

## Opérations relationnelles (4)

## Jointure

- Lien entre relations exprimée par le contenu commun d'un champ
- L'opération de jointure peut-être vue comme une sélection des enregistrements obtenus par le produit cartésien des relations, dont les contenus du champ sur lequel on effectue la jointure sont égaux (**équijointure**)
- Notation  $R = R1 \bowtie_{(R1.X = R2.Y)} R2$  ou  $R = \sigma_p (R1 \times R2)$
- **$\theta$ -jointure** : jointure dont l'opérateur = est remplacé par un autre opérateur
- **semi-jointure** : on ne garde dans R que les attributs correspondant à R1 ou R2
- **jointure externe** : inclut également les enregistrements avec valeurs indéfinies

Code	Nom	Prix	Stock
DT4300	Dell Precision 4300	900	3
DL4200	Dell Latitude 4200	900	3
SV2540	Sony Vaio 2540	1529	2

T1

Ref	Processeur	HDD
DT4300	Core 2 Duo	250
IBMT42	Core 2 Quad	500
SV2540	Pentium	160

T2

Code	Nom	Prix	Stock	Processeur	HDD
DT4300	Dell Precision 4300	900	3	Core 2 Duo	250
SV2540	Sony Vaio 2540	1529	2	Pentium	160

 $T1 \bowtie_{(Code = Ref)} T2$

## Opérations relationnelles (5)

## Division

- Version simplifiée : la relation premier opérande comprend 2 attributs, la seconde comprend un attribut commun avec la première
- Notation  $R = R1[A, B] \div R2[B]$
- La relation résultat comprend :
  - l'attribut A de R1 qui ne figure pas dans R2
  - les tuples formés par la valeur <a> telle qu'il existe dans R1 un tuple <a,b> pour tout <b> appartenant à R2

A	B
a	u
a	x
b	y
b	v
c	u
c	x
c	t
d	x
d	y

T1

B
u
x

T2

A
a
c

 $T1[A,B] \div T2[B]$

# Agrégation (1)

- Permet de regrouper les enregistrements qui contiennent les mêmes valeurs
- Notation  $R = \text{AGG}(R1 ; X ; B)$ 
  - X : liste des attributs de R1 servant à définir le critère de regroupement
  - B : attribut agrégé
  - AGG : fonction d'agrégation (COUNT, SUM, MIN, MAX, AVG)
- Calcul de R :
  - tri suivant les valeurs de X : pour chaque valeur  $\langle x \rangle$  de X, on a un groupe de n-uplets
  - pour chaque groupe, on applique la fonction d'agrégation qui renvoie une valeur y
  - pour chaque groupe, on génère un couple  $\langle x, y \rangle$  à insérer dans R

## Agrégation (2)

Code	Modèle	Cat	Nb
1	Peugeot 107	A	3
2	Renault Clio	B	1
3	Fiat Panda	A	2
4	Peugeot 207	B	5
5	Citroën C4	C	6
6	Volkswagen Polo	B	2

Location

$R = \text{SUM (Location, Cat, Nb)}$

Code	Modèle	Cat	Nb
1	Peugeot 107	A	3
3	Fiat Panda	A	2
2	Renault Clio	B	1
4	Peugeot 207	B	5
6	Volkswagen Polo	B	2
5	Citroën C4	C	6



Cat	Nb
A	3
A	2
B	1
B	5
B	2
C	6



Cat	Nb
A	5
B	8
C	6

# Exercice 5 : film et catalogue (1)

## Enoncé général

On considère les 2 relations Film et Catalogue.

1. trouvez la liste des titres films et leur format
2. est-il possible de faire une jointure sur le champ « Prix » de la relation « Film » et le champ « Numero\_Film » de la relation « Catalogue » ?

Prix	Format	Type	Nombre	Numero_Film
12	4/3	Couleur	3	2
4	16/9	Noir/Blanc	1	4
12	16/9	Couleur	1664	50
35	4/3	Noir/Blanc	890	12
12	16/9	Noir/Blanc	1	12

**Film**

Titre	Numero_Film
Le train qui passe	2
A toi !	4
Les chats du Sénégal	56
Le temps expliqué	111
Les impôts faciles	12

**Catalogue**

## Exercice 5 : film et catalogue (2)

## 1. Jointure sur les champs « Numero\_Film »

Prix	Format	Type	Nombre	Numero_Film (F)	Numero_Film (C)	Titre
12	4/3	Couleur	3	2	2	Le train qui passe !
4	16/9	Noir/Blanc	1	4	4	A toi !
35	4/3	Noir/Blanc	890	12	12	Les impôts faciles
12	16/9	Noir/Blanc	1	12	12	Les impôts faciles

## 1. Projection sur les champs « Titre » et « Format »

Format	Titre
4/3	Le train qui passe !
16/9	A toi !
4/3	Les impôts faciles
16/9	Les impôts faciles

# Exercice 5 : film et catalogue (3)

## 2. Jointure sur les champs « Prix » et « Numero\_Film »

Prix	Format	Type	Nombre	Numero_Film (F)	Numero_Film (C)	Titre
12	4/3	Couleur	3	2	12	Les impôts faciles
4	16/9	Noir/Blanc	1	4	4	A toi !
12	16/9	Couleur	1664	50	12	Les impôts faciles
12	16/9	Noir/Blanc	1	12	12	Les impôts faciles

## Exercice 6 : calcul sur des agrégats (1)

### Enoncé général

On considère la relation Film : quelle est la moyenne des prix de films par format ?

Prix	Format	Type	Nombre	Numero_Film
12	4/3	Couleur	3	2
4	16/9	Noir/Blanc	1	4
12	16/9	Couleur	1664	50
35	4/3	Noir/Blanc	890	12
12	16/9	Noir/Blanc	1	12

Film



## Exercice 6 : calcul sur des agrégats (2)

<b>R = AVG (Film, Format, Prix)</b>
-------------------------------------

Prix	Format	Type	Nombre	Numero_Film
12	4/3	Couleur	3	2
35	4/3	Noir/Blanc	890	12
4	16/9	Noir/Blanc	1	4
12	16/9	Couleur	1664	50
12	16/9	Noir/Blanc	1	12

Film

Prix	Format
12	4/3
35	4/3
4	16/9
12	16/9
12	16/9

Prix	Format
23,5	4/3
9,33	16/9

# Généralités

- La normalisation permet de vérifier la robustesse de la conception d'un modèle et facilite la mémorisation des données
- La normalisation permet d'éviter la redondance et les problèmes sous-jacents de cohérence ou de mise à jour
- La normalisation s'applique à toutes les entités et aux relations porteuses de propriétés
- Il existe 4 principaux types de formes normales, que nous allons détailler dans la suite de cette partie

# Première forme normale

## Principe

- Une relation est en première forme normale si à l'intersection de n'importe quelle ligne  $i$  et de colonne  $k$ , il n'y a qu'une seule valeur pour  $t_j[A_k]$
- La première forme normale interdit donc la **multivaluation**
- Tout champ contient donc une valeur atomique

## Exemple

<u>ISBN</u>	Titre	Auteurs
2100521578	Aide-mémoire Java	Vincent Granet, Jean-Pierre Regourd
2744023108	Javascript et Ajax	Tom Negrino, Doris Smith
2744021822	Java SE version 6	Louis Dirk, Peter Müller

Publication

Décomposition en 3 relations :

- Publication(ISBN, Titre)
- Auteur(NumAuteur, Nom, Prénom)
- Ecrire(ISBN, NumAuteur)

Importance de passer par une  
étape de modélisation !!!!

# Deuxième forme normale

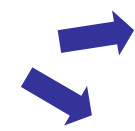
## Principe

- Une relation est en deuxième forme normale si :
  - elle est en première forme normale
  - si un sous-ensemble de la clé n'est pas source de la dépendance fonctionnelle
- Autrement dit si les DF issues de la clé sont **élémentaires**
- Si la clé d'une relation est atomique, elle est en 2<sup>ème</sup> forme normale
- Si la relation n'est pas en 2<sup>ème</sup> forme normale, on la décompose en relations qui seront toutes en 2<sup>ème</sup> forme normale

## Exemple



<u>Article</u>	<u>Fournisseur</u>	Adresse	Prix
Cuisinière	De Dietrich	Strasbourg	599
Hotte	Faure	Paris	159
Hotte	De Dietrich	Strasbourg	349
Robot	Kitchen Aid	New York	549



<u>Article</u>	<u>Fournisseur</u>	Prix
Cuisinière	De Dietrich	599
Hotte	Faure	159
Hotte	De Dietrich	349
Robot	Kitchen Aid	549

<u>Fournisseur</u>	Adresse
De Dietrich	Strasbourg
Faure	Paris
De Dietrich	Strasbourg
Kitchen Aid	New York

# Troisième forme normale

## Principe

- Une relation est en 3<sup>ème</sup> forme normale si :
  - elle est en 2<sup>ème</sup> forme normale
  - si toutes les DF issues de la clé sont **directes**
- $A \rightarrow B$  est directe s'il n'existe pas d'attribut C tel que  $A \rightarrow C$  et  $C \rightarrow B$
- La 3<sup>ème</sup> forme normale interdit donc toute dépendance fonctionnelle dite **transitive** entre les champs
- Si la relation n'est pas en 3<sup>ème</sup> forme normale, on la décompose en relations qui seront toutes en 3<sup>ème</sup> forme normale

## Exemple

<u>Code</u>	Marque	Nom	Couleur
22	Apple	IPod	Noir
599	Apple	IPhone	Blanc
3	Creative	Zen	Rouge
17	Microsoft	Zune	Noir



<u>Code</u>	Nom	Couleur
22	IPod	Noir
599	IPhone	Blanc
3	Zen	Rouge
17	Zune	Noir

<u>Marque</u>	Nom
Apple	IPod
Apple	IPhone
Creative	Zen
Microsoft	Zune



# Forme normale de Boyce-Codd

## Principe

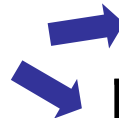
- Une relation est en forme normale de Boyce-Codd si la source de chaque DF existant entre les attributs est clé de la relation
- Il s'agit donc du cas où une partie d'une clé dépend d'un champ
- Lors de la décomposition de la relation, plusieurs choix sont envisageables : il faut choisir la décomposition qui permet de reconstituer l'information sans générer des données supplémentaires

## Exemple

<u>Réf</u>	<u>Marque</u>	Prix	Couleur
22	Apple	49	Noir
599	Apple	49	Blanc
3	Creative	12	Rouge
22	Tesqr	39	Vert
17	Microsoft	59	Noir

~~MP3(Marque, Réf, Prix)~~

~~Marque\_coul(Couleur, Marque)~~



<u>Réf</u>	Prix	<u>Couleur</u>
22	49	Noir
599	49	Blanc
3	12	Rouge
22	39	Vert
17	59	Noir

<u>Marque</u>	<u>Couleur</u>
Apple	Noir
Apple	Blanc
Creative	Rouge
Tesqr	Vert
Microsoft	Noir



# Exercice 7 : deuxième forme normale (1)

## Enoncé général

On considère la relation **Film** présentée ci-après. Il ne peut y avoir 2 fois le même film avec le même format et les DVD sont en 16/9 et les VHS en 4/3.

La relation est-elle en deuxième forme normale ?

Prix	Format	Type	Nombre	Numero_Film	Support
12	4/3	Couleur	3	2	VHS
4	16/9	Noir/Blanc	1	4	DVD
12	16/9	Couleur	1664	56	DVD
35	4/3	Noir/Blanc	890	12	VHS
12	16/9	Noir/Blanc	1	12	DVD

**Film**

# Exercice 7 : deuxième forme normale

Première forme normale ? → oui

Clé(s) ?

- pas de clé candidate atomique
- couple « Numero\_Film – Format » comme clé primaire

Deuxième forme normale ?

- DF entre « Support » et « Format » : un format n'a qu'un support
- On décompose donc la relation :

Prix	<u>Format</u>	Type	Nombre	<u>Numero Film</u>
12	4/3	Couleur	3	2
4	16/9	Noir/Blanc	1	4
12	16/9	Couleur	1664	56
35	4/3	Noir/Blanc	890	12
12	16/9	Noir/Blanc	1	12

<u>Format</u>	<u>Support</u>
4/3	VHS
16/9	DVD

FormatSup

Film



# Généralités

- Il existe 2 familles de techniques pour la construction de schémas relationnels normalisés :
  - à partir d'une abstraction : modèle entité/association ou UML
  - à partir des attributs
- Pour la seconde technique, il existe 3 démarches de construction de schéma normalisés :
  - l'approche par synthèse
  - la méthode du graphe des dépendances fonctionnelles
  - l'approche par décomposition
- En guise d'aperçu, nous allons voir la première approche, par synthèse

# Approche par synthèse (1)

## Principe

- On liste tous les attributs
- On identifie les dépendances fonctionnelles (DF)
- Puis on suit l'algorithme de synthèse (voir diapo suivante)

## Exemple

**Attributs** : P, F, N, PU, C, T

**Dépendances fonctionnelles** :

- $f1 : F \rightarrow N$
- $f2 : P, F, N \rightarrow PU$
- $f3 : P \rightarrow C$
- $f4 : P \rightarrow T$
- $f5 : C \rightarrow T$
- $f6 : N \rightarrow F$
- $f7 : P, C \rightarrow T$

# Approche par synthèse (2)

### Etape 1

*On élimine dans les dépendances fonctionnelles exprimées les attributs redondants (en partie gauche avec au moins 2 attributs)*

### Calcul des attributs redondants

Calcul basé sur l'axiomatisation d'Armstrong :

- axiome 1 : **réflexivité** si  $Y \subseteq X$  alors  $X \rightarrow Y$
- axiome 2 : **augmentation** si  $X \rightarrow Y$ , alors  $\forall Z, XZ \rightarrow YZ$
- axiome 3 : **transitivité** si  $X \rightarrow Y$  et  $Y \rightarrow W$  alors  $X \rightarrow W$

On peut déduire plusieurs règles de ces 3 axiomes :

- **augmentation à gauche** : si  $X \rightarrow Y$ , alors  $\forall Z, XZ \rightarrow Y$
- **union** : si  $X \rightarrow Y$  et  $Y \rightarrow Z$ , alors  $X \rightarrow YZ$
- **pseudo-transitivité** : si  $X \rightarrow Y$  et  $YZ \rightarrow W$  alors  $XZ \rightarrow W$
- **décomposition** : si  $X \rightarrow YZ$  alors  $X \rightarrow Y$  et  $X \rightarrow Z$

# Approche par synthèse (3)

## Etape 1 – suite de l'exemple

f1 :  $F \rightarrow N$

f2 :  $P, F, N \rightarrow PU$

f3 :  $P \rightarrow C$

f4 :  $P \rightarrow T$

f5 :  $C \rightarrow T$

f6 :  $N \rightarrow F$

f7 :  $P, C \rightarrow T$

**f2**

$F \rightarrow N$  donc  $P, F \rightarrow N$  (augmentation)

$F \rightarrow F$  (réflexivité) donc  $P, F \rightarrow F$  (augmentation)

$P \rightarrow P$  (réflexivité) donc  $P, F \rightarrow P$  (augmentation)

$P, F \rightarrow N$  et  $P, F \rightarrow F$  et  $P, F \rightarrow P$  donc  $P, F \rightarrow P, F, N$

$P, F \rightarrow P, F, N$  et  $P, F, N \rightarrow PU$  donc  $P, F \rightarrow PU$  (transitivité)

**$\Rightarrow N$  est donc redondant dans f2**

**f7**

$P \rightarrow C$  (f3) et  $C \rightarrow T$  (f5) donc  $P \rightarrow T$  (transitivité)

**$\Rightarrow C$  est donc redondant dans f7**



f1 :  $F \rightarrow N$

f2' :  **$P, F \rightarrow PU$**

f3 :  $P \rightarrow C$

f4 :  $P \rightarrow T$

f5 :  $C \rightarrow T$

f6 :  $N \rightarrow F$

**f7' :  $P \rightarrow T$**

## Approche par synthèse (4)

### Etape 2

*On élimine les dépendances fonctionnelles redondantes (celles qui peuvent être engendrées par transitivité)*

### Suite de l'exemple

f1 :  $F \rightarrow N$   
f2' :  $P, F \rightarrow PU$   
f3 :  $P \rightarrow C$   
f4 :  $P \rightarrow T$   
f5 :  $C \rightarrow T$   
f6 :  $N \rightarrow F$   
f7' :  $P \rightarrow T$

f4 et f7 sont identiques  
de plus,  $P \rightarrow C$  (f3) et  $C \rightarrow T$  (f5)  $\Rightarrow P \rightarrow T$  (f4)  
 $\Rightarrow$  **f4 et f7 sont redondantes**



f1 :  $F \rightarrow N$   
f2' :  $P, F \rightarrow PU$   
f3 :  $P \rightarrow C$   
f5 :  $C \rightarrow T$   
f6 :  $N \rightarrow F$

# Approche par synthèse (5)

### Etape 3

*Regroupement des dépendances fonctionnelles selon les parties gauches*

### Suite de l'exemple

f1 :  $F \rightarrow N$   
f2' :  $P, F \rightarrow PU$   
f3 :  $P \rightarrow C$   
f5 :  $C \rightarrow T$   
f6 :  $N \rightarrow F$

Toutes les parties gauches sont différentes



$H_1 = \{F \rightarrow N\}$   
 $H_2 = \{P, F \rightarrow PU\}$   
 $H_3 = \{P \rightarrow C\}$   
 $H_4 = \{C \rightarrow T\}$   
 $H_5 = \{N \rightarrow F\}$

## Approche par synthèse (6)

### Etape 4

*Prise en compte des dépendances fonctionnelles doubles permettant de regrouper des ensembles  $H_i$  avec des clés équivalentes*

### Suite de l'exemple

$H_1 = \{F \rightarrow N\}$
$H_2 = \{P, F \rightarrow PU\}$
$H_3 = \{P \rightarrow C\}$
$H_4 = \{C \rightarrow T\}$
$H_5 = \{N \rightarrow F\}$

On peut regrouper  $H_1$  et  $H_5$   
→ il faut choisir la clé primaire



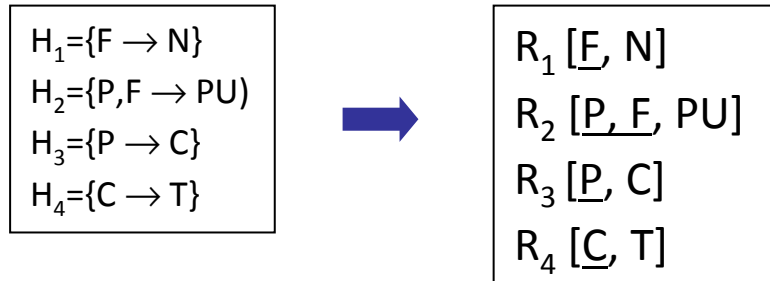
Ici, on peut prendre $F$ ou $N$ , car il n'y a pas d'autre dépendances fonctionnelles avec $F$ ou $N$ comme source
--

# Approche par synthèse (7)

## Etape 5

*Construction des relations*

## Suite de l'exemple





# Généralités

- Nous avons défini dans les parties précédentes les différents opérateurs de l'algèbre relationnelle
- Dans cette partie, nous allons voir quelques exemples afin de comprendre comment exprimer des requêtes avec ces opérateurs
- **Rappels**
  - *union* :  $\cup$
  - *intersection* :  $\cap$
  - *différence* : appartenance à une relation et pas à une autre  $/$
  - *produit cartésien* : combinaison des enregistrements  $\times$
  - *projection* : extraction de certains champs (colonnes)  $R = \Pi_x R1$
  - *selection* : extraction de certains tuples (lignes) suivant un prédicat  $R = \sigma_p R1$
  - *jointure* : lien entre relations par le contenu d'un champ  $R = R1 \bowtie_{(R1.X = R2.Y)} R2$
  - *division*
  - *agrégation* :  $R = \text{AGG}(R1 ; X ; B)$  avec :
    - AGG** fonction d'agrégation
    - X** critère de regroupement
    - B** attribut

## Base de données de travail (1)

- On considère la base de données suivante :
  - **Chercheur** [NC, NOMC]
  - **Projet** [NP, NOMP, NE, BUDGET]
  - **Equipe** [NE, NOME]
  - **Aff** [#NC, #NP]
- Les tables chercheur, Projet et Equipe décrivent les chercheurs, les projets et les équipes. La table **Aff** associe les chercheurs aux projets sur lesquels ils travaillent
- Les règles sont les suivantes :
  - une équipe comprend plusieurs chercheurs
  - un chercheur n'appartient qu'à une équipe
  - une équipe développe plusieurs projets
  - un projet n'est développé que par une équipe
  - les chercheurs d'un projet appartiennent à une même équipe

## Base de données de travail (2)

**Equipe**

NE	NOME
e1	Merlin
e2	Parole

**Chercheur**

NC	NOMC
ch1	Jean
ch2	Jacques
ch3	Paul
ch4	Pierre
ch5	François
ch6	Mathieu
ch7	Jean
ch8	Florent
ch9	Michel

**Aff**

NC	NP		
ch3	p5	ch5	p2
ch2	p5	ch8	p7
ch2	p4	ch8	p1
ch7	p4	ch8	p2
ch6	p4	ch8	p8
ch2	p6		
ch3	p6		
ch6	p6		
ch2	p3		
ch1	p1		
ch4	p1		
ch1	p2		
ch4	p2		
ch1	p8		
ch2	p9		
ch5	p8		
ch4	p8		

**Projet**

NP	NOMP	NE	BUDGET
p7	VACBI	e2	120000
p5	SRI	e1	160000
p1	HYPERMEDIA	e2	130000
p4	BIG	e1	110000
p6	DIABETO	e1	200000
p8	QUAERO	e2	500000
p9	VORTEX	e1	250000
p2	IMAGE	e2	140000
p3	VIDEO	e1	120000

## Questions... et réponses (1)

## Question 1

*Restituer le nom du chercheur « ch1 »*

→  $RQ1 = \Pi_{NOMC} (\sigma_{NC = 'ch1'} Chercheur)$

## Question 2

*Restituer le nom des chercheurs travaillant sur le projet « p5 »*

→ Aff [**NP**, NC]                      Chercheur [NC, **NOMC**]



$R1 = \sigma_{NP = 'p5'} Aff$

$R2 = R1 \Join_{(R1.NC = Chercheur.NC)} Chercheur$

$RQ2 = \Pi_{NOMC} R2$

## Questions... et réponses (2)

### Question 3

*Restituer par chercheur le nombre de projets sur lequel ils travaillent*

→ Agrégation avec :

- AGG = COUNT
- critère de regroupement = chercheur (NC)
- attribut = projet (NP)

**RQ3 = COUNT(Aff ; NC ; NP)**

### Question 4

*Restituer par chercheur le nombre de projets sur lequel ils travaillent (nom du chercheur ↔ nombre de projets)*

→ **RQ3** [NC, COUNT\_NP]                      **Chercheur** [NC, NOMC]



**R1 = RQ3  $\bowtie$  <sub>NC</sub> Chercheur**

**RQ4 =  $\Pi_{\text{NOMC, COUNT\_NP}}$  R1**

## Questions... et réponses (3)

### Question 5

*Restituer les noms des chercheurs qui ne travaillent pas sur le projet « IMAGE »*

$$\rightarrow R1 = \sigma_{NOMP = 'IMAGE'} \text{Projet}$$

$$R2 = \text{Aff} \triangleright_{NP} R1$$

$$R3 = \Pi_{NC} R2$$

$$R4 = \Pi_{NC} \text{Chercheur}$$

$$R5 = R4 - R3$$

$$R6 = \text{Chercheur} \triangleright_{NC} R5$$

$$RQ7 = \Pi_{NOMC} R6$$