**BASE DE DONNEES (DATA BASE)**

Table des matières

[Chapitre 1 : GENERALITES & DEFINITIONS : 2](#_Toc93176336)

[Chapitre 2 : CONCEPTION DES BASES DE DONNEES : 2](#_Toc93176337)

[2. 1 Le dictionnaire 2](#_Toc93176338)

[2. 2 Les éléments constitutifs : 3](#_Toc93176339)

[Chapitre 3 : MERISE : 3](#_Toc93176340)

[Chapitre 4 EXEMPLES : 4](#_Toc93176341)

[EXEMPLE 1 4](#_Toc93176342)

[EXEMPLE 2 : 5](#_Toc93176343)

[EXEMPLE 3 : 6](#_Toc93176344)

[EXEMPLE 4 : 8](#_Toc93176345)

[Chapitre 5 : LES TYPES DE D’ASSOCIATION : 9](#_Toc93176346)

[4.1 Binaire : 9](#_Toc93176347)

[4. 2 Ternaire : 9](#_Toc93176348)

[4. 3 n-aire : 9](#_Toc93176349)

[4. 4 Réflexive : 9](#_Toc93176350)

[Chapitre 6 : DEPENDANCES FONCTIONELLES 9](#_Toc93176351)

[Chapitre 7 : LES REGLES DE PASSAGE DU MODEL E/A VERS LE MODELE RELATIONNEL : 10](#_Toc93176352)

[Chapitre 8 : NORMALISATION DES DONNEES (FORMES NORMALES) 11](#_Toc93176353)

[Chapitre 3 : LES FORMES NORMALES : 11](#_Toc93176354)

[1ere Forme Normale 1FN : 12](#_Toc93176355)

[2eme Forme Normale 2FN : 12](#_Toc93176356)

[3eme Forme Normale 3FN : 12](#_Toc93176357)

# 

# : GENERALITES & DEFINITIONS :

**Définition de base de données :**

Une base = Lieu, espace

Donnée = information

* **Un Système d’Information** **(SI)** : Se sont les moyens humains & matériels et de dispositifs permettant de collecter, stocker, traiter et diffuser les informations nécessaires au sein d’une entreprise.

En fait dans une vie d’une entreprise, elle est continuellement en phase de produire des données.

Si par exemple elle en activités (vente et achat = données) 🡺 nous aurons besoin de stocker, et traiter ces données.

Si elle est en phase de recherche = production de données🡺 on aura besoin de stocker, et traiter aussi.

D’où la nécessité d’organiser ces données pour optimiser leurs manipulations.

* **Persistance des données** = l’information doit être gardée dans le temps, (aucune dépendance des facteurs extérieurs comme l’électricité)

Une base de données se repose sur 4 éléments :

* *Espace mémoire* : La zone utilisée pour écrire et enregistrer une information. Bien évidemment il faut qu’il soit de taille suffisante pour absorber la taille de la donnée qu’on veut stocker.
* *Support persistant*: Disque dur, ROM, CD ROM, etc.
* *Centralisation*: Mettre des données dans un endroit précis avec un point d’entrée, pour exemple un point d’entrée d’un PC c’est son adresse IP.
* *Accessible et sécurisée*: la donnée ne doit pas être accessible à tout le monde mais qu’aux personnes autorisées !

Une base de données donc est une application (SGBD) + R = Système de Gestion de Base données qui ont une Relation entre elles.

Système = Organisation= environnement = ensemble d’outils

Gestion = Organisationnelle, sécurisée, manipuler.

* Donc une base de données est un espace mémoire qui nous permet de stocker l’information dans une façon persistante. Elle doit être aussi accessible (pour traiter les données qu’elle contient) et sécurisée (pour qu’elle ne soit cible d’intrusions). L’Organisation de ces données se fait par la ou les différentes relations qui les lient. Ses relations permettent de rassembler des données se forme d’entités c’est ce qu’on appelle La méthode de conception relationnelle.

# : CONCEPTION DES BASES DE DONNEES :

2. 1 Le dictionnaire : Il donne le besoin à modéliser. C’est des informations brutes extraite du besoin fonctionnel. Une analyse des règles de gestion 🡺 SGF , On enlève tout ce qu’il ne sert pas dans les spécifications fonctionnelles.

## 2. 2 Les éléments constitutifs :

On utilise le dictionnaire pour définir les différents types qui constituent la base de données à savoir : les attributs, les entités, l’identifiant (clé), les associations, et la cardinalité.

**Entités** : ça correspond souvent aux acteurs de la relation (Association),

Exemple on dit : un individu est Un être humain, un individu est une entité

**Attribut** : Il qualifie un sous ensemble (une liste) dans l’entité, Exemple : l’individu est un être humain avec les yeux bleu= un attribut.

**L’identifiant (clé)** : Chaque sous ensemble peut être composé de plusieurs éléments (individus, unités), Exemple : on dit l’individu est un être humain avec des yeux bleu, son nom = sont identifiant est : Karim.

**L’association** : est la ou les relations qui lient les différentes entités dans un modèle. Exemple : on dit il existe deux entités, Les individus et le magasin carrefour, les individus est une entité, le magasin carrefour en est une autre et la relation qui les lient est la **vente (achat)= L’association., il y en avoir une ou plusieurs**

**La cardinalité :** C’est le nombre de fois qu’une relation peut exister entre les entités. On distingue ‘ cas de figures :

0, 1 : Une entité peut exister sans avoir de relation (0) avec l’autre entité, ou une raltion maximum

0, n : Une entité peut exister même si elle n’a pas de relation (0 relation) avec la ou les n entités

1, 1 : une entité qui existe a obligatoirement (1) une seule et unique relation avec l’autre entité.

1, n : une entité qui existe doit avoir au moins une seule relation avec une autre entité, comme elle peut avoir plusieurs relations avec les n entité.

# Chapitre 3 : MERISE :

M = Méthode, E = Etude, R= Réalisation, I= Informatique, S = Systèmes, E= Entreprise.

C’est la Méthode d’étude et de réalisation des systèmes informatiques pour les entreprises

Il est basé sur 3 processus :

**Processus MCD** =Niveau conceptuel : basé sur le besoin fonctionnel : Analyser les règles de gestion (SFG), Recherche des dépendances fonctionnelles

**Processus MLD** = Niveau logique : comment on veut implémenter (élaboration du dictionnaire de données, attributs, entités, associations, les identifiants.

**Processus MPD** : Niveau Logique : la création réelle de la Base de Données.

Dans le chapitre suivant on verra des exemples :

# Chapitre 4 EXEMPLES :

## EXEMPLE 1

1. Un magasin qui vend des fruits avec sa liste de clients,

|  |  |
| --- | --- |
| client | Numéro |
| Personne1 | a |
| Personne2 | b |
| Personne 3 | c |
| Personne 4  Table de clients | d |

1. Fruits : entité, les clients sont une entité, la relation c’est l’achat.

**Achat**

1, 1

0, 1

|  |  |
| --- | --- |
| Produit | Numéro |
| Pomme | 1 |
| Poire | 2 |
| Orange | 3 |
| Cerises | 4 |

Table de produits

0, 1

Signifie que les légumes présents peuvent être ne pas vendus mais doivent être vendus maximum à **un seul** client.

Les clients présents peuvent ne rien acheter (clients potentiels) comme ils peuvent acheter **n** produits.

0, n

Cette configuration on peut la représenter comme suit :

Impossible de l’avoir ce cas avec l’expression **(0, 1)** car on a spécifié qu’un produit ne peut être vendu que une fois (Or la pomme a été déjà achetée par la personne 3)

|  |  |
| --- | --- |
| Pomme | Personne 3 |
| Poire | Pas vendue |
| Orange | Personne 1 |
| cerises | Personne2 |
| 0 achat | Personne 4 |
| **Pomme** | **Personne 4** |

Table de résultat

Si veut

Si on change la cardinalité de **1, 0 vers 1, 1**

**Achat**

1, 1

0, 1

|  |  |
| --- | --- |
| Produit | Numéro |
| Pomme | 1 |
| Poire | 2 |
| Orange | 3 |
| Cerises | 4 |

|  |  |
| --- | --- |
| client | Numéro |
| Personne1 | a |
| Personne2 | b |
| Personne 3 | c |
| Personne 4  Table de clients | d |

|  |  |
| --- | --- |
| Pomme | Personne 3 |
| Orange | Personne 1 |
| cerises | Personne2 |

Le résultat de ce modèle est le suivant

## EXEMPLE 2 :

Des commerciaux qui vendent des produits informatiques (PC, souris, claviers, etc.) pour une entreprise

Cette table est une conception de données pour des ventes (sales) réalisées par des commerciaux avec des prix conseillés.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| sales | Pid(Produit) | CId(client) | dates | Qty | Price ($) |
|  | P1 | C1 | D1 | 100 | 350 |
| P1 | C1 | D2 | 600 | 350 |
| P1 | C2 | D3 | 50 | 350 |
| P2 | C1 | D4 | 40 | 45 |

Table de Sales (Ventes) = c’est une table référentielle qui change rarement c’est les mêmes produits qui se vendent toujours avec les mêmes prix sauf si le prix d’un ou d’un ensemble de produits changent au fil du temps dans ce cas-là nous seront obligés de créer une autre table afin historiser les évolutions des prix à travers le temps. Car si on change le prix directement sur cette table l’ancien prix sera effacé et donc on perd la notion de rentabilité de chaque commercial à travers ces périodes de changements des tarifs.

A titre d’exemple si un commercial vendait des P1 à 400$ avant (il fait 50 $ de plus par rapport au prix conseillé) donc ses clients avaient l’habitude que le produit P1 se vendait à 400 $, jusqu’à une date D’1 où le prix change (Price P1= 420$) donc à ce moment là on crée une ambiguïté sur son chiffre d’affaire réalisé car si on change le prix et là on dit on crée une relation nouvelle ou une nouvelle association entre le produit et son prix, et dans ce cas là il sera judicieux de créer et implémenter cette relation comme suit : La relation c’est « avoir »

Avoir

Price value

PRODUIT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| sales | Pid(Produit) | CId(client) | dates | Qty | Price ($) |
|  | P1 | C1 | D1 | 100 | 350 |
| P1 | C1 | D2 | 600 | 350 |
| P1 | C2 | D3 | 50 | 350 |
| P2 | C1 | D4 | 40 | 45 |

Le résultat est le suivant :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Price adviced | Pid | PRICE | DDV | DFV | Values |
|  | P1 | PR1 | YY/MM/DD | YY/MM/DD | 350$ |
| P1 | PR2 | YY/MM/DD | YY/MM/DD | 420$ |
| P2 | PR3 | YY/MM/DD | YY/MM/DD | 45$ |

**Rappel algorithme** : La fonction qui permet d’insérer la ligne nouvelle date dans le tableau précédent est :

Fonction updateprice (ihm : IHM) : Boléen

Var Id price

Debut

ihm.price

idprice=insert price (ihm.price)

insert advisedPrice(idprice, ihm.Pid, ihm.date)

update Last advisedPrice (ihmdate)

Fin

Il faut noter DDV =< D1’

DFV>= D2’

Si jamais l’intervalle D1’ D2’ se trouve sur deux changements de prix alors il affiche deux lignes.

Une ligne à l’intervalle correspondant au 1er prix qui s’étend de 01/01/2022 à 15/02/2022 qui est de 350 $

Une autre ligne qui s’étend de 16/02/2022 au 30/03/2022 qui est de 420 $

DDV du nouveau prix est la date de DFV de l’ancien prix +1.

## EXEMPLE 3 :

Un moteur composé de plusieurs pièces et chaque pièce est composée de quelques pièces qui peuvent être les mêmes :

1, n

Est composé

Nbre

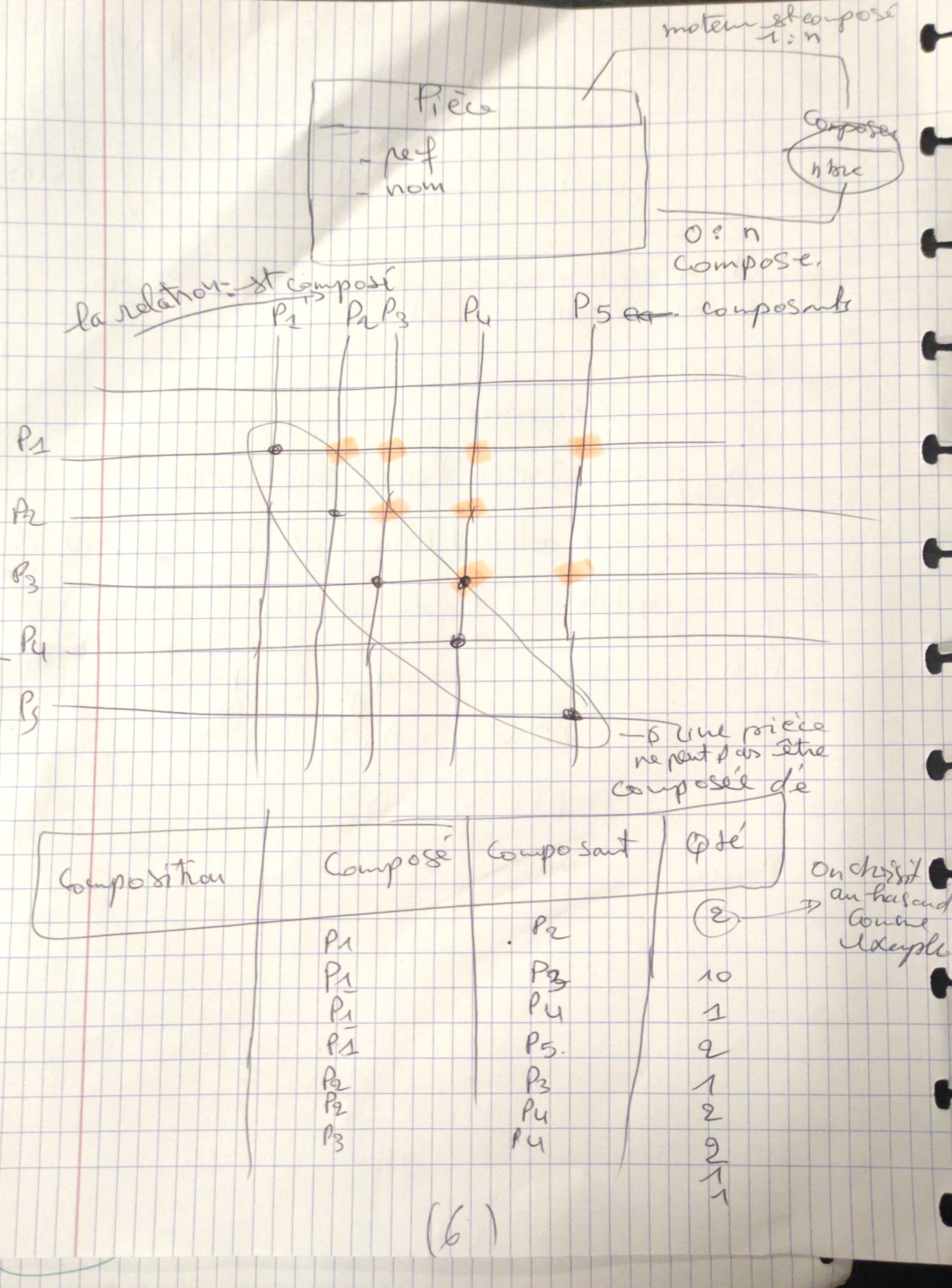
Composer

Piece (p1, P2, P3, P4

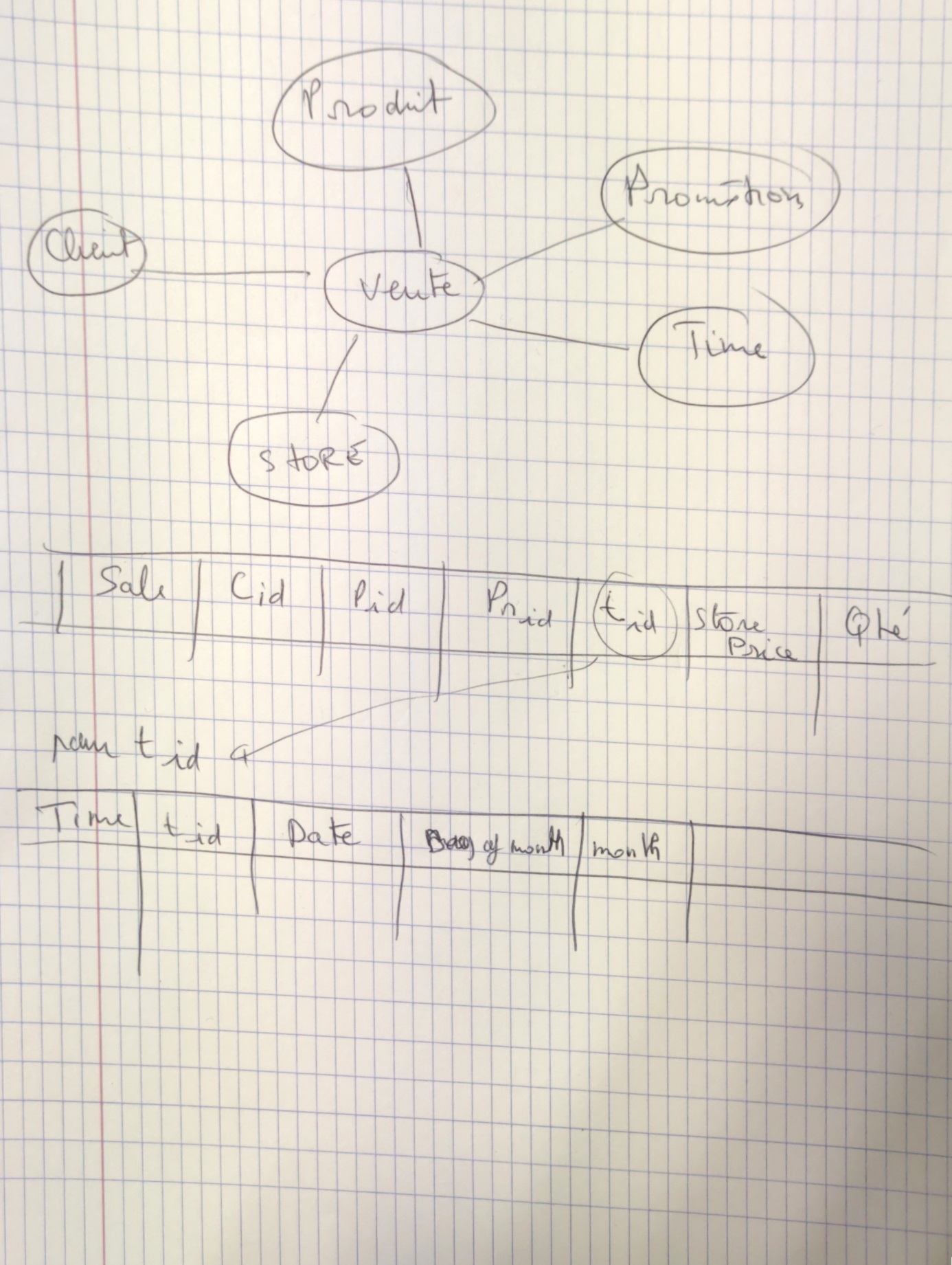
Ref

nom

Compose 



## EXEMPLE 4 :



# Chapitre 5 : LES TYPES DE D’ASSOCIATION :

## 4.1 Binaire :

C’est quand dans la relation il existe deux entités.

## 4. 2 Ternaire :

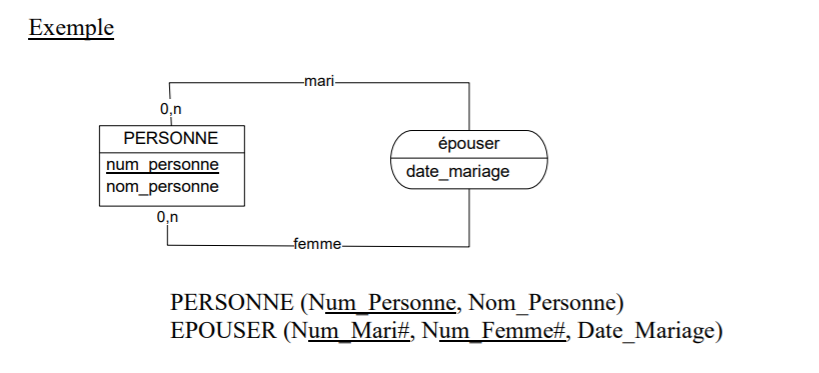
S’il existe au moins une relation entre 3 trois différentes entités. (Les trois entités sont reliées par une seule ou plusieurs relations).

## 4. 3 n-aire :

Relation entre plusieurs entités (plus de trois entités) par une relation.

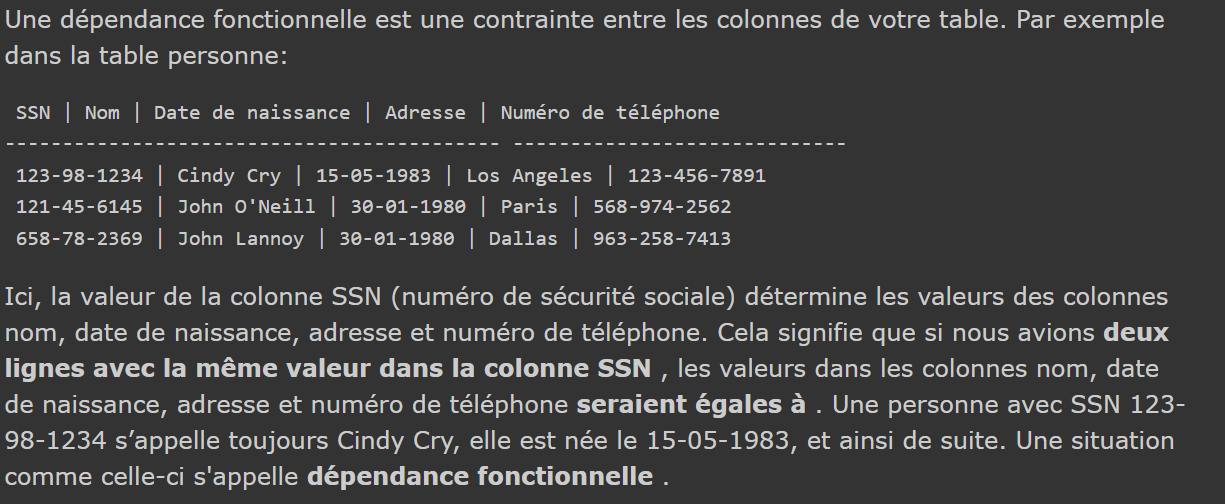
## 4. 4 Réflexive :

C’est une entité qui est en relation avec elle-même comme dans l’exemple des pièces.



# Chapitre 6 : DEPENDANCES FONCTIONELLES

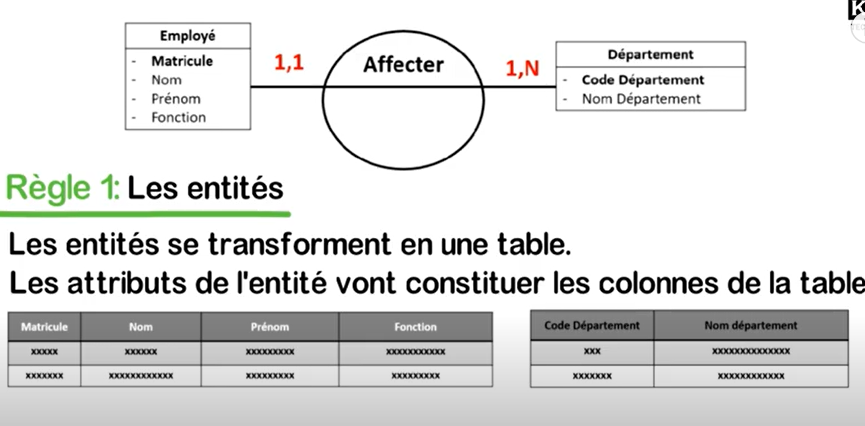
EXEMPLE :



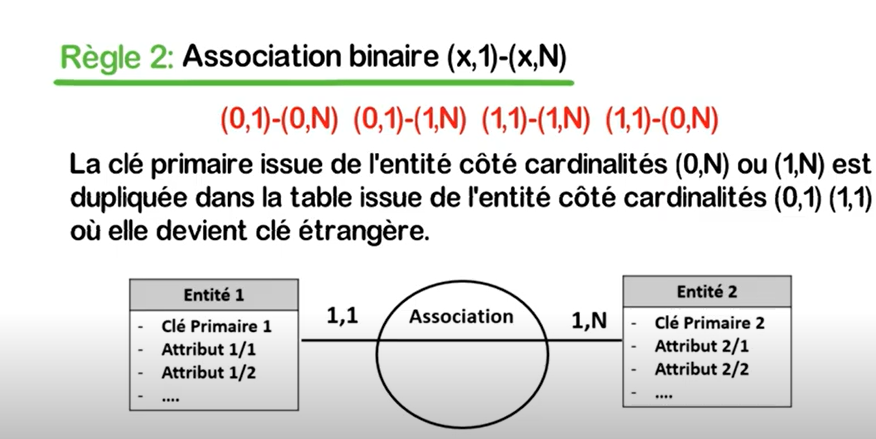
# Chapitre 7 : LES REGLES DE PASSAGE DU MODEL E/A VERS LE MODELE RELATIONNEL :

* **Règle 1** : Chaque entité devient une table de référence.

Les attributs de l’entité deviennent les colonnes de la table, l’identifiant de l’entité devient clé primaire (PK) de la relation. (Tous les attributs de l’entité participent à la relation)



* **Règle 2 :** Si dans l’une des entités reliées par une relation il y a une cardinalité de (0,1) ou (1,1) donc les identifiant de la relation



* **Règle 3 :** dans toute associationoù il n’existe pas de cardinalité (0,1) ou (1,1), la relation devient une table de fait et la clé de cette table c’est les clés des entités qu’elle relie.

# Chapitre 8 : NORMALISATION DES DONNEES (FORMES NORMALES)

L’objectif de la normalisation est de corriger les anomalies liées à la redondance des données qu’on ait eu pendant notre modèle relationnel.

On veut supprimer ces redondances car

* Déjà elles prennent de l’espace
* Affectent les opérations d’insertion de mise à jour et d’insertion de nouvelles données.

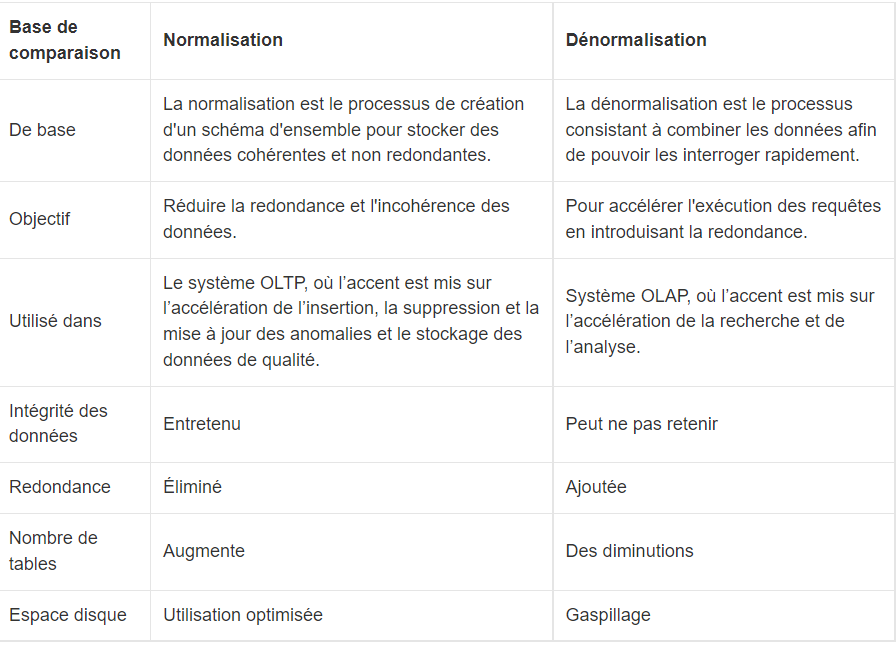
Donc on peut définir la normalisation comme étant un processus qui vise à optimiser et à améliorer le modèle relationnel, il se base sur la réorganisation des tables et des relations qui existent entre elles afin d’éviter la redondance des données.

# : LES FORMES NORMALES :

La normalisation est un processus qui vise à éliminer les redondances et organiser efficacement les informations dans une BDD-Relationnelle.

Il existe 3 formes (niveaux de normalisation), plus la forme normale d’une relation est élevée moins il y a de redondances.

Comparaison entre La normalisation et la Dénormalisation



## 1ere Forme Normale 1FN :

* La première forme normale ne concerne que les tables avec une seule clé primaire non composite (composite = composée de plusieurs colonnes).
* La deuxième condition pour qu’une relation soit en forme normale c’est qu’il faut que ses attributs contiennent tous des valeurs atomiques.

## 2eme Forme Normale 2FN :

* Si elle respecte les conditions de la 1ere FN
* Ça concerne uniquement les tables avec une clé primaire composite (dépendance entre clé primaire et un attribut de la table).

## 3eme Forme Normale 3FN :

* Si elle respecte les conditions de la 2ere FN
* S’il n’existe aucune dépendance fonctionnelles entre les attributs non clé.

Tables en étoiles et tables en flocon de neige

Tables en flocon de neige= tables en tables en etoiles entourées de sous tables de références en.