

# CONCEPTION

Savoir créer une base de données  
relationnelle pour des applications



Compétence demandée :  
Connaître les livrables (opérationnels  
et classiques)

1. Méthode opérationnelle
2. Méthode classique



# 1. METHODE OPERATIONNELLE

boutiques

<u>Id</u>	boutique	nom
-----------	----------	-----

vendeurs

<u>Id</u>	matricule	vendeur	age	boutique_id
-----------	-----------	---------	-----	-------------

vendeurs\_clients

vendeur_id	client_id
------------	-----------

clients

<u>id</u>	client	age
-----------	--------	-----

C'est la **structure** !

Table : boutiques

- id (PK)
- boutique
- nom

Table : vendeurs

- id (PK)
- matricule
- vendeur
- age
- boutique\_id (FK)

Table : vendeurs\_clients

- client\_id (FK)
- vendeur\_id (FK)

Table : clients

- id (PK)
- client
- age





1. Doit-on **stocker** de l'information ?

La demande de changement implique-t-elle de stocker des données ?

2. Quelles sont les **entités/tables** en jeu ?

3. Quelles sont les **relations** ?

Doit-on trouver des « entité2 » à partir des « entité1 » ?

Supprimer les doublons dans les relations

4. Quelles sont les **cardinalités** ?

Transformer les relations  $n-n$  en  $(1-n$  et  $n-1)$

5. Quelles sont les **clés (primaires, étrangères)** ?

6. Quels sont les **attributs** ?



# 1. Doit-on **stocker** de l'information ?

La demande de changement implique-t-elle de stocker des données ?

# 1. Doit-on stocker de l'information ?

La demande de changement implique-t-elle de stocker des données ?

Oui !!!

2. Quelles sont les  
entités/tables en jeu ?

## 2. Quelles sont les entités/tables en jeu ?

Table : boutiques

Table : vendeurs

Table : clients

Table : boutiques

Table : vendeurs

### 3. Quelles sont les relations ?

Doit-on trouver des « entité2 » à partir des « entité1 » ?

Supprimer les doublons dans les relations

Table : clients

2 relations :  
1. vendeurs - boutiques (travailler )  
2. vendeurs - clients (vendre )

Table : boutiques

Table : vendeurs

### 3. Quelles sont les relations ?

Doit-on trouver des « entité2 » à partir des « entité1 » ?

Supprimer les doublons dans les relations

Table : clients



2 relations :

1. vendeurs - boutiques (travailler )

2. vendeurs - clients (vendre )

Table : boutiques

Table : vendeurs

4. Quelles sont les  
cardinalités ?

Table : clients

2 relations :

1. vendeurs - boutiques (travailler,  $n-1$ )
2. vendeurs - clients (vendre,  $n-n$ )

Table : boutiques

Table : vendeurs

4. Quelles sont les  
cardinalités ?

Table : clients

2 relations :

1. vendeurs - boutiques (travailler, n-1)
2. vendeurs - clients (vendre, n-n)

Table : boutiques

Table : vendeurs

5. Quelles sont les clés  
(primaires, étrangères) ?

Table : clients

## 5. Quelles sont les clés (primaires, étrangères) ?

2 relations :

1. vendeurs - boutiques (travailler, n-1)
2. vendeurs - clients (vendre, n-n)

Table : boutiques  
- id (PK)

Table : vendeurs  
- id (PK)

Table : clients  
- id (PK)

## 5. Quelles sont les clés (primaires, étrangères) ?

2 relations :

1. vendeurs - boutiques (travailler, n-1)
2. vendeurs - clients (vendre, n-n)

Table : boutiques  
- id (PK)

Table : vendeurs  
- id (PK)  
- boutique\_id (FK)

Table : clients  
- id (PK)

## 5. Quelles sont les clés (primaires, étrangères) ?

2 relations :

1. vendeurs - boutiques (travailler, n-1)
2. vendeurs - clients (vendre, n-n)

Table : boutiques  
- id (PK)

Table : vendeurs  
- id (PK)  
- boutique\_id (FK)

Table : clients  
- id (PK)

Table : ventes  
- vendeur\_id (FK)  
- client\_id (FK)

## 5. Quelles sont les clés (primaires, étrangères) ?

2 relations :

1. vendeurs - boutiques (travailler, n-1)
2. vendeurs - clients (vendre, n-n)

Table : boutiques  
- id (PK)

Table : vendeurs  
- id (PK)  
- boutique\_id (FK)

Table : clients  
- id (PK)

Table : ventes  
- vendeur\_id (FK)  
- client\_id (FK)

## 6. Quels sont les attributs ?

2 relations :

1. vendeurs - boutiques (travailler, n-1)
2. vendeurs - clients (vendre, n-n)

Table : boutiques  
- id (PK)

Table : vendeurs  
- id (PK)  
- boutique\_id (FK)

Table : clients  
- id (PK)

Table : ventes  
- vendeur\_id (PK)  
- client\_id (PK)



## 6. Quels sont les attributs ?

2 relations :

1. vendeurs - boutiques (travailler, n-1)
2. vendeurs - clients (vendre, n-n)

Table : boutiques

- id (PK)
- lieu
- nom

Table : vendeurs

- id (PK)
- boutique\_id (FK)
- nom
- age

Table : clients

- id (PK)
- nom
- age

Table : ventes

- vendeur\_id (PK)
- client\_id (PK)

# Gagné !!!

2 relations :

1. vendeurs - boutiques (travailler, n-1)
2. vendeurs - clients (vendre, n-n)

Table : boutiques

- id (PK)
- lieu
- nom

Table : vendeurs

- id (PK)
- boutique\_id (FK)
- nom
- age

Table : clients

- id (PK)
- nom
- age

Table : ventes

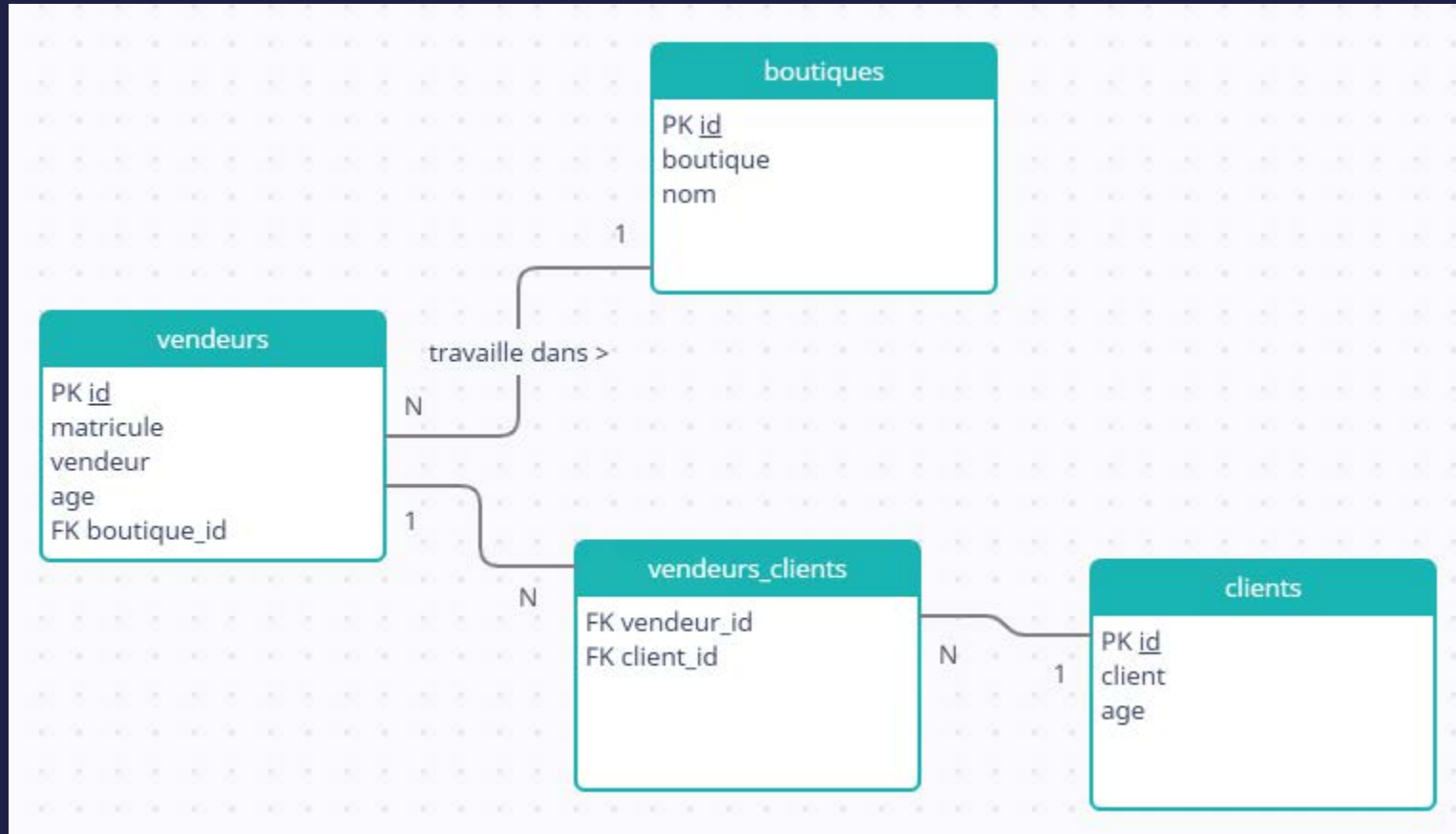
- vendeur\_id (PK)
- client\_id (PK)



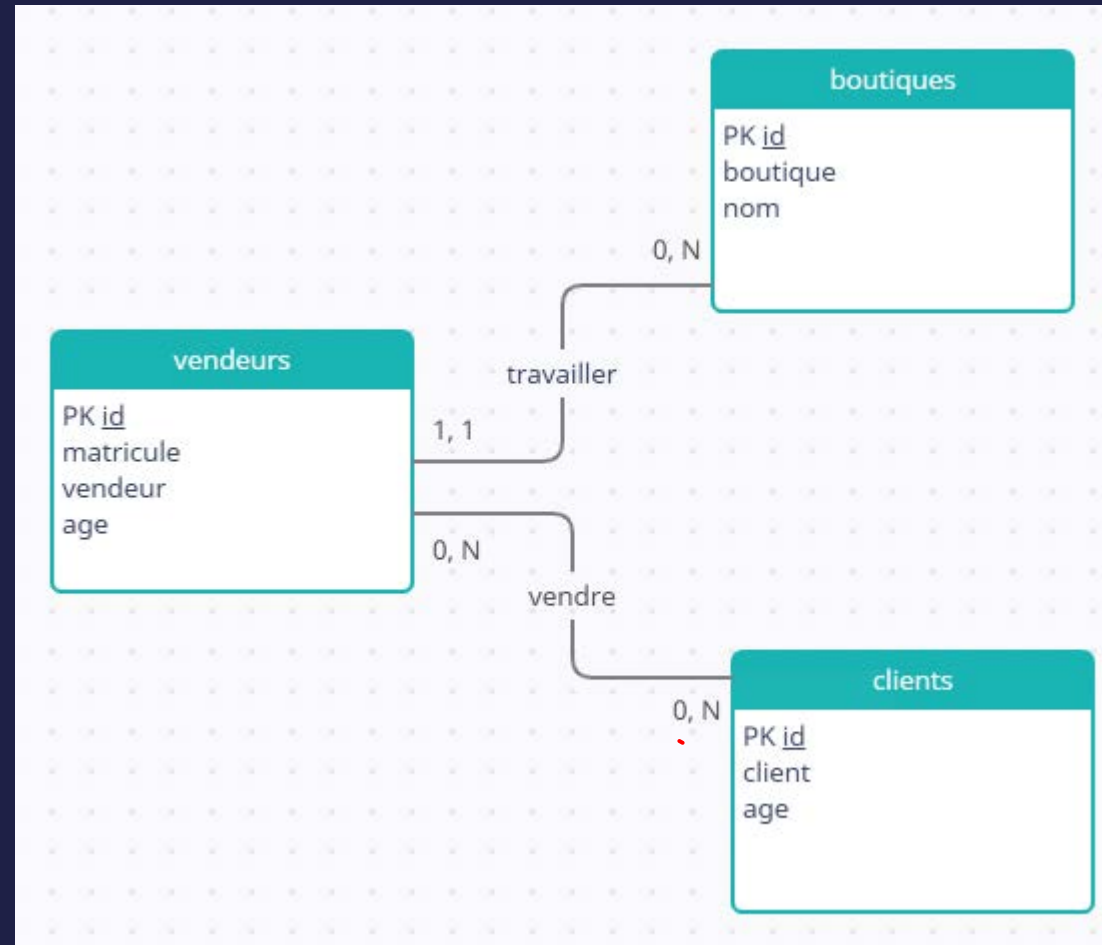
# 3. METHODE CLASSIQUE

# UML ou MERISE

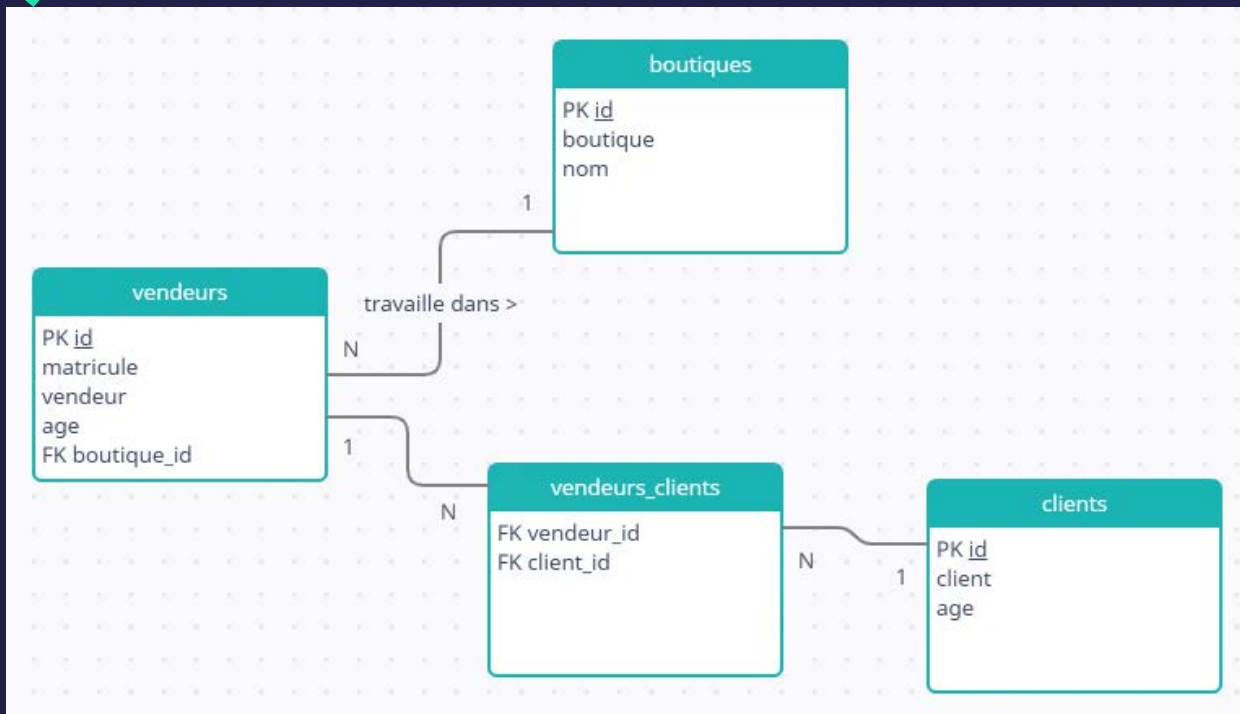
## UML



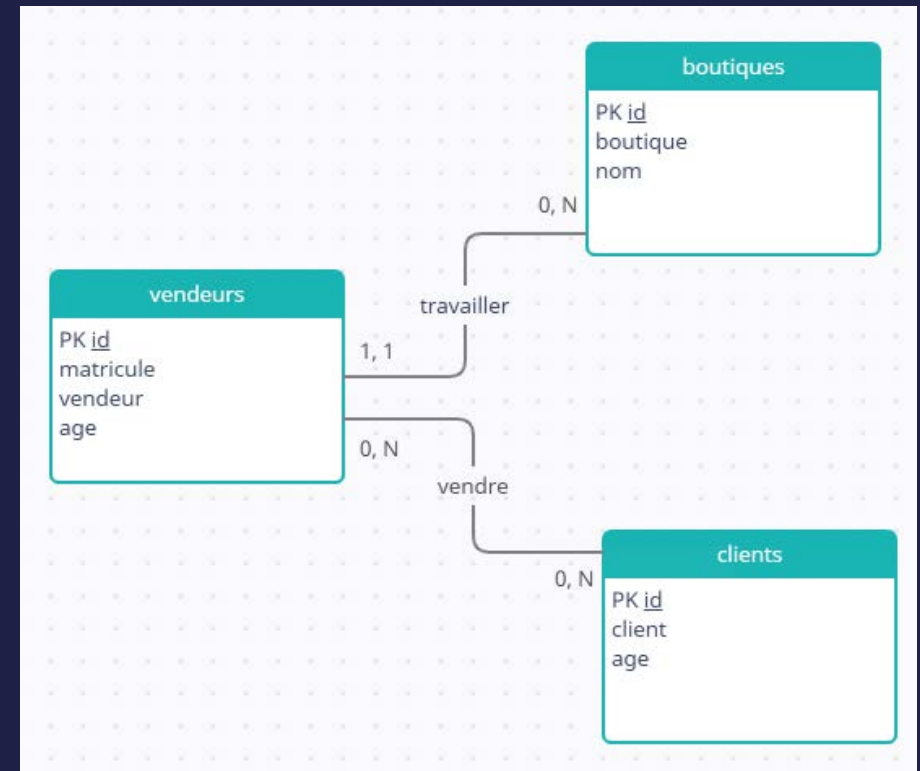
# MERISE



# UML



# MERISE



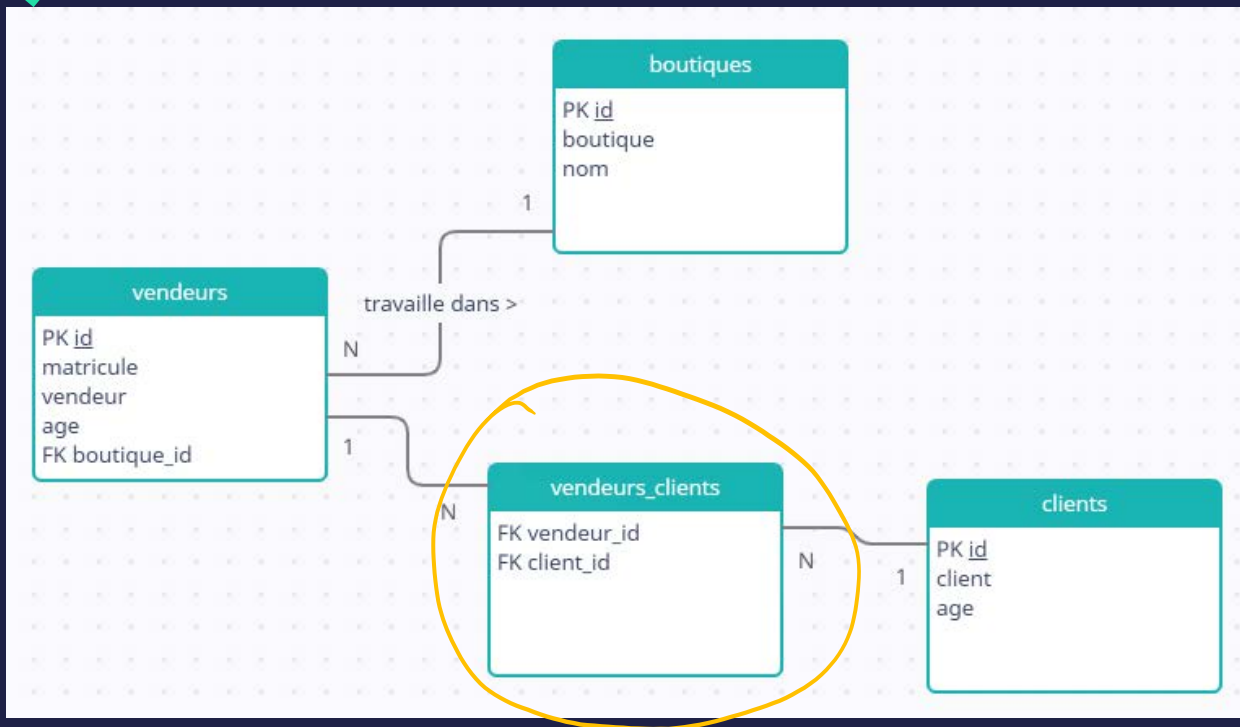


# MERISE

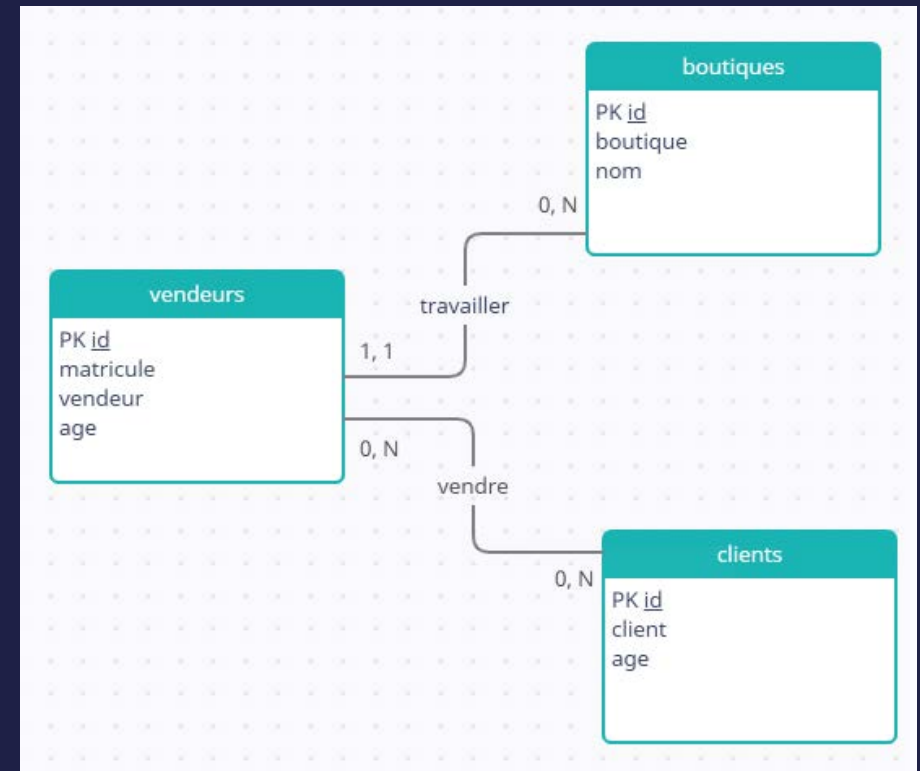
Etape	Type		Contenu	Niveau
1	MCD	Modèle Conceptuel de données	Entités + Relations	Haut
2	MLD	Modèle Logique de données	Entités + Relations + Directions relationnelles	Moyen
3	MPD	Modèle Physique des données	Table + FK + Taille des attributs	Bas

STEP1 : MCD (Modèle Conceptuel des Données) est une représentation graphique de **haut niveau** qui permet facilement et simplement de **comprendre comment les différentes entités sont liés entre eux**

## UML



## MERISE



Parenthèse, l'évolution de l'enseignement classique :

Aujourd'hui, on peut faire figurer la surrogate key dans le MCD. Auparavant et encore aujourd'hui, les dépendances fonctionnelles étaient utilisées pour déterminer la clé.



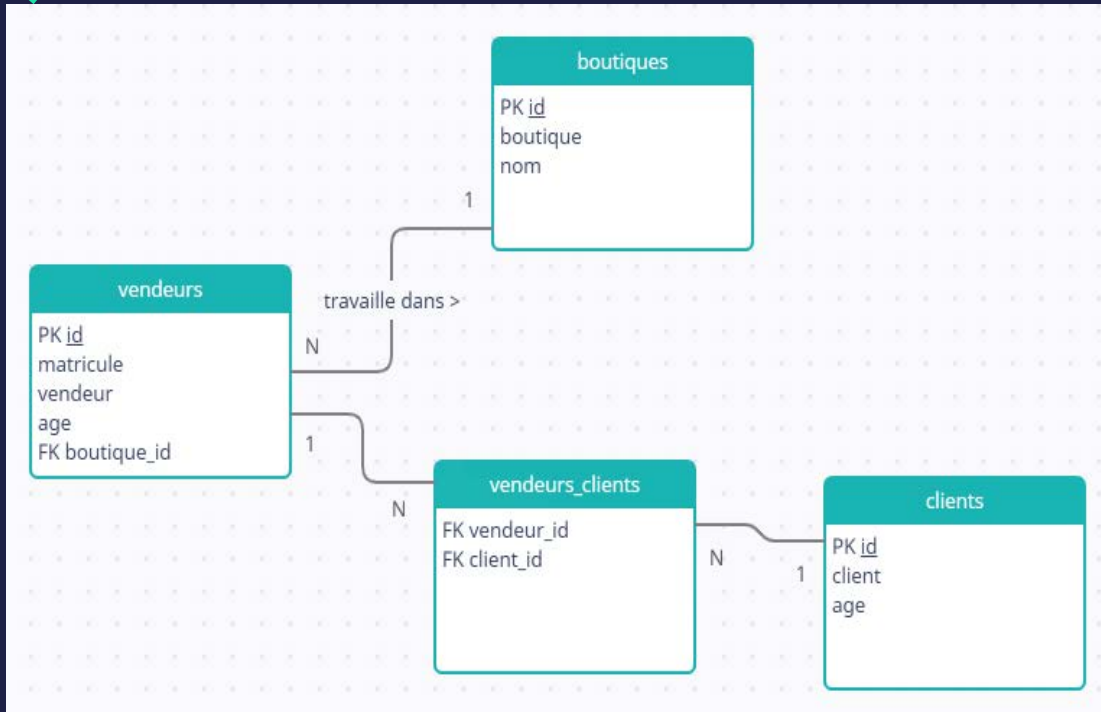
STEP2 : *MLD* (Modèle Logique des Données) est une représentation graphique de *moyen niveau* qui permet facilement et simplement dans *quelle direction est faite* chacune des relations entre les entités

La pointe de la flèche se met où la FK est présente, c'est-à-dire **qu'elle part de la cardinalité multiple** en MERISE.

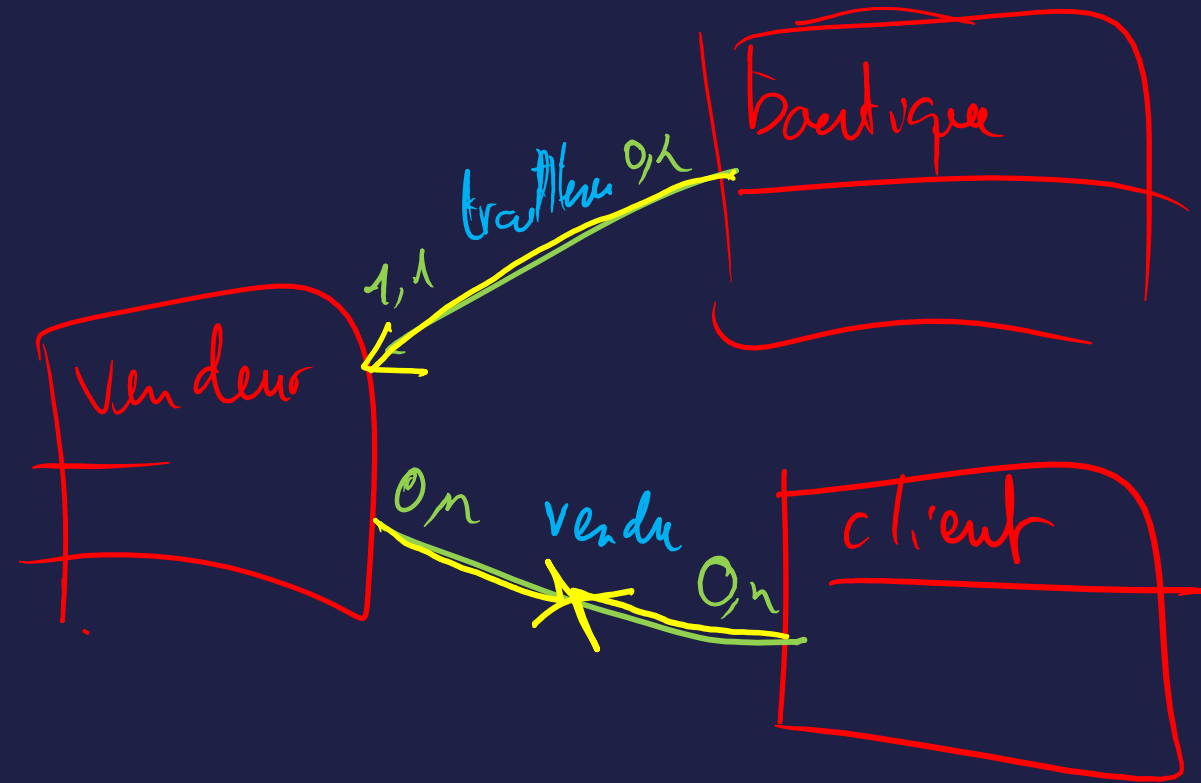
```
Multiple -> Simple
```

```
Multiple -> relation <- Multiple
```

# UML



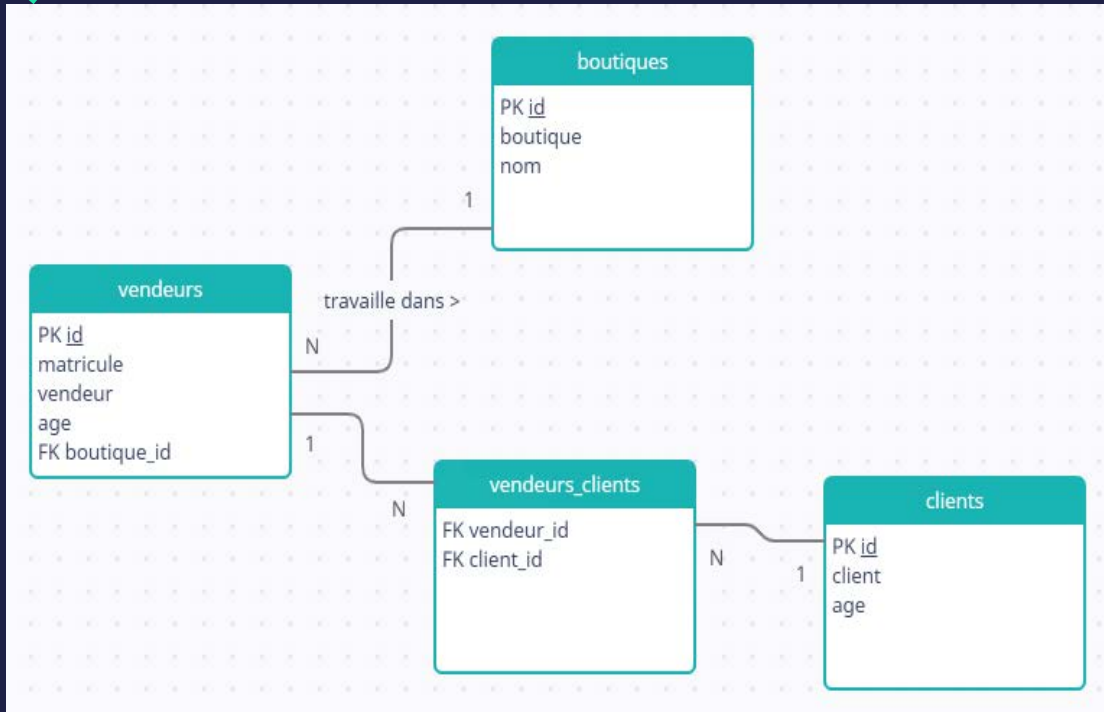
# MERISE



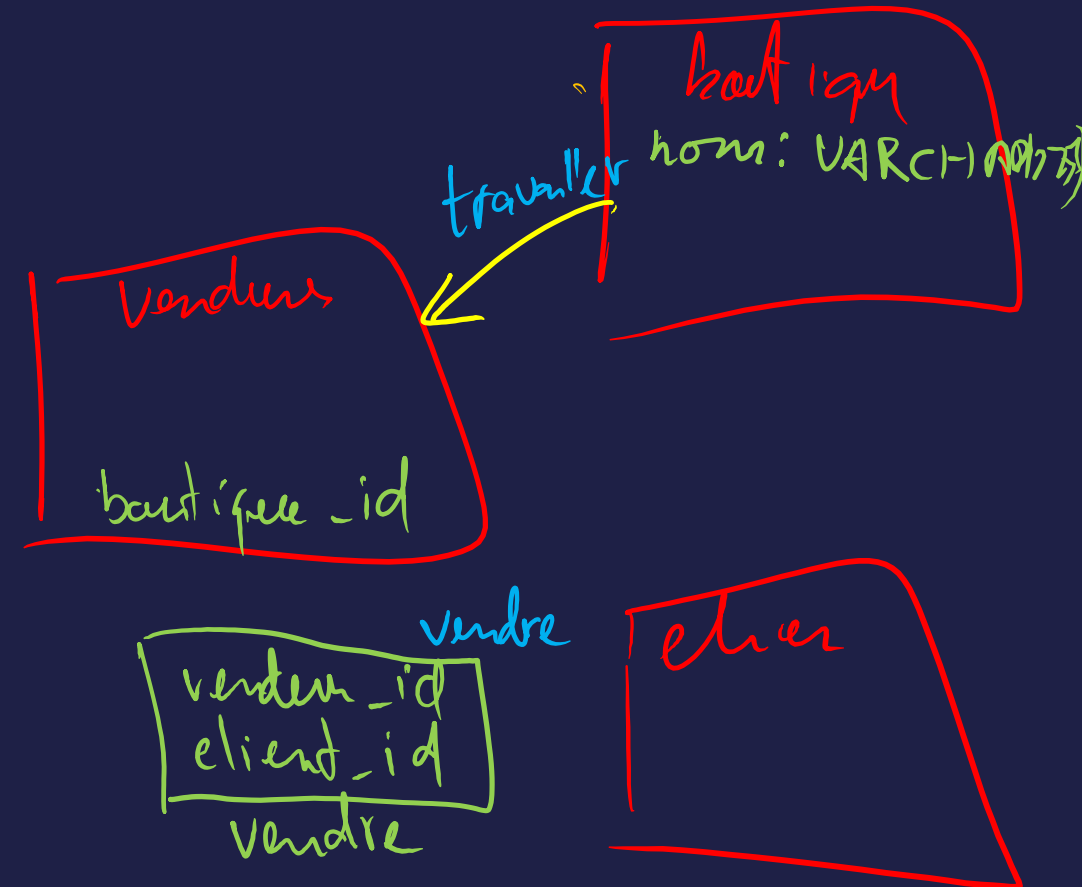


**STEP3 : MPD** (Modèle Physique des Données) est une représentation graphique de **bas niveau** qui consigne toute l'implémentation (peut contenir les tailles des attributs)

# UML



# MERISE





Le processus de réflexion :  
« demande » -> « structure »  
C'est le design, la modélisation ou  
l'architecture