

Cours 03

Conception et optimisation de schéma relationnel

La normalisation

- La conception d'un M E-A représente la vision de la réalité de l'analyste. Le formalisme obtenu, établi avec une méthode, ne garantit pas **justesse** et **optimisation**.
- La **justesse** dépend de la **compétence** et de l'**expérience** de l'analyste.
- L'**optimisation** est obtenue par les **mécanismes** de la normalisation.

Normalisation

- C'est vérifier que la structure devant recevoir des données est organisée pour éviter:
 - des **redondances** d'informations et
 - des **anomalies de conception** (sources inévitables d'incohérence à court ou long terme).

Normalisation

Définition:

Le processus de restructurer le modèle de données logiques pour :

- Eliminer les redondances,
- Organiser les données efficacement,
- Réduire le potentiel d'anomalies pendant les opérations sur les données.

La normalisation

■ Les besoins de normalisation :

- 3 problèmes de cohérence lors de mises à jours de la BD:
 - **anomalie d'insertion** (si on rajoute une commande il faut que l'article existe)
 - **anomalie de suppression** (si on enlève un fournisseur il faut enlever toutes les commandes de ce fournisseur)
 - **anomalie de modification** (si on modifie le numéro d'un article il faut modifier toutes les commandes avec ce numéro d'article)

La normalisation

- La théorie de la normalisation repose sur l'analyse de dépendances entre attributs qui sont à l'origine de phénomènes de redondance.
- La normalisation consiste en des méthodes de décomposition des relations.
- Dans certaines situations, pour des raisons d'efficacité, on *dénormalise*.

Normalisation

- Les **classifications** formelles utilisées pour décrire le **niveau** de normalisation d'une base de données relationnelle sont appelées les **formes normales (FN)**
- Il existe **04** formes normales, sont les plus pratiques et sont à connaître.

Normalisation

- Chaque nouvelle forme normale marque une étape supplémentaire de progression vers des relations présentant de **moins en moins de redondance**
- **Le but..**
est d'obtenir une représentation des données présentant un **minimum de redondance** à l'intérieur de chaque relation et un **maximum d'indépendance** entre les différentes relations

Formes normales

Première forme normale

Deuxième forme normale

Troisième forme normale

Forme normale de Boyce-Codd

Edgar Codd est à l'origine de la définition des formes normales ...1FN, 2FN, 3FN, BCFN.

la normalisation

- La théorie de la normalisation est basée sur les "dépendances fonctionnelles" (DF).
- C.A.D. Dépendance entre les informations
Par exemple, le salaire dépend de la qualification

-
- Les dépendances fonctionnelles traduisent des **contraintes** sur les données
 - Par exemple, on décide que deux individus différents peuvent avoir même nom et prénom mais jamais le même numéro de sécurité sociale.
 - Ces contraintes sont représentatives d'une perception de la réalité et imposent des limites à la base

Dépendance fonctionnelle

Rappel:

- Définition : dépendance fonctionnelle

On dit qu'un attribut **B** dépend fonctionnellement d'un attribut **A** si, étant donné **une valeur de A**, il lui correspond **une unique valeur de B**.

- Notation : $A \twoheadrightarrow B$

- Exemple :

La dépendance fonctionnelle $SS \rightarrow NOM$ signifie qu'à **un numéro** est associé **un nom seulement**.

- *Remarquons qu'une dépendance fonctionnelle n'est généralement pas symétrique, c'est-à-dire que $SS \rightarrow NOM$ n'interdit pas que deux personnes distinctes (correspondant à deux SS différents) portent le même nom.*

Propriétés des dépendances fonctionnelles

- Les dépendances fonctionnelles possèdent trois propriétés fondamentales qu'on appelle couramment les axiomes d'Armstrong car mises en évidence par W. Armstrong. En supposant que \mathbf{X} , \mathbf{Y} , \mathbf{Z} , sont trois ensembles d'attributs, il est utile de faire les remarques suivantes avant d'énoncer les propriétés :
- L'écriture \mathbf{X}, \mathbf{Z} est une écriture simplifiée de l'union ensembliste des deux ensembles d'attributs \mathbf{X} et \mathbf{Z} . Ceci veut dire que l'écriture \mathbf{X}, \mathbf{Z} devra être interprétée comme $\mathbf{X} \cup \mathbf{Z}$.
- Il faut souligner aussi que l'union ensembliste est un opérateur **associatif** et **commutatif** :
- $((\mathbf{X} \cup \mathbf{Y}) \cup \mathbf{Z}) = (\mathbf{X} \cup (\mathbf{Y} \cup \mathbf{Z}))$
- $\mathbf{X} \cup \mathbf{Z} = \mathbf{Z} \cup \mathbf{X}$

Propriétés des dépendances fonctionnelles (5)

- **Réflexivité**

- Si $\mathbf{X} \subseteq \mathbf{Y} \Rightarrow \mathbf{Y} \rightarrow \mathbf{X}$
- **Ex** : $A \subseteq A,B \Rightarrow A,B \rightarrow A$ et $A,B \rightarrow B$

- **Transitivité**

- Si $\mathbf{X} \rightarrow \mathbf{Y}$ et $\mathbf{Y} \rightarrow \mathbf{Z} \Rightarrow \mathbf{X} \rightarrow \mathbf{Z}$
- **Ex** : $A,B \rightarrow C$ et $C \rightarrow D \Rightarrow A,B \rightarrow D$

- **Augmentation**

- Si $\mathbf{X} \rightarrow \mathbf{Y} \Rightarrow \forall \mathbf{Z} \quad \mathbf{X}, \mathbf{Z} \rightarrow \mathbf{Y}, \mathbf{Z}$
- **Ex** : $A,B \rightarrow C \Rightarrow A,B,D \rightarrow C,D$

A partir de ces trois axiomes de base, on peut déduire facilement d'autres propriétés très utiles en pratiques. Les plus remarquables sont :

Propriétés des dépendances fonctionnelles (6)

- **Union des parties droites**
 - Si $X \rightarrow Y$ et $X \rightarrow Z \Rightarrow X \rightarrow Y, Z$
 - **Ex** : $A \rightarrow C$ et $A \rightarrow D \Rightarrow A \rightarrow C, D$
- **Décomposition**
 - Si $X \rightarrow Y, Z \Rightarrow X \rightarrow Y$ et $X \rightarrow Z$
 - **Ex** : $A \rightarrow C, D \Rightarrow A \rightarrow C$ et $A \rightarrow D$
- **Pseudo Transitivité**
 - Si $X \rightarrow Y$ et $Y, W \rightarrow Z \Rightarrow X, W \rightarrow Z$
 - **Ex** : $A \rightarrow C$ et $B, C \rightarrow D \Rightarrow A, B \rightarrow D$
- **Distributivité par rapport à l'union**
 - Si $X \rightarrow U \Rightarrow \forall ui \subseteq U X \rightarrow ui$ (c'est une généralisation de la décomposition)
 - **Ex** : $A \rightarrow C, D, E, F \Rightarrow A \rightarrow C$ et $A \rightarrow D$ et $A \rightarrow E$ et $A \rightarrow F$
 $A, B \rightarrow C, D, E \Rightarrow A, B \rightarrow C$ et $A, B \rightarrow D$ et $A, B \rightarrow E$

Propriétés des dépendances fonctionnelles (7)

Dépendance fonctionnelle élémentaire

- On appelle dépendance fonctionnelle **élémentaire** une **dépendance de la forme : $X \rightarrow A$** , ou **A** est un attribut **unique non inclus dans X** ($A \not\subset X$), **tel que $\forall X' \subset X$, il n'existe pas de dépendance fonctionnelle $X' \rightarrow A$.**
- En d'autres termes, une dépendance fonctionnelle **est élémentaire si elle n'est pas obtenue par augmentation de sa partie gauche.**

Propriétés des dépendances fonctionnelles (8)

- **exemple :**
- Si on considère les deux D.F. :
- **(NUMERO, NOM) → ADRESSE (1)**
- **NUMERO → ADRESSE (2)**
- la D.F. **(1)** est une **D.F. non élémentaire** car **NUMERO \subset (NUMERO,NOM)** et il existe la D.F. **NUMERO → ADRESSE.**
- Par contre la D.F. **(2)** est une **D.F. élémentaire** car elle **vérifie les propriétés d'une D.F. élémentaire**

Propriétés des dépendances fonctionnelles (9)

- **Dépendance fonctionnelle transitive**
- On appelle dépendance fonctionnelle **transitive** **une dépendance de la forme : $X \rightarrow A$, ou A est un attribut unique non inclus dans X ($A \notin X$), tel que :**
 - $\exists Y \notin X, X \rightarrow Y$ et $Y \rightarrow A$ et $Y \not\rightarrow X$
- Dans ce cas, on dit aussi que **A est transitivement dépendant de X . Dans le cas où A n'est pas**
- **transitivement dépendant de X , on dit que A est directement dépendant de X**

Propriétés des dépendances fonctionnelles (10)

- **exemple :**
- Si on considère les D.F. :
- **1 - $A \rightarrow B$ 2- $B \rightarrow C$ 3- $A \rightarrow C$ 4- $D \rightarrow B$**
5- $D \rightarrow C$ 6 $B \rightarrow D$
- La D.F. **$A \rightarrow C$ est transitive car $\exists B \notin A$ tel que :**
- **$A \rightarrow B$ et $B \rightarrow C$ et $B \not\rightarrow A$**
- Par contre la D.F. **$D \rightarrow C$ n'est pas transitive**
puisque $\exists B \notin D$ tel que :
- **$D \rightarrow B$ et $B \rightarrow C$ mais $B \not\rightarrow D$**

Propriétés des dépendances fonctionnelles (12)

- **Dépendance fonctionnelle directe**

On appelle dépendance fonctionnelle **directe** une **dépendance de la forme $X \rightarrow A$** , où **A est un attribut unique non inclus dans X ($A \not\subseteq X$)**, tel **que :**

- **$\forall Y \not\subseteq X$, si $X \rightarrow Y$ et $Y \rightarrow A$ alors la d.f. $Y \rightarrow X$ existe aussi**

Propriétés des dépendances fonctionnelles (13)

- **exemple :**
- Si on considère les D.F. :
- **1- $A \rightarrow B$ 2- $B \rightarrow C$ 3- $A \rightarrow C$ 4- $D \rightarrow B$ 5- $D \rightarrow C$
6- $B \rightarrow D$**
- La D.F. **$A \rightarrow C$ n'est pas directe car pour $B \notin A$ on a : $A \rightarrow B$ et $B \rightarrow C \neq \Rightarrow B \rightarrow A$**
- La D.F. **$D \rightarrow C$ est directe puisque pour $B \notin D$ on a : $D \rightarrow B$ et $B \rightarrow C \Rightarrow B \rightarrow D$ existe**

Propriétés des dépendances fonctionnelles

(14)

- La D.F. $A \rightarrow B$ est directe puisque il n'existe pas d'attribut $Y \notin A$ tel qu'on a en même temps les d.f. $A \rightarrow Y$ et $Y \rightarrow B$ pour vérifier s'il y a ou non contradiction avec la définition. Dans ces cas là , la d.f. est forcément directe.
- Dans le cas où A n'est pas directement dépendant de X (i.e. la d.f. $Y \rightarrow X$ n'existe pas), on dit que A est transitivement dépendant de X .

Propriétés des dépendances fonctionnelles (15)

Dépendance fonctionnelle totale, pleine ou complète

- On appelle dépendance fonctionnelle **totale , pleine ou complète** une dépendance fonctionnelle de la forme $X \rightarrow A$, ou A est un attribut unique non inclus dans X ($A \not\subset X$), telle que $\forall X' \subset X$, il n'existe pas de dépendance fonctionnelle $X' \rightarrow A$. En d'autres termes, la dépendance $X \rightarrow A$ est élémentaire.
- Dans ce cas on dit aussi que **A est pleinement ou totalement ou complètement dépendant de X .**
- Dans le cas contraire, on dit aussi que **A est partiellement dépendant de X (i.e. A dépend d'une partie de X et donc forcément de X aussi par augmentation)**

Propriétés des dépendances fonctionnelles (16)

- **exemple :**
- Si on considère les D.F. :
1- **$A, B \rightarrow C$** 2- **$B \rightarrow C$** 3- **$B, C \rightarrow D$**
- dans la d.f. 1 **C n'est pas totalement dépendant de A, B car on a $B \subset A, B$ tel que $B \rightarrow C$. On peut dire que C est partiellement dépendant de A, B .**
- dans la d.f. 3- **D est totalement dépendant de B, C car il n'existe pas de $X' \subset B, C$ et tel que $X' \rightarrow D$.**

Propriétés des dépendances fonctionnelles (17)

Dépendance fonctionnelle triviale

On appelle dépendance fonctionnelle triviale une dépendance fonctionnelle de la forme $X \rightarrow Y$, tel que $Y \subseteq X$. En d'autres termes, une dépendance fonctionnelle est triviale si elle est obtenue grâce à la propriété de réflexivité.

• **exemple : Si on considère les D.F. :**

1- **$A, B \rightarrow B$** 2- **$B, C \rightarrow C$** 3- **$A, C \rightarrow D$**

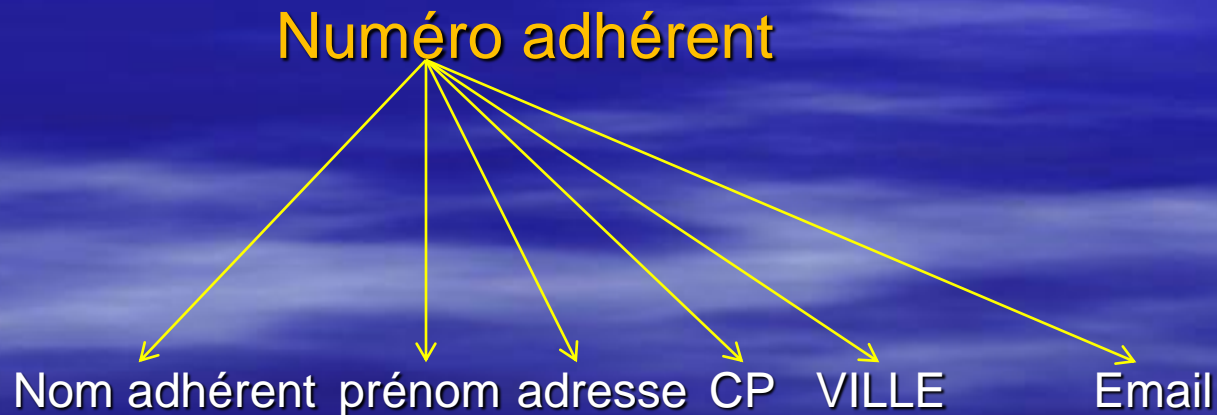
- la d.f. 1 est triviale car on a **$B \subset A, B$**
- la d.f. 2 est triviale car on a **$C \subset B, C$**
- la d.f. 3 n'est pas triviale car on a **$D \not\subset A, C$**

Propriétés des dépendances fonctionnelles (18)

les notions de dépendance fonctionnelle élémentaire, de dépendance fonctionnelle totale, de dépendance fonctionnelle directe vont intervenir dans la définition des formes normales d'une relation et principalement la deuxième et la troisième formes normales.

Les dépendances fonctionnelles

- **Le Graphe des dépendances fonctionnelles**
- Le graphe des dépendances est une étape intéressante car il épure le dictionnaire en ne retenant que les données non déduites et élémentaires et permet une représentation spatiale de ce que sera le futur MCD.



Les dépendances fonctionnelles

- **Méthodologie d'élaboration des dépendances fonctionnelles:**
- L'élaboration des dépendances fonctionnelles est réalisée à l'aide du dictionnaire des données. La démarche consiste à chercher :
 - Les dépendances fonctionnelles formées par deux rubriques, élémentaires et directe.
 - Les dépendances fonctionnelles composées.

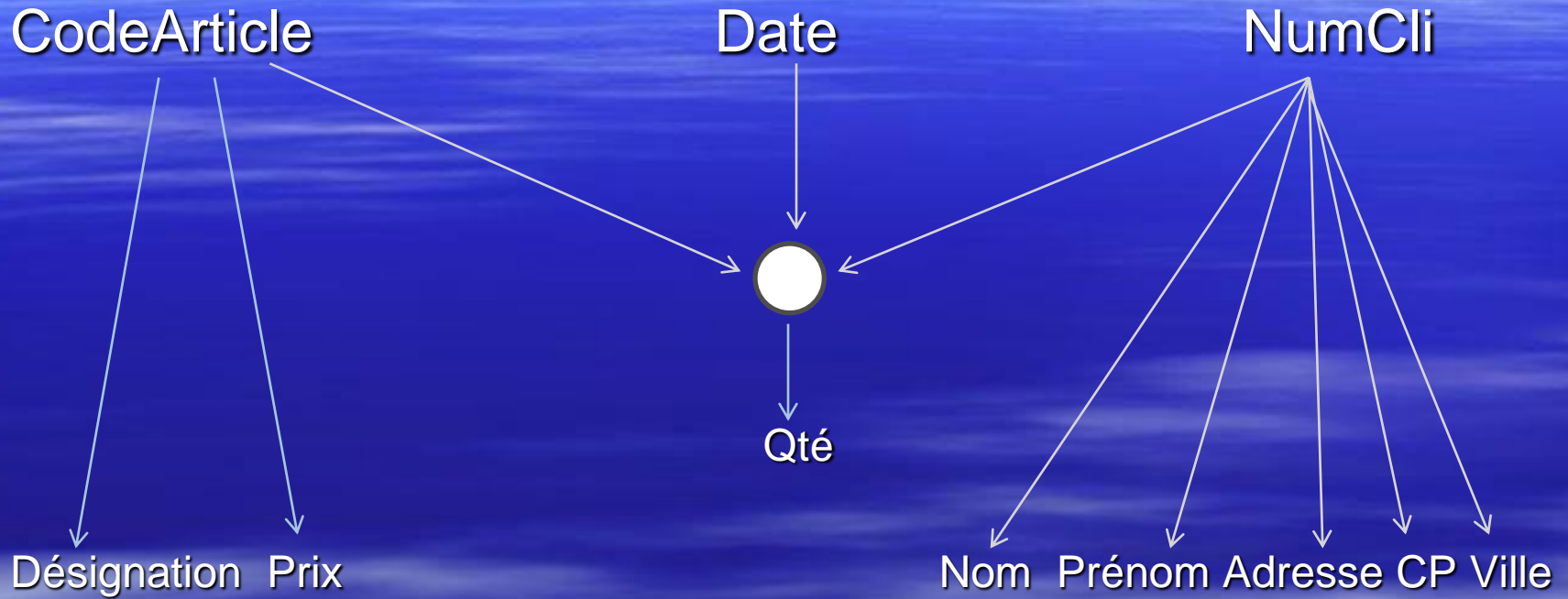
Dépendances fonctionnelles: – cas pratique–

Nom de la donnée	Format	Longueur	Type		Règle de calcul	Règle de gestion	Document
			Elémentaire	Calculée			
NumCli	Numérique		x				Facture
Nom:	Alphabétique	30	x				Facture
Prénom:	Alphabétique	30	x				Facture
Adresse:	Alphanumérique	60	x				Facture
Code postal	Numérique		x				Facture
Ville	Alphabétique	20	x				Facture
Téléphone:	Alphanumérique	14	x				Facture
CodeArticle	Alphanumérique	15	x				Facture
Désignation	Alphabétique	50	x				Facture
Quatité	Numérique		x				Facture
Prix unitaire	Numérique		x				Facture
Date	date		x				Facture
Total ligne	Numérique			x	Prix* Qté		Facture
Total facture	Numérique			x	Somme des Total Ligne		Facture

Dépendances fonctionnelles: – cas pratique–

- NumCli \longrightarrow (Nom, Prénom, Adresse, Code Postal, Ville)
- CodeArticle \longrightarrow (Désignation, Prix unitaire)
- (NumCli, CodeArticle, Date) \longrightarrow Quantité

Dépendances fonctionnelles: – cas pratique–



Normalisation : 1FN

1FN (rubrique élémentaire) :

- Un MR est dit en première forme normale, si toutes les entités sont composées d'attributs *élémentaires ou atomiques* (hors clefs concaténées).

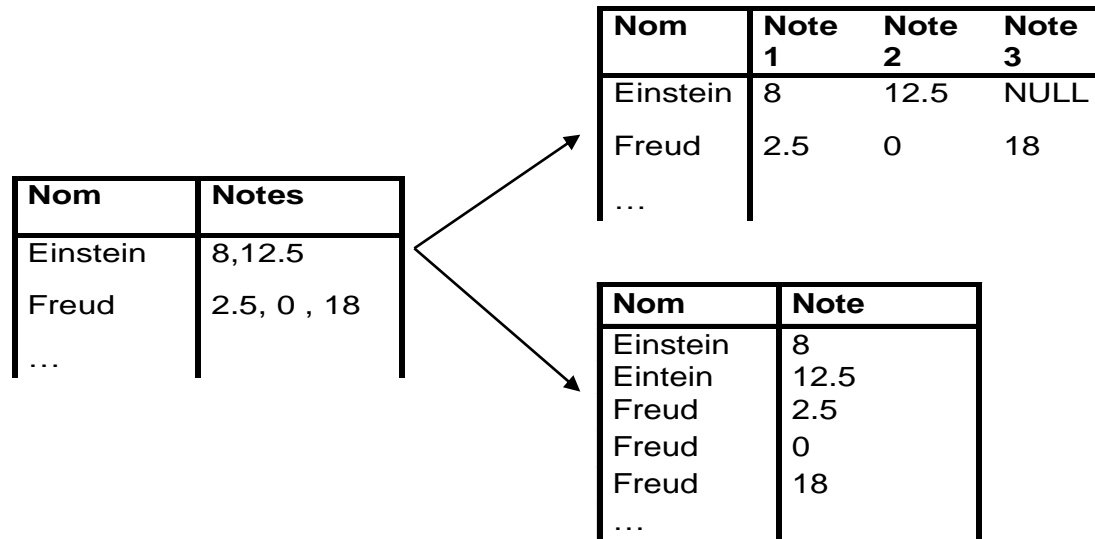
<u>IP</u>	Role
192.168.0.1	client
192.168.0.2	serveur
192.168.0.4	client et serveur

Conséquences :

- un attribut représente **une** donnée élémentaire du monde réel ;
- Un attribut ne peut désigner, ni une donnée **composée** d'entités de *natures différentes*, ni une **liste** de données de *même nature*.

Normalisation : 1FN

- Si on veut avoir les notes d'un étudiant :



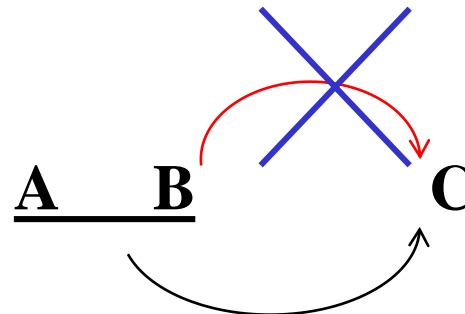
- N'est pas en 1FN. Pour y remédier, on peut soit créer un **attribut par note** si le *nombre maximal de notes est connu* et si le *nombre moyen de notes par personne est proche de ce maximum*, Soit fabriquer autant de **tuples** que de couples (nom, note)

Normalisation : 2FN

2FN (dépendance fonctionnelle élémentaire) :

- *Un MR est dit en deuxième forme normale, si:*
 - *il remplit les conditions de la 1FN et si,*
 - *tout attribut n'appartenant pas à une clé ne dépend pas d'une partie de cette clé.*

A,B -> C
B->C



Normalisation : 2FN

- Considérons la relation PLAGES de schéma suivant : PLAGES (NOMP, REGION, TYPE, POLLUTION)
- Le type est dépendant de la région.
Region → Type

où la clé est (NOMP, REGION).

NOMP, REGION → NOMP

NOMP, REGION → REGION

NOMP, REGION → Type

NOMP, REGION → Region

Est-ce que la relation PLAGES est en 2 FN ?

Non parce qu'il existe

La deuxième forme normale nous impose de distinguer deux relations R1 et R2 de schémas respectifs :

- R1 (NOMP, REGION, POLLUTION);

- R2 (REGION, TYPE).

Normalisation : 3FN

Définition : troisième forme normale:

- Une relation est en troisième forme normale si :
 - elle est en **deuxième forme normale** ;
 - **tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non clé.**
 - cad toute DF dont la partie droite n'est pas une clé doit avoir une partie gauche qui est une clé :



B, C OK



A, B, C

Normalisation : 3FN

- Exemple :

Considérons une table "Departements" dont les attributs sont

ID departement,

Nom departement,

ID manager,

Date embauche manager

et supposons que chaque manager peut gérer un ou plusieurs départements.

ID departement est une clé candidate.

Bien que *Date embauche manager* est fonctionnellement dépendante de $\{ID\ departement\}$, elle est aussi fonctionnellement dépendante de l'attribut *ID manager*.

ID departement → *Date embauche manager*

ID manager → *Date embauche manager* **X**

Ceci signifie que la table n'est pas 3FN.

Departements (ID department, Nom departement, ID manager

Information Manager (ID manager, date embauche manager)

Normalisation : 3FN

Autre exemple:

- Considérons maintenant la relation PLAGÉ de schéma
PLAGE (NP, REGION, TYPE, POLLUTION)
où la clé est NP.

Supposons maintenant comme dans l'exemple précédent que le **type** est **dépendant de la région**.

La troisième forme normale nous impose de distinguer deux relations R1 et R2 de schémas respectifs :

R1 (NP, REGION, POLLUTION) ;
R2 (REGION, TYPE).

Normalisation : 3FN

Autre Exemple :

Enseignant (Nom, Bureau, Batiment, Discipline, telephone)

Avec des contraintes d'intégrité : Un bâtiment héberge des enseignants d'une même discipline; un bureau donné possède un numéro de d'appel unique.

Avec les DF: Batiment → Discipline

Batiment, Bureau → Telephone

Telephone → Bureau, Batiment

Note: la deuxième contrainte définit une relation 1-1, donc une DF reflexive.

Enseignant (Nom, Bureau, Batiment, Discipline, telephone)

- La dépendance Batiment->Discipline ne concerne pas la clé. La relation Enseignant n'est pas 3FN.
- On sort donc l'attribut *Discipline* de la relation Enseignant (dépendant d'un attribut non-clé) et on crée une nouvelle relation Affectation:

Affectation(Batiment, Discipline)

- Les dépendances
Batiment, Bureau->Telephone
Telephone->Bureau, Batiment
ne concernent pas la clé. On crée la nouvelle relation:
Annuaire(Bureau, Batiment, telephone)
- La relation Enseignant d'origine peut être renommée Sièges, avec les attributs restant:
Sièges(Nom, Bureau, Batiment)

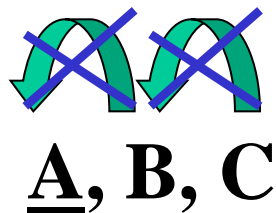
Normalisation : Boyce-Codd

- La FNBC était proposée comme une forme plus simple de 3FN, mais elle est finalement plus stricte que la 3FN.

Normalisation : BCFN

Définition : Boyce-Codd forme normale:

- Une relation est en BCFN si :
 - elle est en 3^{eme} forme normale ;
 - aucun attribut ne dépend d'un attribut non clé.
- Toute relation qui est en BCFN est forcément en 3 FN
- C.a.d. toute DF a pour partie gauche (origine de la flèche) un clé candidate ou primaire entière.
- Une relation en BCFN si quelle que soit la DF, le membre gauche est une clé.



Normalisation : Exemple FNBC

Considérons la relation suivante: Cours (Matiere, Classe, Professeur)

complétée par les règles de gestion suivantes:

un professeur n'enseigne qu'une seule matière,
un classe n'a qu'un seul enseignant par matière

desquelles on déduit les DF suivantes:

Matière, Classe → Professeur

Professeur → Matière

Cette relation est en 3NF, néanmoins il est impossible d'enregistrer un professeur sans classe affectée (puisque classe est une partie de la clé primaire de la table Cours), et la disparition d'une classe peut entraîner la disparition de professeur.

Ceci est du au fait qu'une DF n'ait pas comme origine une clé de la relation.

•

Normalisation : Exemple FNBC

On pourrait alors décomposer la table Cours en 2 tables:
Spécialité (Professeur, Matière)
Enseignant (Classe, Professeur)

Mais la première DF serait alors perdue...

Normalisation : 2^{ième} exemple FNBC

Considérons la relation suivante:

Fournisseur (nom, produit, adresse, prix)

avec les DF suivantes :

DF1: nom \rightarrow adresse

DF2: nom, produit \rightarrow prix

Alors la clé de la relation Fournisseur est [nom, produit].

Cette relation **n'est pas en FNBC**, car dans la DF1, la partie gauche "nom" n'est pas une clé entière.

TD 02 Exercice 03

Soit la relation $R(\text{PROJET}, \text{PIECE}, \text{FOURNISSEUR}, \text{ADRESSE_F}, \text{NB_P})$ modélisant le fait qu'un fournisseur **f** ayant pour adresse **af** ait fourni **n** pièces de type **p** pour le projet **pr**.

Les attributs ont les significations suivantes:

PROJET : numéro de projet

PIECE : nom de pièce

FOURNISSEUR : nom du fournisseur

ADRESSE_F : adresse du fournisseur

NB_P : nombre de pièces fournies

Sachant que les dépendances fonctionnelles suivantes sont vérifiées par cette relation:

FOURNISSEUR \rightarrow ADRESSE_F;

PROJET, PIECE \rightarrow FOURNISSEUR, NB_P

TD 02 Exercice 03

Travail demandé:

- 1- Trouver une clé pour cette relation;
- 2- Dans quelle forme normale se trouve cette relation?
- 3- Si vous juger que la relation doit être décomposée, proposez une décomposition de celle-ci en précisant le type de forme normale et la clé des relations obtenues

TD 02 Exercice 03

Travail demandé:

1- Trouver une clé pour cette relation;

Dans l'énoncé nous avons

FOURNISSEUR → ADRESSE_F;01

PROJET, PIECE → FOURNISSEUR, NB_P02

Par distributivité de 02 nous obtenons

PROJET, PIECE → FOURNISSEUR.....03

PROJET, PIECE → NB_P04

Par transitivité de 03 et 01 nous obtenons

PROJET, PIECE → ADRESSE_F05

Par réflexivité nous obtenons :

PROJET, PIECE → PROJET06

PROJET, PIECE → PIECE07

A partir du 03, 04, 05, 06, 07 nous pouvons conclure que la clé est
PROJET, PIECE.

TD 02 Exercice 03

2- Dans quelle forme normale se trouve cette relation?

Nous remarquons que cette relation est

En première forme normale parce que nous supposons que les valeurs des attributs sont atomiques

En deuxième forme par il n'existe pas une partie de la clé qui détermine un attribut n'appartient pas à la clé.

N'est pas en 3 FN parce qu'il existe un attribut n'appartient pas à la clé (Fournisseur) qui détermine un autre attribut. (DF N° 01)

3- Pour la rendre en 3 eme FN on décompose R en 02 relations.

R1(PROJET, PIECE, FOURNISSEUR, NB_P)

R2(FOURNISSEUR, ADRESSE_F).

TD 02 Exercice 04

On considère la relation AUTEUR(NOM, ADRESSE, TITRE, ANNEE) dont les attributs ont les significations suivantes:

NOM : nom de l'auteur;

ADRESSE : adresse de l'auteur;

TITRE : titre de l'ouvrage écrit par l'auteur;

ANNEE : année d'édition de l'ouvrage.

Sachant que les dépendances fonctionnelles suivantes sont vérifiées par cette relation:

$NOM \rightarrow ADRESSE$;

$NOM, TITRE \rightarrow ANNEE$

TD 02 Exercice 04

- 1- Montrez que le couple d'attributs (NOM, TITRE) constitue une clé pour cette relation.
- 2- Dans quelle forme normale est cette relation et pourquoi ?
- 3- Proposez une décomposition de cette relation en précisant les D.F. et les clés associées aux relations obtenues ainsi que le type de forme normale de chacune.

TD 02 Exercice 04

1- Montrez que le couple d'attributs (NOM, TITRE) constitue une clé pour cette relation.

Dans l'énoncé nous avons

NOM → ADRESSE01

NOM, TITRE → ANNEE.....02

Par réflexivité nous obtenons :

NOM , TITRE → NOM03

NOM, TITRE → TITRE04

Par transitivité de 02 et 01 nous obtenons

NOM,TITRE → ADRESSE05

A partir du 03, 04, 05 nous pouvons conclure que la clé est **NOM, TITRE.**

TD 02 Exercice 04

2- Nous remarquons que cette relation est En première forme normale parce que nous supposons que les valeurs des attributs sont atomiques

N'est pas en deuxième forme par il existe pas une partie de la clé (NOM) qui détermine un attribut n'appartient pas à la clé (ADRESSE)

3-Pour la rendre en 2 FN nous décomposons AUTEUR en R1 et R2 comme suit :

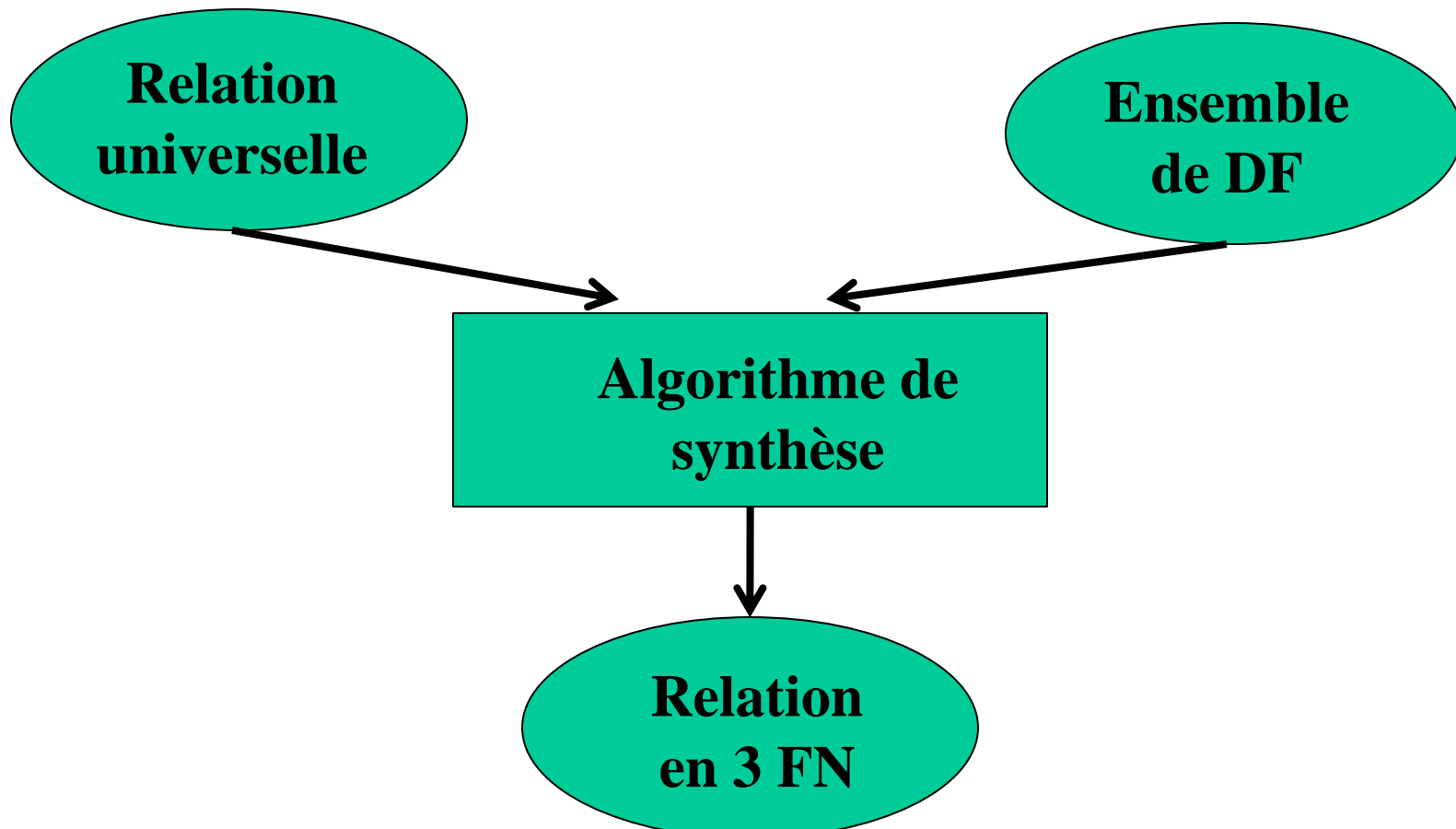
R1(NOM,TITRE,ANNEE)

R2(NOM,ADRESSE).

R1 est 3 FN et R2 est 3 FN.

Algorithme de synthèse (1)

- En appliquant les étapes de cet algorithme, vous pourrez déduire un schéma de relation en 3 FN.



Algorithme de synthèse (2)

- Une relation universelle : est une relation qui contient tous les attributs du domaine d'étude.
- Couverture minimale (irréductible): est un sous ensemble minimum de DF élémentaires permettant de générer toutes les autres.
 - Théorème: Tout ensemble de DF admet une couverture minimale, en général non unique

Algorithme de synthèse (3)

- Exemple :
- 1- code_mod → cod_filière
- 2- cod_filière → libélé_filière
- 3- ~~Code_mod → libélé_filière~~ → **Transitivité de 1 et 2**
- 4- Jour, heure, local → num_ens
- 5- ~~Jour, heure, local → cod_filière~~ → **Transitivité de 1 et 9**
- 6- Jour, heure, local → section
- 7- Jour, heure, local → groupe
- 8- Jour, heure, local → an_étude
- 9- Jour, heure, local → cod_mod
- 10 - ~~Jour, heure, local, num_ens → cod_filière~~

Augmentation de 5

Algorithme de synthèse (4)

- La couverture minimale est la suivante :
- 1- code_mod → cod_filière
- 2- cod_filière → libélé_filière
- 3- Jour, heure, local → num_ens
- 4- Jour, heure, local → section
- 5- Jour, heure, local → groupe
- 6- Jour, heure, local → an_étude
- 7- Jour, heure, local → cod_mod

Algorithme de synthèse (5)

- Etapes de l'algorithme de synthèse
 - Trouver l'ensemble $IRR(F)$ contenant la couverture irréductible (minimale) de F ;
 - Partitionner l'ensemble $IRR(F)$ en sous ensemble de F_i , tels que toutes les DFs de F_i ont la même partie gauche;
 - Pour chaque ensemble F_i de DFs, construire une relation composée de tous les attributs formant les DFs de F_i , la clé de la relation sera la partie gauche commune à ces DFs;
 - Traiter les attributs isolés

Algorithme de synthèse (7)

- Exercice

- 1- Code_mod → libellé_mod
- 2- Code_mod → an_etude
- 3- Code_mod → libellé_filière
- 4- code_filière → libellé_filière
- 5- matricule → nom_et
- 6- matricule → adresse-et
- 7- matricule → dat_nai_et
- 8- num_ens → nom_ens
- 9- num_ens → dat_nai_ens
- 10- num_ens → grade
- 11- num_ens → situation_familiale
- 12- num_ens → salaire
- 13- grade → nbre_heures
- 14- grade → salaire
- 15- cod_mod → coef
- 16- cod_mod → cod_filière
- 17- matricule → prenom_et
- 18- num_ens → prenom_ens
- 19- num_ens → adresse_ens
- 20- num_ens, grade → nbre_heures
- 21- num_ens, grade → salaire
- 22- num_ens → nbre_heures
- 23- matricule → section
- 24- matricule → groupe
- 25- cod_mod, matricule → cod_filière

Algorithme de synthèse (8)

- Concevoir un schéma relationnel en 3FN, à partir de cet ensemble de DFs en appliquant l'algorithme de synthèse.