Normalisation d'une base de données relationnelle

- Un schéma de relation est décrit par la liste de ses attributs et de leurs contraintes d'intégrité éventuelles.
- La constitution de la liste d'attributs du schéma ne peut pas se faire n'importe comment pour ne pas occasionner de redondance, avec toutes ses implications (perte de place, risques d'incohérence et de perte d'informations).
- Les formes normales des relations et les mécanismes pour les construire permettent d'obtenir des relations non redondantes.
- Ces mécanismes sont fondés sur les notions de clés de relations et de dépendances entre données.
- Ils fournissent les conditions d'application d'un processus dit de **normalisation** qui mène à des formes « correctes » de relations qui sont les **formes normales**

- La normalisation est donc le processus de transformation d'une relation ayant des problèmes lors de la mise à jour vers une autre relation n'ayant pas ces problèmes

- L'objectif est donc de définir un bon schéma relationnel qui décrit d'une manière convenable le système d'information d'une entreprise.

- Le processus de normalisation d'une BD n'est pas obligatoire : outil pratique et performant

Exemple :

- Considérons la relation suivante

Livraison (Refproduit,NumService, LibelleProduit, PU, Quantité, Adresse, Capacité)

- Elle est visiblement redondante

RefProduit	LibelleProduit	PU	Quantité	NumService	Adresse	Capacité
P1	CH7	23.510	300	S1	Sousse	9000
P1	CH7	23.510	500	S2	Tunis	6000
P3	VIS12	0.150	900	S4	Sousse	2000

Cette relation présente certaines anomalies :

- **Redondance** : un produit apparaît autant de fois qu'il sera livré par un service
- **Mise à jour** : faute de redondance, les mises à jour conduiront à des risques d'incohérence et de non intégrité.
- **Insertion et suppression** : l'insertion, la suppression ou le transfert d'attributs pourra faire apparaître des valeurs nulles

RefProduit	LibelleProduit	PU	Quantité	NumService	Adresse	Capacité
P1	CH7	23.510	300	S1	Sousse	9000
P1	CH7	23.510	500	S2	Tunis	6000
P3	VIS12	0.150	900	S4	Sousse	2000

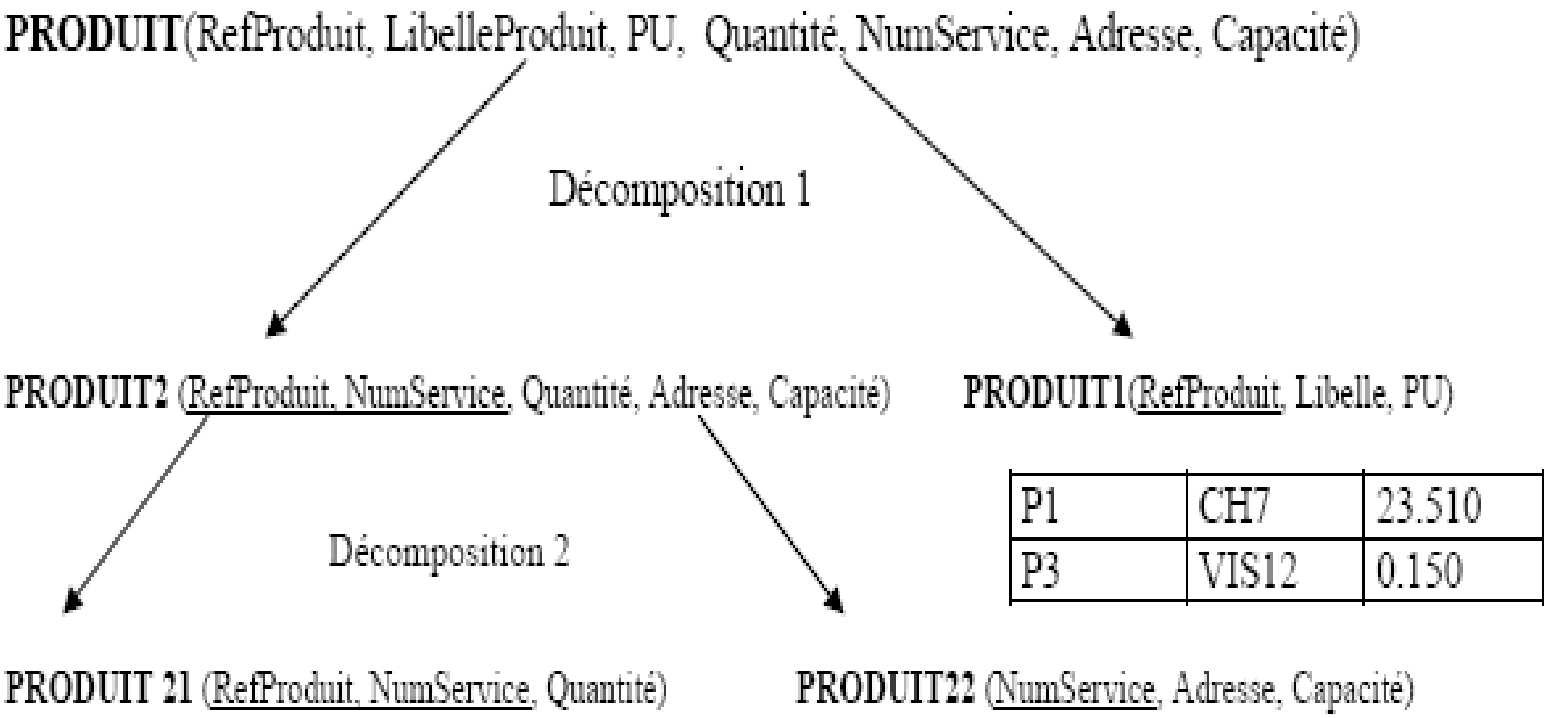
On peut dire qu'une Base de Données relationnelle est 'correcte' ou normalisée si :

- chaque relation décrit une information élémentaire avec les seuls attributs qui lui sont directement liés

- il n'y a pas de redondance d'information qui peuvent produire des problèmes de mise à jour

- La relation Produit peut être décomposée en trois relations nor redondantes :

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



P1	CH7	23.510
P3	VIS12	0.150

P1	S1	300
P1	S2	500
P3	S4	900

S1	Sousse	9000
S2	Tunis	6000
S4	Sousse	2000

Le résultat final de la décomposition est donc les relations suivantes :

- PRODUIT1 (RefProduit, LibelleProduit, PU)

  contient les données relatives aux produits.
- PRODUIT21(RefProduit,NumService, Quantité)

  contient les données relatives aux produits distribués par des services.
- PRODUIT22 (NumService, Adresse, Capacité)

  contient les données relatives aux services.

Dépendance fonctionnelle

Définition

- Un attribut ou une liste d'attributs Y dépend fonctionnellement d'un attribut ou d'une liste d'attributs X dans une relation R, si étant donnée une valeur de X, il ne lui est associé qu'une seule valeur de Y dans tout tuple de R.
- On notera une telle dépendance fonctionnelle :

  X→Y (X détermine Y ou Y dépend fonctionnellement de X).

Exemple:

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

Pour cette relation, les dépendances fonctionnelles suivantes sont vérifiées :

RefProduit → LibelleProduit

NumService → Adresse, Capacité

RefProduit → PU

RefProduit, NumService → Quantité

Propriétés des dépendances fonctionnelles

- Des axiomes et des règles d'inférence permettent de découvrir de nouvelles dépendances à partir d'un ensemble initial. Dans ce que suit nous considérons R une relation.
- Les trois premières propriétés sont connues sous le nom « Axiomes d'Armstrong »

Propriété 1 : Réflexivité

$Y \subseteq X \Rightarrow X \rightarrow Y$

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

Propriété 2 : Augmentation

$X \rightarrow Y \Rightarrow X, Z \rightarrow Y, Z$

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

Propriété 3 : Transitivité

$X \rightarrow Y$ et $Y \rightarrow Z \Rightarrow X \rightarrow Z$.

Propriété 4 : Union

$X \rightarrow Y$ et $X \rightarrow Z \Rightarrow X \rightarrow Y, Z$

Propriété 5 : Pseudo-transitivité:

$X \rightarrow Y$ et $W,Y \rightarrow Z \Rightarrow W,X \rightarrow Z$

Propriété 6 : Décomposition:

$X \rightarrow Y$ et $Z \subseteq Y \Rightarrow X \rightarrow Z$

Dépendance fonctionnelle élémentaire

Une Dépendance fonctionnelle X → Y est élémentaire si pour tout X' ⊂ X la dépendance fonctionnelle X' → Y n'est pas vraie.
En d'autres termes, Y ne dépend pas fonctionnellement d'une partie de X (X est la plus petite quantité d'information donnant Y).

Exemple :

RefProduit, LibelleProduit → PU

N'est pas élémentaire car il suffit d'avoir la référence du produit pour déterminer le prix unitaire.

Dépendance fonctionnelle canonique

- Une Dépendance fonctionnelle X→Y est canonique si sa partie droite ne comporte qu'un seul attribut

- et un ensemble F de dépendances fonctionnelles est canonique si chacune de ses dépendances est canonique.

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 X est une clé pour la relation R(X,Y,Z) si

  - Y et Z dépendent fonctionnellement de X dans R : X→ Y, Z.
  - et X→ Y, Z est élémentaire.

Graphe des Dépendances Fonctionnelles

- Les DF peuvent être représentées à l'aide d'un graphe dont les nœuds sont les attributs impliqués dans les dépendances et les arcs représentent 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 noeud unique.
- De ce fait il est nécessaire d'avoir pour la construction d'un graphe de dépendance fonctionnelle un ensemble canonique de dépendances fonctionnelles.

- Exemple :

F1 : RefProduit ->LibelleProduit

F2 : RefProduit -> PU

F3 : NumService -> Adresse, Capacité

F4 : RefProduit, NumService -> Quantité

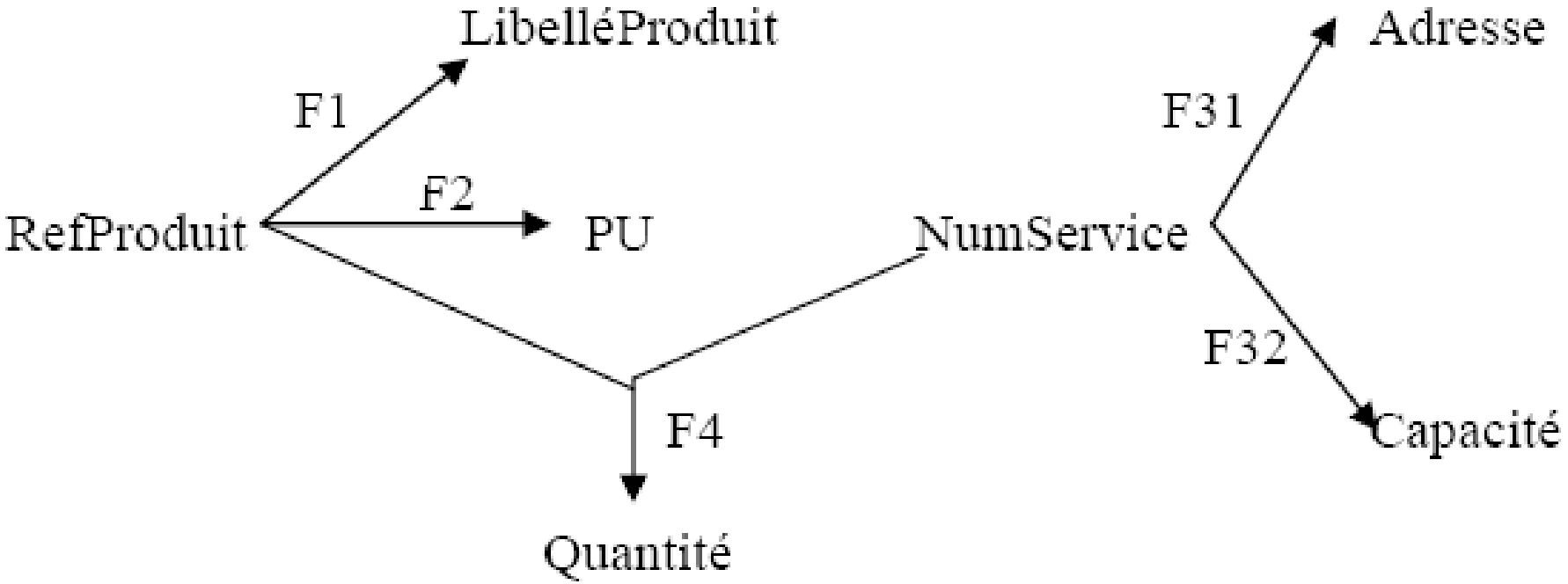
La dépendance fonctionnelle F3 n'est pas canonique, il faut donc la décomposer en deux

dépendances fonctionnelles F31 et F32 :

F31 : NumService -> Adresse

F32 : NumService -> Capacité

Le graphe des dépendances est le suivant :



Graphe des DF

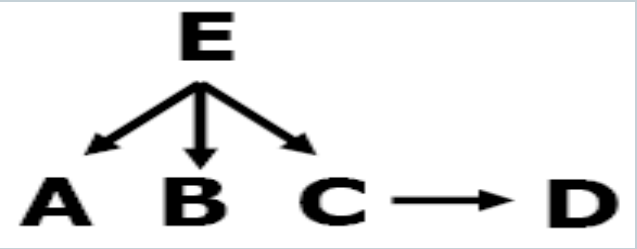
- Pour chaque relation il faut recenser toutes ses DF élémentaires et non déduites.
- On les représente sous forme d'un graphe orienté :

Graphe minimum des DF de la relation

- Une relation peut avoir plusieurs graphes minimum. Ils sont alors équivalents.
- Exemple de graphe minimum : R(A, B, C, D, E)

E→A E → B E → C (≡E → A, B, C)

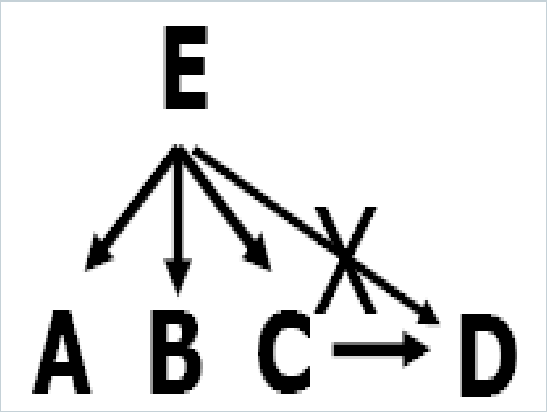
C → D



Utilité du graphe des DF

- Vérifier que le graphe est bien minimum
- Trouver les identifiants de la relation
- Tester si la relation est bonne (bien normalisée)
- Sinon trouver les décompositions

Exemple de graphe non minimum



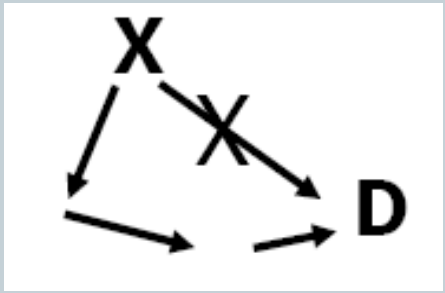
E→D est déduite de E → C et C → D

Il faut supprimer E → D du graphe

Vérifier qu'un graphe est minimum

- DF déduite

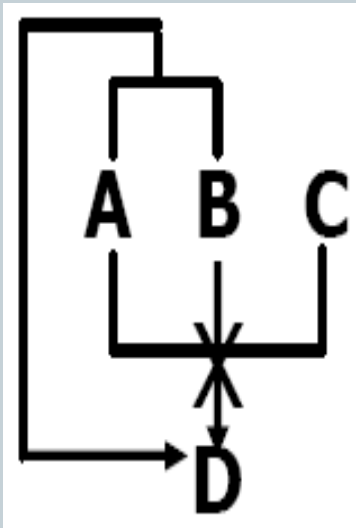
X→D est déduite s'il existe un autre chemin $X \rightarrow A_1 \ldots$ à…$A_n \rightarrow D$



- DF non élémentaire

X → D est non élémentaire s'il existe une DF Y → D telle Y est un sous-ensemble des attributs de X

(A,B,C) → D est non élémentaire

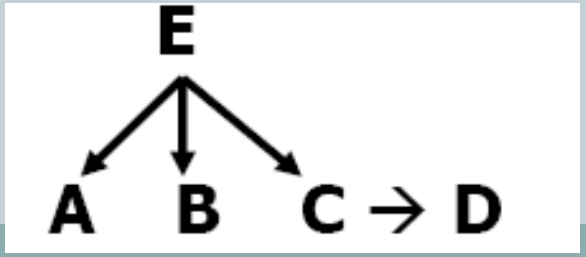


DFs et identifiants

- Le graphe minimum des DF permet de trouver les identifiants de la relation
- L'identifiant d'une relation est l'ensemble (minimal) des nœuds du graphe minimum à partir desquels on peut atteindre tous les autres noeuds (via les DF)
- Preuve :

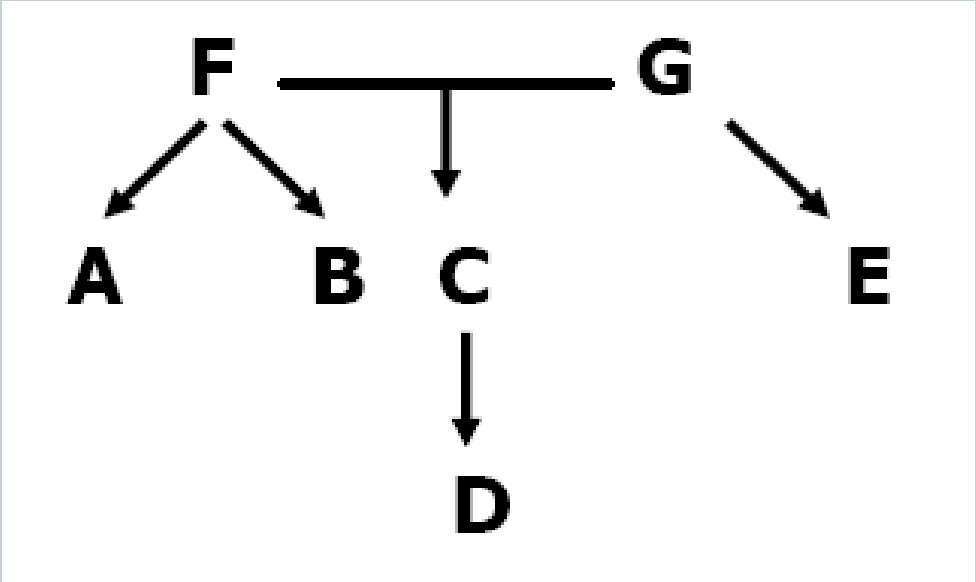
  Pour que ce soit faux il faudrait qu'il y ait deux lignes avec la même valeur de l'« identifiant » et des valeurs différentes pour les autres attributs, ce qui est en contradiction avec les DF.
- Exemple : R (A, B, C, D, E)

  E est l'identifiant de R



DFs et identifiants - Exemple

- Autre exemple : **R (A, B, C, D, E, F, G)**



(F, G) est l'identifiant de R

Fermeture transitive et Couverture Minimale

Fermeture Transitive

La fermeture transitive F+ d'un ensemble F de dépendances fonctionnelles est l'ensemble des dépendances fonctionnelles élémentaires

(X->Y ne dépend pas fonctionnellement d'une partie de X) qui peuvent être produites par application des axiomes d'Armstrong (Les trois premières propriétés des DF : Réflexivité, Augmentation, Transitivité) sur l'ensemble F.

- Exemple :

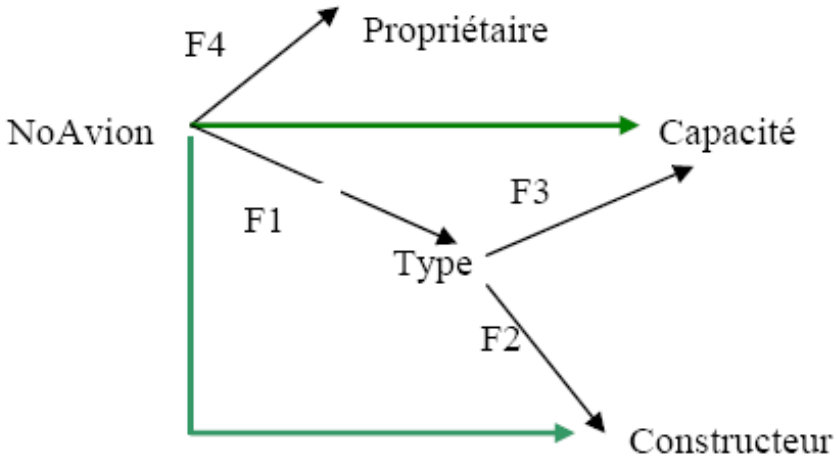
Avion(NoAvion, Type, Constructeur, Capacité, Propriété)

Avec les dépendances fonctionnelles suivantes :

F1 : NoAvion -> Type F4 : NoAvion -> Propriétaire

F2 : Type -> Constructeur F3 : Type -> Capacité

F= {F1, F2, F3, F4}



Les dépendances fonctionnelles en vert sont déduites par transitivité.

- La fermeture transitive de F est
- F+ = F U{NoAvion -> Constructeur} U {NoAvion -> Capacité}

Fermeture transitive et Couverture Minimale

Couverture Minimale

C'est un ensemble F de DF associé à un ensemble d'attributs vérifiant les propriétés suivantes :

1. Aucune dépendance dans F n'est redondante
2. Toute dépendance fonctionnelle élémentaire des attributs est dans la fermeture transitive

Exemple

La couverture minimale de l'ensemble F de l'exemple précédent est

F^ = {NoAvion → Type ; NoAvion → Propriétaire ;
Type → Constructeur ; Type → Capacité}

Les trois premières Formes Normales et la Forme Normale de Boyce Codd

- Les formes normales ont été définies pour permettre la décomposition des relations sans perte d'informations en utilisant la notion de dépendance fonctionnelle.
- Dans ce cours nous présenterons les trois premières formes normales et celle dite de Boyce Codd.

Première Forme Normale (1FN)

- Une relation R est en première forme normale et notée 1FN si chaque attribut de R a un domaine simple, c'est à dire dont les valeurs sont atomiques et monovaluées.
- Cette définition permet d'exclure les relations ayant des attributs dont les valeurs seraient des ensembles ou des listes de valeurs.

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 en la nouvelle relation :

LIVRE (No-ISBN, Titre, Auteur1, Auteur2, Auteur 3, Editeur)

ETUDIANT (Nom, Prénom, Adresse(Rue, Ville))

n'est pas en 1FN car l'attribut Adresse n'est pas atomique. Cette relation peut par exemple être transformée en la nouvelle relation suivante:

ETUDIANT (Nom, Prénom, Rue, Ville)

Deuxième Forme Normale (2FN)

Une relation R est en deuxième forme normale (2FN) si et seulement si:

- elle est en première forme normale,
- et que tout attribut n'appartenant pas à une clé ne dépendra d'aucun sous-ensemble de clé (ne dépend pas d'une partie d'une clé).

Exemple:

- Soit la relation CLIENT avec ses DF :

CLIENT (NomCl, RefProduit, AdrCl, PU)

F1 : NomCl, RefProduit → PU

F2 : NomCl→ AdrCl

La clé de la relation est (NomClt, RefProduit)

- Suite à F2, une partie de la clé (NomClt) détermine un attribut n'appartenant pas à la clé.
- Cette relation n'est donc pas en 2FN. Elle pourra être décomposée en :

R1 (NomCl, AdrCl)

R2 (#NomCl, RefProduit, PU)

Troisième Forme Normale (3FN)

- Une relation R est en troisième forme normale (3FN) si et seulement si
  - elle est en deuxième forme normale,
  - et que tout attribut n'appartenant pas à une clé ne dépendra pas d'un attribut non clé.

- La troisième forme normale permettra d'éliminer les redondances dues aux dépendances transitives.
- La décomposition en 3FN est sans perte d'informations et préserve les DF.

Exemple1 :

CLIENT (NomCl, ChiffreAffaire, Ville, Pays)

Avec les dépendances fonctionnelles suivantes

F1 : NomCl→ChiffreAffaire

F2 : NomCl → Ville

F3 : Ville → Pays

- La relation CLIENT n'est pas en 3FN à cause des dépendances fonctionnelles F2 et F3.
- Cette relation doit être décomposée en deux relations :

CLIENT(NomCl, ChiffreAffaire, #Ville)

ADRESSE(Ville, Pays)

Exemple2 :

VOITURE (NumVoiture, Marque, Type, Puissance, Couleur)

F1 : NumVoiture→ Marque, Type, Puissance, Couleur

F2 : Type → Marque

- n'est pas en 3FN. En effet, l'attribut non clé TYPE détermine MARQUE (F2).
- Cette relation peut ainsi être décomposée en deux relations :

VOITURE (NumVoiture, #Type, Couleur, Puissance)

MODELE (Type, Marque)

Forme Normale de Boyce-codd (BCNF)

- Une relation R est en BCNF si et seulement si les seules dépendances fonctionnelles élémentaires qu'elle comporte sont celles dans lesquelles une clé détermine un attribut.
- En d'autres termes une relation est en BCNF , si elle est en 3FN et qu'aucun attribut membre de la clé ne dépend fonctionnellement d'un attribut non membre de la clé.

Exemple :

ADRESSE (Ville, Rue, CodePostal)

Cette relation présente les DF suivantes :

Ville, Rue→ CodePostal

CodePostal → Ville

- Elle est en 3FN (car elle est en deuxième forme normale, et tout attribut n'appartenant pas à une clé ne dépendra pas d'un attribut non clé).
- Cette relation n'est pas en BCNF car l'attribut "Ville" (qui fait partie de la clé) dépend

  fonctionnellement de CodePostal (qui est un attribut non membre de la clé).

Normalisation par synthèse

- Il est nécessaire d'intégrer dans une BD uniquement des relations avec le degré de forme normale le plus élevé possible.
- Il existent deux algorithmes de conception de schémas relationnels en troisième forme normale ou celle de Boyce Codd.
  - Le premier procède par décomposition et le second par synthèse à partir du graphe des dépendances fonctionnelles. Dans le cadre de ce cours nous allons présenter uniquement le second algorithme.

- Le processus de la normalisation par la synthèse permet de synthétiser des schémas relationnels à partir des attributs de la relation universelle et de ses dépendances fonctionnelles.
- Le principe est le suivant :

1. Trouver une couverture minimale des dépendances fonctionnelles
2. Regrouper les attributs isolés (ceux qui ne sont ni source ni cible d'aucun arc

du graphe) au sein d'une relation dont tous les attributs sont clés ;

3. Tant que le graphe n'est pas vide :

· Rechercher le plus grand ensemble d'attributs X1, ...,Xn dépendant fonctionnellement de X, X étant un attribut ou une liste d'attributs.

· Constituer une relation d'attributs (X,X1,.....,Xn) ; cette relation a X comme clé candidate et elle est forcément en 3FN car X1,.....,Xn dépendent fonctionnellement de X et il n'y a pas de dépendances transitives puisque nous sommes partis de la couverture minimale des dépendances.

· Eliminer les dépendances (X->X1), ...,(X->Xn) du graphe

· Eliminer les attributs isolés dans le graphe

· Refaire à partir de la troisième étape

Un autre type de dépendance

Cours (nomC, prof, livre) en 1FN

nomC	prof	livre
Programmation	Duval	Algorithmes
Programmation	Duval	Progr.1
Programmation	Schmidt	Algorithmes
Programmation	Schmidt	Progr.1
BD	Jouve	Date
BD	Jouve	Ullmann
BD	Jouve	Gardarin
BD	Rochat	Date
BD	Rochat	Ullmann
BD	Rochat	Gardarin

Cours contient beaucoup de redondances

Un autre type de dépendance

- Cours pose des problèmes de maj
  - ajouter un nouveau professeur, Alex, au cours de BD
  - corriger le nom d'un livre
  - ...
- Cependant Cours est déja bien normalisée

  Cours est en BCNF
  - parce qu'il n'y a pas de DF
- Cours aurait deux attributs multivalués
  - profs
  - livres

    indépendants l'un de l'autre

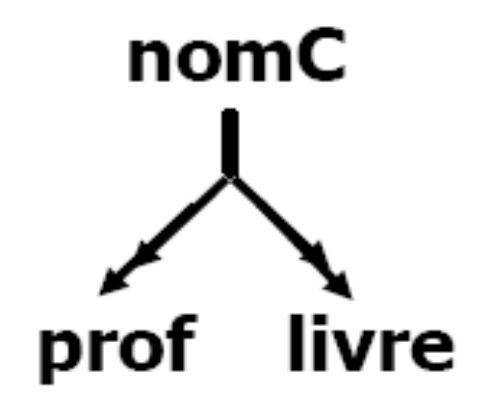
Dépendance multivaluée (DM)

- Définition :
  Soit une relation R (X, Y, Z)
  Il y a dépendance multivaluée
  X --->> Y
  si à toute valeur de X correspond un ensemble de valeurs de Y qui est totalement indépendant de Z
- Propriété :
  S'il y a la DM X-->>Y alors il y a aussi X-->>Z
  On note : X-->>Y|Z

- Remarque : DF est un cas particulier de DM

Graphe des dépendances de Cours

- Cours (nomC, prof, livre)



- La relation Cours contient **une** DM :

  nomC -->> prof | livre

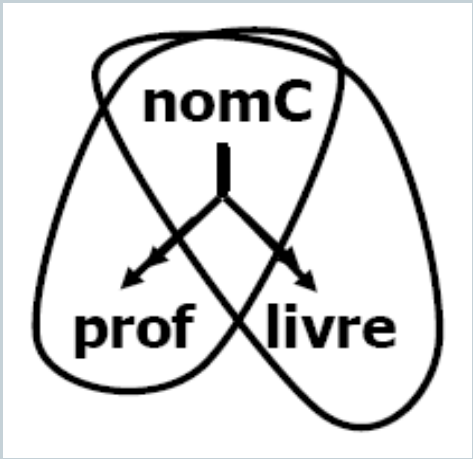
- L'identifiant de livre est (nomC, prof, livre)

4ème forme normale (4FN)

- Sémantique : La 4FN permet de séparer des faits multivalués indépendants qui auraient été réunis dans une même relation
- Définition : R est en 4FN si :
  - elle est en 1ère FN, et
  - si toute DF ou DM de R a pour source un identifiant entier de R
    - Remarque : 4FN implique FNBC
- Autre définition : R est en 4FN si elle est en FNBC et ne contient pas de DM

Décomposition selon une DM

- Exemple : Cours (<u>nomc, prof, livre</u>) FNBC est décomposé en :
  - CoursProf (<u>nomC, prof</u>) 4FN
  - CoursLivre (<u>nomC, livre</u>) 4FN



- Ces deux relations ne contiennent ni DF ni DM