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

Normalisation

BDD 2LMD

Partie 1

Présenté par : Dr BELLEILI Habiba

Plan

- Définitions
- Redondances
  - Anomalies
- Dépendances Fonctionnelles
- Axiomes d’Amstrong
- DF élémentaire / DF augmentée
- Réécriture de DF
- Fermeture transitive
- Couverture minimale / Graphe minimum
- Clé d’une relation

Définitions

Å La théorie de la normalisation est une théorie destinée à concevoir un **bon** schéma d’une base de données sans redondance d’information et sans risques d'anomalie de mise à jour.

Å **Redondance d’information**: les informations sont *répétées à plusieurs endroits* de la base de données

Å Elles constituent des sources de problèmes puisqu’elles sont l’une des Causes **d’incohérence** dans la BDD

Redondances

Livraison (Nofourn, adrF, Noprod, couleur, prixP, date, quantité)

NoFourn	adrF	Noprod	couleur	prixP	date	quantité
1233	**Annaba**	**P56**	**rouge**	147	12/09/19	5
1582	Skikda	P963	vert	245	16/10/19	2
1233	**Annaba**	P69	noir	159	30/11/19	3
1698	Alger	**P56**	**rouge**	258	15/12/19	2
1582	Skikda	P33	noir	168	10/01/20	5
1233	**Annaba**	**P56**	**rouge**	147	12/01/20	3

Anomalies de mise à jour

NoFourn	adrF	Noprod	couleur	prixP	date	quantité
1233	~~Sétif~~ Skikda	P56	rouge	147	12/09/19	5
1582	Annaba	P963	vert	245	16/10/19	2
1233	Sétif	P69	noir	159	30/11/19	3
1698	Alger	P56	rouge	258	15/12/19	2
1582	Skikda	P33	noir	168	10/01/20	5
1233	Sétif	P56	rouge	147	12/01/20	3

incohérence

Redondances

mise à jour simple → INCOHÉRENCE

mise à jour COMPLEXE

Anomalies d'insertion

- A chaque nouvelle livraison je dois insérer le numéro de fournisseur et son adresse

- Si je me trompe dans l'adresse je vais avoir une incohérence

1233	Annaba	P56	rouge	147	12/03/20	3

Redondances

- La relation qui présente des redondances est incorrecte

- Il faut la **décomposer**

- On parle alors de **normalisation**
- La normalisation d'une relation est un processus de décomposition d'une relation présentant des mise à jour **complexes en plusieurs relations à mises à jour simples.**

Exemple

Fournisseur (N°fourn, adrF)

Produit (N°prod, couleur)

NoFourn	adrF	Noprod	couleur	prixP	date	quantité
1233	Annaba	P56	rouge	147	12/09/19	5
1582	Skikda	P963	vert	245	16/10/19	2
1233	Annaba	P69	noir	159	30/11/19	3
1698	Alger	P56	rouge	258	15/12/19	2
1582	Skikda	P33	noir	168	10/01/20	5
1233	Annaba	P56	rouge	147	12/01/20	3

NoFourn	adrF
1233	Annaba
1582	Skikda
1698	Alger

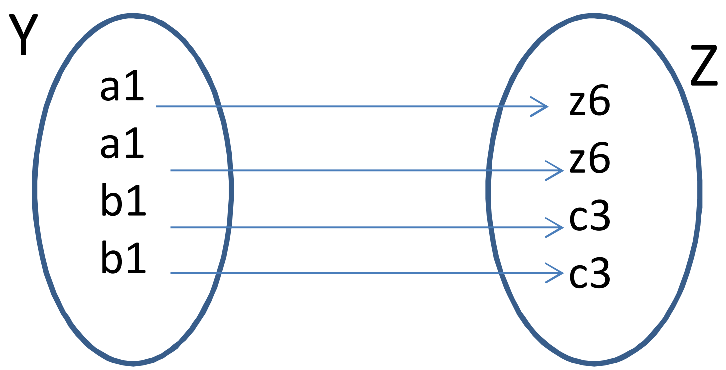
Noprod	couleur
P56	rouge
P963	vert
P69	noir
P33	rouge

Livraison (N°fourn, N°prod, Prix, date, qté)

NoFourn	Noprod	prixP	date	qté
1233	P56	147	12/09/19	5
1582	P963	245	16/10/19	2
1233	P69	159	30/11/19	3
1698	P56	258	15/12/19	2
1582	P33	168	10/01/20	5
1233	P56	147	12/01/20	3

Notion de Dépendances Fonctionnnelles (DF)

- Définition: Les DF permettent d’établir des **liens sémantiques** entre attributs ou groupe d’attributs
- Soit R une relation de schéma R(X,Y,Z,…)
- il existe une **"DF", de Y vers Z,** notée Y→Z, si :
- Etant donné *deux tuples* quelconques de R,
- s'ils ont même valeur pour Y, alors ils ont **nécessairement** même valeur pour Z.



On appelle Y **source** de la DF, et Z **cible** de la DF

Y → Z

Exemple

A	B	C	D	E
a1	b2	c2	d3	e2
a1	b2	c2	d1	e4
a2	b3	c2	d1	e5
a2	b4	c5	d1	e5

A → B n'est pas une DF

B → A est une DF

D → E n'est pas est une DF

E → D est une DF

Propriétés des DF
Axiomes d'Armstrong

Réflexivité: Soient X et Y des attributs :

$X \rightarrow X$ $\quad$ $XY \rightarrow X$ $\quad$ $XY \rightarrow XY$ $\quad$ **triviales**

$Y \rightarrow Y$ $\quad$ $YX \rightarrow Y$

Augmentation: Soient X, Y et Z des attributs :

$$X \rightarrow Y \Longrightarrow \begin{cases} XZ \rightarrow YZ \\ XZ \rightarrow Y \end{cases}$$

Transitivité: Soient X, Y et Z des attributs :

$$\left.\begin{array}{l} X \rightarrow Y \text{ et} \\ Y \rightarrow Z \end{array}\right\} \Longrightarrow X \rightarrow Z$$

Axiomes d'Amstrong

Å **Pseudo-transitivité**: Soient, W, X, Y et Z des attributs

X→Y et WY→Z ⇒ WX→Z

Démonstration:

X→Y ⇒ WX → WY (réflexivité)

WX → WY et WY→Z ⇒ WX→ Z (transitivité)

Å **Union**: Soient X, Y et Z des attributs

X→Y et **X→Z** ⇒ X→YZ

transitivité

Démonstration

Réflexivité triviale

Augmentation X→Y

(X→Y et X→Z ⇒ X→XX et XX→XY ⟹ **X→XY**

X→XY et **YX→YZ** ⇒ X→YZ)

Augmentation X→Z

transitivité

DF élémentaire

- une DF, X -> B est **élémentaire si B est un attribut** unique, et si X est un ensemble minimum d'attributs (ou un attribut unique)
  - NoFourn,Noprod,date → quantité est élémentaire
- DF non élémentaires:
  - Côté source :
    - J'ai A→B alors AX→B est non élémentaire (augmentée)
    - la cible est inclus dans la source : AB→A A ⊂AB. Elle est dite triviale
  - Côté cible : la cible est un groupe d'attributs AB→C,B C,B est un groupe d'attributs.

DF augmentée

- Une DF **non élémentaire** est dite Augmentée : si X→Y alors quelque soit A A,X→ Y est une DF Augmentée
  - NoFourn→adrF est élémentaire
  - NoFour,Noprod→adrF est augmentée

Réécriture de DF

- On peut toujours réécrire un ensemble de DF en un ensemble de DFE (DF élémentaires):
  - en supprimant les DF triviales obtenues par **réflexivité,**
  - en décomposant les DF à **partie droite non atomique** en plusieurs DFE
- AB→A n'est pas considérée car c'est une DF triviale obtenu par réflexivité.
- A→B,C sera réécrite: A→B et A→C
- AB→CB est décomposée en
  - AB→C et
  - ~~AB→B~~

triviale

Fermeture Transitive

- On appelle fermeture transitive F+ d'un ensemble F de DFE, l'ensemble de toutes les DFE qui peuvent être **composées par transitivité** ou **pseudo transitivité** à partir des DFE de F
  - F = {A→B, B→C, B→D, A→E}.
  - La fermeture transitive de F est F+ = { A→B, B→C, B→D, A→E, A→C, A→D }
- La fermeture transitive permet de retrouver toutes les DFE

Couverture minimale des DFE

- La couverture minimale (CM)d’un ensemble de DFE (notée DFE*) est un sous-ensemble minimum des DFE permettant de générer toutes les autres DFE.
- Tout ensemble de DFE admet au moins une CM
- Un ensemble de DFE peut avoir plusieurs CM

F={A→B, A→C,B→C,C→B}

$\left.\begin{array}{l} A \rightarrow B \\ B \rightarrow C \end{array}\right\} \cancel{A \rightarrow C}$

DFE*={A→B, ~~A→C~~, B→C, C→B}

2 Couvertures Minimales

$\left.\begin{array}{l} A \rightarrow C \\ C \rightarrow B \end{array}\right\} \cancel{A \rightarrow B}$

DFE*={~~A→B~~, A→C, B→C, C→B}

Couverture minimale: Algorithme

Å **Entrée:** *F* un ensemble de dépendances fonctionnelles

Å **Sortie:** G une couverture minimale de *F*

Å **Début**

1.G := *F*

2.**Décomposer**: Pour chaque DF ∈ G , **appliquer la règle de décomposition** (axiome d'Armstrong)

X-->ABC sera décomposé en X-->A ; X-->B; X-->C

3. Déterminer les **DFs élémentaires en supprimant les DF augmentées**: Supprimer les attributs en surnombre à gauche :

Pour tout X --> Y, s'il existe dans G un Z⊆X tel que Z-->Y alors remplacer X-->Y par Z-->Y

4. **Supprimer les DF déduites** :

Une DF X-->A est déduite si elle peut être retrouvée par transitivité ou pseudo transitivité

si X-->Z et Z-->A alors (par transitivité) X-->A

si X-->Y et Y,Z-->A alors (par pseudo transitivité) X,Z --> A voir diapo 12

Fin

Graphe minimum des DF

Å On appelle **graphe minimum des DF de la relation, tout ensemble de DF élémentaires non déduites,**
   ï **DF élémentaires → Pas de DF augmentée**
   ï **DF non déduite → par transitivité**

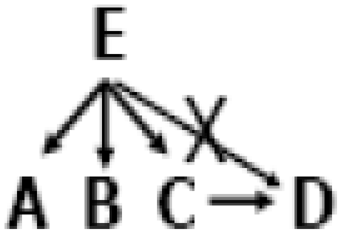
Å les DF **augmentées et déduites doivent être supprimer du graphe des DF pour** obtenir un graphe minimum des DF.

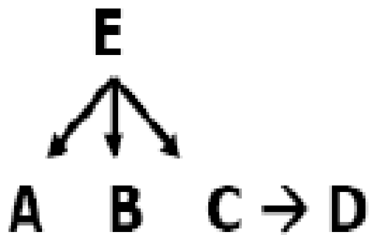
Le graphe minimum des DF sert essentiellement à définir des relations normalisées.

DF déduite ?

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

- ï établir un graphe de toutes les DF, (non minimum)
- ï supprimer la DF X→Y du graphe,
- ï 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.

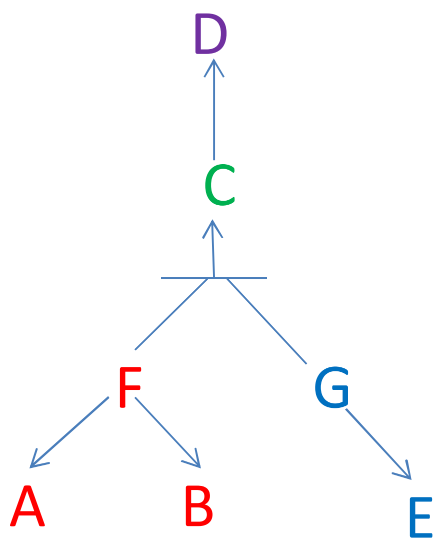


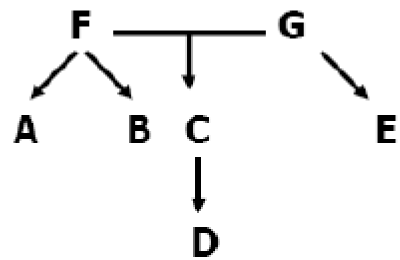


Exemple

Å R(A,B,C,D,E,F,G)

DF={F→A,
F→B,
G→E,
F,G→C,
C→D}





Clé d’une relation

- La clé d’une relation peut être cherchée à partir d’un ensemble initial de DF (Couverture Minimale ou quelconque)
- Les méthodes pour trouver la clé d’une relation:
  - L’ Algorithme de la fermeture transitive sur **les attributs** via l’ensemble de DF
  - Ou en appliquant les règles d’Armstrong sur l’ensemble des DF
  - Ou intuitivement
  - à partir de la **superclé**

Clé: Algorithme par fermeture transitive d'un attribut

- ***Données:*** *F un ensemble de DF et X un ensemble d'attributs*
- ***Résultat:*** *X^+ fermeture transitive de X*
- ***Algorithme de saturation:***

1. *Initialiser $(X)^+$ à X,*
2. *Trouver une DF $\in$ F possédant en partie gauche des attributs inclus dans $(X)^+$,*
3. *Ajouter dans $(X)^+$ les attributs placés en partie droite de la DF*
4. *Répéter les étapes 2) et 3) jusqu'à ce que $(X)^+$ n'évolue plus.*

Clé par fermeture transitive sur les attributs

- R(A,B,C,D,E,F)
- F={AB→C, C→A,BC→D, ACD→B, BE→C, CE→BD, CE→FA, D→EF}
- Clé (s) de la relation R

$(C)^+$={C} // initialisation

- $1^{ière}$ itération $(C)^+$ ={C,A},
- $2^{ième}$ itération **$(C)^+$= {CA}** reste inchangé je m'arrête // C^+ ={C,A}

(D)+ = {D}

- $1^{ière}$ itération (D)+ ={D,E,F} (D→EF) ,
- $2^{ième}$ itération **(D)+ = {D,E,F}** reste inchangé je m'arrête

(AB)+ = {A,B}

- $1^{ière}$ itération (AB)+ ={A,B,C} (AB→C), (AB)+ ={A,B,C,D}(BC→D), (AB)+ ={A,B,C,D,E,F} (D→EF)
- $2^{ième}$ itération (AB)+= ={A,B,C,D,E,F} reste inchangé ARRET //de plus tous les attributs sont obtenus à partir de AB donc **(AB) une clé candidate**

(BC)+ ={B,C}

- $1^{ière}$ itération(B,C)+={BCAD} (C→A,BC→D) (BC)+={B,C,A,D,E,F} (D→EF)
- $2^{ième}$ itération (BC))+={B,C,A,D,E,F} reste inchangé ARRET // **(BC) clé candidate** (génère tous les attributs)

(BE)+ = {B,E}

$1^{ière}$ itération (BE)+={B,E,C} (BE→C), (BE)+={B,E,C,A,D,F},

$2^{ième}$ itération (BE)+={B,E,C,A,D,F} reste inchangé ARRET // **(BE) clé candidate**

(CE)+ = {C,E}

- $1^{ière}$ itération (CE)+={C,E,B,D,A,F},
- 2itération (CE)+={C,E,B,D,A,F}, reste inchangé ARRET **// (CE) clé candidate**

Superclé

- On Appelle une **superclé** d'une relation R une clé contenant:
  - Tous les attributs de la relation
  - Ou, pour optimiser, c'est l'union de toutes les parties gauches des DF

  - Exemples:
  - R(A,B,C,D,E) F={A→BD, CD→E, B→E}
  - Superclé= (ABCDE) ou (ABCD)

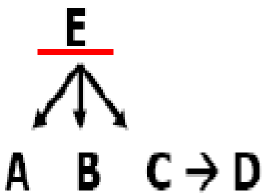
Clé par réduction de la superclé

- On peut obtenir la clé d’une relation par réduction de la superclé
  - Exemple:R(A,B,C,D,E) F={A→B, CD→E, B→E}
  - Superclé= (ABCDE)
  - ABCDE $\xrightarrow{A\to B}$ ACDE $\xrightarrow{CD\to E}$ ACD (2 étapes)
  - ACD est une clé

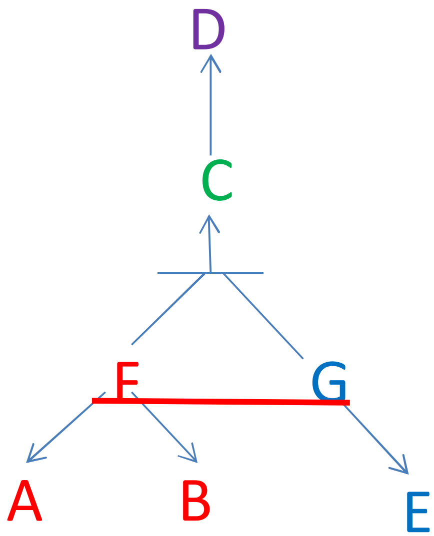
Une meilleure superclé est l’union des parties gauches des DF = ACD~~B~~ $\xrightarrow[A\to B]{}$ ACD (1 seule étape)

Clés d’une relation: a partir GM

- Les Clés peuvent aussi être cherchées à partir du graphe minimum des DF,
- Les Clés correspondent à l’ensemble minimum d’attributs qui nous permettent, en suivant, les DF d’atteindre tous les autres attributs.



La Clé est E



La Clé est (F+G)

Clés Candidates et clé primaires

Å Si une relation comporte plusieurs clés, chacune est dite clé candidate

Å On choisit une en particulier pour être la clé primaire.

Å Toutes les clés candidates sont des clés, pas seulement la clé primaire.

Å *Les clés candidates se déterminent mutuellement*

ï *Clé1 →Clé2 et aussi Clé2 →Clé1 (Clé1 ⟷ Clé2)*

Å **Si** une relation R n'admet aucune clé K (sous ensemble des attributs A1..An de R)

Å **alors** la clé K=A1..An est composée de *tous les attributs de R.*

Fin Normalisation
Partie 1