

**UNIVERSITE BADJI MOKHTAR-ANNABA**  
FACULTÉ DES SCIENCES DE L'INGENIEUR  
DÉPARTEMENT D'INFORMATIQUE



# Normalisation

BDD 2LMD

Partie 2: Les formes Normales

Présenté par : Dr BELLEILI Habiba

# Définitions

- “ Les formes normales ont pour objectif de définir la décomposition des schémas relationnels, tout en **préservant les DF** et sans **perdre d'informations**,
- “ On s'intéresse à 4 formes Normales:
  - . la première forme normale
  - . la deuxième forme normale
  - . la troisième forme normale
  - . La forme normale de Boyce Codd (FNBC)

# Décomposition sans Perte d'information

“ Une décomposition d'une relation  $R(X,Y,Z)$  en deux relations

$R1 = \pi[X,Y]R$  et  $R2 = \pi[X,Z]R$  est dite "sans perte d'information" si

$R = R1 \text{ join } R2$

$\pi [\text{NomEmp, Adresse, poste}] \text{ Employé}$

$\text{Employé} (\text{NomEmp, Adresse, poste, age})$

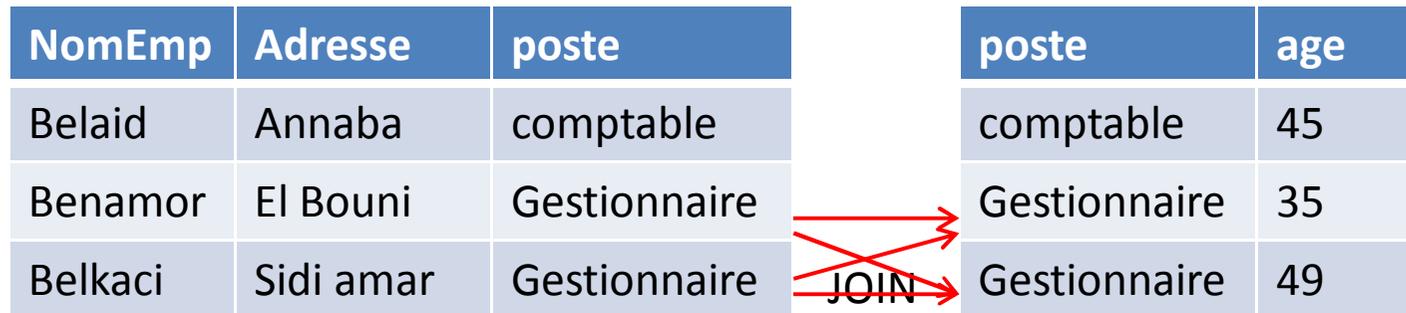
NomEmp	Adresse	Poste	age
Belaid	Annaba	Comptable	45
Benamor	El Bouni	Gestionnaire	35
Belkaci	Sidi amar	Gestionnaire	49

NomEmp	Adresse	poste
Belaid	Annaba	comptable
Benamor	El Bouni	Gestionnaire
Belkaci	Sidi amar	Gestionnaire

$\pi[\text{poste, age}] \text{ Employé}$

poste	age
comptable	45
Gestionnaire	35
Gestionnaire	49

NomEmp	Adresse	Poste	age
Belaïd	Annaba	Comptable	45
Benamor	El Bouni	Gestionnaire	35
Belkaci	Sidi amar	Gestionnaire	49



NomEmp	Adresse	poste	age
Belaïd	Annaba	comptable	45
Banamor	Elbouni	Gestionnaire	35
Banamor	El Bouni	Gestionnaire	49
Belkaci	Sidi Amar	Gestionnaire	35
Belkaci	Sidi Amar	Gestionnaire	49

≠

=

Perte  
d'information

# Théorème de Heath

“ Toute relation  $R(X,Y,Z)$  est décomposable **sans perte d'information** en  $R_1=\pi[X,Y]R$  et  $R_2=\pi[X,Z]R$  s'il y a dans  $R$  une dépendance fonctionnelle de  $X$  vers  $Y$  ( $X \rightarrow Y$ ) et de  $X$  vers  $Z$  ( $X \rightarrow Z$ ).

“ Dans l'exemple:  
il n'y a pas de DF **poste  $\rightarrow$  age**

poste	age
comptable	45
Gestionnaire	35
Gestionnaire	49

“ Il **ne faut jamais faire** une décomposition avec **perte de données** car les informations contenues dans la base sont alors *fausses* et il est **impossible de rétablir les bonnes données**.

# Perte de dépendances fonctionnelles

- “ Lorsque l'on décompose R en deux relations R1 et R2.
- “ Les DF (sources et cibles) de la relation R doivent être préservés dans les relations R1 et R2
- “ Une décomposition en plusieurs relations est dite **sans pertes de dépendances** si l'on peut retrouver les DF de départ dans les relations résultant de la décomposition.
- “ Si on décompose avec une perte de dépendances, on ne peut travailler indépendamment sur chacune des relations de la décomposition sans s'occuper des autres relations.
- “ Lors des MAJ on ne peut s'assurer que les dépendances (perdues) sont vérifiées qu'en consultant les autres relations, ce qui occasionne des traitements coûteux .
- “ Il faut préférer les décompositions **sans pertes de dépendances**. Mais une perte de dépendances est moins grave qu'une perte de données.

# LES FORMES NORMALES

# Méthode

- “ Normaliser une relation qui présente des redondances consiste à décomposer cette relation en relations sans redondances (relations normalisées).
- “ La méthode à suivre est la suivante:
  - 1/ vérifier que la relation est en 1<sup>ière</sup> forme normale**
  - 2/ établir son **graphe minimum des dépendances**;
  - 3/ déterminer, à l'aide du graphe, **toutes ses clés**;
  - 4/ déterminer, à l'aide du graphe, sa forme normale**
  - 5/ si la relation n'est pas normalisée, décomposer, à l'aide du graphe, la relation en relations mieux normalisées.**

# 1<sup>ière</sup> forme normale

- “ Une relation est en 1<sup>ière</sup> FN si elle possède **au moins une clé** et si tous ses attributs sont **atomiques**.
- “ **Un attribut est atomique** si il ne contient qu'une seule valeur pour un tuple donné, et donc s'il ne regroupe pas un ensemble de plusieurs valeurs.

## Définition 2

“ Une relation est en **1<sup>ère</sup> forme normale** si **chaque valeur de chaque attribut de chaque tuple** est une valeur simple (tous les attributs sont simples et monovalués).

# Attribut simple/composé

- “ Attribut simple: est un attribut présentant une seule information,
- “ Le contraire est un attribut composé.

Exemple:

- . L'adresse peut être considérée comme un attribut composé: 1, boulevard du 1<sup>ier</sup> Novembre 23000 Annaba Algérie
- . Le numéro de compte bancaire: code banque, code agence, numéro de compte

# Attribut monovalué/multivalué

- “ Attribut **monovalué**: est attribut qui étant donnée un tuple ne peut avoir qu'une seule valeur pour cet attribut. Exemple :
  - . Numéro d'étudiant: un étudiant ne peut avoir qu'un seul numéro
  - . le nom de l'étudiant: un étudiant ne peut avoir qu'un seul nom
  - . la date de naissance: une personne n'a qu'une seule date de naissance
  
- “ Un attribut qui n'est pas monovalué est dit **multi-valué**:
  
- “ Un attribut est dit multi-valué s'il peut présenter *pour un même tuple* plusieurs valeurs possibles. Exemple:
  - . Numéro de téléphone: une société a plusieurs numéro de téléphone
  - . Prénom: une personne peut avoir plusieurs prénoms
  - . Numéro de compte: une personne peut avoir plusieurs numéro de compte (courant, épargne, devise)

# Exemple

- ” Société (codeS, nomSociété, ville, tél)
- ” Cette relation n’est pas en 1FN puisque l’attribut téléphone n’est pas atomique il est multi-valué.  
Exemple:  
(12330, ‘SONELGAZ’, ‘Annaba’, ‘038403023’, ‘038662310’, ‘038583030’,...)
- ” Pour obtenir la 1FN on doit créer une relation qui contient la clé de la relation **concaténée** avec l’attribut multi-valué:

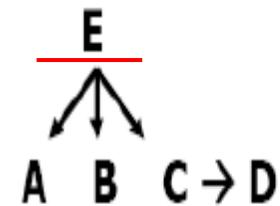
**R(codeS,tél)** (12330,’038403023’),  
(12330,’038662310’),  
(12330,’038583030’)....

**Société (codeS,nomsociété, ville)**

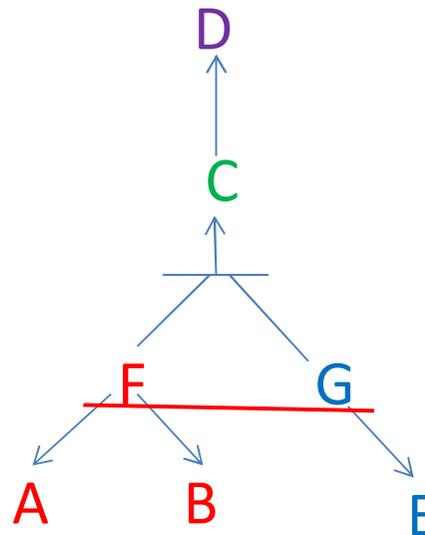
## 2<sup>ème</sup> FN

“ Une relation est en 2<sup>ème</sup> FN si :

- Elle est en 1FN, et
- Chaque attribut **qui ne fait pas partie de la clé dépend de la clé entière et non pas d'une partie de la clé.**



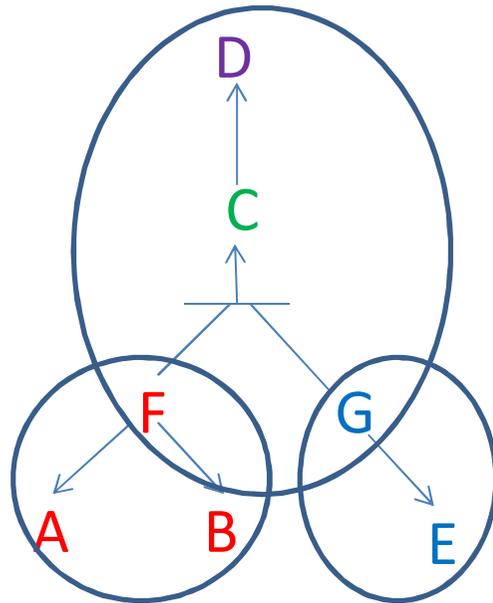
2 FN



N'est pas en 2 FN

# Décomposition en 2FN

F+G Clé



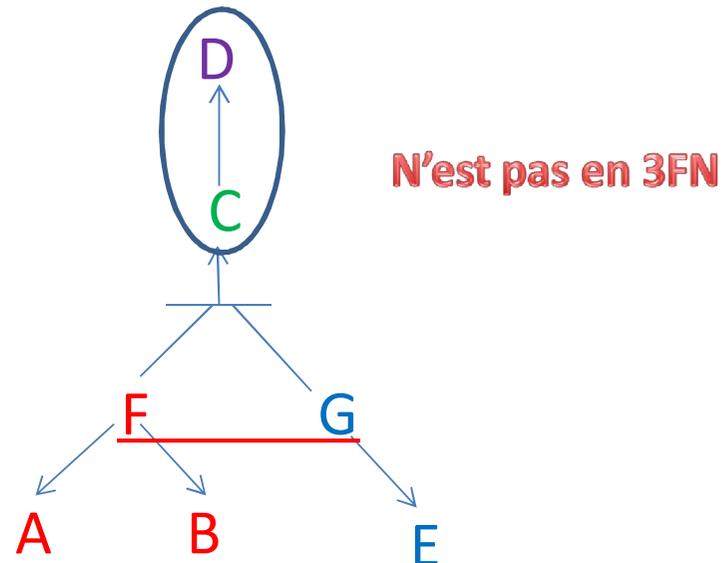
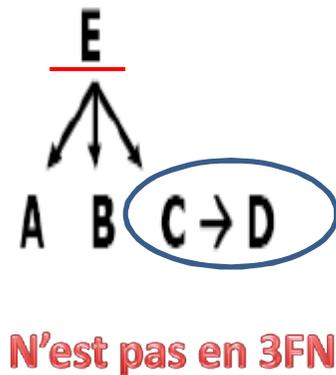
**R1(F,A,B)**

**R2(G,E)**

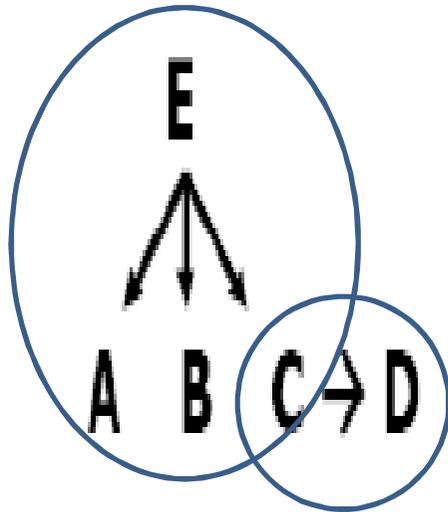
**R3(F,G,C,D)**

# 3<sup>ème</sup> FN

- “ Une relation est en 3<sup>ème</sup> FN si
- Elle est en 2FN, et
  - Chaque attribut qui ne fait pas partie d'aucune Clé dépend **directement de la clé**.

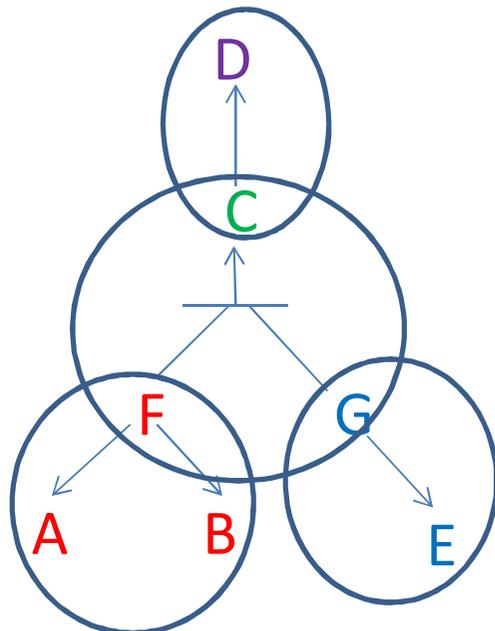


# Décomposition en 3FN



$R1(\underline{C}, D)$

$R2(\underline{E}, A, B, C)$



$R1(\underline{C}, D)$

$R2(\underline{F}, A, B)$

$R3(\underline{G}, E)$

$R4(\underline{F}, G, C)$

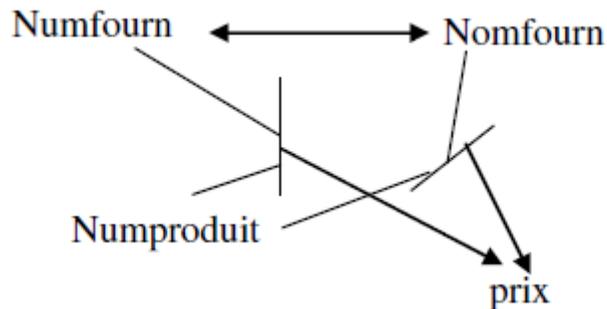
# Décomposition en 3FN

## Propriétés

- “ Toute relation peut toujours être décomposée en 3FN **sans aucune perte d'information**:
  - . sans perte de **DF** et
  - . sans perte de **données**.
- “ Cependant une relation en 3FN peut encore présenter des **redondances**.
- “ C'est le cas où la relation présente **plusieurs clés candidates**

# Exemple

- . catalogue(Numfourn, Nomfourn, Numproduit, prix)
- ” Avec le graphe minimum des dépendances suivant :



Cette relation présente deux clés:  
**(numfourn, numproduit)** et  
**(nomfourn, numproduit)**

La relation catalogue est en 3FN

**Elle présente toujours des redondances**

Pour ce type de relation en fait appel à  
la Forme Normale de Boyce Cod (FNBC) 19

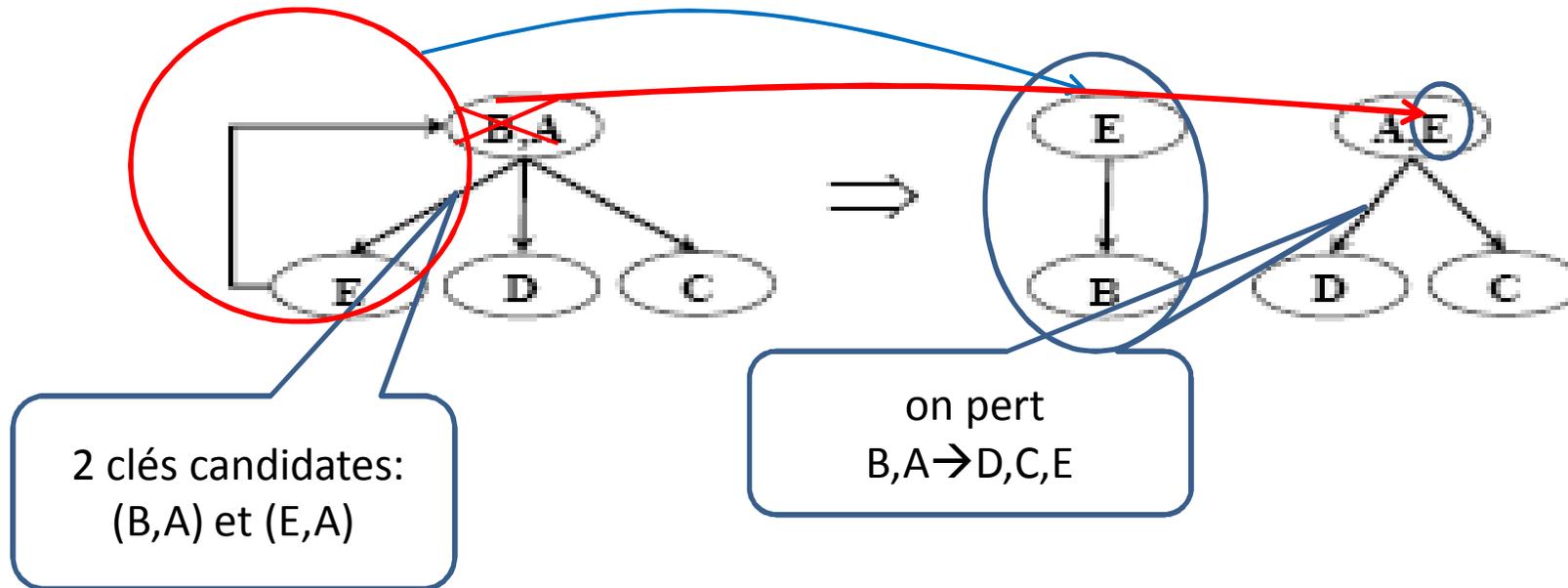
# FNBC (Forme Normale de Boyce Codd)

- “ Une relation est en FNBC si elle est en **3FN** et:
- “ si toute source de DF est **une clé entière**
- “ il n'existe pas d'attribut non clé qui détermine une partie de la clé

# Exemple

- “ Dans la relation catalogue présentant 2 clés candidates: **(numfourn+numproduit)** et **(nomfourn+ numproduit)**, les DF
  - . **numfourn** → nomfourn présente une source qui n'est pas Clé entière
  - . **nomfourn** → numfourn aussi
  - . Donc la relation catalogue n'est pas en FNBC
- “ Décomposition:
  - . R1(numFour, nomFour) avec deux clés candidates numfour et nomfour elle est en FNBC
  - . R2(numfourn,numproduit,prix) **NumFourn,Numproduit** → prix, R2 est en FNBC

# Décomposition en FNBC



- 1. Isoler la DF **problématique** dans une nouvelle relation
- 2. Éliminer la cible de cette DF et la remplacer par sa source dans la relation initiale.

# FNBC

## propriétés

- “ Une relation en FNBC est en 3ième forme normale, le contraire n’est pas vrai.
- “ La normalisation en FNBC peut conduire à des **pertes d’information ou de DF.**

# Exemple

“ Enseignant (NEtud,Matière,Prof)

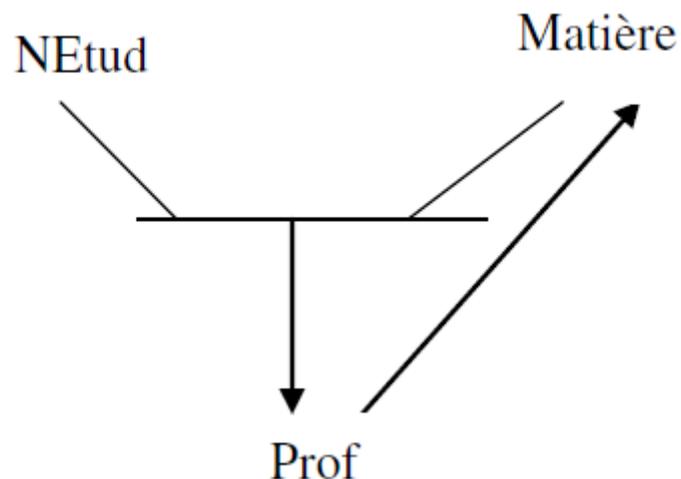
Avec les règles de gestion :

1) chaque professeur enseigne une seule matière

“ Prof → Matière

2) pour chaque matière, chaque étudiant suit les cours **d'un seul professeur**

“ NEtud, Matière → Prof



(NEtud+Matière) et  
(NEtud+Prof).

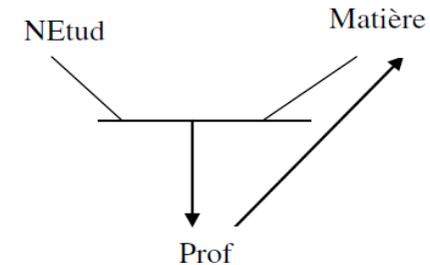
Elle est en 3 FN

Elle n'est pas en FNBC: en  
FNBC toutes les DFE sont  
issues d'une clé c'est pas le  
cas de la DF Prof → Matière

## Exemple (suite)

- “ Enseignant (NEtud,Matière,Prof)
  - . (123,'BDD',789), (123,'LM',896),(258,'BDD',789),(126,'LM',896)
- “ Cette relation présente **des redondances** dues au fait que la DF **prof → matière** n'est pas représentée par une relation qui lui ai dédiée.
- “ Si un enseignant change de spécialité, il faut faire cette MAJ dans tous les tuples où apparaît ce professeur.

# Décomposition en FNBC



- ” Enseignant (NEtud,Matière,Prof)
- ” Décomposition en FNBC:
  - . R1 (Prof,Matière) prof → matière //R1 en FNBC
  - . R2 (NEtud,Prof) //R2 est en FNBC
- ” La décomposition en FNBC conduit à **la perte** de DF (**NEtud,Matière**)→Prof
- ” Cette perte de DF va permettre d’insérer le fait qu’un même étudiant suit deux cours portant sur la même matière avec deux professeurs différents, ce qui est interdit dans la relation enseignant (règle de gestion 2)
  - . R1: (1258,’BDD’), (1358,’BDD’)
  - . R2: (125,1258), (~~125~~,1358) (126, 1258), (~~126~~, 1358)
- ” **Une décomposition en FNBC ne préserve pas toujours les DF.**

# **FIN NORMALISATION (PARTIE 2)**