

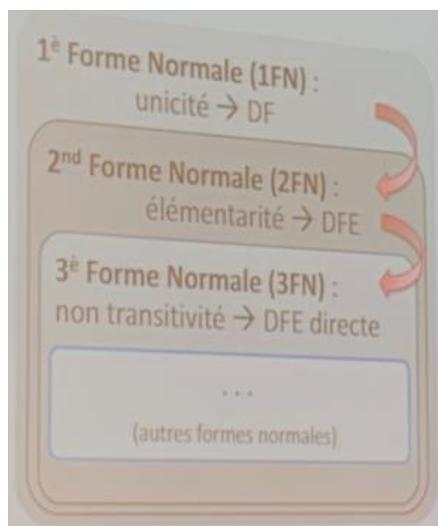
## Normalisation

- **Éliminer les redondances (une bonne Bdd)**
- **Décomposer les relations** sans perte des données
  - **Un seul fait dans un seul lieu** : une seule notion sémantique par relation

Chaque table parle d'une chose

- Comment ? Formes Normales
  - 1) « **Supprimer** » les **DFs redondantes**
  - 2) « **Supprimer** » les **DFs autres** que celles liées **aux clés**
  - 3) **Associer** l'ensemble des **attributs dépendants** d'un même attribut (ou d'un même ensemble = clés primaires) dans une **même table**

### I. Les formes Normales



- **Plus** une relation est **normalisée**, **moins** elle a de **redondances**
- Autres formes normales
  - Forme de Boyces-Codd (BCNF)
  - 4<sup>ème</sup> Forme Normale (4FN) DFs multivaluées
  - 5<sup>ème</sup> Forme Normale (5FN) DFs de jointure

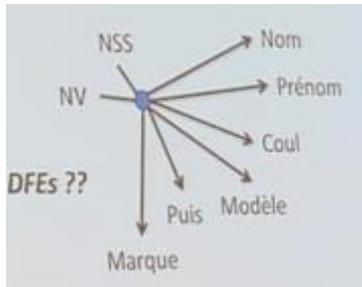
### Normalisation : 1FN

- Une relation est en **1FN** si :
  - Tout **attribut dépend** fonctionnel<sup>t</sup> de la clé (clé primaire)
  - Tous les **attributs** sont **atomiques (une seule info)**



Uniquement **DFE** entre la clé et les attributs non-clé

Propriétaire (NSS, NV, Nom, Prénom, Modèle, Marque, Puis, Coul) en 2FN ?



1FN ? Oui

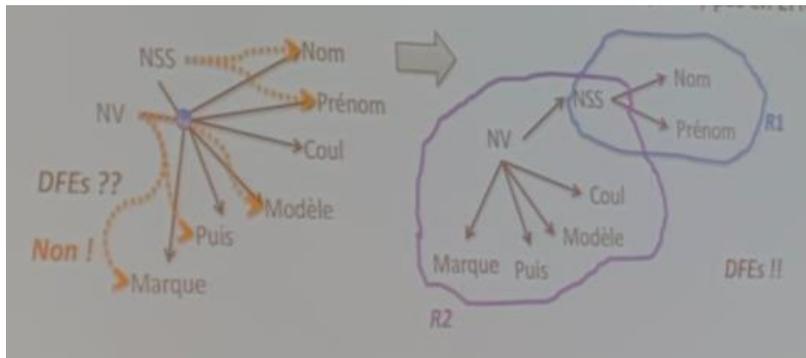
DF élémentaire ? Non (pas besoin de NSS)

Rendre en 2FN :

- 1) Trouver les DFE
- 2) Trouver les DF directes
- 3) Décomposer Propriétaire en R1 (NSS, Nom, Prénom)

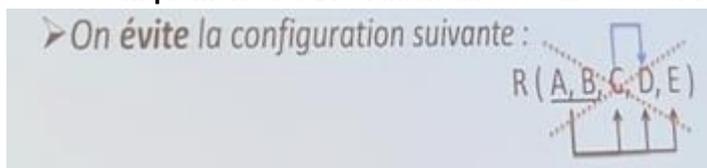
R2 (NV, NSS, Coul, Modèle, Marque, Puis)

(on regarde la clé primaire, on prend tous les attributs déterminants dans une relation)



### Normalisation : 3FN

- Une relation est en **3FN** si :
  - Elle est en 2FN (DFE)
  - Il n'existe aucune DF entre les attributs non-clé
  - **Uniquement des DF élémentaires et directes** entre les attributs non-clé et la clé



- **3FN** : uniquement **DF élémentaires et directes**

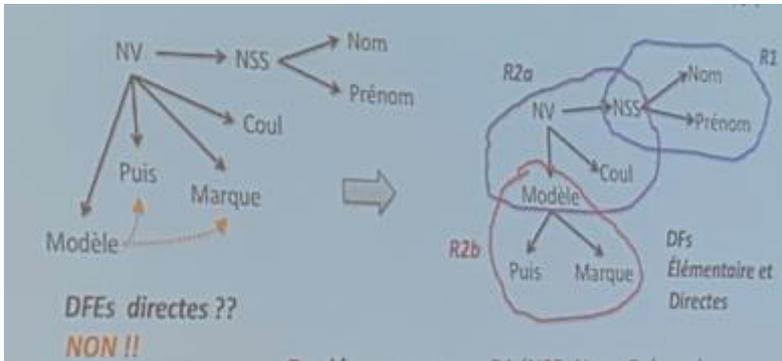
Les relations **R1 (NSS, Nom, Prénom)** et

**R2 (NV, NSS, Modèle, Marque, Puis, Coul)** sont en 3FN ?

2FN -> Oui

DFE et direct ? non (puis et marque en trop)

Ensuite on peut décomposer la relation



**On décompose en R1 (NSS, Nom, Prénom)**

R2a (NV, NSS, Modèle, Coul)

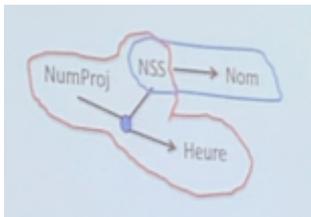
R2b (Modèle, Marque, Puis)

Les relations

R1( NSS, Nom) et R2 (NSS, NumProj, Heures) sont-elles en 3FN ?

NSS -> Nom

NSS, NumProj -> Heures



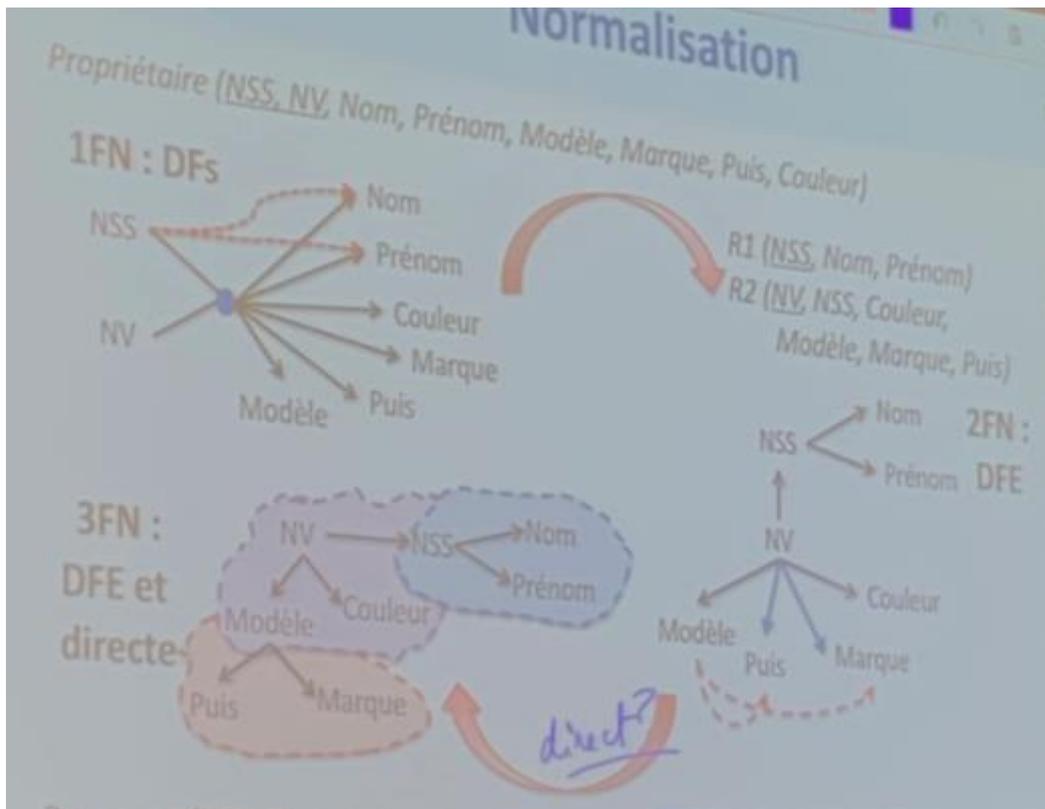
NSS	Nom
101	Durand
122	Dupond
120	Remy

NSS	NumProj	Heures
101	1	18
122	2	6.5
120	2	8.5

✓ Oui. Uniquement DF élémentaires et directes

**Récapitulatif :**

Propriétaire (NSS, NV, Nom, Prénom, Modèle, Marque, Puis, Couleur)



On a trouvé une couverture minimale.

- **Etapas recommandées :**

- Analyser la réalité, reconnaître les phénomènes pertinents

➔ **Identifier les DFs**

- Dessiner le **graphe des DFs**
- Rendre les **DFs élémentaires et directs** (couverture minimale)
- En déduire les **collections de relations** en 3FN (on regroupe les attributs dépendants d'un même attribut)

Exemple : base de données « Film »

- **Attributs**

- Nfilm
- Titre
- Année
- Ncass
- Type
- Etat
- AnAchat

- ➔ Un film a un titre et est produit à une année précise (comment on identifie un film ?)
- ➔ Une cassette correspond à un DVD, BlueRay ou VHS ( type acheté à une certaine année, et ayant un état précis (bon, mauvais, HS...) ) ( comment on identifie une cassette ?)
- ➔ **Une cassette correspond à un seul film**

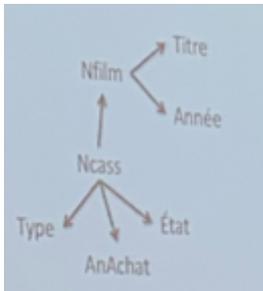
1ère étape : Les DFs ? 1FN

Ensemble des DFs :

NFilm -> Titre, Année

Ncass -> Type, Etat, AnAchat

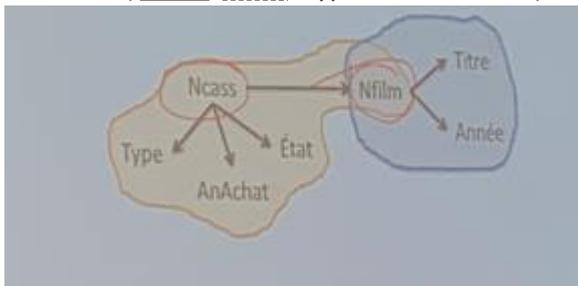
Ncass -> NFilm (combien ?)



Pour le graphe, mettre les attributs au centre.

Relation en 3FN

- Film (Nfilm, Titre, Année)
- Cassettes ( Ncass, Nfilm, Type, Etat, Anachat)



### Exemple 1 :

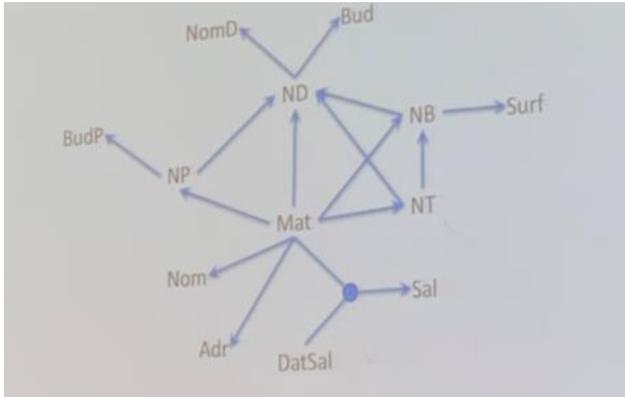
Départements => **identification des DFs**

1.

- Un employé a un nom et une adresse (ici on ajoute un numéro matricule – numéro unique)
  - o Matr -> Nom Matr -> ADR
- Un employé travaille pour un département sur un projet donné (comment identifier ? Numéro)
  - o Matr -> ND , Matr -> NP
- Un employé a un téléphone et il occupe un bureau
  - o Matr -> NT , Matr -> NB
- Un employé a un salaire à une certaine date (Historisation)
  - o Matr, DateSal -> Salaire
- Un département a un budget et un nom
  - o ND -> Bud, ND-> NomD
- Un projet est sous la responsabilité d'un département et il a un budget propre
  - o NP -> ND , NP-> BudP
- Un bureau appartient à un département, et il a une surface
  - o NB-> ND, NB-> Surf
- Un téléphone est associé à un département et à un bureau

- NT -> ND, NT -> NB

Graphe des DFs :



**DFs élémentaires et directes :**

- **DFs élémentaires**

- Contrôler les DFs en partant d'un ensemble
  - Matr, DatSal -> Sal vérifier si Matr -> Sal = Non (historique) ou DateSal -> Sal = Non
    - ⇒ DFE

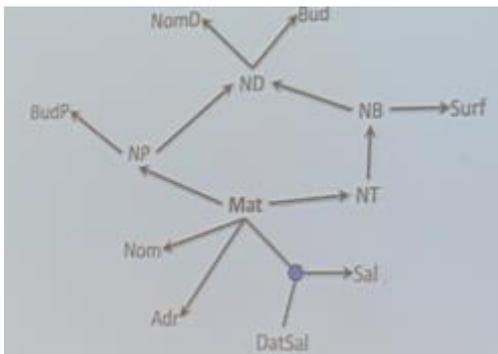
- **DFs Directes**

- Contrôler pour chaque X -> Y, s'il n'existe pas un Z tel que X -> Z et Z -> Y
  - Matr -> ND regarder si les DFs de ND et Matr : NP -> ND
    - Matr -> NP

Matr -> NP -> ND => NON

On coupe Matr -> NB et NT -> ND et Matr -> ND

Donc on a le graphe :



Chaque groupe des DFs avec **le même ensemble d'attributs à gauche** forme une relation

On ne peut pas couper NP -> ND et Matr -> NP car on perdrait de l'information

**Employé** (Matr, NT, NP, Nom, Adr)

**Département** (ND, NomD, Bud)

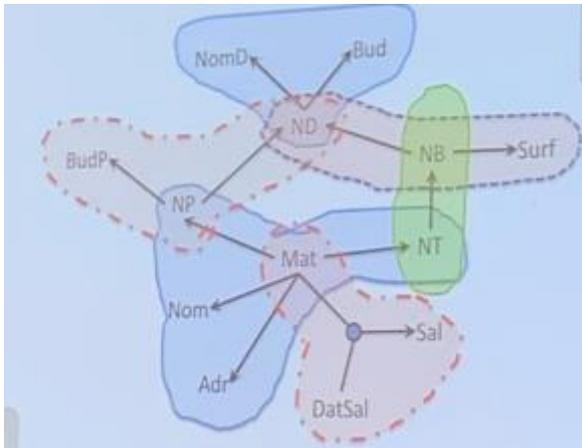
**Budget** (NB, ND, Surf)

**Projet** (NP, BudP, ND)

**Salaire** (**Matr**, DatSal, Sal)

**Téléphone** (NT, NB)

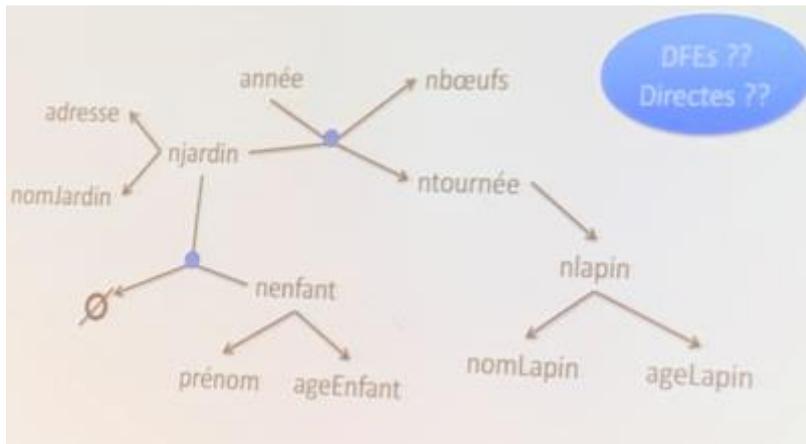
Gras = clé étrangère



### Exemple 2 :

- Le lapin de pâques
  - ➔ Le lapin de pâques veut informatiser son système de gestion. Il veut connaître les informations sur les jardins, les enfants et les tournées à effectuer
  - Un enfant dont on connaît l'âge et le prénom, est identifier par un numéro unique.
    - Nenfant -> Age Nenfant -> Prénom
  - Un jardin est identifié par un numéro. On connaît son adresse et aussi son nom
    - Njardin -> NomJardin, Adresse
  - Un enfant pourra aller chercher des œufs dans **plusieurs jardins**  
Inversement, un jardin peut être visité par **plusieurs enfants**
    - Njardin, nenfant -> ∅
  - Une tournée est toujours réalisée par **un lapin** livreur (le lapin de paques a bien besoin d'aide). On veut connaître l'âge et le nom de ces lapins.
    - Ntournée -> nlapin et nlapin -> nomLapin, âgeLapin
  - Une tournée va être faite pour **différents jardins** afin d'optimiser le coût de transport
    - Ntournée ≠ Njardin
  - On veut aussi pouvoir connaître la **quantité d'œufs** reçue chaque année par un jardin (historisation)
    - Njardin, année -> nboeufs
  - Une tournée évolue au cours des ans (travaux sur un jardin, chien de garde peu aimable...), en englobant plus ou moins de jardins.
  - On souhaite savoir quels jardins ont été visités chaque année dans une tournée.
    - Njardin, Année -> ntournée

Graphe de dépendances fonctionnelles :



→ Les DFs sont élémentaires et directes

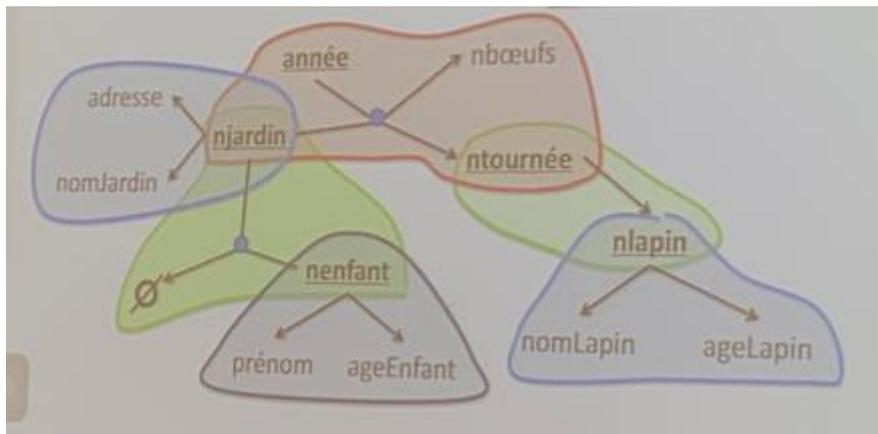
**Jardin** (njardin, nomjardin, adresse)

**Enfant** (nenfant, prénom, ageEnfant)

**Enf\_jardin** (nenfant, njardin)

**Tournée** (ntournée, nlapin)

**Visites** (njardin, année, nboeufs, ntournées)



Si on voudrait organiser les lapins en équipes ?

- Chaque lapin a un lapin chef
- Comment on identifie un lapin ?