

# Partie 1

## MCD – MLD



### S2.04 – Fabrication

**AKIN Berkan**  
**GERIFAUD Axel**  
**SERVONNET Mathieu**

BUT INFORMATIQUE



# Sommaire

1.	Introduction.....	2
2.	Répartition du travail .....	2
3.	MCD, MLD Relationnel, Création de la base .....	2
3.1	MCD .....	2
3.2	MLD .....	3
3.3	Création de la base .....	6
4.	Les Contraintes .....	6

## 1. Introduction

La société Foxconne Industries nous a contacté pour que l'on leur crée un système de base de données selon leurs besoins. Pour cela nous avons modéliser une base de données, créée un scripte pour pouvoir utiliser cette base, implémenter des vérifications dans le scripte, nous avons ensuite rempli cette base avec des données.

## 2. Répartition du travail

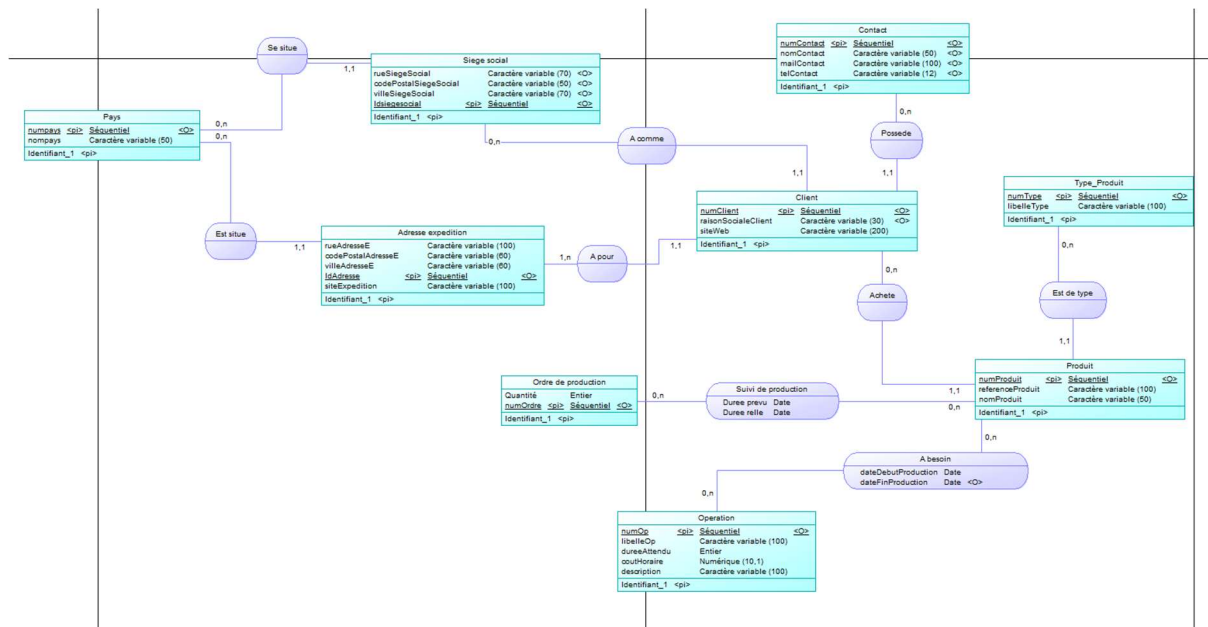
Taches	MCD-MLD	Création du script	Les contraintes
<b>Axel</b>		X	X
<b>Mathieu</b>	X		X
<b>Berkan</b>	X	X	

X = a contribué à la réalisation de la tâche

## 3. MCD, MLD Relationnel, Création de la base

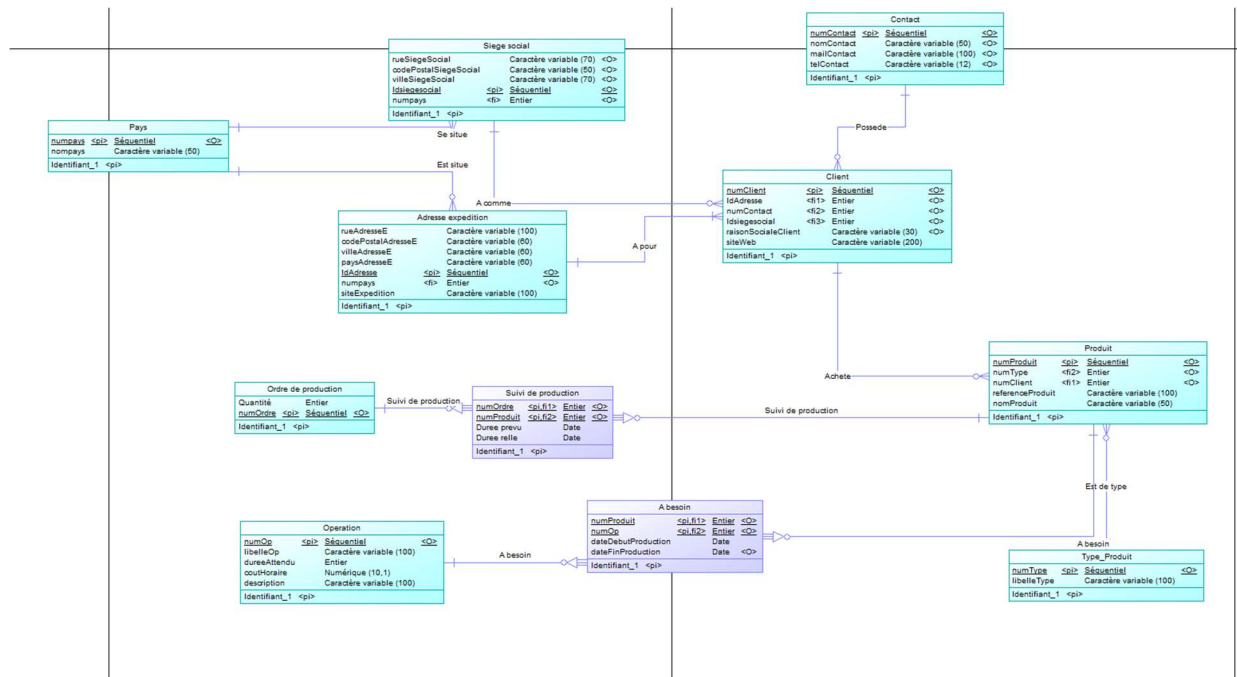
### 3.1 MCD

Pour répondre à ce besoin nous avons réalisé le MCD suivant :



### 3.2 MLD

Nous avons ensuite converti le MCD en MLD :



### 3.3 Explication des tables du MLD

Pays			
<u>numPays</u>	<pi>	Séquentiel	<O>
nomPays		Caractère variable (50)	
Identifiant_1	<pi>		

Pour la table pays nous avons donc un numéro de pays qui sera l'identifiant de cette table car il est unique puis il y a un libellé du pays qui va permettre de savoir de quelle pays il s'agit.

Siege social			
rueSiegeSocial		Caractère variable (70)	<O>
codePostalSiegeSocial		Caractère variable (50)	<O>
villeSiegeSocial		Caractère variable (70)	<O>
<u>Idsiegesocial</u>	<pi>	Séquentiel	<O>
numPays	<fi>	Entier	<O>
Identifiant_1	<pi>		

Pour ce qui est du siège social nous avons un ID qui correspond à la clé primaire puis tous les attributs d'une adresse tel que la rue, le code postal, la ville et il récupère en clé étrangère le numéro du pays pour pouvoir identifier le pays dans lequel il est implanté.

Adresse expedition			
rueAdresseE		Caractère variable (100)	
codePostalAdresseE		Caractère variable (60)	
villeAdresseE		Caractère variable (60)	
<u>IdAdresse</u>	<pi>	<u>Séquentiel</u>	<O>
numpays	<fi>	Entier	<O>
siteExpedition		Caractère variable (100)	
Identifiant_1	<pi>		

Pour l'adresse d'expédition nous avons aussi un ID comme clé primaire puis tous les attributs d'une adresse avec la rue, le code postal, la ville. Pour finir le site d'expédition et le numéro de pays en clé étrangère qu'il récupère de la table pays.

Contact			
<u>numContact</u>	<pi>	<u>Séquentiel</u>	<O>
nomContact		Caractère variable (50)	<O>
mailContact		Caractère variable (100)	<O>
telContact		Caractère variable (12)	<O>
Identifiant_1	<pi>		

Pour la table contact il y a un numéro de contact qui est la clé primaire puis toutes les informations essentielles avec le nom, le mail, et le numéro de téléphone.

Client			
<u>numClient</u>	<pi>	<u>Séquentiel</u>	<O>
IdAdresse	<fi1>	Entier	<O>
numContact	<fi2>	Entier	<O>
Idsiegesocial	<fi3>	Entier	<O>
raisonSocialeClient		Caractère variable (30)	<O>
siteWeb		Caractère variable (200)	
Identifiant_1	<pi>		

Pour la table client celle-ci va récupérer en clé étrangère l'id adresse, le numéro de contact et l'id siège social pour récupérer les informations sur l'adresse d'expédition le contact et le siège social. Nous aurons ensuite comme clé primaire un numéro de client puis leur raison sociale et pour finir leur site web.

Type_Produit			
<u>numType</u>	<pi>	<u>Séquentiel</u>	<O>
libelleType		Caractère variable (100)	
Identifiant_1	<pi>		

Pour la table type de produit, il y a comme clé primaire le numéro de type et un libelle du type qui va permettre de savoir de qu'elle type il s'agit.



Produit			
<u>numProduit</u>	<pi>	<u>Séquentiel</u>	<O>
numType	<fi2>	Entier	<O>
numClient	<fi1>	Entier	<O>
referenceProduit		Caractère variable (100)	
nomProduit		Caractère variable (50)	
Identifiant_1	<pi>		

Pour la table Produit elle récupère avec la clé étrangère le numéro de client et le numéro de type. Il y a ensuite un numéro de produit qui sera l'identifiant de la table. Ainsi qu'un nom du produit et une référence du produit.

Operation			
<u>numOp</u>	<pi>	<u>Séquentiel</u>	<O>
libelleOp		Caractère variable (100)	
dureeAttendu		Entier	
coutHoraire		Numérique (10,1)	
description		Caractère variable (100)	
Identifiant_1	<pi>		

Pour la table Opération il y aura un numéro d'opération comme clé primaire. Puis un libelle ainsi qu'une durée attendue, un cout horaire et une description pour finir.

A besoin			
<u>numProduit</u>	<pi,fi1>	<u>Entier</u>	<O>
<u>numOp</u>	<pi,fi2>	<u>Entier</u>	<O>
dateDebutProduction		Date	
dateFinProduction		Date	<O>
Identifiant_1	<pi>		

Pour la table A besoin celle-ci récupère le numéro de produit et le numéro d'opération grâce à des clés étrangères ces 2 attributs seront aussi les clés primaires de la table. Il y aura en plus de ses informations la date de début de production et la date de fin de production.

Ordre de production			
Quantité		Entier	
<u>numOrdre</u>	<pi>	<u>Séquentiel</u>	<O>
Identifiant_1	<pi>		

Pour la table Ordre de Production celle-ci sera identifiée par un numéro d'ordre et une quantité sera aussi renseigné.

Suivi de production			
<u>numOrdre</u>	<pi,fi1>	<u>Entier</u>	<O>
<u>numProduit</u>	<pi,fi2>	<u>Entier</u>	<O>
Duree prevu		Date	
Duree relle		Date	
Identifiant_1	<pi>		

Pour cette table suivi de production nous aurons comme clés primaires le numéro d'ordre ainsi que le numéro de produit qui sont des clés étrangères également. En plus de ces attributs il y aura aussi la durée prévu et la durée réelle comme attribut.

### 3.4 Création de la base

---

Pour la création de la base nous avons utilisé les outils de Power AMC pour générer une base PostgreSQL SQL à partir de notre MCD converti en MLD puis en MPD. Après le script nous avons supprimé tous les index ainsi que modifié les ordres drop table en rajoutant des « if exists » et des « cascade » pour que le script fonctionne à chaque fois qu'on l'exécute.

## 4. Les Contraintes

---

Pour ce qui est des contraintes à rajouter nous avons donc rajouté alter table contenant des check ou des unique directement dans les tables comme cela est présent dans les captures d'écrans ci-dessous :

```
alter table A_BESOIN
  add constraint CK_A_BESOIN_DATE_FIN check (DATEFINPRODUCTION > DATEDEBUTPRODUCTION);

alter table OPERATION
  add constraint CK_OPERATION_COUTHORAIRE check (COUTHORAIRE between 0 and 10);

alter table CONTACT
  add constraint CK_CONTACT_MAILCONTACT check (MAILCONTACT like '%@%.%');

alter table CLIENT
  add constraint CK_CLIENT_SITEWEB check (SITEWEB like 'https://%' or SITEWEB like 'http://');

alter table ORDRE_DE_PRODUCTION
  add constraint CK_ORDREDEPRODUCTION_QUANTITE check (QUANTITE between 1000 and 1000000);

TELCONTACT          VARCHAR(20)          not null UNIQUE,
REFERENCEPRODUIT    VARCHAR(200)         null UNIQUE,
```

## 5. Insertion des données

---

Après avoir réalisé les MCD-MLD ainsi que la base de données nous devons maintenant insérer les données de l'entreprise pour pouvoir répondre à certain besoin démontré via les requêtes. Il y aura donc dans un premier temps l'insertion de données puis dans un second temps les requêtes pour répondre aux besoins.

### 5.1 Les méthodes de génération de données

---

Avant d'insérer les données nous avons rajouté avec des requêtes SQL les champs Latitude et longitude dans la table adresse expédition ainsi que supprimé le fait que le code postal soit nul.

```
ALTER TABLE ADRESSE_EXPEDITION ALTER COLUMN CODEPOSTALADRESSEE DROP NOT NULL;
```

```
alter table ADRESSE_EXPEDITION
add latitude decimal(8,6), ADD longitude decimal(8,6)
```

Pour les insertions des données nous avons utilisés plusieurs méthodes :

- La création de fichier CSV sous Excel
- La génération de données avec des sites tel que generate data.
- Des requêtes SQL « Cross Join »

Nous avons donc généré les tables client, adresse d'expédition, siège social, opération, ordre de production, suivi de production, produit et type de produit.

Nous avons utilisé le cross join pour la table a besoin car celle-ci dépend du numéro de produit et du numéro d'opération voici une partie du cross join en question :

```
select p.numProduit,o.numOp from operation o
cross join produit p
where p.numproduit = 1 and (o.numOp between 1 and 20)
union
select p.numProduit,o.numOp from operation o
cross join produit p
where p.numproduit = 2 and (o.numOp between 21 and 40)
```

Celui-ci continue jusqu'au numéro de produit 15 et l'opération 300. (Voir fichier texte en pièce jointe).

Pour ce qui est de la table pays nous avons récupéré des fichiers csv contenant tous les pays directement fournis sur internet.

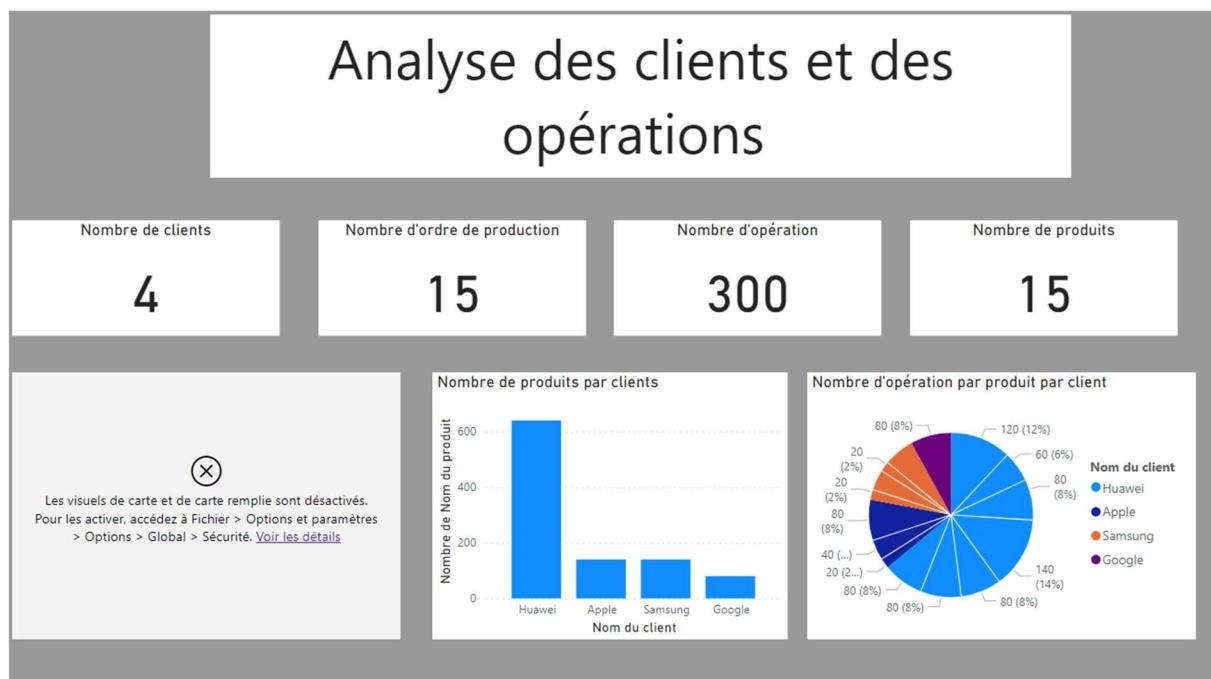
## 5.2 L'ordre des insertions

Pour l'ordre des insertions nous avons suivi l'ordre suivant :

- 1- Pays : il ne dépend d'aucune autre table
- 1- Opération : il ne dépend d'aucune autre table
- 1- Ordre de production : il ne dépend d'aucune autre table
- 1- Type de produit : il ne dépend d'aucune autre table
- 1- Contact : il ne dépend d'aucune autre table
- 2- Adresse d'expédition : Il dépend uniquement de la table pays insérer précédemment
- 2- Siège social : Il dépend uniquement de la table pays déjà insérer précédemment
- 3- Client : Il dépend des tables adresse d'expédition, siège social et contact insérer précédemment
- 3- Produit : Il dépend des table client et type produit insérer précédemment
- 4- Suivi de production : Il dépend des table ordre de production et produit insérer précédemment
- 4- A besoin : Il dépend des table produit et opération insérer précédemment



## 6. Rapport Power BI



## Analyse des types et des coût horaires

Nombre de Types

15

Coût horaire minimal

0,10

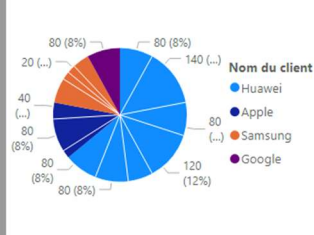
Coût horaire moyen

5,21

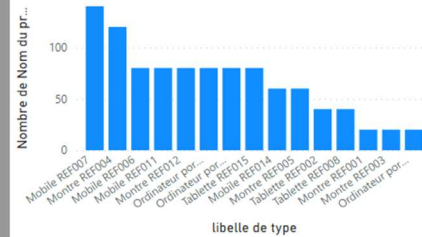
Coût horaire maximal

10,00

Nombre de Type de produit par clients



Nombre de produit par type



Moyenne de cout horaire par clients

