Université d'Alger 1
Benyoucef Benkhadda

Faculté des Sciences
Département MI

BASES DE DONNÉES

COURS:

LES FORMES NORMALES

MODÈLE RELATIONNEL

Présenté par:
Dr. A BOUTORH

Licence 2

Informatique

LA THÉORIE DE LA NORMALISATION

- Elle met en évidence les relations **indésirables**
- Elle définit les critères des relations **désirables** appelées les **formes normales**.
- Propriétés indésirables des relations:
  - **Redondances**
  - **Valeurs NULL**
- Elle définit le processus de normalisation permettant de **décomposer** une relation non normalisée en un ensemble équivalent de relations normalisées.

NORMALISATION

- **Normaliser une relation qui pose des problèmes de mise-à-jour (*relation non normalisée*), consiste à décomposer cette relation en relations sans problèmes (relations normalisées). La méthode à suivre est la suivante:**

- **1/ vérifier que la relation est en première forme normale**

- **2/ établir son graphe minimum des dépendances;**

- **3/ déterminer, à l'aide du graphe, tous ses identifiants;**

- **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**

DÉCOMPOSITION

- **Objectifs:**

✓ Décomposer les relations du schéma relationnel *sans perte d'informations*

✓ Obtenir des relations *canoniques* ou de base du monde réel

✓ Aboutir au schéma relationnel *normalisé*

- Le schéma de départ est le schéma universel de la base

- Par raffinement successifs on obtient des sous relations sans perte d'informations et qui ne seront pas affectées lors des mises à jour « **non redondance** »

NORMALISATION

- **Le processus de normalisation est le processus de transformation d'une relation *posant des problèmes* lors des mises à jour en relations *sans* ces problèmes.**

- **La capacité d'une relation à représenter le monde réel sans générer des problèmes est connue par *la qualité d'une relation*.**

- **La qualité d'une relation est mesurée par son *degré de normalisation*.**

NORMALISATION

- **Une relation peut être, *de la moins bonne à la meilleure*, en :**

➢ **1ère Forme Normale**

➢ **2ème Forme Normale**

➢ **3ème Forme Normale**

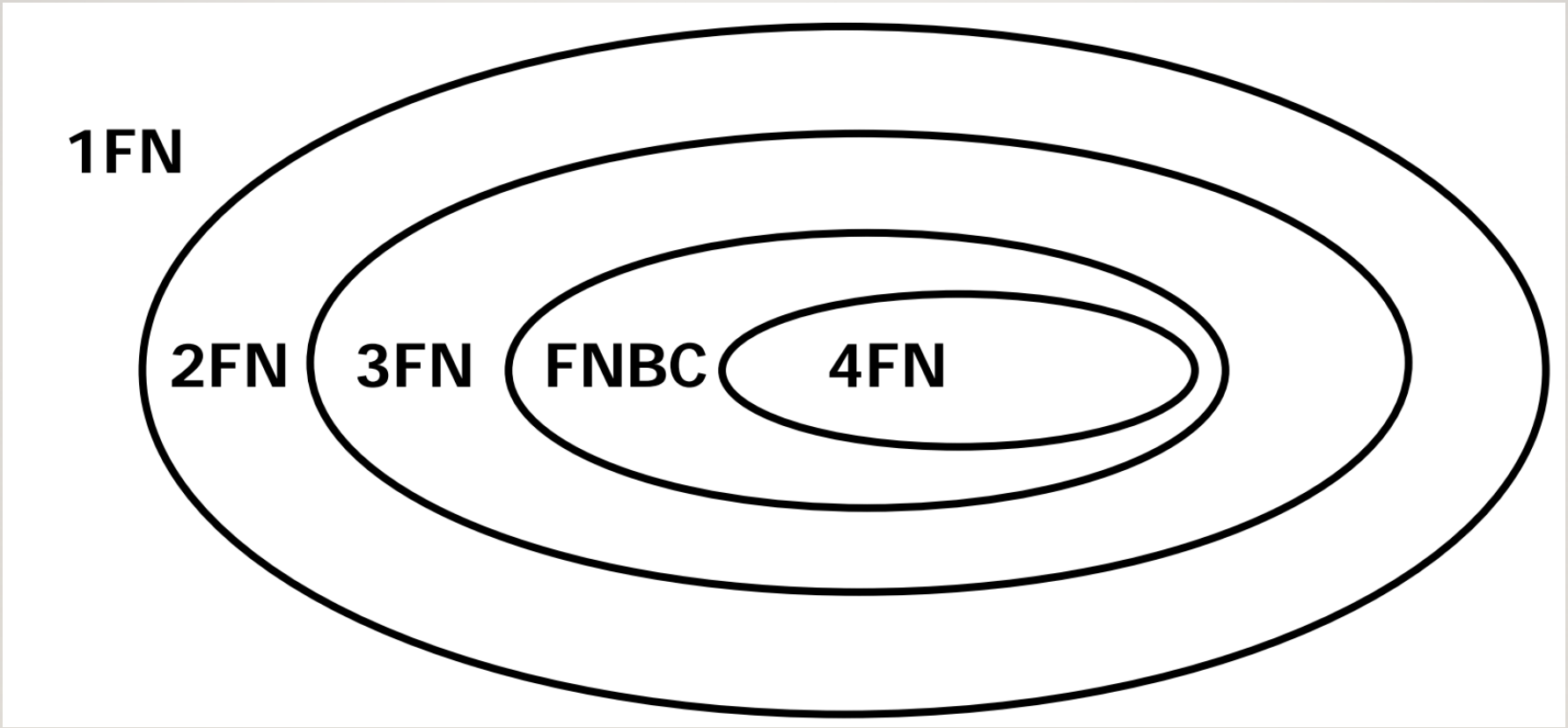
➢ **en Forme Normale de Boyce Codd**

➢ **4ème Forme Normale**

➢ **…**

❖ **Chaque forme normale implique les précédentes.**

NORMALISATION



Référence

LES FORMES NORMALES

- **Les critères sont de plus en plus *restrictifs*.**

$$FNj \Rightarrow FNi \quad (j > i)$$

- **Une *bonne relation* peut être considérée comme une fonction de la *clé primaire* vers les *attributs restants***

DÉPENDANCES FONCTIONNELLES

- **Définition :** Etant donné une relation, **R (X , Y , Z)**,

  il existe une **dépendance fonctionnelle**, ou « **DF** »,

  de **Y** vers **Z** , Notée **Y → Z** si :

✓étant donné **deux tuples** quelconques de **R**,

s'ils ont **même valeur** pour **Y**, alors ils ont nécessairement **même valeur** pour **Z**.

✓**Y source** de la **DF**, et **Z cible** de la **DF**

DÉPENDANCES FONCTIONNELLES

- **Exemple: La relation Produit (NP, NomP, Poids, Couleur), il y a les DF suivantes en supposant qu'il n'existe pas deux produits de même nom:**

- **NP →NomP,** **NomP → NP,**
- **NP →Poids,** **NomP → Poids,**
- **NP →Couleur,** **NomP → Couleur,**
- **NP → (NomP, Poids, Couleur),** **(NP, NomP) →Poids,**
- **(NP, NomP) →Couleur**

DÉPENDANCE FONCTIONNELLE ÉLÉMENTAIRE

- **Définition :** une DF, X → B, est une dépendance fonctionnelle élémentaire si B est un attribut unique, et si X est un ensemble minimum d'attributs (ou un attribut unique).

Exemples : Dans la relation Produit, les DF :

NP → (couleur, poids) et (NP, NomP) →Poids ne sont pas élémentaires.

Mais les DF : NP → Couleur, NP → Poids, NP →NomP , NomP → Couleur, NomP → Poids, NomP →NP sont élémentaires.

- Relation: Livraison (NP, NF, Date, Qté) La DF :

(NP, NF, date) → Qté est élémentaire

DÉPENDANCES FONCTIONNELLES

- **Chaque DF traduit un fait du monde réel.**
- **Les DF élémentaires traduisent des faits atomiques.**
- **La DF, NP → Couleur, signifie que: *chaque produit, identifié par son numéro, est dune seule couleur*.**
- **La DF, (NP, NF, Date) → Qté, signifie qu'*un même fournisseur ne peut livrer plusieurs fois le même jour le même produit avec des quantités différentes*.**

GRAPHE DES DÉPENDANCES FONCTIONNELLES

- **Propriété des dépendances fonctionnelles:**

Si dans une relation, on a les deux dépendances fonctionnelles, X→Y et Y→Z, alors on a aussi la dépendance fonctionnelle X→Z qui est dite déduite des deux autres.

- **Définition :**

Etant donné une relation et un ensemble F de DF portant sur les attributs de cette relation, on appelle graphe minimum des DF de la relation, tout ensemble de DF élémentaires non déduites, équivalent à F en ce sens que toute DF de F peut être déduite des DF du graphe.

DÉPENDANCES FONCTIONNELLES ET GRAPHE DES DÉPENDANCES FONCTIONNELLES

- **Une méthode pour savoir si une DF, X→Y, est déduite des autres est la suivante:**

➢ **Etablir un graphe (non minimum) des DF,**

➢ **Supprimer la DF X→Y du graphe,**

➢ **Parcourir tous les chemins possibles partant de X et suivant les DF. La DF , X → Y, est déduite si un (ou plusieurs) de ces chemins atteint Y.**

Le graphe minimum des DF sert essentiellement à définir des relations normalisées.

DÉPENDANCES FONCTIONNELLES ET GRAPHE DES DÉPENDANCES FONCTIONNELLES

- Tout graphe (minimum ou pas) des DF peut être employé pour la recherche des identifiants des relations de la façon suivante:

➢ Chercher, sur le graphe des DF de la relation, tout ensemble minimum d'attributs, X, tel que tous les chemins partant de X et suivant les DF atteignent tous les autres attributs du graphe.

➢ Alors X est un identifiant de la relation

DÉCOMPOSITION D'UNE RELATION

- **Etant donné une relation non satisfaisante, en ce sens qu'elle implique des *répétitions* au niveau de sa population et qu'elle pose des *problèmes* lors des *insertions*/ *modifications*/ *suppressions* de tuples.**

- **Exemple: Livraison (NP, NF, Date, TélF, Qté)**

Exemples de problèmes « *Livraison* »: s'il **n'y a plus** de **livraison** pour un fournisseur son numéro de **téléphone** est **perdu**. S'il existe **N livraisons** pour un fournisseur, le numéro **TélF** (téléphone du fournisseur) est **répété N fois**.

- **Objectif: Trouver un ensemble de relations satisfaisantes et qui décrive les mêmes informations**

DÉCOMPOSITION D'UNE RELATION

❖ **Objectif:** Trouver un ensemble de relations satisfaisantes et qui décrive les mêmes informations

❖ La méthode consiste à décomposer la relation en deux (ou plusieurs) relations. Pour cela, il est nécessaire de disposer de deux opérations qui permettent:

➢ L'une, de découper une relation en sous relations *« Projection »*,

➢ Et l'autre, de recomposer la relation à partir de ses sous-relations *« Jointure »*.

DÉCOMPOSITION D'UNE RELATION

- On peut décomposer une relation en un ensemble de relations projetées, si on peut retrouver la relation initiale à partir des *relations projetées* en faisant leur jointure.

- Ainsi, toute requête, qu'elle soit posée à la relation initiale ou aux relations issues de la décomposition, donnera le même résultat.
- La relation initiale et les relations issues de la décomposition constituent deux bases de données équivalentes.

- Définition: 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 $R = R1 * R2$.

- **Exemple : La relation Personne (NP, NomP, VilleP)**

NP	NomP	VilleP
12	Ahmed	Alger
45	Amel	Oran
33	Ahmed	Setif

❖ **La décomposition D1 de la relation Personne en:**

➢ $R1 = \pi_{NP, NomP}$ Personne R1 (NP, NomP)

➢ $R2 = \pi_{NP, VilleP}$ Personne R2 (NP, VilleP)

Est une décomposition sans perte d'information, car R1 * R2 = Personne

On remarque cependant que cette décomposition est inutile, car la relation Personne est bonne.

❖ **La décomposition D2 de la relation Personne en:**

➢ $R3 = \pi_{NP, NomP}$ Personne R3 (NP, NomP)

➢ $R4 = \pi_{NomP, VilleP}$ Personne R4 (NomP, VilleP)

Est une décomposition avec perte d'information, car la Jointure sur l'attribut *'nom'* a multiplié les informations, il y'a plus de tuples, en effet on ne sait plus quels sont les bons et les faux tuples.

R3 * R4

NP	NomP	VilleP
12	Ahmed	Alger
12	Ahmed	Setif
45	Amel	Oran
33	Ahmed	Alger
33	Ahmed	Stif

DÉCOMPOSITION D'UNE RELATION

- Les décompositions sans perte d'informations assurent que toute requête posée à la base de données initiale (*avant décomposition*) et la requête équivalente posée sur la base de données issue de la décomposition donne le même résultat

DÉCOMPOSITION D'UNE RELATION

- **Théorème de Heath**: Toute relation R (X, Y, Z) est décomposable sans perte d'information en $R1=\pi_{X,Y}R$ et $R2=\pi_{X,Z}R$ s'il y a dans R une dépendance fonctionnelle de X vers Y (X→Y).

- **Principe de la démonstration**:

➢ R1*R2 contient au moins tous les tuples de R, puisque tout tuple (x,y,z) de R crée un tuple (x,y) dans R1 et un tuple (x,z) dans R2, qui sont concaténés par la jointure en (x, y, z).

➢ R1*R2 ne peut pas contenir de tuples en plus de ceux de R. Ceci est démontré par l'absurde. Soit (x,y,z) un tuple de R1*R2, qui n'appartient pas à R. (x,y,z) provient de deux tuples de R: (x,y,z') et (x,y',z). Etant donné que (x,y,z) n'appartient pas à R, on a: $z' \neq z$ et $y' \neq y$, ce qui est contraire à la DF: X→Y.

1ÈRE FORME NORMALE 1FN

- **Une relation est en 1FN si tout attribut est Atomique (non–décomposable)**
- **Une relation est en première forme normale (1FN) si chaque valeur de chaque attribut de chaque tuple est une valeur simple (tous les attributs sont simples et monovalués).**

❖ **Contre-Exemple:**

➢ **Etudiant (NE, NomE, PrenomE, Liste-Notes)**

Un attribut ne peut pas être un ensemble de valeurs.

❖ **Décomposition:**

➢ **Etudiant (NE, NomE, PrenomE)**

➢ **Note (NE, N Matière, NoteE)**

2ÈME FORME NORMALE 2FN

- **Une relation est en 2FN si :**

✓ **Elle est en 1FN**

✓ **Tout attribut n'appartenant pas à la clé, ne dépend pas d'une partie de la clé**

- **Une relation est en 2FN Si chaque attribut qui ne fait partie d'aucun identifiant dépend de tout identifiant entier (et non pas d'une partie de l'identifiant).**
- **C'est la phase d'identification des clés**
- **Cette étape évite certaines redondances.**
- **Tout attribut doit dépendre fonctionnellement de la totalité de la clé.**

2ÈME FORME NORMALE 2FN

❖ **Contre-Exemple:**

Une relation en 1FN qui n'est pas en 2FN

➢ **COMMANDE (Date, N_Client, N_Prod, Qte, PrixUHT)**

Elle n'est pas en 2FN, car la clé = (Date, N_Client, N_Prod)
et le PrixUHT ne dépend que de N_Prod.

❖ **Décomposition:**

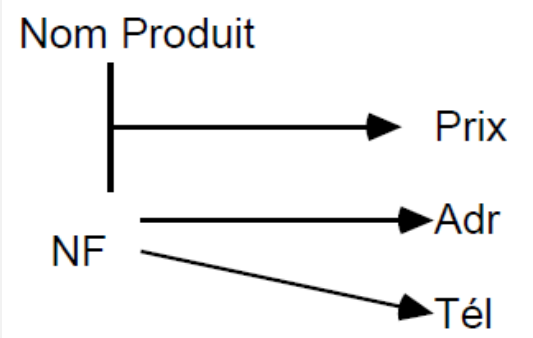
➢ **Cammande (Date, N_Client, N_Prod, Qte)**

➢ **Produit (N_Prod, PrixUHT)**

2ÈME FORME NORMALE 2FN

❖ **Soit une relation qui est en 1FN, mais qui n'est pas en 2FN:**

- **Fournisseur1 (NF, NomProd, Adrs, Tel, Prix)**
- **Graphe minimum des DF de Fournisseur1 :**



❖ **Une telle relation pose des problèmes :**

➢ **Redondances: s'il existe 100 produits pour un fournisseur**

on va répéter 100 fois le *NomProd*, l'*Adrs*, *Tel*

➢ **Problème m-à-j pour les insertions: quand on veut rajouter un produit, il faut rentrer à nouveau l'*Adrs* et *Tel* du Fournisseur.**

➢ **Problème pour les suppressions: si on supprime (momentanément) la liste des produits d'un fournisseur, alors on supprime aussi le fournisseur.**

➢ **Problème de mise à jour des tuples: si un fournisseur change d'adresse ou de téléphone, il faut faire cette mise à jour sur tous les 100 tuples !**

2ÈME FORME NORMALE 2FN

- Ces problèmes sont dus au fait que la relation n'est pas en 2FN.
- On décompose Fournisseur1 en deux relations de la façon suivante :

➢ Pour chaque source de DF, on crée une relation ayant pour attributs la source et tous les attributs en dépendance fonctionnelle directe de cette source.

➢ S'assurer qu'une (au moins) des deux sources est entièrement contenue dans les attributs communs aux deux relations créées (*théorème de Heath*).

❖ On obtient les relations suivantes qui sont en 2FN :

- Fournisseur (NF, Adr, Tel)
- Catalogue (NF, NomProduit, Prix)
- Cette décomposition est :
- Sans perte d'information (NF est l'identifiant de la relation Fournisseur)
- Sans perte de dépendance fonctionnelle (les DF sont soit dans l'une, soit dans l'autre des deux relations décomposées).

3ÈME FORME NORMALE 3FN

- **Une relation est en 3FN si :**

✓ **Elle est en 2FN.**

✓ **Tout attribut n'appartenant pas à la clé, ne dépend pas d'un attribut non clé.**

- **Ceci correspond à la non transitivité des DF, ce qui évite les. redondances.**
- **En 3FN une relation préserve les DF sans perte d'informations.**
- **Une relation est en 3FN si elle est en 1FN et si chaque attribut qui ne fait partie d'aucun identifiant dépend uniquement des identifiants entiers.**
- ***NB : On peut toujours décomposer en 3FN sans perte ni d'information, ni de DF; ce qui n'est pas vrai des formes normales plus poussées. D'où l'intérêt de cette 3FN.***

3ÈME FORME NORMALE 3FN

❖ Contre-Exemple:

Une relation en 2FN qui n'est pas en 3FN

➢ **VOITURE (Matricule_V, Marque_V, Modèle_V, Puissance_V)**

- **On vérifie qu'elle est en 2FN,**
- **Elle n'est pas en 3FN car la clé = (Matricule_V) et la Puissance_V dépend de (Marque_V et Modèle_V).**

❖ Décomposition:

➢ **Voiture (Matricule_V, Marque_V, Modèle_V)**

➢ **Modèle (Marque_V, Modèle_V, Puissance_V)**

3ÈME FORME NORMALE 3FN

- Soit une relation qui est en 2FN : Fournisseur2 (NF, Pays, Ville)
- Ensemble des DF : { NF → Ville , Ville → Pays }
- On suppose qu'on n'a dans la base de données que des grandes villes de différents noms. La DF : NF → Pays est une DF déduite.
- Graphe minimum des DF de Fournisseur2 :

NF → Ville → Pays

- Dans la relation Fournisseur2, il y a redondance : le pays d'une ville est répété, ce qui cause des problèmes de mise à jour.
- On décompose donc en :

➢ Fourn (NF, Ville) Géo (Ville, Pays)

- Cette décomposition est sans perte d'information (Ville est identifiant pour Géo), et sans perte de DF (les DF non déduites sont soit dans Fourn, soit dans Géo).

3ÈME FORME NORMALE DE BOYCE–CODD FNBC

- **Une relation est en FNBC si :**

✓ **Elle est en 1FN.**

✓ **SSI les seuls DF élémentaires sont celles dans lesquelles une clé détermine un attribut.**

- **FNBC signifie qu'on ne peut pas avoir un attribut (ou groupe d'attributs) déterminant un autre attribut et diffèrent de la clé.**
- **Ceci évite les redondances dans l'extension de la relation: mêmes valeurs pour certains attributs de n-uplets différents.**

- **FNBC est plus fin que 3FN : FNBC → 3FN**

3ÈME FORME NORMALE DE BOYCE–CODD FNBC

❖ **Contre-Exemple:**

Une relation en 3FN qui n'est pas en FNBC

➢ **CODEPOSTAL (Ville, Rue, Code)**

- **On vérifie qu'elle est en 3FN,**
- **Elle n'est pas en FNBC car la clé = (Ville, Rue) et Ville dépend de (Code).**

Code → Ville

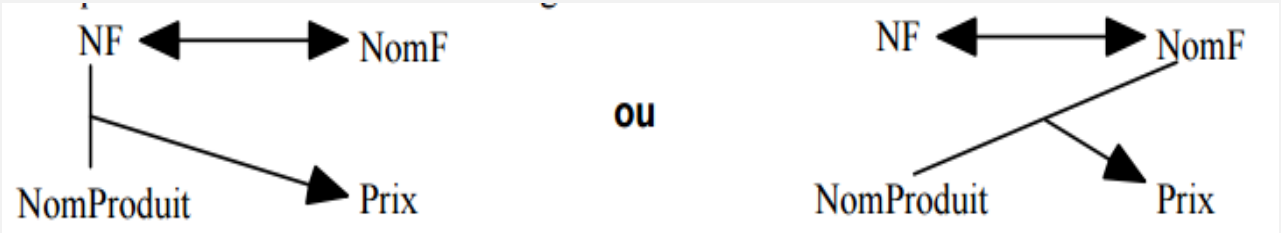
3ÈME FORME NORMALE DE BOYCE–CODD FNBC

Relations à plusieurs identifiants

- **Soit une relation qui est en 3FN (on suppose qu'il n'y a pas mêmes noms de fournisseurs) : Catalogue3 (NF, NomF, NomProduit, Prix)**

Identifiants : (NF + NomProduit) , (NomF + NomProduit)

- **Cette relation possède deux graphes minimum des DF. Ils sont équivalents et qui conduisent à des décompositions équivalentes.**



- **Dans la relation Catalogue3, il y a redondance entre NF et NomF, ce qui génère des problèmes lors des mises à jour. Par exemple si un fournisseur change de nom, il faut mettre à jour son nom dans tous les tuples correspondants à ses produits.**
- **On décompose donc Catalogue3 en deux relations de forme normale de Boyce Codd:**

➢ **Fournisseur (NF , NomF) qui a deux identifiants : (NF) et (NomF)**

➢ **Catalogue (NF, NomProduit, Prix) qui a un identifiant composé:(NF + NomProduit)**

3ÈME FORME NORMALE DE BOYCE–CODD FNBC

Relations à plusieurs identifiants

- **Définition :**

« Une relation est en forme normale de Boyce Codd (FNBC) si elle est en première forme normale (1FN) et si toute source complète de DF est un identifiant entier »

- La relation Catalogue3 n'est pas en forme normale de Boyce Codd parce que l'attribut NF est source complète de DF (NF → NomF) et n'est pas un identifiant entier. Il en est de même pour NomF.

- On remarque qu'une relation en forme normale de Boyce Codd, est en troisième forme normale.

3ÈME FORME NORMALE DE BOYCE–CODD FNBC

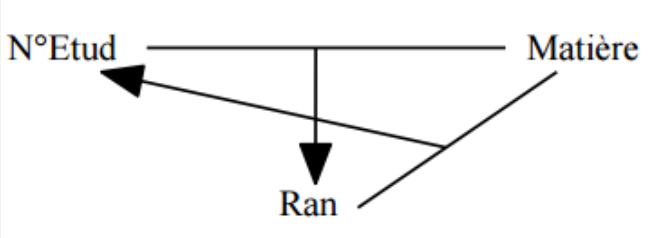
Relations à plusieurs identifiants

- **Autre exemple La relation: Place (N°Etud, Matière, Rang)**

Représente le rang obtenu par chaque étudiant pour chaque matière. Il y a deux identifiants :

(N°Etud + Matière) , (Matière + Rang).

Graphe minimum des DF de Place :



❖ **La relation Place est-elle en 3FN ?**

➢ **Oui, car il n'y a pas d'attribut qui ne fasse pas partie d'un identifiant.**

➢ **Elle est aussi en Boyce Codd. On ne décompose donc pas**

3ÈME FORME NORMALE DE BOYCE–CODD FNBC

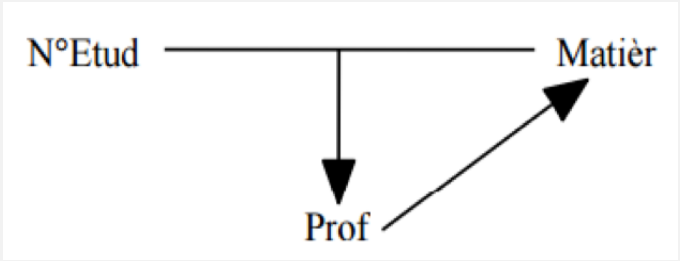
Relations à plusieurs identifiants

- **Autre exemple La relation: Enseignement (N°Etud, Matière, Prof)**

❑ **On suppose que chaque professeur enseigne une seule matière.**

❑ **Chaque étudiant suit les cours d'un seul professeur.**

- **Graphe minimum des DF d'Enseignement :**



❖ **La relation Enseignement a 2 identifiants:**

(N°Etud + Matière) et (N°Etud + Prof)

❖ **Elle est en 3FN puisque tout attribut fait partie d'un identifiant.**

❖ **Elle n'est pas en Boyce Codd puisque l'attribut Prof est source de DF et ne constitue pas un *identifiant entier*.**

- **Autre exemple La relation: Enseignement (N°Etud, Matière, Prof)**
- Cette relation présente des inconvénients *(redondance, problème de mise à jour)* dus au fait que la **DF Prof → Matière** n'est pas traduite par la relation.
- Notamment, si un professeur *change* de spécialité il faut penser à faire cette mise à jour dans tous les tuples où apparaît ce professeur.
- Cependant, on ne peut pas décomposer la relation enseignement *sans perdre* de **DF,** par exemple la **décomposition**: (**Prof**, Matière) (**N°Etud**, Prof)

Perd la **DF (N°Etud, Matière) → Prof.**

- Cette décomposition permet d'insérer le fait qu'un même étudiant suit deux cours pourtant sur la même matière avec deux professeurs différents ce qu'***interdisait*** la relation Enseignement.
- Il n'y a pas de solution idéale dans ce cas. Pratiquement, on ***conserve*** la relation initiale et on ***rajoute*** une *condition d'intégrité* spécifiant le fait qu'*un professeur ne peut enseigner qu'une seule matière.*

MÉTHODES DE DÉCOMPOSITION

- **Pour décomposer une relation non normalisée en un ensemble de relations normalisées, plusieurs méthodes conduisant à des algorithmes ont été proposées, mais aucune d'entre elles, appliquée de façon purement automatique, n'est totalement satisfaisante.**

- **En effet, elles peuvent proposer en résultat des relations normalisées, mais qui ne sont pas sémantiquement significatives.**

I- DÉCOMPOSITION EN BOYCE CODD (FNBC) SANS PERTE D'INFORMATION

❖ **Il est possible de décomposer toute relation en forme normale de FNBC *sans* perte d'information en appliquant récursivement le théorème de Heath :** *Tant que dans une des relations obtenues par décomposition, R(X,Y,Z), il existe une* **DF X→Z**, *on décompose* **R en $\pi_{X,Y}(R)$ et $\pi_{X,Z}(R)$.**

❖ **Cette méthode présente plusieurs inconvénients :**

- **Elle peut conduire à trop décomposer, notamment en décomposant une relation qui est déjà en *Boyce Codd*.**

Une décomposition n'est utile que si la relation initiale *n'est pas assez normalisée* et que les relations issues de la décomposition sont *plus normalisées*. Un schéma trop décomposé complique l'emploi de la base de données, car les requêtes seront plus complexes (*nombreuses jointures*).

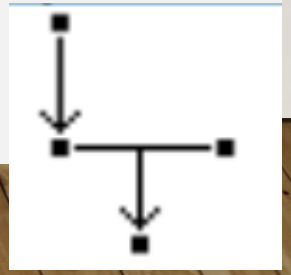
- **Elle peut perdre des DFs, voir par exemple le cas de la relation: *Enseignement*. Les dépendances perdues doivent alors être ajoutées au *schéma décomposé* sous la forme de contraintes d'intégrité.**

2- DÉCOMPOSITION EN TROISIÈME FORME NORMALE (3FN) SANS PERTE D'INFORMATION NI DE DÉPENDANCE FONCTIONNELLE

- **Il est possible de décomposer toute relation R en 3FN sans *perte d'information* ni de DF, en suivant la méthode ci-dessous :**

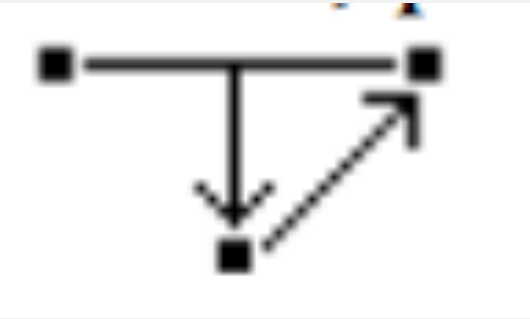
❖ **Créer pour chaque source de DF une relation comprenant comme attributs la *source et toutes les cibles de cette source*;**

❖ **Si aucun des *identifiants* de R n'est présent dans au moins une des relations créées, ajouter une relation supplémentaire, constituée des attributs composant un des identifiants de R. Cette relation supplémentaire supprimera les tuples excédentaires lors de la jointure des relations issues de la décomposition pour reconstituer la relation initiale.**



- **Cette méthode de décomposition a cependant l'*inconvénient* de générer parfois des décompositions redondantes. Par exemple dans le cas de la relation *Enseignement (N°Etud, Matière, Prof)*.**

- **Il faut donc ensuite supprimer du résultat les relations qui sont incluses dans (c'est-à-dire égales à une projection d) une autre relation.**



EXERCICE

- **Soit la relation: Cours (Annee, Etudiant, Groupe, Professeur, Matiere)**

➢ **1) Exprimer chaque contrainte en dépendance fonctionnelle:**

a) Chaque année, un étudiant est placé dans un groupe

Rep: **{Annee, Etudiant} → {Groupe}**

b) Pour un groupe et matière donnée, un professeur unique.

Rep: **{Groupe, Matiere} → {Professeur}**

c) Les professeurs n'enseignent qu' une seule matière

Rep: **{Professeur} → {Matiere}**

➢ **2) Donner toutes les clés minimales possibles:**

a) {Annee, Etudiant}→{Groupe} ET {Groupe, Matiere}→{Professeur}

Donc {Annee, Etudiant, Matiere}→{Professeur} (Clé 1)

b) On a Aussi {Professeur} → {Matiere}

Donc {Annee, Etudiant, Professeur}→{Matiere} (Clé 2)

DF = { {Annee, Etudiant} → {Groupe} ; {Groupe, Matiere} → {Professeur} ; {Professeur} → {Matiere} }

- 3) Déterminer la forme normale de la relation cours

- Non 2FN : La clef {Etudiant, Annee, Professeur} contient {Professeur} qui détermine {Matiere}.
- Non 2FN : La clef {Etudiant, Annee, Professeur} contient {Etudiant, Annee} qui détermine {Groupe}.

- 4) Normaliser la relation cours pour obtenir une décomposition, sans perte d'information et sans perte de dépendance fonctionnelle, en un ensemble de relations en 3FN. Et préciser si elle est en FNBC.

- 1. {Etudiant, Annee} → {Groupe} donne

  Inscrits (Etudiant, Annee, Groupe) 3NF et FNBC.

  2. {Groupe, Matiere} → {Professeur} donne

  Repartition (Groupe, Matiere, Professeur) 3NF et non FNBC.

BIBLIOGRAPHIE

- Concepts et langages des Bases de Données Relationnelles, SUPPORT DE COURS SGBD1. IUT de Nice – Département INFORMATIQUE

- Chapitre 5 NORMALISATION D'UNE RELATION, Cours de Bases de données avancées. Université de Lausanne, Ecole des Hautes Etudes Commerciales (HEC).