

INFO-H-403 Bases de données  
Séance d'exercices 10  
Normalisation

F. Servais et B. Verhaegen

19 décembre 2007

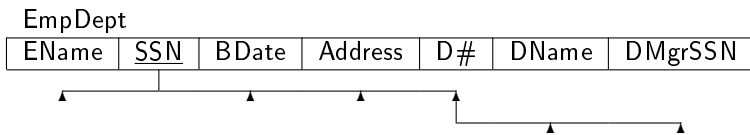
# But de la normalisation

EmpDept

ENAME	<u>SSN</u>	BDate	Address	DNumber	DName	DMgr
-------	------------	-------	---------	---------	-------	------

- ▶ Objectif : construire un schéma relationnel évitant la redondance
- ▶ La redondance implique des anomalies lors de
  - ▶ l'insertion (nouvel employé, nouveau département)
  - ▶ la suppression (du dernier employé d'un département)
  - ▶ la modification (changement de manager)

# Dépendances fonctionnelles (DF)



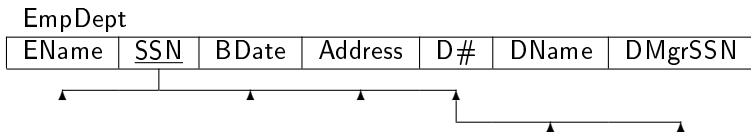
DF1 :  $SSN \rightarrow \{EName, BDate, Address, D\# \}$

DF2 :  $D\# \rightarrow \{DName, DMgrSSN \}$

Soit  $R(A_1, \dots, A_n)$  avec  $X, Y \subseteq \{A_1, \dots, A_n\}$

Il y a une **dépendance fonctionnelle**  $X \rightarrow Y$  (X détermine Y)  
si pour chaque paire de tuple  $t_1, t_2$  de R,  
si  $t_1[X] = t_2[X]$  alors  $t_1[Y] = t_2[Y]$ .

## Dépendances fonctionnelles : exemple



DF1 :  $SSN \rightarrow \{EName, BDate, Address, D\# \}$

DF2 :  $D\# \rightarrow \{DName, DMgrSSN \}$

EmpDept

EName	<u>SSN</u>	BDate	Address	D#	DName	DMgrSSN
Smith	1234	21/07/39	...	1	Research	1234
Narayan	6668	18/01/43	...	1	Research	1234
English	4534	8/05/53	...	2	Account	4534
Wong	9788	30/11/49	...	3	Admin	9788
Zelaya	6677	23/08/60	...	3	Admin	9788

# Règles d'inférences des DF

## Axiomes d'Armstrong

- ▶ (Réflexivité) Si  $Y \subseteq X$ , alors  $X \rightarrow Y$
- ▶ (Augmentation) Si  $X \rightarrow Y$ , alors  $XZ \rightarrow YZ$  (et  $XZ \rightarrow Y$ )
- ▶ (Transitivité) Si  $X \rightarrow Y$  et  $Y \rightarrow Z$ , alors  $X \rightarrow Z$

## Règles supplémentaires

- ▶ (Décomposition) Si  $X \rightarrow YZ$ , alors  $X \rightarrow Y$  et  $X \rightarrow Z$
- ▶ (Union) Si  $X \rightarrow Y$  et  $X \rightarrow Z$ , alors  $X \rightarrow YZ$
- ▶ (Pseudotransitivité) Si  $X \rightarrow Y$  et  $WY \rightarrow Z$ , alors  $WX \rightarrow Z$

Les 3 dernières règles sont des conséquences des 3 premières.

## Couverture minimale d'un ensemble $F$ de DF

Un ensemble  $F$  de DF **couvre** un ensemble  $G$  de DF si on peut inférer chaque DF de  $G$  avec les DF de  $F$ .

Deux ensembles de DF  $F$  et  $G$  sont **équivalents** si  $F$  couvre  $G$  et  $G$  couvre  $F$ .

La **couverture minimale** d'un ensemble  $F$  de DF est un ensemble minimal de DF qui est équivalent à  $F$ . On parle aussi de **graphe minimum des dépendances**.

# Couverture minimale : algorithme

## Trouver une couverture minimale $G$ de $F$

- (1)  $G \leftarrow F$
- (2) Remplacer chaque DF  $X \rightarrow \{A_1, \dots, A_n\}$  dans  $G$  par  $n$  DF  $X \rightarrow A_i$
- (3) Pour chaque DF  $X \rightarrow A$  dans  $G$ ,  
pour chaque attribut  $B$  de  $X$   
si  $(G - \{X \rightarrow A\}) \cup \{(X - \{B\}) \rightarrow A\}$  est équivalent à  $G$   
alors remplacer  $X \rightarrow A$  par  $(X - \{B\}) \rightarrow A$  dans  $G$
- (4) Pour chaque DF  $X \rightarrow A$  restante dans  $G$   
si  $(G - \{X \rightarrow A\})$  est équivalent à  $G$   
alors enlever  $X \rightarrow A$  de  $G$

## Fermeture d'un ensemble d'attribut $X$

La **fermeture d'un ensemble d'attributs  $X$** ,  $(X)^+$  représente l'ensemble des attributs de  $R$  qui peuvent être déduits de  $X$  à partir d'un ensemble  $F$  de DF et des règles d'inférences.

$Y$  est inclus dans  $(X)^+$  si et seulement si  $X \rightarrow Y$

Algorithme :

1.  $(X)^+ \leftarrow X$
2. trouver une DF dans  $F$  dont la partie gauche est incluse dans  $(X)^+$
3. ajouter dans  $(X)^+$  la partie droite de cette DF
4. répéter les opérations 2 et 3 jusqu'à ce que  $(X)^+$  n'évolue plus



# Première forme normale

Une relation  $R$  est en **première forme normale** si :

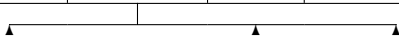
- ▶  $R$  respecte la définition du modèle relationnel
- ▶  $R$  ne possède pas d'attribut composés ou multivalués

Toutes les relations que l'on a vu jusqu'à présent respectent la première forme normale.

## Première forme normale : exemple

Department

<u>DName</u>	<u>DNumber</u>	DMgr	{DLocations}
--------------	----------------	------	--------------



Department

<u>DName</u>	<u>DNumber</u>	DMgr	{DLocations}
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

↓ 1NF Normalization

Department

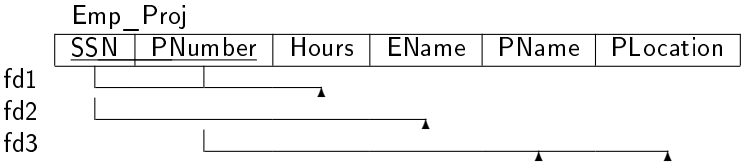
<u>DName</u>	<u>DNumber</u>	<u>DLocations</u>	DMgr
Research	5	Bellaire	333445555
Research	5	Sugarland	333445555
Research	5	Houston	333445555
Administration	4	Stafford	987654321
Headquarters	1	Houston	888665555

## Deuxième forme normale

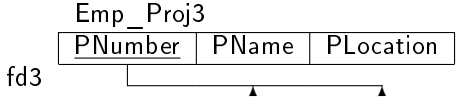
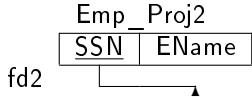
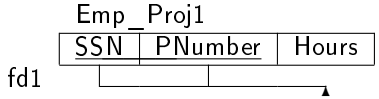
Une relation  $R$  est en **deuxième forme normale** si :

- ▶  $R$  est en première forme normale
- ▶ il n'y a pas d'attribut ne faisant pas partie d'une clé qui dépend d'une partie de cette clé

# Deuxième forme normale : exemple



↓ 2NF Normalization

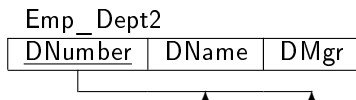
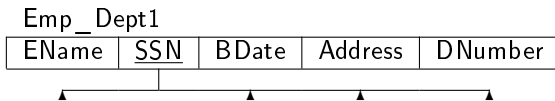
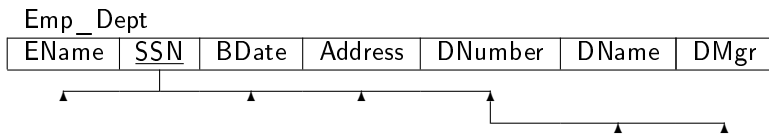


# Troisième forme normale

Une relation  $R$  est en **troisième forme normale** si :

- ▶  $R$  est en deuxième forme normale
- ▶ il n'y a pas d'attribut ne faisant pas partie d'une clé qui dépend transitivement de cette clé

## Troisième forme normale : exemple



# Forme normale de Boyce-Codd (BCNF)

Une relation  $R$  est en **BCNF** si :

- ▶  $R$  est en troisième forme normale
- ▶ la partie gauche de chaque DF est une clé candidate entière

La plupart des relations en troisième forme normale sont en BCNF.

# Décomposition

Lors de la décomposition, il faut veiller à ne pas perdre d'informations.

Pour cela, il faut utiliser des décompositions ayant la propriété de jointure sans perte (Lossless Join Property) et l'algorithme de décomposition combinée (Combined Decomposition Algorithm) vus au cours.



## Perte d'information (spurious tuples)

EmpProj

<u>SSN</u>	<u>P#</u>	Hours	EName	PName	PLocation
1234	1	32.5	Smith	ProjX	Bellaire
1234	2	7.5	Smith	ProjY	Sugarland
6668	3	40.0	Narayan	ProjZ	Houston
4534	1	20.0	English	ProjX	Bellaire
4534	2	20.0	English	ProjY	Sugarland



EmpProj1

<u>SSN</u>	<u>P#</u>	Hours	PName	PLocation
1234	1	32.5	ProjX	Bellaire
1234	2	7.5	ProjY	Sugarland
6668	3	40.0	ProjZ	Houston
4534	1	20.0	ProjX	Bellaire
4534	2	20.0	ProjY	Sugarland

EmpLocs

<u>EName</u>	<u>PLocation</u>
Smith	Bellaire
Smith	Sugarland
Narayan	Houston
English	Bellaire
English	Sugarland

## Perte d'information (spurious tuples)

EmpProj1 ⋈ EmpLocs

	<u>SSN</u>	<u>P#</u>	Hours	PName	PLocation	EName
	1234	1	32.5	ProjX	Bellaire	Smith
*	1234	1	32.5	ProjX	Bellaire	English
	1234	2	7.5	ProjY	Sugarland	Smith
*	1234	2	7.5	ProjY	Sugarland	English
	6668	3	40.0	ProjZ	Houston	Narayan
*	4534	1	20.0	ProjX	Bellaire	Smith
	4534	1	20.0	ProjX	Bellaire	English
	4534	2	20.0	ProjY	Sugarland	Smith
*	4534	2	20.0	ProjY	Sugarland	English

# Algorithme de décomposition combinée

## Décomposition sans perte d'information et sans perte de dépendance fonctionnelle

1. trouver un ensemble minimum  $G$  de  $F$
2. pour chaque  $X$  d'une DF  $X \rightarrow A$  de  $G$   
créer une relation dans  $D$  avec les attributs  $\{X \cup A_1 \cup \dots \cup A_k\}$   
où  $X \rightarrow A_1, \dots, X \rightarrow A_k$  sont les seules DF de  $G$  avec  $X$   
comme partie gauche ( $X$  est la clé de cette relation)
3. si aucune des relations de  $D$  ne contient une clé de  $R$ , créer une relation contenant les attributs qui forment une clé de  $R$