

# Formes normales des Base de données relationnelles

**André Miralles**

# Normalisation des données

## » Objectif

> Éviter les anomalies découlant d'une mauvaise modélisation des données

+ Traduit par

- Des anomalies de lecture
- Des anomalies d'écriture
- De la redondance des données
- Des contre-performances au niveau des requêtes

## » Normalisation des données

> Vérifier la robustesse de la conception pour améliorer le modèle (et donc obtenir une meilleure représentation)

> Faciliter le stockage des données en évitant la redondance

+ Faciliter le problème sous-jacent de mise à jour des données ou de cohérence

» **Forme normale** est un type de relation particulier entre les entités.



# Normalisation des données

» La normalisation s'applique à **toutes les entités** et aux **relations porteuses de propriétés**

» Propriétés des formes normales

- > Elles **s'emboîtent** les **unes** dans les **autres**, tant et si bien que le respect d'une forme normale de niveau supérieur implique le respect des formes normales des niveaux inférieurs

**3 premières sont les plus connues et les plus utilisées**

» Dans le modèle relationnel (OLRP), il existe huit formes normales

- > la **première** forme normale notée **1FN** (1NF en anglais)
- > la **deuxième** forme normale notée **2FN** (2NF en anglais)
- > la **troisième** forme normale notée **3FN** (3NF en anglais)
- > la forme normale de **Boyce Codd** notée **FNBC** (BCNF en anglais)
- > la **quatrième** forme normale notée **4FN** (4NF en anglais)
- > la **cinquième** forme normale notée **5FN** (5NF en anglais)
- > la forme normale **domaine clé** notée **FNDC** (DKNF en anglais)
- > la **sixième** forme normale notée **6FN** (6NF en anglais) rarement présentée

# 1NF – Première forme normale

## » Forme normale satisfaite si

> Valeur d'un attribut est atomique pour une valeur de clé primaire

## » Unicité des données pour une valeur de clé primaire

> Pour un triplet a1, a2, a3, le triplet a4, a5, a6 est unique

$$R (\underline{a1}, \underline{a2}, \underline{a3} \rightarrow a4, a5, a6)$$

## » Exemple de Normalisation

> Partitionnement des tuples

Produit	Fournisseur	
téléviseur	VIDEO SA	HITEK LTD



Produit	Fournisseur
téléviseur	VIDEO SA
téléviseur	HITEK LTD



# 2NF – Deuxième forme normale


## » Forme normale satisfaite si

- > **Première forme normale** satisfaite
- > Chacun des **attributs non impliqués dans la clé primaire** (a4, a5, a6) ne doivent pas dépendre uniquement d'un sous-ensemble d'attributs de la clé primaire

## » Exemple de Violation

- > L'attribut a5 dépend uniquement de a2

$R(\underline{a1}, \underline{a2}, \underline{a3}, a4, a5, a6)$



## » Exemple de Normalisation

- > **Partitionnement des tables**

<u>Produit</u>	<u>Fournisseur</u>	Adresse fournisseur
téléviseur	VIDEO SA	13 rue du cherche-midi
écran plat	VIDEO SA	13 rue du cherche-midi
téléviseur	HITEK LTD	25 Bond Street



<u>Produit</u>	<u>Fournisseur</u>
téléviseur	VIDEO SA
téléviseur	HITEK LTD
écran plat	VIDEO SA

<u>Fournisseur</u>	Adresse fournisseur
VIDEO SA	13 rue du cherche-midi
HITEK LTD	25 Bond Street

# 3NF – Troisième forme normale


## » Forme normale satisfaite si

- > **Seconde forme normale** satisfaite
- > Les **attributs non impliqués dans la clé primaire** (a4, a5, a6) ne doivent **pas** avoir dépendance entre eux

## » Exemple de Violation

- > L'attribut a6 dépend de a4

$R(\underline{a1}, \underline{a2}, \underline{a3}, a4, a5, a6)$



## » 1° Exemple de Normalisation

- > L'attribut **NOM\_CLIENT** dépend de **CLIENT\_ID**

<u>COMMANDE_ID</u>	<u>CLIENT_ID</u>	NOM_CLIENT
1	1	Durand

>



➔ **CLIENT\_ID** devient clé primaire

<u>COMMANDE_ID</u>	<u>CLIENT_ID</u>	NOM_CLIENT
1	1	Durand

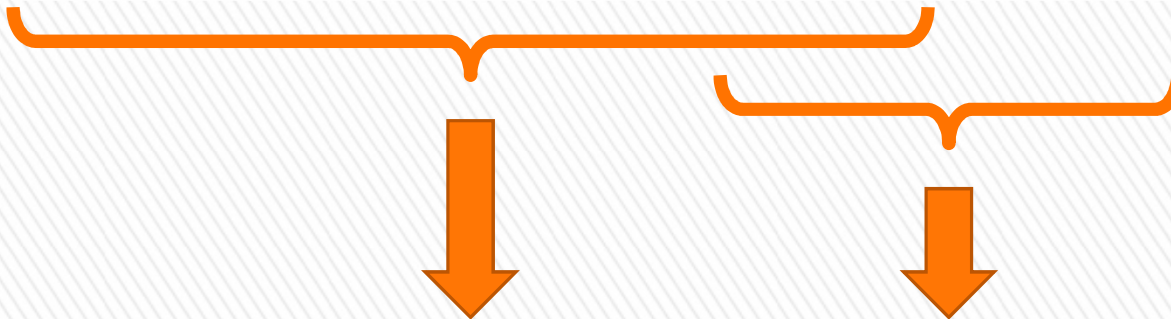


# 3NF – Troisième forme normale

## » 2° Exemple de Normalisation

### > Partitionnement des tables

<u>Fournisseur</u>	Adresse fournisseur	Ville	Pays
VIDEO SA	13 rue du cherche-midi	PARIS	FRANCE
HITEK LTD	25 Bond Street	LONDON	ENGLAND



<u>Fournisseur</u>	Adresse fournisseur	Ville
VIDEO SA	13 rue du cherche-midi	PARIS
HITEK LTD	25 Bond Street	LONDON

<u>Ville</u>	Pays
PARIS	FRANCE
LONDON	ENGLAND

# Avantages / Inconvénient de la Normalisation

## » Principaux avantages

- > **Limiter les redondances** de données (multiples écritures)
- > **Diminuer la volumétrie** globale
  - + Réduction de espace disque nécessaire
  - + Mais surtout la taille du cache (les SGBDR travaillant en mémoire)
- > **Interdire les incohérences de données** venant des redondances dont une partie seulement a été mise à jour
  - + Les données deviendraient inutilisables (différentes versions d'une même information, sans que l'on sache laquelle est valide)
- > **Limiter fortement le nombre et la durée des mises à jour** qui sont des processus bloquants (écritures)

**Application des Formes Normales 1NF et 2NF  
conduit partition des tuples (1NF) ou des tables (2NF, 3NF)**

**Vérification des Formes Normales  
doit être effectuée pour TOUTES les données**



# Avantages / Inconvénient de la Normalisation

## » Principal inconvénient

- > **Temps d'accès aux données plus longs** lorsque les requêtes sont complexes (nombreuses jointures)
  - + Temps de 30 min à 24h

## » Comment contourner cet inconvénient ?

- > **Petites** bases de données ==> Utiliser les 3<sup>o</sup> formes normales
- > Bases de données **plus importantes**
  - + Nécessité de **dénormaliser** de façon intelligente
    - Utiliser un **benchmark** pour prouver les **gains de temps**

# Avantages / Inconvénient de la Normalisation

## » 5 principes pour dénormaliser de façon intelligente une base de données

- ~~> Rajouter des colonnes calculées persistantes, voire des index calculés~~
- > Mettre en place des index couvrants et/ou filtrés
- > Utiliser la compression des données
- > Utiliser des vues matérialisées (Oracle ou PostgreSQL) ou indexées (SQL Server)
- > Partitionner les très grandes tables

**Mise en œuvre de la technologie BI / ID**  
(Business Intelligence / Informatique Décisionnelle)



# Conclusion

**Problème majeur est que la vision des Modèles Relationnels et Entités Associations (MERISE) est une vision donnée**



**Formalisme Objet et le Langage UML**

# Quelques sites utiles

## » SQL

- > [https://fr.wikipedia.org/wiki/Forme normale \(bases de donn%C3%A9es relationnelles\)](https://fr.wikipedia.org/wiki/Forme_normale_(bases_de_donn%C3%A9es_relationnelles))