'année et comment améliorer les ventes ?
- Comment gérer efficacement le listing des clients pour les pousser à faire des achats en ligne ?

### 3. Les besoins exprimés

#### Quels sont les besoins qui découlent de ces problématiques ?

L'artiste souhaiterait pouvoir poster ses dessins digitaux sur une plateforme en ligne où elle pourrait les trier par catégorie et les assigner à des produits dérivés qui seront vendus et livrés à des clients après avoir réalisé une commande sur son site internet. Elle aimerait de même pouvoir gérer les fournisseurs qui seront chargés d'imprimer ses illustrations sur des produits (posters, tableaux, tee-shirts, sweatshirt, tote bags, coque de téléphone) . Un fournisseur peu lui vendre un type de produits ou plusieurs.

Pour pouvoir gérer sa comptabilité elle voudrait d'une part pouvoir enregistrer les factures faites avec ces fournisseurs pour l'achat de nouveaux produits pour alimenter ses stocks et d'une autre suivre l'évolutions de ses ventes (produits les plus vendu, chiffre d'affaire mensuel, ext ...). Enfin, elle a exprimé son besoin de suivre la quantité de chaque produits en réserve afin d'être alerté en cas de rupture de stock.





## Les contraintes

- La solution doit pouvoir être accessible même sans connaissances en informatique pour qu'elle puisse être gérée par n'importe quelles personnes
- La solution doit pouvoir fonctionner dans un environnement Windows, Mac, Linux

## 4. Les livrables

Quels sont les livrables et résultats attendus ?

3 livrables sont attendus :

- La solution *packagée* dans des conteneurs type Docker
- Le cahier des charges fonctionnel et technique
- La documentation contenant les procédures d'installation et de déploiement (fichier `readme.md` à la racine du projet)



## 5. Solutions

### 5.1. Solutions fonctionnelles

Quelles sont les solutions fonctionnelles pour répondre aux besoins ?

Les solutions fonctionnelles pour répondre à la création de ce site e-commerce sont :

- Un système de gestion des produits et des stocks disponibles
  - Liste de tous les produits avec leurs caractéristiques (prix, fournisseur, etc ...)
  - Possibilité de voir les produits vendus, ceux en rupture de stock et ceux encore disponibles
- Un système de gestion des factures faites pour les fournisseurs
  - Liste de tous les fournisseurs et des produits achetés chez eux
  - Liste de toutes les factures d'achats de produits
- Gestion des clients et des commandes clients
  - Gestion des inscriptions sur site et réalisation de commande côté client
  - Liste de tous les clients et de leurs achats sur le site
  - Liste de toutes les commandes réalisées par les clients pouvant être triés par dates ou montant
- Gestion des illustrations disponibles
  - Ajout et suppression d'illustrations en spécifiant leurs types

### 5.2. Solutions techniques

Quelles sont les solutions techniques et sont-elles adaptées aux besoins ?

D'un point de vue technique, nous proposons d'intégrer des technologies accessibles et reconnus, performantes et peu onéreuses :

Pour la base de donnée on opte pour une base de donnée PostgreSQL qui permettra d'assurer la manipulation des données (modifications, suppression, calcul etc..). Accessible au grand public depuis une vingtaine d'année maintenant, PostgreSQL est libre de droit et sans coût de licence.





Nous utiliserons le **serveur web Apache** pour l'affichage de notre interface web. Avec ses nombreux modules disponibles, Apache est aujourd'hui le serveur le plus populaire.

Concernant la génération dynamique des pages web, nous opterons pour le langage de programmation **PHP**. Ce choix s'explique par sa simplicité et sa rapidité de mise en œuvre pour la création de site web. PHP possède de plus des bibliothèques très fournies.

Nous installerons le **framework Symfony** pour la création de notre projet pour toutes les fonctionnalités qu'il propose, la sécurité qu'il apporte et l'environnement de travail qu'il propose.

Enfin, afin de pouvoir déplacer le site internet et son backend d'administration sur n'importe quelle machine sans avoir à se soucier des versions et des paquets installés. Nous utiliserons **Docker** pour le packaging de l'application dans un environnement maîtrisé et cloisonné.

### 5.3. Livrables

**Quels sont les livrables logiciels prévus ? Détaillez par rapport aux livrables exprimés par le Client qui peuvent être génériques**

Le livrable est un environnement Docker contenant l'ensemble des services, base de données et IHMs. Le livrable pourra s'installer sur une machine GNU/Linux avec Docker et docker-compose installés.

Le fichier README décrit la procédure d'installation de l'application.

- Stopper les services : docker-compose down
- Redémarrer les services : docker-compose up

2



Hortalia

# CAHIER DE CONCEPTION

TECHNIQUE



# Sommaire

<b>1.</b>	<b>Dictionnaire des données.....</b>	<b>12</b>
<b>2.</b>	<b>Conception du modèle entité association.....</b>	<b>13</b>
2.1.	Matrice simplifiée des dépendances fonctionnelles.....	13
2.2.	Liste des dépendances fonctionnelles.....	15
2.3.	Modèle Entité Association (MEA).....	18
<b>3.</b>	<b>Approche relationnelle des données.....</b>	<b>19</b>
3.1.	Schémas relationnel.....	19
3.2.	Modèle relationnel des données (MDC) .....	20
<b>4.</b>	<b>Contraintes particulières.....</b>	<b>21</b>
4.1.	Contraintes de domaine.....	21
4.2.	Autres contraintes définis par tables .....	22
<b>5.</b>	<b>Liste des fonctionnalités prises en charge par votre BD.....</b>	<b>25</b>
5.1.	Fonctions & trigger.....	25
5.2.	Procédures.....	26
5.3.	Vues.....	27



## 1. Dictionnaire des données

Quels sont les données essentielles ?

n°	Label données	Type
1	refProduit	VARCHAR(255)
2	labelProduit	VARCHAR(255)
3	typeProduit	VARCHAR(60)
4	descriptionProduit	TEXT
5	imageProduit	VARCHAR(255)
6	prixProduitCdeUser	FLOAT
7	stockProduit	INT
8	labelIllust	VARCHAR(255)
9	typeIllust	VARCHAR(40)
10	descriptionIllust	TEXT
11	imageillust	VARCHAR(255)
12	refCdeUser	INT
13	dateCdeUser	DATE
14	QteProduitCdeUser	INT
15	mailUser	VARCHAR(255)
16	nomUser	VARCHAR(255)
17	prenomUser	VARCHAR(60)
18	dateNaissanceUser	DATE
19	paysUser	VARCHAR(50)
20	villeUser	VARCHAR(50)
21	cpUser	VARCHAR(15)
22	adresseUser	TEXT
23	telUser	VARCHAR(25)
24	roleUser	VARCHAR(40)

n°	Label données	Type
25	mailFournisseur	VARCHAR(255)
26	labelFournisseur	VARCHAR(255)
27	CodePaysFournisseur	VARCHAR(5)
28	refProduitFournisseur	INT
29	labelProduitFournisseur	VARCHAR(255)
30	typeProduitFournisseur	VARCHAR(60)
31	prixProduitFournisseur	FLOAT
32	refAchatFournisseur	INT
33	dateAchatFournisseur	DATE
34	QteAchatFournisseur	INT
35	refLivraison	INT
36	labelTransporteur	VARCHAR(255)
37	tarifLivraison	FLOAT
38	dateLivraison	DATE



Clés primaires



## 2. Conception du Modèle Entité Association (MEA)

Comment les données sont-elles liées entre elles ?

### 2.1. Matrice simplifiée des dépendances fonctionnelles (page 1)

x/y	1	8	12	15	25	28	32	35	1+12	1+32	n°	Label données
1	X		X				X				1	refProduit
2	X		X				X				2	labelProduit
3	X		X				X				3	typeProduit
4	X		X				X				4	descriptionProduit
5	X		X				X				5	imageProduit
6	X		X				X				6	prixProduitCdeUser
7	X		X				X				7	stockProduit
8	X	x									8	labelIllust
9	X	x									9	typeIllust
10	X	x									10	descriptionIllust
11	x	x									11	imageillust
12			x					X			12	refCdeUser
13			X					X			13	dateCdeUser
14									x		14	QteProduitCdeUser
15			x	x							15	mailUser
16			x	X							16	nomUser
17			x	X							17	prenomUser
18			x	X							18	dateNaissanceUser
19			X	X							19	paysUser
20			X	X							20	villeUser
21			x	x							21	cpUser





## 2.1 Matrice simplifiée des dépendances fonctionnelles (page 2)

x/y	1	8	12	15	25	28	32	35	1 + 12	1 + 32
22			x	X						
23			x	X						
24			x	x						
25	X				X	x	X			
26	X				X	X	X			
27	X				X	X	x			
28						x				
29						X				
30						X				
31						x				
32	X						X			
33	X						X			
34										x
35			X					X		
36			X					X		
37			X					X		
38			x					x		

n°	Label données
22	adresseUser
23	telUser
24	roleUser
25	mailFournisseur
26	labelFournisseur
27	CodePaysFournisseur
28	refProduitFournisseur
29	labelProduitFournisseur
30	typeProduitFournisseur
31	prixProduitFournisseur
32	refAchatFournisseur
33	dateAchatFournisseur
34	QteAchatFournisseur
35	refLivraison
36	labelTransporteur
37	tarifLivraison
38	dateLivraison





## 2.2. Liste des dépendances fonctionnelles



Suppression des  
dépendances  
fonctionnelles réflexives



Suppression des  
dépendances  
fonctionnelles  
transitives



Dépendances  
fonctionnelles utiles  
pour construction du  
MEA



Hortalia

# Cahier de conception

Produit	Illustration	Commande	User
#Produit	#Illust	#Produit	#User
1 -> 1	8 -> 8	12-> 1	15 -> 15
1 -> 2	8 -> 9	12 -> 2	15 -> 16
1-> 3	8-> 10	12-> 3	15 -> 17
1 -> 4	8-> 11	12 -> 4	15 -> 18
1 -> 5		12 -> 5	15 -> 19
1 -> 6		12 -> 6	15 -> 20
1 -> 7		12 -> 7	15 -> 21
#Illust		#Cde user	15 -> 22
1 -> 8		12 -> 12	15 -> 23
1 -> 9		12 -> 13	15 -> 24
1 -> 10		#User	
1 -> 11		12 -> 15	
		12 -> 16	
		12 -> 17	
		12 -> 18	
		12 -> 19	
		12 -> 20	
		12 -> 21	
		12 -> 22	
		12 -> 23	
		12 -> 24	

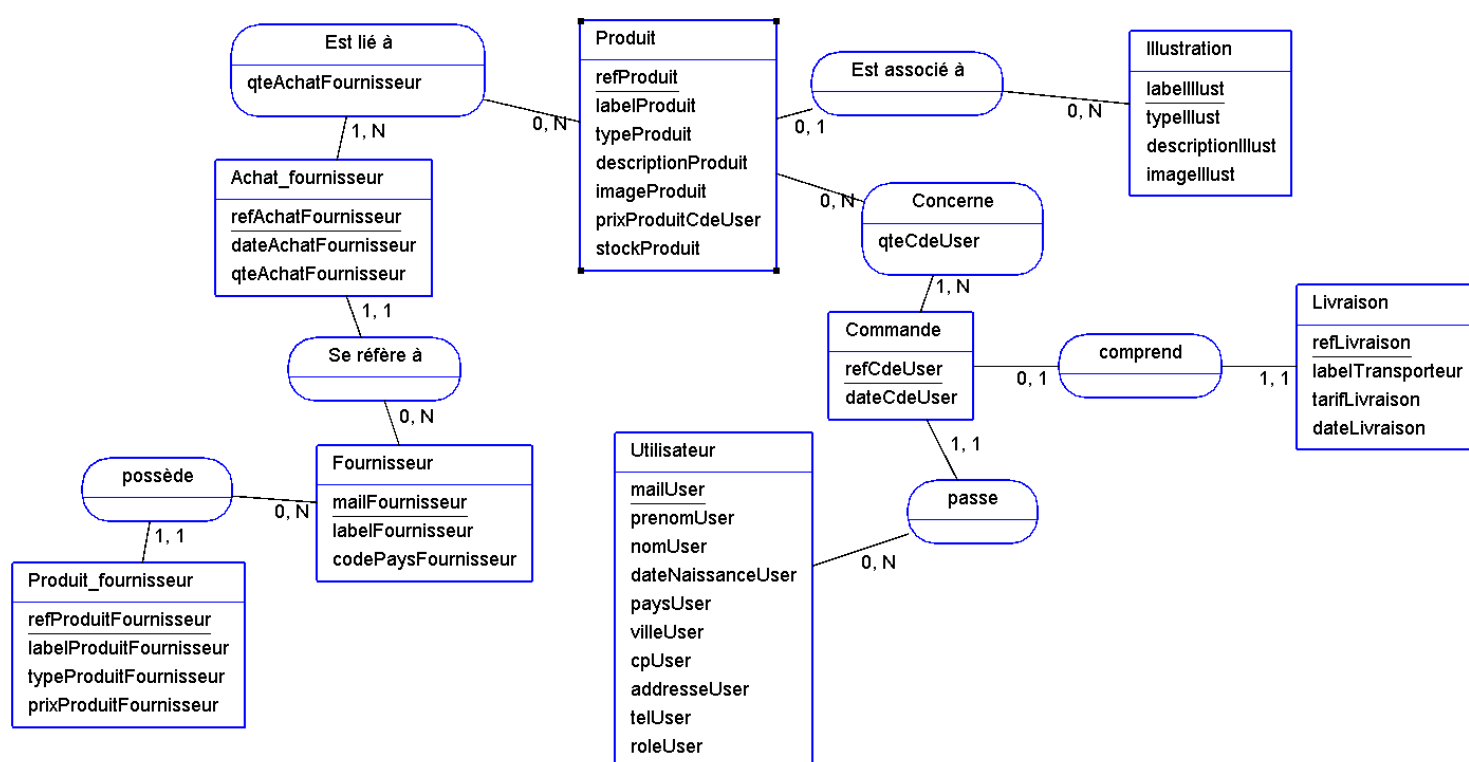


# Cahier de conception

[illegible]



## 2.3. Modèle Entité Association (MEA)



### Note 1

Deux tables « Produit » sont créées car la table **Produit** fait référence aux produits que possède la boutique et sur lesquels sont assignés des illustrations.

La table **Produit fournisseur** quant à elle représente les produits « vierges » sans impressions d'illustration proposée par les fournisseurs. (certains ont déjà été acheté par la boutique et d'autres non)

### Note 2

Un produit peut avoir qu'une illustration car la cliente a spécifié sa volonté de créer un produit à part entière pour chaque illustration différente (de telle sorte qu'elle puisse faire évoluer les tarifs des produits en fonction de l'illustration, modifier l'affichage des produits ou autres paramètres).



## 3. Approche relationnelle des données

Quel est l'autre modèle de représentation ?

### 3.1. Schémas relationnel

**Produit** ( refProduit, labelProduit, typeProduit, descriptionProduit, imageProduit, prixProduitCdeUser, stockProduit, #labelIllust)

**Illustration** (labelIllust, typeIllust, descriptionIllust, imageIllust)

**Commande** (refCdeUser, dateCdeUser, #mailUser)

**Livraison** (refLivraison, labelTransporteur, tarifLivraison, dateLivraison, #refCdeUser)

**Utilisateur** ( mailUser, nomUser, prenomUser, dateNaissanceUser, paysUser, villeUser , cpUser, adresseUser, telUser, roleUser)

**Fournisseur** (mailFournisseur, labelFournisseur, CdePayspaysFournisseur)

**Produit\_fournisseur** (refProduitFournisseur, labelProduitFournisseur, typeProduitFournisseur, prixProduitFournisseur, #refFournisseur )

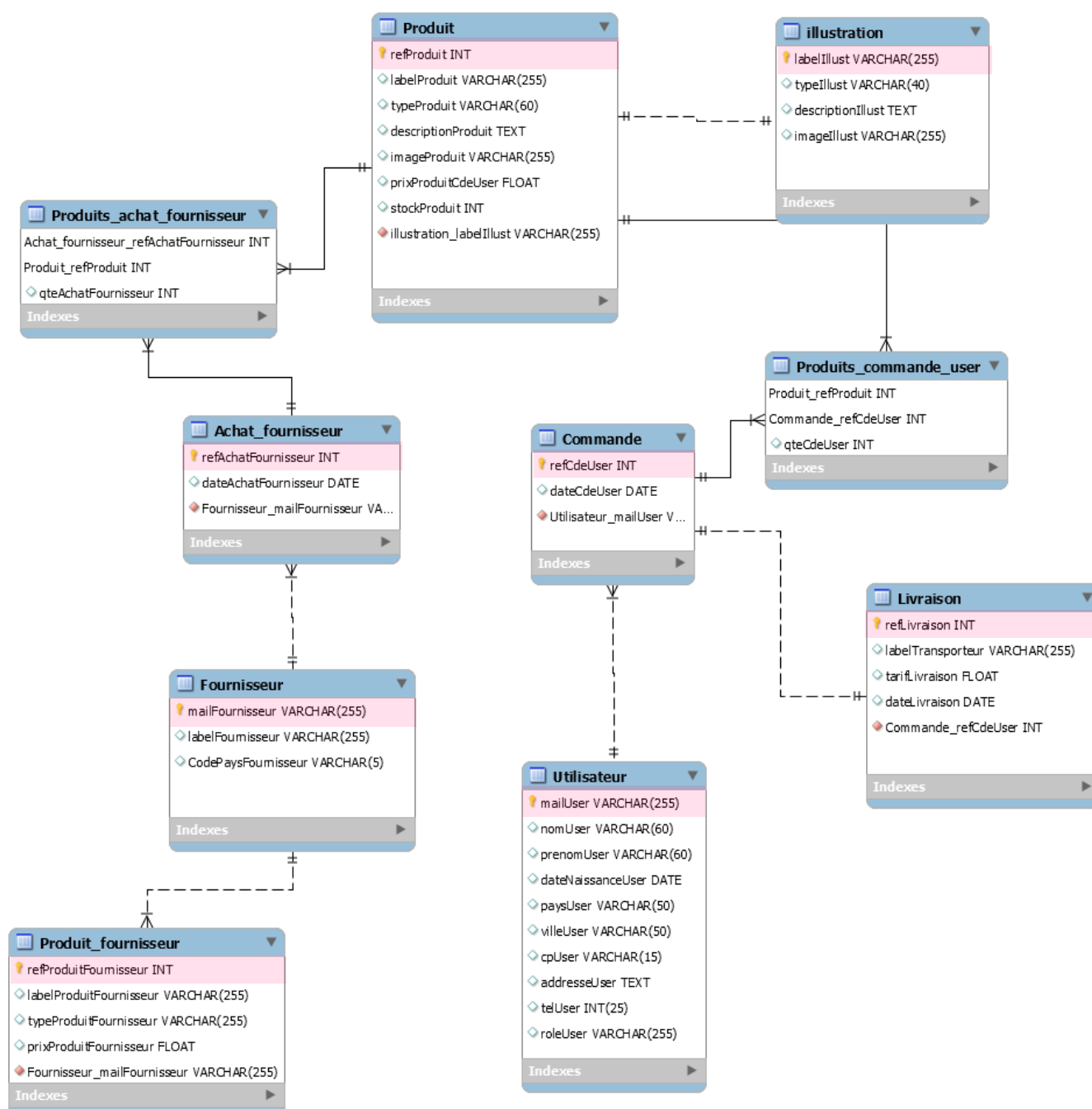
**Achat\_fournisseur** (refAchatFournisseur, dateAchatFournisseur, #refFournisseur)

**Produits\_Commande\_User** (#refCdeUser, #refProduit, qteCdeUser)

**Produits\_Achat\_Fournisseur** (#refAchatFournisseur, #refProduit, qteAchatFournisseur)



## 3.2. Modèle relationnel données (MCD)







## 4. Contraintes particulières

Il est maintenant temps de définir quels sont les contraintes d'intégrité de la base de données. On rappelle qu'une contrainte d'intégrité est une règle qui définit la cohérence d'une donnée ou d'un ensemble de données. Les contraintes sont définies lors de la création des tables et peuvent être appliquées sur une colonne unique ou sur une table lorsque la contrainte porte sur une ou plusieurs colonnes.

On définit plusieurs types de contraintes.

### 4.1. Contraintes de domaine

On a défini nos types de données lors de la création de notre dictionnaire de données en précisant lorsque cela est nécessaire le nombre de caractère maximal de la donnée. Toutes les données existantes et futures respectent les contraintes précisées ci-dessous.

Tous les champs date sont au format « DATE » (ex: dateAchatFournisseur, dateCdeUser, dateLivraison, etc..). Pour les dates qui correspondent à la date et à l'heure actuelle on mettra dans les champs pour valeur par défaut la fonction « now() ».

Toutes les références sont en format « INT » en auto-increment (attribut SERIAL sous postgresql) pour qu'elles se génèrent toutes seules sans avoir à les remplir (ex: refCdeUser, refAchatFournisseur, refLivraison, ect ...).

Tous les prix sont en format « FLOAT » pour pouvoir réaliser des calculs et être très précis (ex: Prix produit, PrixProduitFournisseur, tarifLivraison, ect ...)

Pour les longs textes on spécifie le format «TEXT» (ex: descriptionProduit, **descriptionIllustration**, adresseUser, etc ...)

Pour toutes données qui traites des chaînes de caractère relativement courtes on opte pour le type de donnée « VARCHAR » (ex : nomUser, mailUser, typeProduitFournisseur, etc...)

- Pour les codes postales (cp) on donne 15 caractères maximum
- Pour les téléphones 25 cactères maximum
- Pour les villes et pays on donne 50 caractères maximum
- Pour les noms et prénoms ont donne 60 caractères maximum



- Pour les types 40 caractères maximum
- Pour les labels, mails, images on donne la valeur maximale qui est à 255 caractères

Pour les longs textes de description (ex: descriptionProduit, descriptionIllust) on attribut le type TEXT.

Enfin, pour toutes les données qui traitent de quantité on opte pour le format « INT » (ex: qteCdeUser, qteAchatFournisseur)

## 4.2. Autres contraintes définis par tables

### Table illustration :

- (CLÉ PRIMAIRE) : chaque illustration doit avoir un label d'illustration unique qui ne peut pas être nulle
- (NON NULL) : le label, le type et l'image de l'illustration ne peuvent pas être null
- (CHECK) : les types d'illustrations ne peuvent prendre uniquement les valeurs 'sketch', 'crayon gris', 'crayon de couleurs', 'encre', 'acrylique', 'aquarelle', 'digitale')

### Table Produit :

- (CLÉ PRIMAIRE) : chaque produit doit avoir une référence de produit unique qui ne peut pas être nulle
- (NON NULL) : la référence, le label, le type et le prix du produit ne peuvent pas être null.
- (UNIQUE) : un produit ne peut pas avoir le même label et la même illustration associée. La combinaison de ces deux champs doit être unique.
- (CLÉ ÉTRANGÈRE) : La table produit doit avoir une clé étrangère référente à la table illustration.
- (CHECK) : les types de produits ne peuvent prendre uniquement les valeurs ('décoration intérieur', 'vêtement', 'coque', 'accessoire', 'papeterie', 'art mural', 'non matériel', 'fond écran', 'tuto' )

### Table Commande :

- (CLÉ PRIMAIRE) : chaque commande doit avoir une référence de commande unique qui ne peut pas être nulle
- (NON NULL) : la référence, la date et la clé étrangère du produit vers l'utilisateur ne peuvent pas être nulle
- (CLÉ ÉTRANGÈRE) : La table commande doit avoir une clé étrangère référente à la table user.



## Table Produits\_commande\_user :

- (CLÉ PRIMAIRE) & (CLÉ ÉTRANGÈRE) : Chaque produits de commande user doit avoir une paire de référence de commande et de référence de produit unique et non null qui forment chacune leur clé étrangère pour accéder aux tables produits et commande
- (NON NULL) : La quantité de produit ne peut pas être nulle

## Table Utilisateur :

- (CLÉ PRIMAIRE) : Les utilisateurs ne peuvent pas avoir la même adresse mail et le champ ne peut pas être null. Chaque adresse mail doit être unique.
- (NON NULL) : toutes les données de la table utilisateur ne peuvent pas être nulles à l'exception de l'adresse et du numéro de téléphone.
- (CHECK) : le champ mailFournisseur doit respecter la syntaxe des mails.
- (CHECK) : L'âge des utilisateurs doit être supérieur à 0 ans et inférieur à 120 ans (réaliser le calcul à partir de la date de naissance et de l'année actuelle)
- (CHECK) : L'utilisateur peut prendre 2 valeurs de roles : «client» ou « admin»

## Table livraison :

- (CLÉ PRIMAIRE) : Chaque livraison doit avoir une référence de livraison unique qui ne peut pas être null.
- (NON NULL) : tous les champs de la table livraison ne peuvent pas être nulles
- (CHECK) : Il ne peut pas y avoir plus de livraison que de commandes passés et une commande est associé à une et une seule livraison
- (CHECK) : les labels des transporteur ne peuvent prendre uniquement les valeurs ('courrier electronique', 'colissimo', 'chronospost','DPD', 'mondial relay', 'relais colis', 'DHL', 'FedEX', 'UPS')
- (CLÉ ÉTRANGÈRE) : La table livraison doit avoir une clé étrangère référente à la table commande.

## Table fournisseur :

- (CLÉ PRIMAIRE) : Les fournisseurs ne peuvent pas avoir la même adresse mail et le champ ne peut pas être null. Chaque adresse mail doit être unique.
- (NON NULL) : le label ne peut pas être nul.
- (CHECK) : le champ mailFournisseur doit respecter la syntaxe des mails.



## **Table Produit\_Fournisseur :**

- (CLÉ PRIMAIRE) : Chaque produit fournisseur doit avoir une référence de produit fournisseur unique qui ne peut pas être null.
- (NON NULL) : tous les champs de la table ne peuvent pas être nuls.
- (CLÉ ÉTRANGÈRE) : La table produit\_fournisseur doit avoir une clé étrangère référente à la table fournisseur.
- (CHECK) : les types de produits fournisseur ne peuvent prendre uniquement les valeurs ('décoration intérieur', 'vêtement', 'coque', 'accessoire', 'papeterie', 'art mural', 'non matériel', 'fond écran', 'tuto' )

## **Table Achat\_Fournisseur :**

- (CLÉ PRIMAIRE) : Chaque achat fournisseur doit avoir une référence d'achat fournisseur unique qui ne peut pas être null.
- (NON NULL) : la référence et la clé étrangère vers la table fournisseur ne peuvent pas être nulles.
- (CLÉ ÉTRANGÈRE) : La table achat fournisseur doit avoir une clé étrangère référente à la table fournisseur.

## **Table Produits commande fournisseur**

- (CLÉ PRIMAIRE) & (CLÉ ÉTRANGÈRE) : Chaque produits d'achat fournisseur doit avoir une paire de référence d'achat et de référence de produit unique et non nul.
- (NON NULL) : la quantité ne peut pas être nulle.



## 5. Liste des fonctionnalités prises en charge par la BD

Quels fonctionnalités prises en compte ?

### 5.1. Fonctions & Triggers

L'artiste a émis la volonté d'avoir une boutique « autonome ». De telle sorte que la boutique puisse se gérer au maximum toute seule pour que l'artiste puisse de son côté se consacrer à son activité principale : le dessin. Pour cela quelques fonctions ont été intégrés pour lui faciliter au maximum la tâche.

Une fonction **verifStockProduit()** qui permet d'empêcher de réaliser une commande lorsque le stock de produit disponible est inférieur à la quantité demandée par l'utilisateur en générant une erreur. Cette fonction est automatisée par le trigger du même nom (*verifStockProduit*) qui lance la fonction à chaque insertion ou modification de la table `produits_commande_user`.

Une fonction **miseAJoursStockProduitCommandeUser()** qui diminue automatiquement le stock de produit disponible après chaque commande passée par un utilisateur (décrémentement du stock). Cette fonction est automatisée par le trigger du même nom (*miseAJoursStockProduitCommandeUser()*) qui lance la fonction à chaque insertion ou modification de la table `produits_commande_user`.

Une fonction **miseAJoursStockProduitAchatFournisseur()** qui permet d'augmenter le stock de produits lorsqu'un achat fournisseur est réalisé. Cette fonction est automatisée par le trigger du même nom (*miseAJoursStockProduitAchatFournisseur*) qui lance la fonction à chaque insertion ou modification de la table `produits_commande_user`.

Une fonction **passerAchatFournisseur()** qui passe une commande auprès d'un fournisseur et le trigger du même nom (*passerAchatFournisseur*) qui permet de lancer cette fonction lorsque le stock du produit concerné est nul.

AMELIORATION possible : fonction/procedure pour passer l'appel en choisissant le fournisseur qui produit le type de produit en rupture de stock (1<sup>er</sup> sur la liste)





Une fonction ***afficherCommandeUser()*** qui permet d'afficher les informations importantes sur la commande, le client et la livraison. Cette fonction est automatisée par le trigger du même nom (*afficherCommandeUser*) qui lance la fonction à chaque insertion ou modification de la classe produits\_commande\_user.

## 5.2. Procédures

Des procédures sont implémentées dans le projet dans l'optique de faciliter l'ajout de nouvelles informations dans la base de données.

La procédure ***commandeUser*** (*mail\_user VARCHAR, ref\_produit INT, qte\_produit\_commande INT*) est implémenté afin de simplifier l'insertion des nouvelles commande utilisateur.

ex : `call commandeUser('houssam.imani@gmail.com', 2, 1);`

La procédure ***ajouterProduitCommandeUser***(*mail\_user VARCHAR, ref\_produit INT, qte\_produit\_commande INT*) est implémenté en complément de celle du dessus afin d'ajouter un produit à une commande déjà existante. Le numéro de la commande pris en compte est celui qui renvoie à la commande la plus récente de l'utilisateur en question.

ex : `call ajouterProduitCommandeUser('houssam.imani@gmail.com', 1, 3);`

La procédure ***commandefournisseur***(*f\_fournisseur\_email VARCHAR, p\_ref\_produit INT, a\_qte\_achat\_fournisseur INT*) est implémenté afin de simplifier l'insertion d'un nouvel achat fournisseur.

`call commandefournisseur('contact-printful@printful.com', 2, 5)`

**AMELIORATION possible :** Stopper la procédure si le fournisseur choisi ne propose pas ce type de produit-- Si le type de produit est défini comme 'non-matériel', 'tuto' ou 'fond écran' il est impossible de réaliser un achat fournisseur

De la même manière que pour les commandes utilisateurs. La procédure ***commandefournisseurAjoutProduit***(*f\_fournisseur\_email VARCHAR, p\_ref\_produit INT, a\_qte\_achat\_fournisseur INT*) est implémenté en complément de procédure *commandefournisseur* pour ajouter un produit différent au dernier achat fournisseur réalisé.





## 5.3. Vues

Des vues sont implémentées dans le projet dans l'optique d'afficher plus simplement les informations importantes et faciliter la lecture des données de la base .

On implémente une vue **commandesParProduits** qui affiche les commandes triées par produits différents commandés.

Cette vue permet de voir :

- ✓ Les informations de base sur les commandes : refCdeUser, dateCdeUser, qteCdeUser
- ✓ Quel client est à l'origine de la commande : mailUser
- ✓ Les informations de base sur les produits commandés : labelProduit, illustration\_labelIllust, prixProduit (prix unitaire)
- ✓ Le prix total prix pour la quantité commandée
- ✓ Le prix total par référence de commande pour tous les produits différents commandés
- ✓ Tester la vue avec la commande sql : `select * FROM commandesParProduits;`

On implémente une vue **commandesParRefCdeUser** qui affiche les commandes triées par référence de commande et utilisateur

Cette vue permet de voir :

- ✓ Le numéro et la date des dernières commandes réalisés (sans répétitions)
- ✓ Le client à l'origine de la commande (mailUser)
- ✓ Le nombre de produits différents commandés pour cette commande