

Normalisation

- I. Normalisation
- II. Dépendance fonctionnelle (DF)
- III. Les formes Normale

Normalisation

Niveau conceptuel (EA)

Schéma EA

VALIDATION

Schéma EA valide

*Niveau logique
(Relationnel)*

Schéma relationnel
normalisé

NORMALISATION

Normalisation

TRADUCTION
EA - R

Schéma Relationnel

Normalisation

La normalisation est un processus de transformation d'une relation posant des problèmes lors de la mise à jour en relation ne posant pas de problème.

On mesure la qualité d'une relation par son degré de normalisation N, la valeur de N est présenté en fonction des formes normales.

Normalisation

- ✓ Un schéma de relation est décrit par la liste de ses attributs et de leurs contraintes d'intégrité.
- ✓ La constitution de la liste d'attributs du schéma **ne peut pas se faire** n'importe comment **pour ne pas provoquer de redondance**, avec toutes ses implications (perte de place, risques d'incohérence et de perte d'informations).

Normalisation

✓ Objectif du Schéma Logique (relationnel) :

- Décrire une BD qui va **être effectivement exploitée** par le Système d'Information de l'Entreprise
- Définition (LDD), Interrogation (LID), Contrôle (LCD) et Mise-à-Jour « MAJ » (LMD)

✓ Exigences au niveau exploitation de la BD

- Les MAJ doivent conserver **la cohérence de la BD**
- ⌚ Plus la BD contient de **redondances**, plus les MAJ avec maintien de cohérence est difficile.

Normalisation

✓ Anomalie de modification

id	nom	adresse	cnum	desc	note
124	Jean	Paris	F234	Philo I	A
456	Emma	Lyon	F234	Philo I	B
789	Paul	Marseille	M321	Analyse I	C
124	Jean	Paris	M321	Analyse I	A
789	Paul	Marseille	CS24	BD I	B

- Une modification sur une ligne peut nécessiter des modifications sur d'autres lignes.
- **Exemple :** on souhaite modifier l'adresse de Paul : deux lignes sont impactées.

Normalisation

✓ Anomalie de suppression

id	nom	adresse	cnum	desc	note
124	Jean	Paris	F234	Philo I	A
456	Emma	Lyon	F234	Philo I	B
789	Paul	Marseille	M321	Analyse I	C
124	Jean	Paris	M321	Analyse I	A
789	Paul	Marseille	CS24	BD I	B

- Certaines informations dépendent de l'existence d'autres informations.
- Exemple : le cours 'CS24' dépend de l'inscription de Paul.

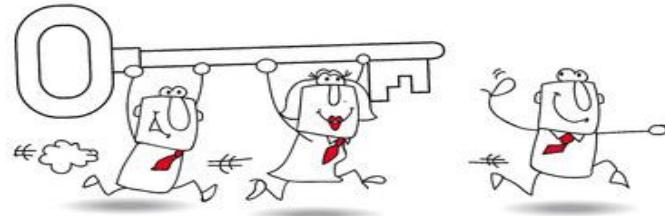
Normalisation

✓ Anomalie d'insertion

id	nom	adresse	cnum	desc	note
124	Jean	Paris	F234	Philo I	A
456	Emma	Lyon	F234	Philo I	B
789	Paul	Marseille	M321	Analyse I	C
124	Jean	Paris	M321	Analyse I	A
789	Paul	Marseille	CS24	BD I	B
145	Evariste	Aubenas	???	???	???

- La possibilité d'enregistrer un tuple implique la connaissance de toutes les informations qui lui sont liées : problème de valeurs manquantes.
- Exemple : soit '145, Evariste, Aubenas' un nouvel étudiant.
 - On ne peut l'insérer que si l'on connaît un de ses cours et sa note dans ce cours, à moins de permettre les valeurs nulles.

Normalisation



Le moyen qui permet d'éviter ces problèmes est **l'étude des dépendances** (fonctionnelles, multi-valuées, de jointure...)

Normalisation

✓ Pour éviter les anomalies

- Le principe est de décomposer les relations de telle sorte d'éviter les anomalies;
- c'est-à-dire de transformer une relation en plusieurs relations

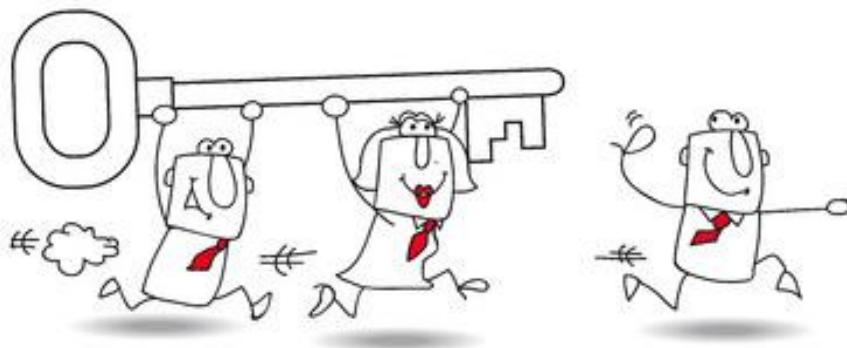
⌚ Difficulté: Le risque en décomposant est de perdre de l'information :

- ⌚ Pouvoir retrouver toutes les informations initiales
- ⌚ Avoir les mêmes dépendances satisfaites

Normalisation

- ✓ Construire des BD Relationnelles cohérentes et **sans redondance** d'informations
 - Éviter les éventuelles anomalies
- ✓ On peut dire qu'une BD Relationnelle est « **correcte** » ou **normalisée** si :
 - Chaque Relation décrit une **information élémentaire** à travers les Attributs qui lui sont **directement liés**
 - Il n'y a **pas des Redondances** d'informations qui peuvent produire des problèmes de Mises-à-Jour (MAJ)

Normalisation



LA **NORMALISATION**

- ✓ La **Normalisation** est basée sur la **notion de Dépendance Fonctionnelle (DF)**
- **DF** : Permet de décomposer des Relations **sans pertes** d'informations

Normalisation

Exemple : La Relation « PRODUIT » peut être décomposée en trois relations non redondantes

PRODUIT(RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

Décomposition 1

PRODUIT1(RefProduit, Libelle, PU)

P1	CH7	23.510
P3	VIS12	0.150

PRODUIT2 (RefProduit, NumService, Quantité, Adresse, Capacité)

Décomposition 2

PRODUIT22 (NumService, Adresse, Capacité)

PRODUIT 21 (#RefProduit, #NumService, Quantité)

P1	S1	300
P1	S2	500
P3	S4	900

S1	Sousse	9000
S2	Tunis	6000
S4	Sousse	2000

Normalisation

✓ Définition des DF

- Soit une relation R [..., A, B, ...], on dit qu'il **existe une DF entre les 2 attributs A et B de la relation R**, si à toute valeur de **A il ne lui est associé qu'une seule valeur de B**.
- On note une telle DF

A → B (A détermine B ou B dépend fonctionnellement de A).

✓ Exemple

PRODUIT (RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

Pour cette relation, les dépendances fonctionnelles suivantes sont vérifiées :

RefProduit → LibelleProduit

RefProduit → PU

NumService → Adresse, Capacité

RefProduit, NumService → Quantité

Normalisation

✓ Axiomes et règles d'inférence

- Découvrir ***de nouvelles dépendances*** à partir d'un ensemble initial.
- Les trois premières propriétés sont connues sous le nom « **Axiomes d'Armstrong** »

Propriété 1 : Réflexivité

$B \text{ inclus-ou-égal } A$ alors $A \rightarrow B$

Tout ensemble d'attributs détermine lui-même ou une partie de lui-même.

Propriété 2 : Augmentation

$A \rightarrow B$ alors $A, C \rightarrow B, C$

Si A détermine B , les deux ensembles d'attributs peuvent être enrichis par un même troisième.

Propriété 3 : Transitivité

$A \rightarrow B$ et $B \rightarrow C$ alors $A \rightarrow C$

Normalisation

✓ Axiomes et règles d'inférence

- Découvrir ***de nouvelles dépendances*** à partir d'un ensemble initial.
- Les trois premières propriétés sont connues sous le nom « **Axiomes d'Armstrong** »

Propriété 1 : Réflexivité

$B \text{ inclus-ou-égal } A$ alors $A \rightarrow B$

Tout ensemble d'attributs détermine lui-même ou une partie de lui-même.

Propriété 2 : Augmentation

$A \rightarrow B$ alors $A, C \rightarrow B, C$

Si A détermine B , les deux ensembles d'attributs peuvent être enrichis par un même troisième.

Propriété 3 : Transitivité

$A \rightarrow B$ et $B \rightarrow C$ alors $A \rightarrow C$

Normalisation

✓ Axiomes et règles d'inférence

Propriété 4 : Union

$A \rightarrow B$ et $A \rightarrow C$ alors $A \rightarrow B,C$

Propriété 5 : Pseudo-transitivité

$A \rightarrow B$ et $C,B \rightarrow D$ alors $C,A \rightarrow D$

Propriété 6 : Décomposition

$A \rightarrow B$ et $C \subset B$ alors $A \rightarrow C$

Normalisation

✓ Les DF élémentaires

- Une Dépendance fonctionnelle $A \rightarrow B$ est **élémentaire**
 - si pour tout $A' \subseteq A$ la dépendance fonctionnelle $A' \rightarrow B$ n'est pas vraie.
- En d'autres termes, B **ne dépend pas fonctionnellement** d'une partie de A
 - (A est la plus petite quantité d'information donnant B).
- Une DF de la forme $X, Y \rightarrow Z$ est **élémentaire**
 - SI : $X \rightarrow Z$ n'est pas une DF
 $Y \rightarrow Z$ n'est pas une DF

→Ainsi tous les attributs à gauche sont nécessaires pour déterminer l'attribut à droite de la flèche

Normalisation

✓ Les DF élémentaires

- Exemples :

NCIN, NV → DateAchat est une **DF élémentaire**

- NCIN, NV → Marque **n'est pas une DF élémentaire**,
 - car la NV → Marque est une DF
- RefProduit, LibelleProduit → PU **n'est pas une DF élémentaire**,
 - car il suffit d'avoir la référence du produit pour déterminer le prix unitaire

Normalisation

✓ Les DF élémentaires

- Une dépendance fonctionnelle $A \rightarrow B$ est **canonique**
 - si sa partie droite ne comporte qu'un seul attribut.
- Un ensemble F de dépendances fonctionnelles est canonique si chacune de ses dépendances est canonique.
 - Exemple

NumService → Adresse, Capacité : **n'est pas canonique**

RefProduit → LibelleProduit: **est une DF canonique**

Normalisation

✓ Les DF directes

- Une DF : $X \rightarrow Z$ est **directe**
- SI elle n'est pas déduite par une transitivité:
 - Il n'existe pas un Attribut Y tel que : $X \rightarrow Y$ et $Y \rightarrow Z$
- Exemples :
 - NV → Modèle **est une DF directe**
 - Modèle → Marque **est une DF directe**
 - NV → Marque **est une DF indirecte**

Normalisation

✓ Rappelons

- La clé **d'une relation** est l'ensemble d'attributs dont les valeurs permettent de caractériser les n-uplets de la relation de **manière unique**.
- Formellement :
 - Un attribut ou une liste d'attributs A est **une clé** pour la relation $R(A, B, C)$ si
B et C **dépendent fonctionnellement de A** dans R : $A \rightarrow B, C$
et $A \rightarrow B, C$ est élémentaire.
 - Une relation peut avoir plusieurs clés. Une clé sera choisie et désignée comme clé primaire. Les autres seront appelées clés candidates.

Normalisation

✓ Graphe de DF

- Présentation des DF à l'aide d'un graphe dont les nœuds sont les attributs impliqués dans les dépendances et les arcs les dépendances elles-mêmes.
- Les arcs sont orientés de la partie gauche de la dépendance vers sa partie droite.
- L'origine d'un arc peut être multiple mais sa cible doit être un nœud unique.
- De ce fait il est nécessaire d'avoir pour la construction d'un graphe de dépendance fonctionnelle un ensemble canonique de dépendances fonctionnelles.

Normalisation

✓ Graphe de DF

- Exemple :

F1 : RefProduit → LibelleProduit

F2 : RefProduit → PU

F3 : NumService → Adresse, Capacité

F4 : RefProduit, NumService → Quantité

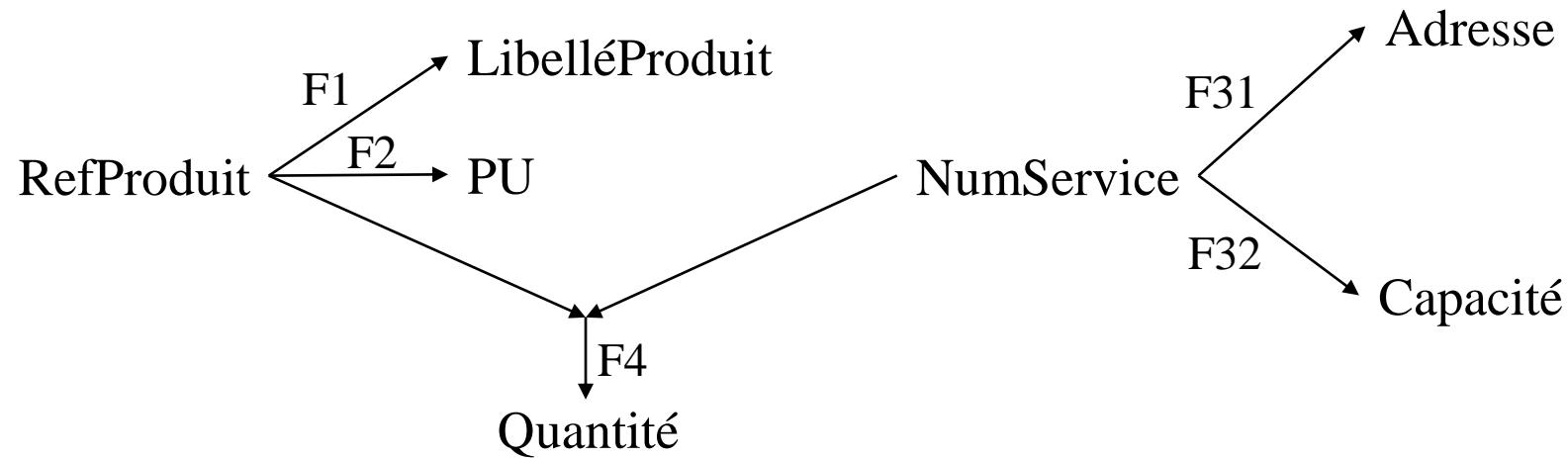
→ La dépendance fonctionnelle F3 **n'est pas canonique**, il faut donc la décomposer en deux dépendances fonctionnelles F31 et F32 :

F31 : NumService → Adresse

F32 : NumService → Capacité

Normalisation

✓ Graphe de DF



- Le graphe des dépendances fonctionnelles d'une relation R permet de trouver **les clés de R**
 - L'ensemble (minimal) des noeuds du graphe à partir desquels on peut atteindre tous les autres noeuds (via les dépendances fonctionnelles).

Normalisation

✓ La fermeture transitive

- La Fermeture Transitive d'un ensemble de Dépendances Fonctionnelles F ,
 - l'ensemble F^+ qui est **l'union de F** et de l'ensemble des Dépendances Fonctionnelles déduites par la **Transitivité**
- Définition Mathématique
 - $F = \{\text{DF élémentaires}\}$
 - $F^+ = F \cup \{ \text{DF Transitives de } F \}$

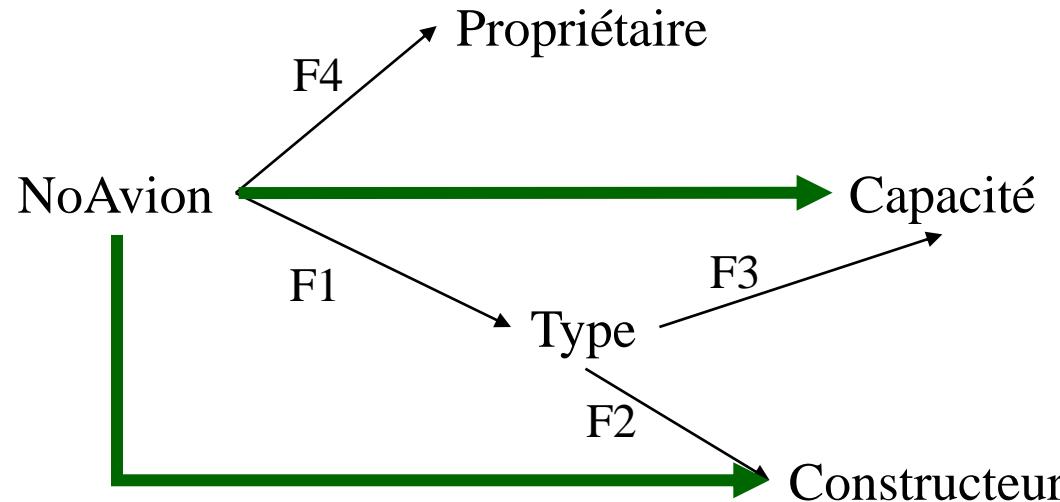
Normalisation

✓ La fermeture transitive

- L'ensemble des **dépendances fonctionnelles élémentaires** qui peuvent être produites par **application des axiomes d'Armstrong** sur l'ensemble F.
- Exemple
 - Avion(NoAvion, Type, Constructeur, Capacité, Propriété)
 - Avec les dépendances fonctionnelles suivantes :
 - $F1 : \text{NoAvion} \rightarrow \text{Type}$
 - $F2 : \text{Type} \rightarrow \text{Constructeur}$
 - $F3 : \text{Type} \rightarrow \text{Capacité}$
 - $F4 : \text{NoAvion} \rightarrow \text{Propriétaire}$
 - $F = \{F1, F2, F3, F4\}$

Normalisation

✓ La fermeture transitive



→ Les dépendances fonctionnelles en gras sont déduites par transitivité.

Normalisation

✓ La fermeture transitive

- Commande (numCde, dateCde, montant, numCli, nomCli, adrCli)

- $F = \{ \text{numCde} \rightarrow \text{dateCde} ;$

- $\text{numCde} \rightarrow \text{montant} ;$

- $\text{numCde} \rightarrow \text{numCli} ;$

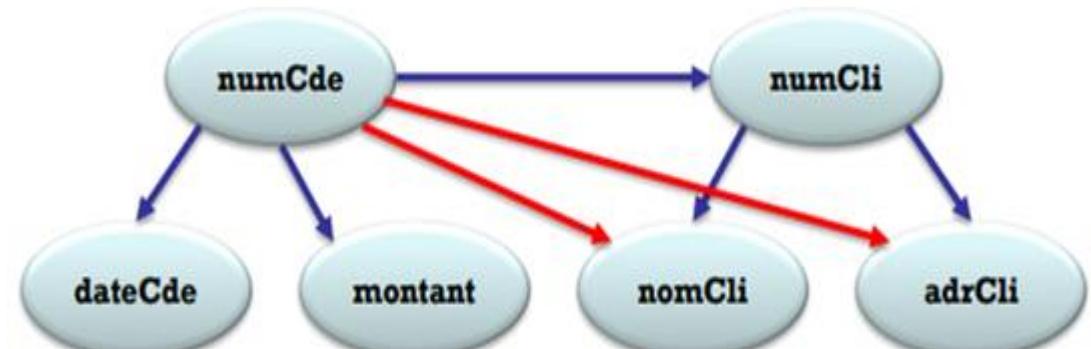
- $\text{numCli} \rightarrow \text{nomCli} ;$

- $\text{numCli} \rightarrow \text{adrCli} \}$

- $F^+ = F \cup \{ \text{numCde} \rightarrow \text{nomCli};$

- $\text{numCde} \rightarrow \text{adrCli} \}$

→ Graphe des DF obtenues à partir de F^+



Normalisation

✓ Le processus de Normalisation

- Les formes normales ont été définies pour permettre la **décomposition** des relations **sans perte d'informations** en utilisant la notion de **dépendance fonctionnelle**.
- La normalisation des relations est l'application d'un **ensemble de règles prédefinies**, introduites dans le modèle relationnel afin de garantir la **cohérence** de la base lors des différentes opérations de manipulation et de MAJ.
- Le processus est basé sur 5 niveaux ou 5 formes normales appelées dans l'ordre 1FN, 2FN, 3FN, 4FN et 5FN.

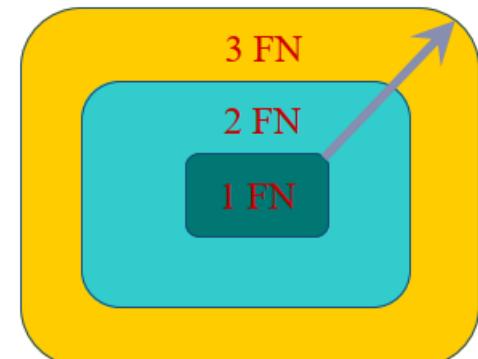
Normalisation

✓ Le processus de Normalisation

- Le processus de normalisation passe par plusieurs étapes constituant les différentes formes normales qui se

présentent comme des règles de construction.

- On mesure la qualité d'une relation par son degré de normalisation (1FN, 2FN, 3FN).
- Les différentes formes normales sont dépendantes et structurées. En effet une relation ne peut être en 2FN que si elle est déjà en 1FN.



Normalisation

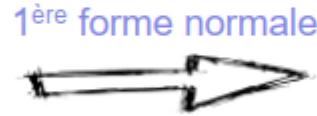
✓ Le processus de Normalisation: **1 FN**

■ Une relation R est dite **normalisée (1FN)** lorsque

- Elle possède une clé
- tous ses domaines sont **simples et atomiques** (non décomposable)
 - aucun d'eux n'est un ensemble d'ensembles.

Exemple :

Etudiant
Nom
Prénom
Age
Université

1^{ère} forme normale


Etudiant
<u>CNE</u>
Nom
Prénom
Age



Université
<u>IdUniv</u>
Nom
Adresse

Normalisation

- ✓ Le processus de Normalisation: **1 FN**

- Exemple
 - Produit (NumP, NomP, **adrF**)
 - SI « adrF » est elle-même **composée** de (Rue, Ville, CodePostal)
ALORS Fournisseur **n'est pas en 1FN**
 - Pour rendre Fournisseur en 1FN, il faut créer autant d'Attributs que ceux qui composent l'Adresse Fournisseur
(NumF, NomF, Rue, Ville, CodePostal)
 - Exemple
 - LIVRE (No-ISBN, Titre, Auteurs, Editeur)**
 - Cette relation **n'est pas en 1FN** car l'attribut "Auteurs" est **multivalué**.
 - Un auteur ne peut pas y être traité d'une façon individuelle (exemple: tri des livres par nom d'auteur).
 - Cette relation peut par exemple être transformée vers la nouvelle relation :
LIVRE (No-ISBN, Titre, Auteur1, Auteur2, Auteur 3, Editeur)

Normalisation

✓ Le processus de Normalisation: **2 FN**

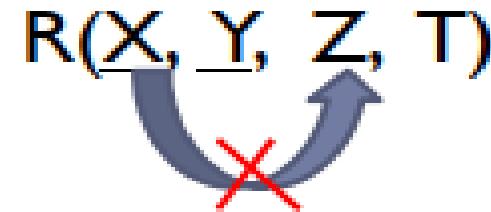
▪ Une Relation R est en 2FN si et seulement si :

▪ R est en 1FN

▪ Chaque Attribut **non clé** de R est en **DF élémentaire (totale)** avec **la clé**

▪ 2FN → Tout Attribut **n'appartenant pas à la clé ne dépend pas d'une partie de la clé**

▪ 2FN → Chaque Attribut qui **ne fait pas partie de l'identifiant doit dépendre** de **l'identifiant entier**



Exemple :

Location
<u>IdClient</u>
<u>IdAppartement</u>
Montant
AdresseAppartement



Location
<u>IdClient</u>
<u>IdAppartement</u>
Montant



Appartement
<u>IdAppartement</u>
AdresseAppartement

Normalisation

✓ Le processus de Normalisation: **2 FN**

- Exemple: Soit la relation CLIENT avec ses DF

- CLIENT (NumCl, AdrCl, RefProduit, PU) avec

- F1 : NumCl, RefProduit → PU
 - F2 : NumCl → AdrCl

- **1 FN:** Tout les attributs sont simples

- La clé de la relation est (NumClt, RefProduit)

→ Suite à F1, une partie de la clé (NumClt) détermine un attribut n'appartenant pas à la clé. **Cette relation n'est donc pas en 2FN.**

- Elle pourra être décomposée en :

- CLIENT (NumCl, AdrCl)
 - PRODUIT (RefProduit, #NumCl, PU)

Normalisation

✓ Le processus de Normalisation: 3 FN

- Une Relation R est en 3FN si et seulement si :
 - R est en 2FN
 - Il n'existe aucune DF **Transitive entre les Attributs non clé**
 - 3FN → Tout Attribut **n'appartenant pas à la clé** ne dépend pas d'un autre A:
 - 3FN → Permet d'éliminer des Sous-Relations incluses dans une Relation

→ Importance de la 3FN

- Toute Relation peut être décomposée en 3FN sans aucune perte : de DF et d'informations
- Il faut donc toujours faire des Schémas au moins en 3FN

Exemple :

Appartement
<u>IdApt</u>
AdresseApt
CIN_Propriétaire
NOM_Propriétaire

3^{ème} forme normale

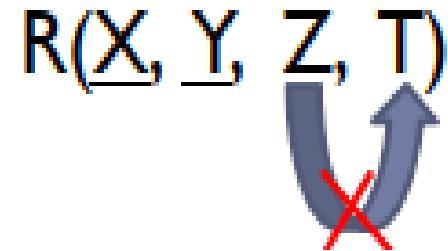


The diagram shows the decomposition of the 'Appartement' relation into two smaller relations. An arrow points from the original 'Appartement' relation to a plus sign (+), which then connects to two new tables: 'Appartement' and 'Propriétaire'. The 'Appartement' table contains attributes 'IdApt', 'AdresseApt', and 'CIN_Propriétaire'. The 'Propriétaire' table contains attributes 'CIN_Propriétaire' and 'NOM_Propriétaire'.

Appartement
<u>IdApt</u>
AdresseApt
CIN_Propriétaire



Propriétaire
CIN_Propriétaire
NOM_Propriétaire



Normalisation

✓ Considérons la Relation « Fournisseur » ayant le Schéma suivant :

Fournisseur (NumFrs, NomFrs, Ville, CodePostal)

✓ Les DF sont :

- ✓ **DF1** : NumFrs → NomFrs, Ville, CodePostal
- ✓ **DF2** : CodePostal → Ville

✓ **Question** : Que pensez-vous de « Fournisseur »?

« Fournisseur » est en **2FN** (i.e. DF1) **mais n'est pas en 3FN**

(DF2: l'Attribut *non clé* « Ville » *dépend* d'un autre Attribut *non clé* « CodePostal »)

✓ **Question** : Comment rendre « Fournisseur » en 3FN ?

➤ Solution : Pour passer cette Relation à 3FN, il faut la décomposer comme suit :

- Fournisseur (NumFrs, NomFrs, CodePostal)
- Adresse (CodePostal, Ville)

→ Les nouvelles Relations « Fournisseur » et « Adresse » sont **en 3FN**

Normalisation --- Résumé

- ✓ Que faire si une relation n'est pas « normalisée » ? **DECOMPOSITION**
- ✓ La relation doit être remplacée par un ensemble de relations (plus petites : moins d'attributs)
- ✓ Une « bonne » décomposition est une décomposition
 - 1) sans perte d'information
 - 2) sans perte de DF : toute DF doit être dans l'une des relations obtenues par décomposition.