

Cours 5 : La normalisation

Rabii EL GHORFI

Module : Technique de programmation avancées

Département : Mathématiques, informatique et géomatique (MIG)

EHTP 2017-2018

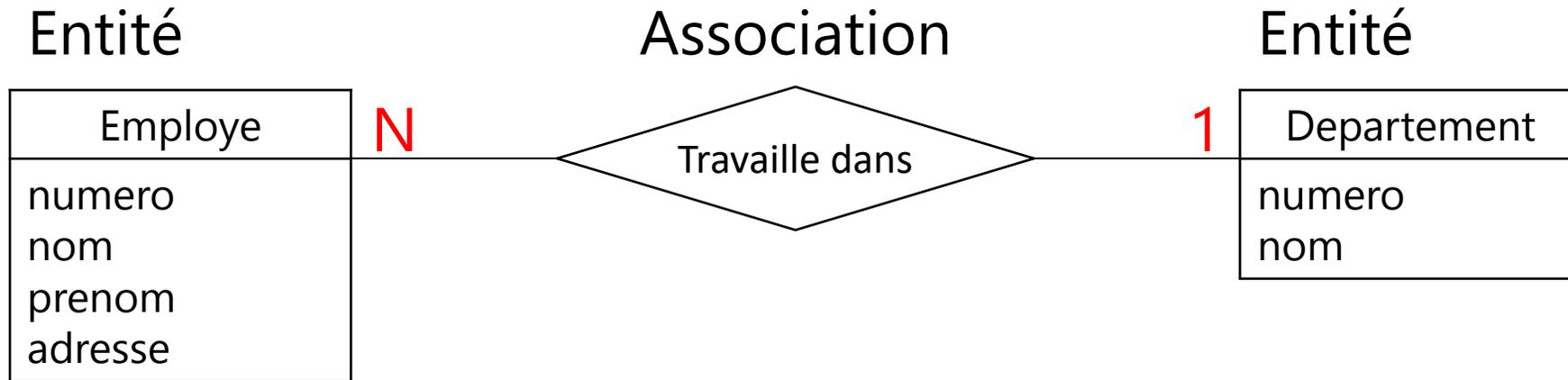


Principaux axes du cours

- Passage des diagrammes E-A aux tables
- Dépendances fonctionnelles (DF)
- Fermeture d'un ensemble de DF
- Normalisation : 1NF, 2NF, 3NF, BCNF
- Normalisation : Avantages et inconvénients
- Dénormalisation

Diagrammes E-A → tables (1)

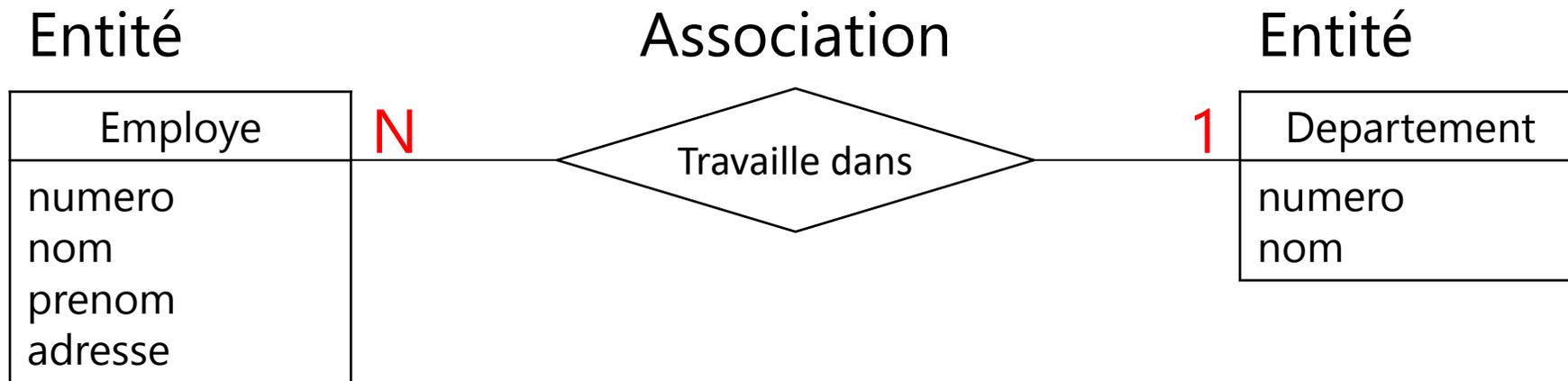
- Le modèle E-A (Entité-Association) fournit une description graphique des données



- Dans ce modèle, il existe différents types d'association entre les entités
 - L'association N-1
 - L'association N-N
 - L'association 1-1

Diagrammes E-A → tables (2)

- L'association N-1 :



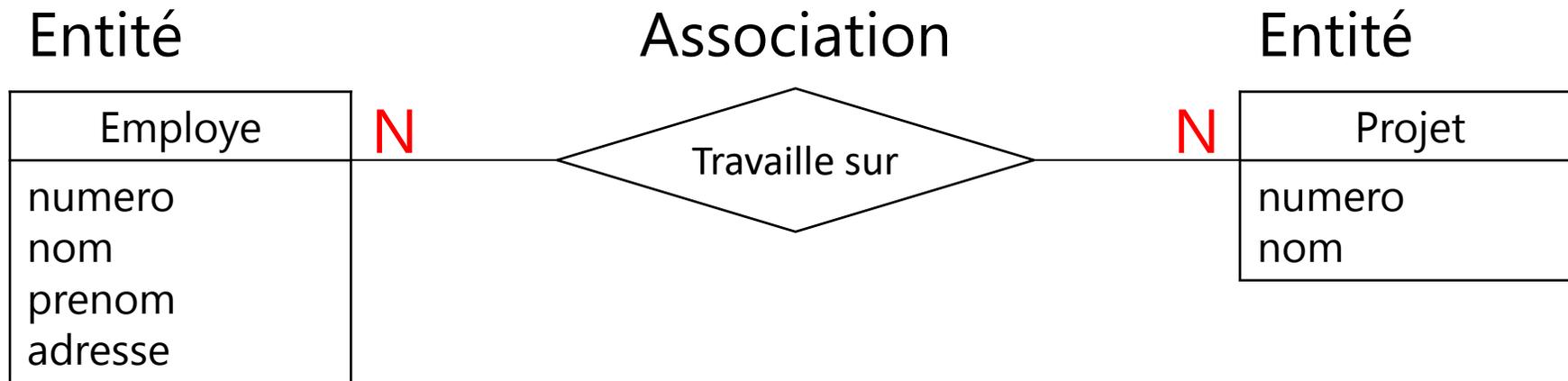
Règle de passage : Inclure la clé primaire de l'entité de cardinalité 1 dans l'autre entité de cardinalité n

Employe(numE, nom, prenom, adresse#, numD#)

Departement(numD, nom)

Diagrammes E-A → tables (3)

- L'association N-N :



Règle de passage : Créer une nouvelle table et inclure la clé primaire de chaque entité dans cette table

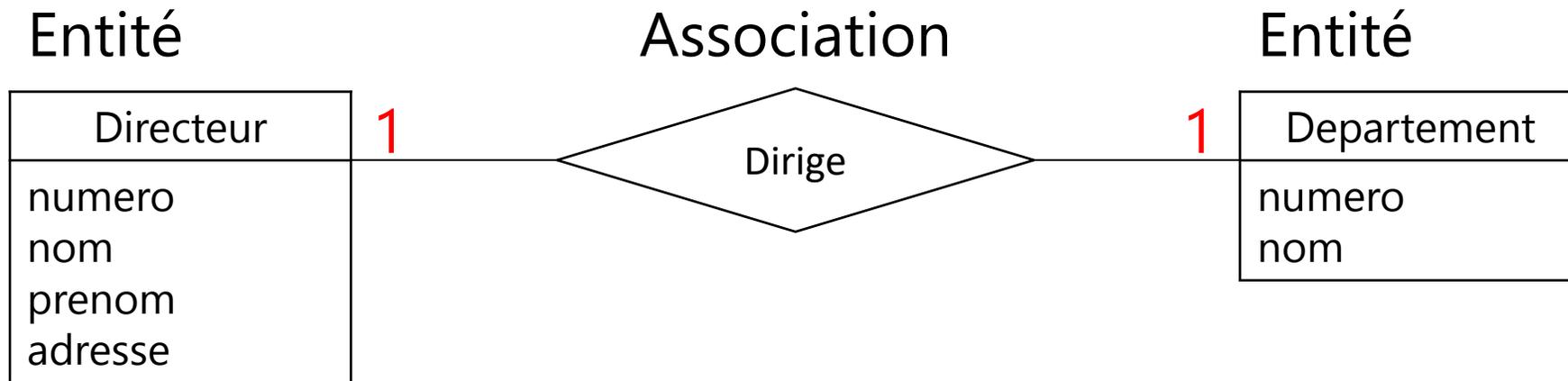
Employe(numE, nom, prenom, adresse#)

Projet(numP, nom)

EmployeProj(numE#, numP#)

Diagrammes E-A → tables (4)

- L'association 1-1 :



Règle de passage : inclure la clé primaire d'une entité dans l'autre entité **ou** combiner les deux entités

Solution 1 : Directeur(numE, nom, prenom, adresse#, numD#)

Departement(numD, nomDep)

Solution 2 : Directeur(numero, nom, prenom, adresse#, numD, nomDep)

Dépendances fonctionnelles (1)

- Une dépendance fonctionnelle concerne deux attributs ou deux groupes d'attributs notés X et Y

Syntaxe : $X \rightarrow Y$

Notation : X détermine Y ou Y a une dépendance fonctionnelle de X

Signification : Si on connaît la valeur de X , on peut connaître celle de Y

- Exemple : K est une clé candidate de la table R , Si on a

$\forall X$ attribut de R : $K \rightarrow X$

Dépendances fonctionnelles (2)

- On constate la dépendance fonctionnelle :

ville → pays

Table : Adresse

numero	numRue	nomRue	ville	pays
201	120	Nakhil	Rabat	Maroc
202	7	Anfa	Casablanca	Maroc
801	3	Lapaix	Paris	France
203	8	Alfath	Rabat	Maroc

- Il est inutile de répéter le couple (Rabat, Maroc)
- L'information Ville est suffisante pour déduire Pays

Dépendances fonctionnelles (3)

- A partir des données présentes dans la table R déduire quelques dépendances fonctionnelles

Table : R

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b3	c2	d3
a3	b3	c2	d4

$A \rightarrow A$: triviale

$A \rightarrow C$: vrai

$C \rightarrow A$: pas vrai

$AC \rightarrow D$: pas vrai

$AB \rightarrow D$: vrai (pas certain, val différentes)

Attention : l'ajout de nouveaux tuples peut tout changer

Dépendances fonctionnelles (4)

- Propriétés de base

Réflexivité :

Si Y est inclus dans X c.à.d. $X = (Y, Z, \dots)$ alors $X \rightarrow Y$

Augmentation :

Si $X \rightarrow Y$ alors $WX \rightarrow WY$

Transitivité :

Si $X \rightarrow Y$ et $Y \rightarrow Z$, alors $X \rightarrow Z$

Attention : W, X, Y, Z représentent soit un attribut soit un groupe d'attributs

Dépendances fonctionnelles (5)

- Propriétés déduites

Pseudo transitivité :

Si $X \rightarrow Y$ et $YW \rightarrow Z$, alors $XW \rightarrow Z$

Union :

Si $X \rightarrow Y$ et $X \rightarrow Z$, alors $X \rightarrow YZ$

Décomposition :

Si $X \rightarrow YZ$, alors $X \rightarrow Y$ et $X \rightarrow Z$

Remarque : Les propriétés déduites découlent des propriétés de base

Fermeture d'un ensemble de DF (1)

- Une fermeture F^+ est l'ensemble de toutes les dépendances fonctionnelles logiquement impliquées par F

- Exemple :

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

$F :$

- $A \rightarrow BC$
- $CD \rightarrow E$
- $B \rightarrow D$
- $E \rightarrow A$

- Quelques éléments de la fermeture :

$A \rightarrow BC \Rightarrow A \rightarrow B$ et $A \rightarrow C$ (décomposition)

$A \rightarrow B$ et $B \rightarrow D \Rightarrow A \rightarrow D$ (transitivité)

$A \rightarrow C$ et $A \rightarrow D \Rightarrow A \rightarrow CD$ (union)

$\{A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow CD\} \subset F^+$

- En pratique, on utilise un algorithme pour trouver tous les éléments de F^+

Fermeture d'un ensemble de DF (2)

- Les clés candidates X sont les éléments de la fermeture F^+ tels que $X \rightarrow R$
Dans notre exemple : X clé candidate si $X \rightarrow ABCDE$

- Exemple :

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

$F :$
 $A \rightarrow BC$
 $CD \rightarrow E$
 $B \rightarrow D$
 $E \rightarrow A$

$A \rightarrow CD$ et $CD \rightarrow E \Rightarrow A \rightarrow E$ (transitivité)

$A \rightarrow A, A \rightarrow B, A \rightarrow C, A \rightarrow D$ et $A \rightarrow E \Rightarrow A \rightarrow ABCDE$ (union)

$E \rightarrow A$ et $A \rightarrow ABCDE \Rightarrow E \rightarrow ABCDE$ (transitivité)

$CD \rightarrow E$ et $E \rightarrow ABCDE \Rightarrow CD \rightarrow ABCDE$ (transitivité)

$B \rightarrow D \Rightarrow BC \rightarrow CD$ (augmentation)

$BC \rightarrow CD$ et $CD \rightarrow ABCDE \Rightarrow BC \rightarrow ABCDE$ (transitivité)

- Les clés candidates trouvées sont : A, E, CD, BC

Normalisation (1)

- Processus de décomposition d'une table R en plusieurs tables R1, R2, ..., Rn en se basant sur la notion de dépendance fonctionnelle
- La décomposition se fait sans perte d'information
- La jointure des tables obtenues après la normalisation est équivalente à la table décomposée $R = R1 |X| R2 |X| \dots |X| Rn$

Objectifs :

- Éviter de stocker des données redondantes
- Simplifier la mise à jour des données
- Éviter les problèmes d'incohérences

Normalisation (2)

- Codd (mathématicien et chercheur à IBM) a proposé en 1972 trois formes normales
 - 1NF : première forme normale
 - 2NF : deuxième forme normale
 - 3NF : troisième forme normale
- Puis en 1974, Codd a proposé en collaboration avec Boyce une nouvelle forme normale
 - BCNF : forme normale de Boyce-Codd (3.5NF)

Normalisation (3)

- Avec le temps, d'autres formes normales sont apparues
 - 4NF : quatrième forme normale
 - 5NF : cinquième forme normale
 - PJNF : forme normale à projections jointives
 - DKN : forme normale à domaines-clés
- Ces formes normales traitent de cas particuliers non résolus par les autres formes normales

Normalisation 1NF (1)

- Une relation est en 1NF si tous les attributs sont atomiques

Table : Personne

numero	nom	prenom	age	adresse
1001	Loudad	Yassine	20	120, Nakhil, Rabat, Maroc
1002	Labrini	Fayrouz	30	7, Anfa, Casa, Maroc
1003	Tarek	Laamiri	32	8, Alfath, Rabat, Maroc



adresse n'est pas atomique

Table : Personne

numero	nom	prenom	age	numA
1001	Loudad	Yassine	20	201
1002	Labrini	Fayrouz	30	202
1003	Tarek	Laamiri	32	203

Table : Adresse

numA	numRue	nomRue	ville	pays
201	120	Nakhil	Rabat	Maroc
202	7	Anfa	Casa	Maroc
203	8	Alfath	Rabat	Maroc

Normalisation 1NF (2)

- Une relation est en 1NF si tous les attributs sont atomiques

Avant la normalisation :

Personne (numero, nom, prenom, age, adresse)

Résultat de la normalisation 1NF :

Personne (numero, nom, prenom, age, numA#)

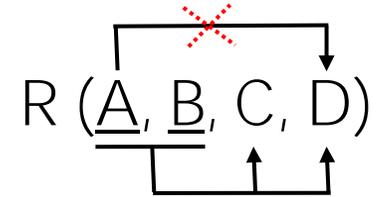
Adresse (numA, numRue, nomRue, ville, pays)

Normalisation 2NF (1)

- Une relation est en 2NF si elle est en 1NF **et** si il n'y a pas de DF entre un sous-ensemble de la clé et des attributs non-clés

Table : AdressePersonne

numP	numA	taxeHab	numId
1001	201	4000	AE33633
1002	202	5000	AA11334
1003	203	3000	FJ45373



numP → numId



DF entre un sous ensemble de la clé et un attribut

Table : AdressePersonne

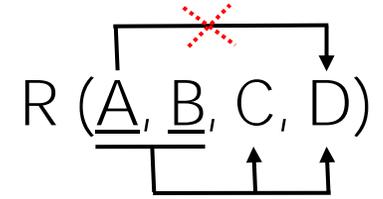
numP	numA	taxeHab
1001	201	4000
1002	202	5000
1003	203	3000

Table : Personne

numP	nom	prenom	age	numId
1001	Loudad	Yassine	20	AE33633
1002	Labrini	Fayrouz	30	AA11334
1003	Tarek	Laamiri	32	FJ45373

Normalisation 2NF (2)

- Une relation est en 2NF si elle est en 1NF **et** si il n'y a pas de DF entre un sous-ensemble de la clé et des attributs non-clés



Avant la normalisation :

AdressePersonne (numP#, numA#, taxeHab, numId)

Résultat de la normalisation 2NF :

AdressePersonne (numP#, numA#, taxeHab)

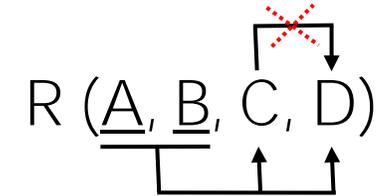
Personne (numP, nom, prenom, age, numId)

Normalisation 3NF (1)

- Une relation est en 3NF si elle est en 2NF **et** si il n'y a pas de DF entre un ensemble non inclus dans la clé et des attributs non-clés

Table : Adresse

numA	numRue	nomRue	ville	pays
201	120	Nakhil	Rabat	Maroc
202	7	Anfa	Casa	Maroc
203	8	Alfath	Rabat	Maroc



ville → pays

← DF entre deux attributs non-clés

Table : AdressePersonne

numA	numRue	nomRue	ville
201	120	Nakhil	Rabat
202	7	Anfa	Casa
203	8	Alfath	Rabat

Table : Ville

ville	pays
Rabat	Maroc
Casa	Maroc

Normalisation 3NF (2)

- Une relation est en 3NF si elle est en 2NF **et** si il n'y a pas de DF entre un ensemble non inclus dans la clé et des attributs non-clés

Avant la normalisation :

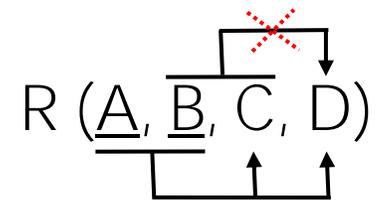
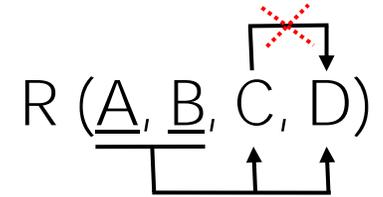
Adresse (numA, numRue, nomRue, ville, pays)

Résultat de la normalisation 3NF :

Adresse (numA, numRue, nomRue, ville#)

Ville (ville, pays)

- Remarque : En 3NF, la DF suivante n'est pas permise



Normalisation BCNF (1)

- Une relation est en BCNF si les seules DF sont celles de la clé qui détermine les attributs non-clés, par conséquent $3NF \subset BCNF$

Table : CoupeDuMonde

paysOrg	annee	vainqueur	villeFinale
Allemagne	1974	Allemagne	Munich
France	1998	France	Paris
Allemagne	2006	Italie	Berlin

villeFinale \rightarrow paysOrg

DF différente de
clé \rightarrow attribut non-clé

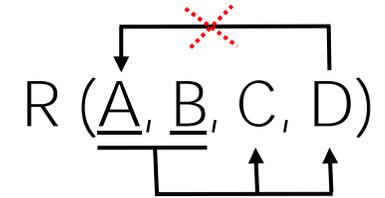


Table : CoupeDuMonde

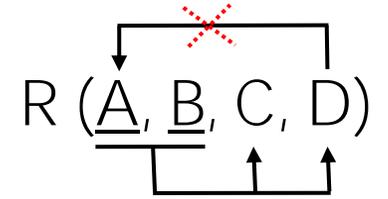
villeFinale	annee	vainqueur
Munich	1974	Allemagne
Paris	1998	France
Berlin	2006	Italie

Table : VilleFinale

villeFinale	paysOrg
Munich	Allemagne
Paris	France
Berlin	Allemagne

Normalisation BCNF (2)

- Une relation est en BCNF si les seules DF sont celles de la clé qui détermine les attributs non-clés, par conséquent $3NF \subset BCNF$



Avant la normalisation :

CoupeDuMonde (paysOrg, annee, vainqueur, villeFinale)

Résultat de la normalisation BCNF :

CoupeDuMonde (villeFinale#, annee, vainqueur)

VilleFinale (villeFinale, paysOrg)

- Remarque : En BCNF, on peut être amené à changer la clé de la relation

Avantages de la normalisation

- **Redondance** réduite
 - moins de données dans la BD (base de données)
 - plus petite est la BD et plus rapides sont les E/S (Entrées/Sorties)
- Petites relations et **petits tuples**
 - plus de tuples peuvent être affichés ou imprimés sur une page
 - plus de tuples peuvent être traités en une même E/S
 - plus de tuples peuvent être mis en mémoire cache
- **Consistance** de la base de donnée
 - Lors de la modification d'une donnée, toutes les répliquions de cette donnée sont mises à jour automatiquement

Inconvénients de la normalisation

- Nombre **élevé** de relations de petites tailles
 - la performance n'est pas optimisée
 - plusieurs jointures pour retrouver l'information
 - les jointures peuvent être coûteuses en temps CPU (E/S)
- Pour cette raison, on a recours à la dénormalisation

Dénormalisation (1)

- La dénormalisation
 - s'effectue en général après la normalisation
 - vise à s'éloigner intentionnellement de la normalisation
 - a pour objectif d'améliorer les **performances** de la BD
 - peut être appliquée sur des attributs ou des relations entières
 - requiert de connaître comment les données seront utilisées
- La dénormalisation peut aussi être utilisé pour ajuster la BD à des applications particulières

Dénormalisation (2)

- Avantages de la dénormalisation
 - minimise le besoin de jointures
 - réduit le nombre de clés étrangères (Foreign Key)
 - réduit le nombre d'index
 - réduit le nombre de relations
- Le choix d'un bon schéma sera toujours un compromis entre la performance de la BD (moins de jointures) et les anomalies de mises à jour (redondance de données)
 - Le concepteur doit faire une étude sur la fréquence des mises à jours de certaines données pour justifier la redondance des données