

CHAPITRE 2 :

Conception de base de données

Objectifs du cours :

Présentation et définition des concepts de base liées à la conception de bases de données relationnelles : relations, clés, dépendance fonctionnelles, formes normales...

Introduction :

La plupart des recherches de ces 30 dernières années se sont basées sur l'approche relationnelle. Elle représente aujourd'hui la tendance principale du marché. La raison est que le modèle relationnel est fondé sur des bases mathématiques.

Un système relationnel est un système dans lequel :

1. l'utilisateur perçoit les données comme des tables,
2. les opérateurs génèrent de nouvelles tables résultats à partir des tables existantes.

1. Définition d'une Relation :

Définition Soient D_1, D_2, \dots, D_n des ensembles de valeurs (domaines) pas nécessairement distincts. Une relation $R (D_1, D_2, \dots, D_n)$ est un sous ensemble du produit cartésien des domaines D_1, D_2, \dots, D_n , i.e., $R (D_1, D_2, \dots, D_n) \subset D_1 \times D_2 \times \dots \times D_n$.

Un tuple ou n-uplet d'une relation est $\langle d_1, d_2, \dots, d_n \rangle : d_1 \in D_1, \dots, d_n \in D_n$.

2. Notions de clé : Clé candidate - Clé primaire – Superclé - Clé étrangère :

a) Clé candidate :

Définition : Soit R une relation, une clé candidate C pour cette relation est un sous ensemble d'attributs de R tel que :

1. Il n'existe pas deux n-uplets distincts dans R ayant la même valeur C (propriété d'unicité)
2. Aucun sous-ensemble strict de C ne possède la propriété d'unicité (propriété d'irréductibilité).

Remarques :

- Chaque relation possède, au moins, une clé candidate parce que les relations ne contiennent pas de n-uplets dupliqués. Il s'ensuit qu'au moins la combinaison de tous les attributs de la relation possède la propriété d'unicité, de là, soit cette combinaison possède la propriété d'irréductibilité et est donc une clé candidate (en fait la seule clé candidate), soit il existe au moins un sous-ensemble strict de cette combinaison qui a la propriété d'unicité et également la propriété d'irréductibilité

b) Clé primaire :

Si la relation a plusieurs clés candidates, le modèle relationnel exige que seule une de ces clés candidates soit choisie comme clé pour cette relation qu'on appellera **clé primaire**; les autres clés sont appelées **clés alternatives**.

c) Super-clé :

Définition : Une super-clé pour une relation R est un ensemble d'attribut de R qui inclut au moins une clé candidate de R .

Remarque :

- Une super-clé a la propriété d'unicité mais pas la propriété d'irréductibilité/
- Une clé candidate est une super-clé, l'inverse n'est pas vrai.

d) Clé étrangère :

Définition: Soit R1 et R2 deux relations, on appelle clé étrangère ou clé externe dans R1, tout attribut (ou groupe d'attributs) X de R1 qui apparaît comme clé primaire ou clé unique Y dans R2.

Remarques :

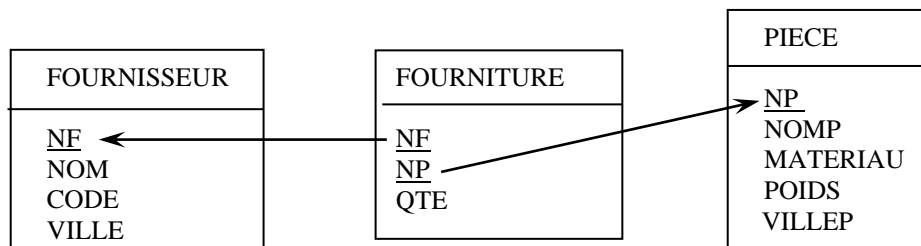
- R2 est appelée relation primaire et R1 relation secondaire.
- X et Y doivent être du même domaine (même type de données).

Exemple : Soit la base de données suivante décrivant les départements à l'aide de la relation DEPT et les employés travaillant dans ces départements à l'aide de la relation EMP :

DEPT(NDEPT, DNOM, BUDGET)

EMP(NEMP, ENOM, NDEPT, SAL)

L'attribut NDEPT de la relation EMP est une clé étrangère qui fait référence à la clé primaire NDEPT de la relation DEPT.

Exemple2 :

Dans cet exemple, la clé primaire de FOURNITURE est composée de deux attributs (NF, NP) qui forme chacun une clé étrangère : NF est une clé étrangère qui fait référence à la clé primaire de FOURNISSEUR et NP est une clé étrangère qui fait référence à la clé primaire de PIECE

3. Concept de dépendance fonctionnelle

Définition : Soit R une relation, X un attribut (ou un groupe d'attributs) de R et Y un attribut (ou groupe d'attributs) de R. On dit que Y **dépend fonctionnellement** de X (ou X détermine Y) si et seulement si, étant donné une valeur de X dans R, il lui correspond une valeur unique de Y dans R, \forall l'instant considéré (pour n'importe quelle extension de R).

On note : $X \rightarrow Y$

Le membre gauche d'une dépendance fonctionnelle (ici X) est appelée **déterminant** et le membre droit (ici Y) **dépendant**

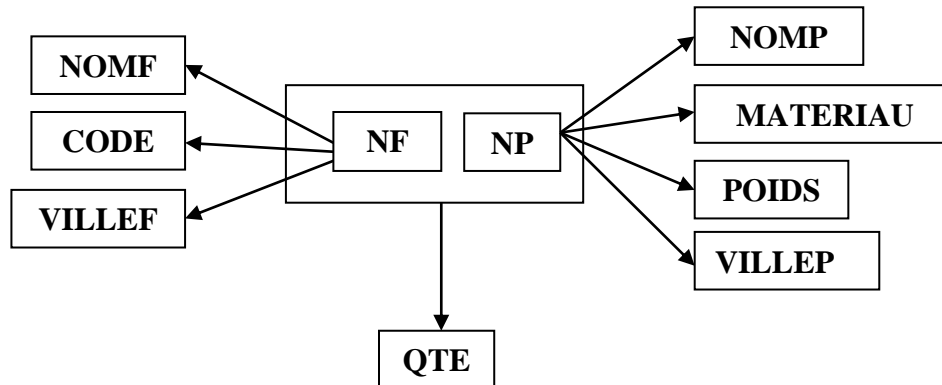
Voici quelques dépendances fonctionnelles qui s'appliquent à la relation VFOURNITURE :

$\{NF, NP\} \rightarrow QTE$ $\{NF, NP\} \rightarrow VILLE$ $\{NF, NP\} \rightarrow \{VILLE, QTE\}$
 $\{NF, NP\} \rightarrow NF$ $\{NF, NP\} \rightarrow NP$ $\{NF, NP\} \rightarrow \{NF, NP, VILLE, QTE\}$

Les dépendances fonctionnelles : $NF \rightarrow QTE$ et $QTE \rightarrow NF$ existent dans VFOURNITURE, mais ne sont pas valides à tout instant. En effet, l'énoncé "pour un fournisseur donné, toute cargaison à la même quantité de pièces livrées", est vrai pour l'exemple mais n'est pas vrai tout le temps.

Remarque :

Si X est une clé candidate de la relation R alors tous les attributs Y de la relation R sont en dépendance fonctionnelle avec X.

Diagramme de dépendances fonctionnelles

Les nœuds sont les attributs et les arcs les DFs.

4. Axiomes d'Armstrong :

Les axiomes d'Armstrong (cité ci-dessous) sont des règles d'inférences permettant de dériver des DFs à partir d'autres DFs.

Soit R une relation. Soient A, B, C trois sous-ensembles d'attributs de R,

1. **Réflexivité** : Si B est un sous-ensemble de A, alors $A \rightarrow B$ (DF triviale)
2. **Augmentation** : Si $A \rightarrow B$, alors $AC \rightarrow BC$
3. **Transitivité** : Si $A \rightarrow B$ et $B \rightarrow C$, alors $A \rightarrow C$

(A,B est écrite AB pour simplifier l'écriture)

Plusieurs règles supplémentaires peuvent être dérivées des trois règles représentées ci-dessus.

Soit D est un sous-ensemble d'attributs de R

1. **Pseudo - Transitivité**: Si $A \rightarrow B$ et $CB \rightarrow D$ alors $CA \rightarrow D$
2. **Décomposition** : Si $A \rightarrow BC$, alors $A \rightarrow B$ et $A \rightarrow C$
3. **Union** : Si $A \rightarrow B$ et $A \rightarrow C$ alors $A \rightarrow BC$
4. **Composition** : Si $A \rightarrow B$ et $C \rightarrow D$ alors $AC \rightarrow BD$

Exemple :

Soit R (A, B, C, D, E, F) une relation et soit G un ensemble de DFs satisfaites par R,

$$G = \{ \begin{matrix} A \rightarrow BC, & B \rightarrow E, & CD \rightarrow EF \\ (1) & (2) & (3) \end{matrix} \}$$

- On peut montrer que $A \rightarrow E$ est satisfaite par R en la dérivant à partir des DFs de G en utilisant les règles d'inférence :

$A \rightarrow BC$ (1) par décomposition on obtient : $A \rightarrow B$ (4) et $A \rightarrow C$ (5)

$A \rightarrow B$ (4) et $B \rightarrow E$ (2) par transitivité, on obtient $A \rightarrow E$ (6)

- De même, on peut montrer que $AD \rightarrow F$ est satisfaite par R :

$A \rightarrow C$ (5) par augmentation, on obtient $AD \rightarrow CD$ (6)

$AD \rightarrow CD$ (6) et $CD \rightarrow EF$ (3) par transitivité, on obtient $AD \rightarrow EF$ (7)

(Ou bien (5) et (3) donnent par pseudo-transitivité $AD \rightarrow EF$ (7))

La décomposition de (7) donne $AD \rightarrow E$ et $AD \rightarrow F$

Donc $AD \rightarrow F$ est bien satisfaite par R.

Remarque :

Si nous connaissons les dépendances fonctionnelles d'une relation et nous voulons déterminer les clés candidates, il suffit de considérer les super-clés qui sont irréductibles (ou rendre irréductibles les super-clés)

Exemple précédent :

$R(A, B, C, D, E, F)$ et $G = \{ A \rightarrow BC, B \rightarrow E, CD \rightarrow EF \}$

Soit la superclé formée de l'ensemble des attributs de R : $ABCDEF$

$A B C D E F$ réduite à $A B C D$ en utilisant $(CD \rightarrow EF)$

$A B C D$ réduite à $A D$ en utilisant $(A \rightarrow BC)$

$A D$ unique et irréductible : AD est une clé candidate et c'est la seule.

5. Fermeture de dépendances fonctionnelles/ Fermeture d'attributs :

a) Fermeture de dépendances fonctionnelles :

Soit R une relation et F un ensemble de dépendances fonctionnelles satisfaites par R.

On appelle **fermeture** de F (noté F^+), l'ensemble de toutes les dépendances fonctionnelles qui sont impliquées (dérivées) par les DFs de F.

Remarques :

-Les règles d'inférences sont un moyen de calculer la fermeture F^+ .

-Le calcul de F^+ peut être coûteux en temps, car le cardinal de F^+ peut être très grand même si F est restreint.

-Par contre il y a un moyen efficace pour déterminer si une DF donnée appartient à F^+ . Pour cela, nous avons besoin d'introduire la notion de **fermeture d'attributs**.

b) Fermeture d'attributs :

Soit R une relation, F un ensemble de dépendances fonctionnelles satisfaites par R.

Soit A un attribut (ou groupes d'attributs) de R. On appelle **fermeture** de A (noté A^+), l'ensemble de tous les attributs de R qui sont en dépendance fonctionnelle avec A.

L'algorithme général suivant permet de calculer cette fermeture :

```

Début
  A+ = {A}
  Tant que Vrai
    faire
      pour chaque DF  $X \rightarrow Y$  de  $F$ 
        faire :
          Si  $X$  est un sous-ensemble de  $A^+$ 
            alors  $A^+ = A^+ \cup \{Y\}$ 
          Fsi ;
        fin ;
      Si  $A^+$  n'est pas modifié pendant cette itération
        alors quitter la boucle ; /* calcul terminé */
    Fait;
  Fin ;

```

Exemple :

Soit $R(A, B, C, D, E, F)$ et $G = \{ B \rightarrow E, A \rightarrow BC, CD \rightarrow EF \}$

Calculons B^+

$B^+ = \{B\}$

1ère itération :

$B \rightarrow E$ B appartient à B^+ , alors $B^+ = \{B, E\}$

$A \rightarrow BC$ A n'appartient pas à B^+

$CD \rightarrow EF$ CD n'appartient pas à B^+

$B^+ = \{B, E\}$, B^+ a changé

2ème itération :

$B \rightarrow E$ B appartient à B^+ , alors $B^+ = \{B, E\}$

$A \rightarrow BC$ A n'appartient pas à B^+

$CD \rightarrow EF$ CD n'appartient pas à B^+

$B^+ = \{B, E\}$, B^+ n'a pas changé, donc Arrêt

$B^+ = \{B, E\}$

Calculons AD^+

$AD^+ = \{A, D\}$

1ère itération :

$B \rightarrow E$ B n'appartient pas à AD^+

$A \rightarrow BC$ A appartient à AD^+ , alors $AD^+ = \{A, D, B, C\}$

$CD \rightarrow EF$ CD appartient à AD^+ , alors $AD^+ = \{A, D, B, C, E, F\}$

AD^+ a changé

2ème itération : $AD^+ = \{A, D, B, C, E, F\}$

$B \rightarrow E$ B appartient à AD^+ alors $AD^+ = \{A, D, B, C, E, F\}$

$A \rightarrow BC$ A appartient à AD^+ , alors $AD^+ = \{A, D, B, C, E, F\}$

$CD \rightarrow EF$ CD appartient à AD^+ , alors $AD^+ = \{A, D, B, C, E, F\}$

AD^+ n'a pas changé, donc Arrêt

$AD^+ = \{A, D, B, C, E, F\}$

Ainsi, nous avons un moyen simple de déterminer si une dépendance fonctionnelle $X \rightarrow Y$ est dérivable d'un ensemble F de DFs (ie appartient à sa fermeture F^+).

Corollaire :

Etant donné un ensemble F de dépendances fonctionnelles, $X \rightarrow Y$ est dérivable de F si et seulement si Y est un sous-ensemble de la fermeture de X pour $F(X^+)$.

Remarque :

-Si X est une clé candidate ou une superclé de la relation R alors $X^+ =$ l'ensemble des attributs de R .

6. Notions de couverture d'un ensemble de DFs : Couverture & Couverture minimale**a) Couverture :**

Soit R une relation et F_1, F_2 deux ensembles de DFs définies dans R ,
 F_1 est une couverture de F_2 ssi F_2^+ est inclut dans F_1^+

Remarque : pour montrer que F_1 est une couverture de F_2 , il suffit de montrer que toute DF de F_2 appartient à F_1^+ (est dérivable à partir des DFS de F_1)

Exemple :

$R(A, B, C, D, E, F)$

$F_1 = \{ (1) A \rightarrow BC, (2) B \rightarrow D \}$

$F_2 = \{ (3) A \rightarrow BD, (4) AB \rightarrow C, (5) B \rightarrow D \}$

Montrons que F_2 est couverture de F_1 :

- (1) $A \rightarrow BC$?

(3) donne par décomposition (6) $A \rightarrow B$ et (7) $A \rightarrow D$

(6) et (4) donnent par pseudo-transitivité (8) $A \rightarrow C$

(6) et (8) donnent par union $A \rightarrow BC$ donc la df (1) est dérivable par F_2 (appartient à F_2^+)

- (2) est la même que (5) donc la df (2) est dérivable par F_2 (appartient à F_2^+)

Donc **F2 couverture de F1**

b) Ensembles de DFs équivalents :

Soit R une relation, F_1 et F_2 ensembles de DFs définies dans R , F_1 et F_2 sont équivalents ssi $F_1^+ = F_2^+$

Ou encore, ssi F_1 est une couverture de F_2 et F_2 est une couverture de F_1 .

c) Couverture minimale : (Ensemble irréductible de DFs)

Un ensemble de dépendances fonctionnelles est irréductible ssi il satisfait les trois propriétés suivantes :

1. Le membre droit (le dépendant) de chaque dépendance fonctionnelle de F contient un seul attribut (Un ensemble singleton).
2. Le membre gauche (le déterminant) de chaque dépendance fonctionnelle de F est irréductible, (c'est à dire, aucun attribut ne peut être enlevé du déterminant sans changer la fermeture F^+). Nous dirons qu'une telle dépendance fonctionnelle est **irréductible à gauche** ou élémentaire.
3. Aucune dépendance fonctionnelle n'est redondante (aucune Df ne peut être supprimée de F sans changer la fermeture de F^+).

Exemple : Soit la relation Etudiant (Matricule, Nom, Prénom, Adresse)

$\{\text{Matricule} \rightarrow \text{Nom}, \text{Matricule} \rightarrow \text{Prénom}, \text{Matricule} \rightarrow \text{Adresse}\}$

Il est facile de voir que cet ensemble de dépendances fonctionnelles est irréductible. Chaque Df a le membre droit constitué d'un seul attribut et le membre gauche irréductible. Aucune Df ne peut être supprimée sans changer la fermeture.

Les ensembles suivants ne sont pas irréductibles :

1. $\text{Matricule} \rightarrow \{\text{Nom}, \text{Prénom}\}$: le membre droit n'est pas un singleton
 $\text{Matricule} \rightarrow \text{Adresse}$
2. $\{\text{Matricule}, \text{Adresse}\} \rightarrow \text{Nom}$: Cette DF peut être simplifiée en enlevant Adresse du membre gauche sans changer la fermeture.
 $\text{Matricule} \rightarrow \text{Prénom}$
 $\text{Matricule} \rightarrow \text{Adresse}$
3. $\text{Matricule} \rightarrow \text{Nom}, \text{Prénom}$: Cette Df peut être supprimée sans changer la fermeture.
 $\text{Matricule} \rightarrow \text{Nom}$
 $\text{Matricule} \rightarrow \text{Prénom}$
 $\text{Matricule} \rightarrow \text{Adresse}$

Remarque :

Pour chaque ensemble de Dfs, il existe au moins un ensemble équivalent qui est irréductible.

En effet : Soit F , l'ensemble des Dfs.

Grâce à la règle de décomposition, chaque Df de F peut être transformée en un ensemble de Dfs ayant un membre droit singleton.

Pour chaque Df f de F ayant un membre gauche composé de plus d'un attribut, on examine chaque attribut A du membre gauche de f ; si F et l'ensemble des Dfs obtenus en éliminant A du membre gauche de f sont équivalents, on supprime A du membre gauche de f .

Pour chaque Df f restante dans F , si F et $F - \{f\}$ sont équivalents, alors on supprime f de F .

L'ensemble des Dfs final est irréductible et est équivalent à l'ensemble F initial.

Exemple :

Soit $R(A, B, C, D)$ et soient les dépendances fonctionnelles

$A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C, AC \rightarrow D$

Nous calculons un ensemble de dépendances fonctionnelles irréductible équivalent :

- | | | |
|---------------------|--|---|
| $A \rightarrow B^*$ | | • Une des deux occurrences de $A \rightarrow B$ doit disparaître |
| $A \rightarrow C$ | | • C peut être éliminée de $AC \rightarrow D$ car nous avons $A \rightarrow C$ et donc $A \rightarrow AC$ (par augmentation) et nous avons $AC \rightarrow D$ et donc $A \rightarrow D$ (par transitivité); ainsi, l'attribut C dans le membre gauche de $AC \rightarrow D$ est redondant. |
| $B \rightarrow C$ | | • $AB \rightarrow C$ peut être éliminée, car On a $A \rightarrow C$, donc $AB \rightarrow CB$ (par augmentation), et donc $AB \rightarrow C$ (par décomposition). |
| $A \rightarrow B^*$ | | • $A \rightarrow C$ est impliquée par les DF $A \rightarrow B$ et $B \rightarrow C$, elle peut être éliminée |
| $AB \rightarrow C$ | | |
| $AC \rightarrow D$ | | |

Il nous reste $A \rightarrow B, B \rightarrow C, A \rightarrow D$ Cet ensemble est irréductible.

Considérons l'ensemble suivant :

$A \rightarrow B$

$B \rightarrow C$

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$

$D \rightarrow E$

Cet ensemble s'écrit :

$A \rightarrow B$

$B \rightarrow C$

$A \rightarrow C$

$A \rightarrow D$

$A \rightarrow E$

$D \rightarrow E$

La DF ($A \rightarrow C$) n'est pas élémentaire car ($A \rightarrow B$) donc on peut l'enlever.

La DF ($A \rightarrow D$) est conséquence de ($A \rightarrow B$) (car alors ($A \rightarrow B \rightarrow C$)) et de ($B \rightarrow C$), donc elle est redondante. De même ($A \rightarrow E$) est redondante car elle se déduit de ($A \rightarrow B$), ($B \rightarrow C$) et ($D \rightarrow E$).

On obtient l'ensemble minimal suivant :

$A \rightarrow B$

$B \rightarrow C$

$D \rightarrow E$

7. Décomposition

Lors d'une conception d'une base de données relationnelle la première étape consiste à définir son schéma relationnel. Il existe plusieurs possibilités de schémas, il faut choisir les meilleurs. Encore faut-il définir les critères qui les qualifient et concevoir les algorithmes qui permettent de les obtenir.

Codd propose en 1971 les bases d'une démarche de normalisation qui conduit à décomposer une relation selon plusieurs schémas plus restreints, normalisés, sans perte d'informations. Cette démarche repose aujourd'hui sur un ensemble rigoureux de règles et la mise en œuvre d'algorithmes performants. Consiste à décomposer une relation R en un ensemble de relations normalisées.

L'objectif est d'éviter les redondances (répétition d'informations) sans causer une perte d'information.

Exemple Soit la relation VFOURNITURE (NF, VILLE, NP, QTE) dont l'extension est la suivante :

NF	VILLE	NP	QTE
F1	Alger	P1	100
F1	Alger	P2	100
F1	Alger	P3	100
F2	Tunis	P1	200
F2	Tunis	P2	200
F3	Tunis	P2	300

Les principales anomalies de mise à jour sont les suivantes :

- anomalie de modification: si on doit changer une information redondante, par exemple la ville d'un fournisseur, il faut changer cette information dans plusieurs tuples.
- anomalie d'insertion : pour ajouter les informations d'un fournisseur il faut qu'il ait fourni au moins une pièce

- anomalie de suppression : si un ensemble de valeurs devient vide, on peut perdre d'autres informations par effet de bord. Par exemple, si on supprime la pièce P2, on perd les informations sur F3 .

Pour éviter ces anomalies, il faut supprimer la redondance en décomposant le schéma relationnel. On utilise pour cela les dépendances fonctionnelles.

Qualité d'une décomposition

La qualité d'une décomposition est liée à deux aspects :

- être sans perte d'information,
- être préservative des DFs

a) Décomposition sans perte d'information

Considérons la relation fournisseur réduite uniquement aux attributs (NF, CODE, VILLE) avec l'extension suivante :

NF	CODE	VILLE
F3	130	Tunis
F5	130	Oran

Et l'ensemble de DF suivant $\{NF \rightarrow CODE, NF \rightarrow VILLE\}$

Soit les deux décompositions possibles de cette relation

a) FC

NF	CODE
F 3	130
F5	130

FV

NF	VILLE
F3	Tunis
F5	Oran

b) FC

NF	CODE
F3	130
F5	130

CV

CODE	VILLE
130	Tunis
130	Oran

Nous constatons que

1. dans le cas a) aucune information n'est perdue ; les deux relations FC et FV nous indiquent que le fournisseur F3 a le code 130 et Tunis comme adresse et le fournisseur F5 a un code 130 et Oran comme adresse. Cette décomposition est bien sans perte d'information.
2. Dans le cas b), au contraire, des informations sont définitivement perdues ; nous pouvons encore dire que les deux fournisseurs ont un code de 130, mais nous ne pouvons plus dire quelle est la ville de chaque fournisseur. La seconde décomposition est une décomposition avec perte d'information.

Nous remarquons que dans le cas a) aucune information n'est perdue car si nous effectuons la jointure des relations FC et FV, nous obtenons de nouveau la relation initiale fournisseur. Dans le cas b) au contraire, si nous effectuons la jointure des deux relations FC et CV, nous n'obtenons pas de nouveau la relation initiale fournisseur, et par conséquent nous avons perdu des informations.

Principe de décomposition :

Théorème de Heath : Soit la relation $R(A, B, C)$ où A, B et C sont des ensembles d'attributs de R . Si R satisfait la DF $A \rightarrow B$, alors R est égale à la jointure de sa projection sur $\{A, B\}$ et $\{A, C\}$

Autrement dit, la décomposition de R en $R_1(A, B)$ et $R_2(A, C)$ est sans perte d'information.

Exemple: la décomposition 1 de l'exemple précédent est sans perte d'information en considérant la DF : $NF \rightarrow VILLE$

b)Préservation des DFs :

Soit R une relation et F son ensemble de DFs, et soit une décomposition R_1, R_2, \dots, R_n de R , on note F_i l'ensemble de DFs de R qui portent sur les attributs de R_i ; cette décomposition préserve les DFs si et seulement si F est équivalent à $(F_1 \cup F_2 \cup \dots \cup F_n)$

C'est-à-dire si $G = F_1 \cup F_2 \cup \dots \cup F_n$ alors $F^+ = G^+$

Exemple :

la décomposition 2 de l'exemple précédent FC: $\{NF \rightarrow CODE\}$, CV : pas de Df : ne préserve les DF car on a perdu $NF \rightarrow VILLE$ (F^+ différent de $\{NF \rightarrow CODE\}^+$)

la décomposition 1 de l'exemple précédent : FC: $\{NF \rightarrow CODE\}$, FV : $\{NF \rightarrow VILLE\}$ préserve les DF

$F^+ = (\{NF \rightarrow CODE\} \cup \{NF \rightarrow VILLE\})^+$

8. Normalisation

8.1. Les trois formes normales

a) Première forme normale (1FN)

Définition : Une relation est en 1 FN si tous les domaines sous-jacents contiennent uniquement des valeurs scalaires (valeurs atomiques).

(Scalaire : plus petite unité sémantique de données telle que la valeur du numéro d'un fournisseur, du poids d'une pièce...)

Exemple : $R_1(NF, CODE, VILLE, NP, QTE)$ est en 1FN

b) Deuxième forme normale (2FN)

Définition : Une relation est en 2FN si et seulement si elle est en 1FN et si chaque attribut non clé est en dépendance irréductible (élémentaire) avec la clé.

Définition Soit R une relation, X et Y des sous ensembles d'attributs de R . Y est en DF élémentaire avec X ssi il n'existe pas X' sous ensemble strict de X tq $X' \rightarrow Y$.

Exemple : $R_1(\underline{NF}, \underline{NP}, CODE, VILLE, QTE)$

$NF \rightarrow CODE$,

$NF \rightarrow VILLE$

$NF, NP \rightarrow QTE$

R_1 n'est pas en 2FN

Une relation qui est 1FN et pas 2FN peut être décomposée en un ensemble de relations 2FN équivalentes (sans pertes d'informations).

Ainsi R1 est décomposée en R2 et R3 tq

R2(NF, CODE, VILLE)

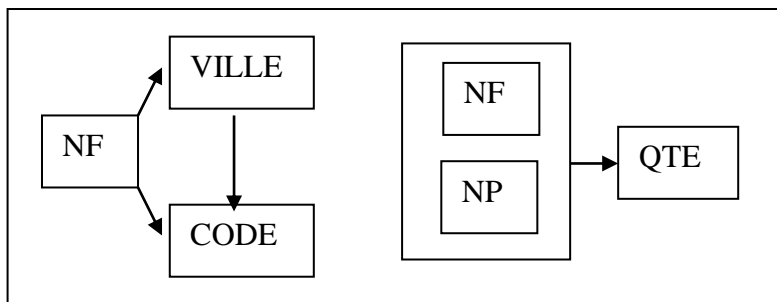
R3(NF, NP, QTE)

Les relations Seconde et fourniture sont en 2FN. Les clés sont NF pour Seconde et (NF, NP) pour Fourniture. La relation Première n'est pas en 2FN.

SECONDE (NF, CODE, VILLE)

FOURNITURE (NF, NP, QTE)

Avec le diagramme des DFs :



Et l'extension est :

SECONDE

NF	CODE	VILLE
F1	120	Alger
F2	110	Tunis
F3	110	Tunis
F4	120	Alger
F5	130	Oran

Les informations concernant les fournisseurs ont été regroupées dans Seconde, par conséquent, les informations concernant F5 sont dans seconde et pas dans fourniture. Avec ces nouvelles relations, les problèmes précédents sont résolus.

FOURNITURE

NF	NP	QTE
F1	P1	300
F1	P2	200
F1	P3	400
F1.	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P2	200
F4	P4	300
F4	P5	400

Résumé :

Une première étape de la procédure de normalisation consiste à effectuer des projections pour supprimer les dépendances fonctionnelles qui ne sont pas irréductibles. Ainsi, soit la relation R (A, B, C, D) avec la clé primaire (A, B) et la DF $A \rightarrow D$

Les principes de normalisation recommandent de remplacer R par ses deux projections R1 et R2 avec :

R1 (A, D) et la clé primaire A

R2 (A, B, C) et la clé primaire (A, B)

La relation R peut être retrouvée en effectuant la jointure de R1 et R2 sur la clé de R2.

Cette décomposition pose néanmoins des problèmes dans le cas de la relation Seconde. En effet, la dépendance CODE sur NF, bien qu'elle soit fonctionnelle et en fait irréductible, est transitive (via VILLE). Chaque valeur de ville à son tour détermine une valeur de CODE.

D'une manière générale, lorsque les DFs $A \rightarrow B$ et $B \rightarrow C$ sont valides alors $A \rightarrow C$ est valide aussi, et les dépendances transitives posent aussi des problèmes lors des mises à jour.

Exemple :

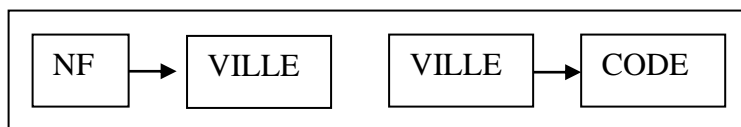
Concernant la redondance VILLE – CODE correspondant à la DF VILLE \rightarrow CODE

- On ne peut pas exprimer le fait qu'une ville particulière a un code particulier tant qu'il n'y a pas de fournisseur qui habite dans cette ville.
- S'il n'y a qu'un seul n-uplet pour une ville particulière de la relation Seconde, et si on le supprime, on supprime non seulement l'information pour le fournisseur concerné mais également le fait que cette ville a un code particulier.
- La valeur du code pour une ville donnée apparaît en général, plusieurs fois dans la relation Seconde, ce qui pose problème lors des mises à jour.

De nouveau, la solution à ces problèmes est de remplacer la relation initiale (Seconde) par deux projections :

FV (NF, VILLE) et VC (VILLE, CODE)

Le diagramme des DFs est :



FV	
NF	VILLE
F1	Alger
F2	Tunis
F3	Tunis
F4	Alger
F5	Oran

VC	
VILLE	CODE
Alger	130
Oran	120
Tunis	110
Annaba	150

L'opération est réversible puisque Seconde est la jointure de FV et VC sur VILLE. L'effet de cette normalisation a été d'éliminer la dépendance transitive de CODE sur NF.

Le problème ici est que le code ne décrit pas le fournisseur mais la ville dans laquelle habite le fournisseur.

Définition 9. Relation 3FN. Une relation est en 3FN si et seulement si elle est en 2FN et si chaque attribut non clé est en dépendance non transitive avec la clé primaire.

Les relations FV et VC sont en 3FN.

Résumé : Effectuer des projections pour éliminer les dépendances transitives.

Si R (A, B, C) et A est une clé primaire et B \rightarrow C est une DF .

Les principes de normalisation recommandent de remplacer R par ses deux projections R1 et R2 de la façon suivante : R1 (B, C) avec B comme clé primaire et R2 (A, B) avec A comme clé primaire.

La relation R peut être retrouvée en effectuant la jointure de R1 et R2 sur la clé étrangère de R2 qui est aussi la clé primaire de R1.

Il a été prouvé que toute relation a une décomposition en 3FN sans perte d'informations et qui préserve les DFs.

Plusieurs approches permettent cette décomposition : nous présenterons une approche basée sur l'algorithme de Berstein.

Algorithme de synthèse (BERNSTEIN)

Cet algorithme est basé sur la décomposition de la relation universelle. **La relation universelle** est une relation définie sur tous les attributs.

ENTREE : $R = \{A, F\}$, $A = \{\text{ensemble de tous les attributs}\}$, $F = \{\text{ensemble des DFs}\}$

SORTIE : $D = \{R_1, R_2, \dots, R_n\}$ tel que $R_i = \{X_i, Y_i\}_{i=1, \dots, n}$ en 3FN

ETAPE 1 : Rechercher une couverture irréductible de F notée F'

ETAPE 2 : Partitionner F' en groupes F'_1, F'_2, \dots, F'_k tels que toutes les DFs d'un même groupe aient même partie gauche X_i

ETAPE 3 : Pour chaque groupe F'_i construire un schéma $R_i = \{X_i, Y_i\}$ où Y_i est l'ensemble de tous les attributs apparaissant à droite du groupe F'_i

ETAPE 4 : Soient A_1, A_2, A_n les attributs de R (s'il y en a qui n'ont pas encore été pris en compte. Ajouter à D le schéma de relation constitué de la projection de R sur ces attributs.

ETAPES 5 : Si aucune relation de D n'inclut une clé candidate de R , ajouter à D le schéma de relation constitué d'une clé candidate de R .

3.3.3 Forme normale de Boyce/ Codd

Dans cette partie, il est remis en cause l'hypothèse simplificatrice que chaque relation a une seule clé candidate. En fait le cas général impose qu'une relation peut avoir plusieurs clés candidates. Nous traiterons dans ce paragraphe le cas général où :

- Une relation peut avoir plusieurs clés candidates,
- Les clés candidates peuvent être composées
- Elles se chevauchent, c'est-à-dire qu'elles ont au moins un attribut commun.

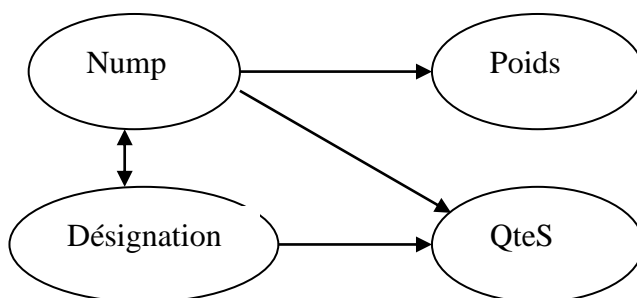
Nous substituons alors à la troisième forme normale une forme plus forte appelée **forme normale de Boyce/ Codd ou BCNF** :

Définition 10. Relation en BCNF. Une relation est en BCNF si et seulement si les seuls déterminants sont des clés candidates.

La BCNF impose qu'il ne peut pas exister dans une relation, d'autres dfs que celles qui partent des clés candidates.

Exemple1 :

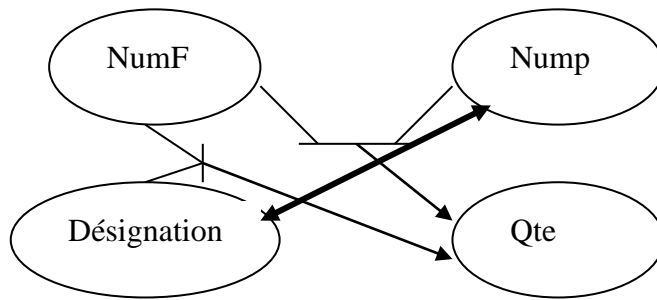
Produit (NumP, Designation, Poids, QteS) et NumP et Désignation sont deux clés candidates



Cette relation est en BCNF, car les seules flèches partent des deux clés candidates.

Exemple2 :

Four-produit (Numf, Nump, Désignation, Qte) avec 2 clés candidates : (Numf, Nump) et (Numf, Désignation) : Cette relation est-elle en BCNF ?



Cette relation n'est pas en BCNF, car il existe deux flèches entre des parties de clés.

Soit une extension de cette relation :

Numf	NumP	Désignation	Qte
F1	P1	Clou	100
F2	P1	Clou	200
F3	P1	Clou	400
F4	P1	Clou	100

La redondance désignation avec Nump, introduit une anomalie de mise-à-jour : en effet si on veut changer la désignation pour le produit P1, il faut rechercher toutes les occurrences dans la table sinon il y aurait incohérence.

Solution : décomposition en deux relations en BCNF:

R1(Nump, désignation) et R2 (Numf, Nump, Qte)

Ou bien :

R1 (Nump, désignation) et R2 (Numf, Désignation, Qte)

- *Fin Chapitre 2* -