

NORMALISATION ET FORMES NORMALES

NORMALISATION ET FORMES NORMALES

1. Motivation: Les problèmes liés à la modélisation
2. Définitions
3. Les dépendances fonctionnelles
4. Les Formes normales

Motivation: Les problèmes liés à la modélisation

Exemple1: gestion des Clients inscrits dans des centres sportifs

NUM_CENTRE	NOM_CENTRE	COUT_INSCRI	NUM_CLIENT	PREN_CLIENT	NOM_CLIENT
103	Gym Center	230,00	002	Omar	SHERIFF
107	Sport Center	260,00	002	Omar	SHERIFF
107	Sport Center	260,00	004	Hedi	MUFTI
107	Sport Center	260,00	007	Syn	KONNERY
107	Sport Center	260,00	008	Woody	HELENE

Motivation: Les problèmes liés à la modélisation

Exemple1: gestion des Clients inscrits dans des centres sportifs

NUM_CENTRE	NOM_CENTRE	COUT_INSCRI	NUM_CLIENT	PREN_CLIENT	NOM_CLIENT
103	Gym Center	230,00	002	Omar	SHERIFF
107	Sport Center	260,00	002	Omar	SHERIFF
107	Sport Center	260,00	004	Hedi	MUFTI
107	Sport Center	260,00	007	Syn	KONNERY
107	Sport Center	260,00	008	Woody	HELENE

Problème : Cette table contient des informations redondantes. Elle génère les problèmes de mise à jour suivants :

- **Anomalie de modification** : la modification du coût d'inscription (COUT_INSCRI) pour le centre 107 nécessite la modification de plusieurs lignes au lieu d'une seule.
- **Anomalie de suppression** : la suppression dans la table du client n°002 peut entraîner la perte des informations concernant le centre 103.

Motivation: Les problèmes liés à la modélisation

Exemple : BD représentant un Parc Automobile

NSS	Nom	Prénom	Adr.	NVH	Marque	Type	Puiss	Coul	Date	Prix
16607	Jacques	Paul	Paris	90222275	Renault	Clio	5	Blanc	1/1/01	12 500
24906	Martin	Marie	Lille	23AE59	Renault	Clio	5	Beige	3/4/02	16 000
24906	Martin	Marie	Lille	45UK59	Peugeot	504	7	Gris	2/7/95	10 000
15405	Durand	Olivier	Lille	782AC06	Peugeot	504	7	Vert	12/4/96	11 000
15405	Durand	Olivier	Nice	BMW	BMW	520	10	Gris	11/7/90	12 000
15 405	Durand	Olivier	Nice	Citroen	Citroen	CX	8	Noir	10/1/98	8 000

Motivation: Les problèmes liés à la modélisation

Exemple : BD représentant un Parc Automobile

NSS	Nom	Prénom	Adr.	NVH	Marque	Type	Puiss	Coul	Date	Prix
16607	Jacques	Paul	Paris	902ZZ75	Renault	Clio	5	Blanc	1/1/01	12 500
24906	Martin	Marie	Liège	23AE59	Renault	Clio	5	Beige	3/4/02	16 000
24906	Martin	Marie	Liège	45UK59	Peugeot	504	7	Gris	2/7/95	10 000
15405	Durand	Olivier	Liège	782AC06	Peugeot	504	7	Vert	12/4/96	11 000
15405	Durand	Olivier	Nice	BMW	BMW	520	10	Gris	11/7/90	12 000
15 405	Durand	Olivier	Nice	Citroen	Citroen	CX	8	Noir	10/1/98	8 000

- Redondance des données : Incohérence en mise à jour
- Anomalie d'insertion : Comment représenter les personnes sans véhicule
- Anomalie de suppression : Perte d'information

Motivation: Les problèmes liés à la modélisation

Nécessité d'une "bonne" conception



Appliquer des règles

Solution basée sur les Dépendances Fonctionnelles & Normalisation

Définitions

La **normalisation** est un processus qui permet de s'assurer de la “**Bonne conception**” du Schéma de la BD (c'est à dire de la non redondance de ses données).

L'**objectif** de la normalisation est de limiter **le nombre de redondances** et les **anomalies** lors de la mise à jour de Base de Données.

Définitions

Concepts

- **Les formes normales** définissent des relations bien conçues.
- **Les dépendances fonctionnelles** traduisent des contraintes sur les données.
- **Processus de normalisation** : Décomposer progressivement les relations jusqu'à obtenir des relations normalisées.

Définitions

Concepts

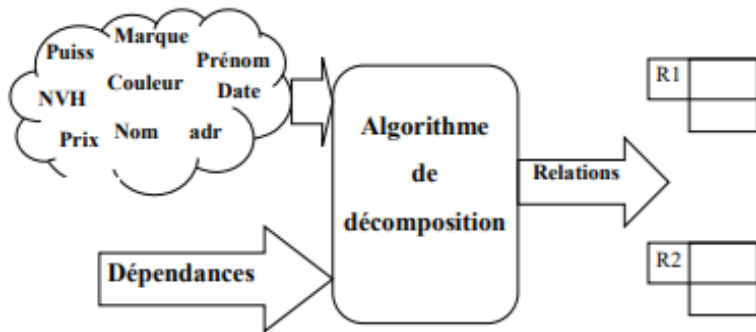
- **Les formes normales** définissent des relations bien conçues.
- **Les dépendances fonctionnelles** traduisent des contraintes sur les données.
- **Processus de normalisation** : Décomposer progressivement les relations jusqu'à obtenir des relations normalisées.

La normalisation

- Consiste à faire **éclater** une relation en **plusieurs relations** plus réduites grâce aux **formes normales**, qui sont basées sur les **dépendances fonctionnelles** entre attributs.

Algorithmes de décomposition

Relation Universelle



Définitions

Les Dépendances Fonctionnelles

Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation et X et Y deux sous-ensembles d'attributs de R .

On dit que $X \rightarrow Y$ (X détermine fonctionnellement Y , ou Y dépend fonctionnellement de X) si, quelle que soit l'extension de R , pour tous tuples t_1 et t_2 de R :

$$t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Ce qui signifie qu'à chaque valeur de X correspond toujours une seule valeur de Y .

Exemples

Etudiant

- $\text{uid} \rightarrow \text{nom}$
- $\text{nom} \not\rightarrow \text{uid}$

- À chaque **uid** correspond un unique étudiant :

$$\{\text{uid}\} \rightarrow \{\text{uid}, \text{login}, \text{nom}, \text{prenom}, \text{groupe}, \text{option}, \text{chef_doption}\}$$

- À chaque **login** correspond un unique étudiant :

$$\{\text{login}\} \rightarrow \{\text{uid}, \text{login}, \text{nom}, \text{prenom}, \text{groupe}, \text{option}, \text{chef_doption}\}$$

- À chaque **option** correspond un seul **chef_doption** :

$$\{\text{option}\} \rightarrow \{\text{chef_doption}\}$$

Exemples

Voiture

- $(\text{marque}, \text{type}) \rightarrow \text{puissance}$
- $\text{marque} \not\rightarrow \text{puissance}$
- $\text{puissance} \not\rightarrow \text{type}$

Propriétaire

- $n \text{ voiture} \rightarrow n \text{ prop}$
- $n \text{ prop} \not\rightarrow n \text{ voiture}$
- $(n \text{ voiture}, n \text{ prop}) \rightarrow \text{date-achat}$

Les Dépendances Fonctionnelles (DF)

Une DF représente une contrainte de la base de données :

- Il ne peut y avoir, à un instant donné, dans une salle qu'un seul cours :

$$\{\text{salle}, \text{date}\} \rightarrow \text{cours}$$

- Un étudiant choisit une et une seule option, il ne peut pas choisir deux options :

$$\text{uid} \rightarrow \text{option}$$

Les Dépendances Fonctionnelles (DF)

Une DF représente une contrainte de la base de données :

- Il ne peut y avoir, à un instant donné, dans une salle qu'un seul cours :

$$\{\text{salle}, \text{date}\} \rightarrow \text{cours}$$

- Un étudiant choisit une et une seule option, il ne peut pas choisir deux options :

$$\text{uid} \rightarrow \text{option}$$

- À chaque **groupe**, **professeur**, **cours** et **date** correspond une unique **salle** :

$$\{\text{groupe}, \text{uidprof}, \text{nomcours}, \text{date}\} \rightarrow \{\text{salle}\}$$

Les Dépendances Fonctionnelles (DF)

Utilisation des DFs pour déterminer les Clés d'une table

L'ensemble des attributs C est une **clé** d'une relation R si et seulement si :

- C détermine fonctionnellement tous les attributs de R .
- Il n'existe pas un sous-ensemble propre de C qui détermine tous les attributs de R .

Les Dépendances Fonctionnelles (DF)

Exemples des clés

- ⊙ Etudiants(uid, login, nom, prenom, groupe, option, chef_doption)

- {uid} → {uid, login, nom, prenom, groupe, option, chef_doption}



- {option} → {chef_doption}

- {login} → {uid, login, nom, prenom, groupe, option, chef_doption}



- Les Clés possibles sont : uid ou bien login

Les Dépendances Fonctionnelles (DF)

Propriétés des Dépendances Fonctionnelles (DF): Axiomes d'amstrong

- la réflexivité : $X \rightarrow X$
- l'augmentation : Si $X \rightarrow Y$, alors $X, Z \rightarrow Y$ (Quelque soit Z)
- la transitivité : Si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$

- la pseudo-transitivité : Si $X \rightarrow Y$ et $Y, T \rightarrow Z$ alors $X, T \rightarrow Z$
- l'union : Si $X \rightarrow Y$ et $X \rightarrow Z$ alors $X \rightarrow Y, Z$
- la décomposition : Si $X \rightarrow Y, Z$ alors $X \rightarrow Y$ et $X \rightarrow Z$

Les Dépendances Fonctionnelles: Dépendance Élémentaire

Dépendance Élémentaire

Une DF $X \rightarrow Y$ est dite **élémentaire** s'il n'existe pas un sous-ensemble Z inclus dans X tel que :

$$Z \rightarrow Y$$

En d'autres termes, il n'y a pas d'attribut superflu dans la partie gauche de la dépendance fonctionnelle.

Les Dépendances Fonctionnelles (DF)

Dépendance Élémentaire Exemple: Cours(groupe, uidprof, nomcours, date, salle)

Les deux DF suivantes sont-elles élémentaires ?

- nomcours \rightarrow uidprof ? 
- {groupe, uidprof, nomcours, date} \rightarrow salle ? **X**

Cette dépendance n'est pas élémentaire car:
{nomcours, date} \rightarrow salle

Les Dépendances Fonctionnelles: Dépendance directe

Dans une relation R , une dépendance fonctionnelle (DF) $X \rightarrow Y$ est **directe** :
*s'il n'existe pas un ensemble d'attributs Z tel que $X \rightarrow Z$ et $Z \rightarrow Y$,
avec Z distinct de X et de Y .*

Exemples : Etant donné le schéma de relation :

Etudiants(uid, login, nom, prenom, groupe, option, chef_doption)

Les DF suivantes sont directes:

$\text{uid} \rightarrow \text{login}$

$\text{uid} \rightarrow \text{nom}$

...

$\text{uid} \rightarrow \text{option}$

$\text{option} \rightarrow \text{chef_doption}$

Les Dépendances Fonctionnelles: Dépendance directe

Dans une relation R , une dépendance fonctionnelle (DF) $X \rightarrow Y$ est **directe** :
s'il n'existe pas un ensemble d'attributs Z tel que $X \rightarrow Z$ et $Z \rightarrow Y$, avec Z distinct de X et de Y .

Exemples : Etant donné le schéma de relation :

Etudiants(uid, login, nom, prenom, groupe, option, chef_doption)

Les DF suivantes sont directes:

uid \rightarrow login

uid \rightarrow nom

...

uid \rightarrow option

option \rightarrow chef_doption

Cependant La DF (**uid \rightarrow chef_doption**) n'est pas directe car :

(b) (**uid \rightarrow option**) et (**option \rightarrow chef_doption**)

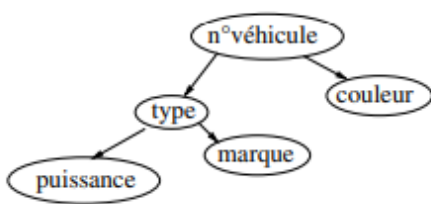
Représentation des dépendances fonctionnelles (DF)

1. **Le graphe de dépendances fonctionnelles** est une représentation graphique de toutes les dépendances fonctionnelles.
Permet de vérifier si une table est normalisée.
2. **Fermeture transitive (F^+)** Ensemble des DF élémentaires augmenté de celles obtenues par transitivité

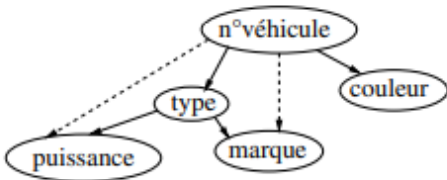
Représentation des dépendances fonctionnelles (DF)

Exemple : voiture(n_véhicule,type,marque,puissance,couleur)

$F = \{n_véhicule \rightarrow type, type \rightarrow marque, type \rightarrow puissance, n_véhicule \rightarrow couleur\}$



Graphe des DF

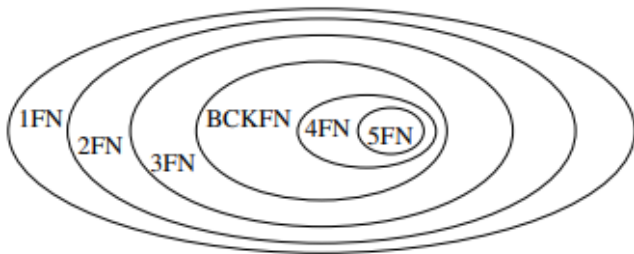


Fermeture transitive

Normalisation des Bases de Données

Normalisation

Consiste à faire **éclater** une relation en plusieurs relations plus réduites grâce aux **formes normales**, qui sont basées sur **les dépendances fonctionnelles** entre attributs.



Normalisation des Bases de Données

La Première Forme Normale

Une relation est dite en **première forme normale (1FN)** :

1. Si chacun de ses attributs dépend fonctionnellement de la clé,
2. et, si tout attribut contient une valeur atomique (non multiple, non composé).

1NF-Exemple 1

Livre (No-ISBN, Titre, Auteurs, Editeur)

- Cette relation n'est **pas en 1FN** car l'attribut Auteurs est multivalué.
- Un auteur ne peut pas y être traité d'une façon individuelle (exemple : tri des livres par nom d'auteur).

Cette relation peut, par exemple, être transformée en la nouvelle relation :

Livre (No-ISBN, Titre, Auteur1, Auteur2, Auteur3, Editeur)

1NF-Exemple 2

Cette relation n'est pas en 1FN car l'attribut Adresse n'est pas atomique

Nom	Adresse	Ville
Yasmine Dali	21 rue de Paris	Sfax, 3000
Ghassen Yahyaoui	45 avenue de la liberté	Sfax, 3004
Fouad Rayane	27 rue des Fleurs	Sfax, 3006



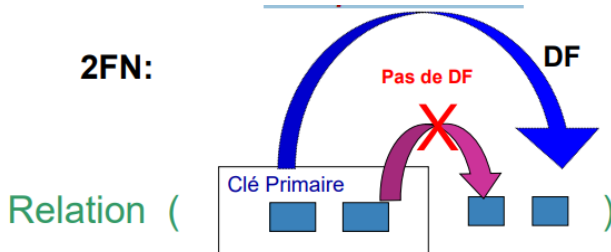
Nom	Prénom	adresse	Ville	Code Postal
Dali	Yasmine	21 rue de Paris	Sfax	3000
Yahyaoui	Ghassen	45 avenue de la liberté	Sfax	3004
Rayane	Rayane	27 rue des Fleurs	Sfax	3006

Normalisation des Bases de Données

La deuxième Forme Normale

Une relation est dite en **deuxième forme normale (2FN)** :

1. Si elle est en **première forme normale (1FN)**,
2. Tout attribut non clé **ne dépend pas d'une partie de la clé**.



2NF-Exemple 1

fournisseur(nom,adresse,article,prix)

DF : { nom,article \rightarrow prix; nom \rightarrow adresse }

- clé = nom,article

- nom \rightarrow adresse

\Rightarrow pas en 2FN

prod_pharma(produit,type,client,remise)

DF : { produit \rightarrow type; type,client \rightarrow remise }

- clé = produit,client

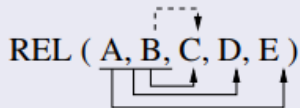
- produit \rightarrow type \Rightarrow pas en 2FN

Règle de décomposition en 2FN

1. Regrouper dans une table les champs dépendant de la totalité de la clé.
2. Regrouper dans une autre table, les champs dépendant d'une la partie de la clé et faire de cette partie la clé de cette nouvelle table

Principe de la décomposition

Sortir la(les) DF qui empêche(nt) que la relation soit en 2FN :



--> DF contradictoire avec la 2FN

R1(B,C)

Reste(A,B,D,E)

La deuxième Forme Normale: Décomposition

La relation **Cours** (groupe, nomcours, date, uidprof, salle) n'est **pas en deuxième forme normale (2FN)**

La deuxième Forme Normale: Décomposition

La relation **Cours** (groupe, nomcours, date, uidprof, salle) n'est **pas en deuxième forme normale (2FN)** car:

- La dépendance fonctionnelle $\{\text{groupe, nomcours, date}\} \rightarrow \text{uidprof}$ n'est pas élémentaire,
- puisque $\text{nomcours} \rightarrow \text{uidprof}$.

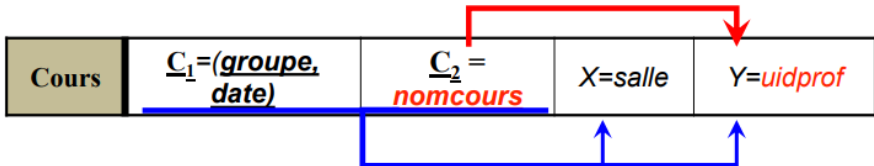
En d'autres termes, l'attribut uidprof ne dépend que d'une partie de la clé (nomcours).

Solution : ?

La deuxième Forme Normale: Décomposition

Exemple de Décomposition

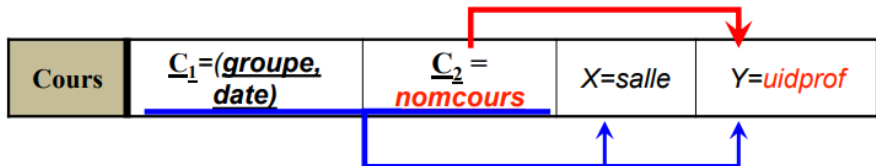
La relation *Cours*(*groupe*, *nomcours*, *date*, *uidprof*, *salle*) n'est pas en deuxième forme normale :



La deuxième Forme Normale: Décomposition

Exemple de Décomposition

La relation *Cours*(groupe, nomcours, date, uidprof, salle) n'est pas en deuxième forme normale :



Solution : décomposer la relation *Cours* en deux relations *Cours_Salle* et *Cours_Prof* suivantes :

1. **Cours_Salle** (groupe, date, nomcours, salle)
2. **Cours_Prof** (nomcours, uidprof)

La deuxième Forme Normale: Décomposition

Exemple de Décomposition

MembreProjet (première forme normale)

<u>NumEmp</u>	<u>NumP</u>	Nom	Ville
E7	P1	Paul	Bulle
E7	P9	Paul	Bulle
E1	P7	Sali	Paris
E1	P11	Sali	Paris
E1	P9	Sali	Paris

<u>NumEmp</u>	Nom	Ville
E7	Paul	Bulle
E1	Sali	Paris

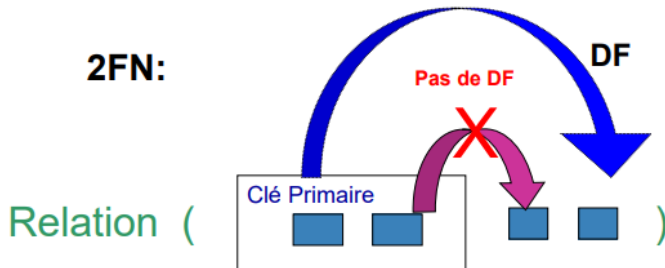
Employe
(2FN)

<u>NumEmp</u>	<u>NumP</u>
E7	P1
E7	P9
E1	P7
E1	P11
E1	P9

Appartenance
(2FN)

- La clé de la table MembreProjet est (NumEmp, NumP). On remarque que les deux attributs non clés (Nom et Ville) sont fonctionnellement dépendants d'une partie de la clé seulement, à savoir NumEmp \rightarrow Nom et NumEmp \rightarrow Ville, donc la relation MembreProjet n'est pas en 2FN
 - Il faut isoler les attributs NumEmp, Nom et Ville dans une relation
 - Il faut isoler les attributs NumEmp et NumP dans une autre relation

La deuxième Forme Normale: Décomposition



Remarques :

- Une relation, en 1FN, dont la clé primaire est composée d'un seul attribut est forcément en 2FN:

-> **Exemple:** Etudiants(uid, login, nom, prenom, groupe, option, chef_doption)

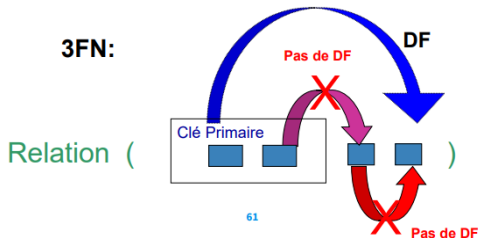
est forcément en 2FN.

Normalisation des Bases de Données

Définition

Une relation est en **troisième forme normale** si

- elle est en 2FN
- il n'y a pas d'attribut n'appartenant pas à la clé qui dépende de la clé par transitivité (*pas de dépendance par transitivité*)
ou il n'existe pas de DF $MG \rightarrow MD$ telle que $MG \cap \text{clé} = \emptyset$



La troisième Forme Normale: Exemple

Exemples:

La relation **Etudiants** ci-dessous n'est pas en **troisième forme normale (3NF)** :

Etudiants(uid, login, nom, prenom, groupe, option, chef_doption)

Pourquoi ?

La troisième Forme Normale: Exemple

Exemples:

La relation Etudiants ci-dessus n'est pas en troisième forme normale car:

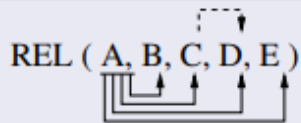
- La clé primaire est uid,
- Les dépendances fonctionnelles suivantes sont vérifiées :
 uid → option
 et option → chef_doption

Donc, la dépendance fonctionnelle entre la clé primaire uid et l'attribut chef_doption **n'est pas directe.**



Règle de décomposition en 3FN

Principe de la décomposition



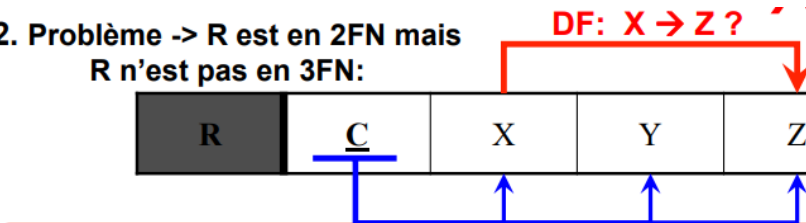
--> DF contradictoire avec la 3FN

R1(C,D)

Reste(A,B,C,E)

La troisième Forme Normale: Décomposition

2. Problème -> R est en 2FN mais
R n'est pas en 3FN:

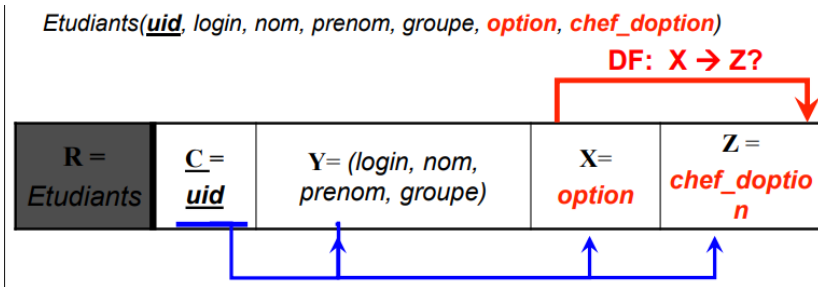


Solution

Une telle relation *R* doit être décomposée en :

- $R_1(C, X, Y)$
- $R_2(X, Z)$

La troisième Forme Normale: Décomposition



car *chef_option* ne dépend pas directement de la clé primaire *uid*, mais d'un attribut non-clé.

La relation **Etudiants** doit être décomposée en deux relations **Etudiants** et **Option** :

- **Etudiants**(*uid*, login, nom, prenom, groupe, *option*)
- **Option**(*option*, *chef_option*)

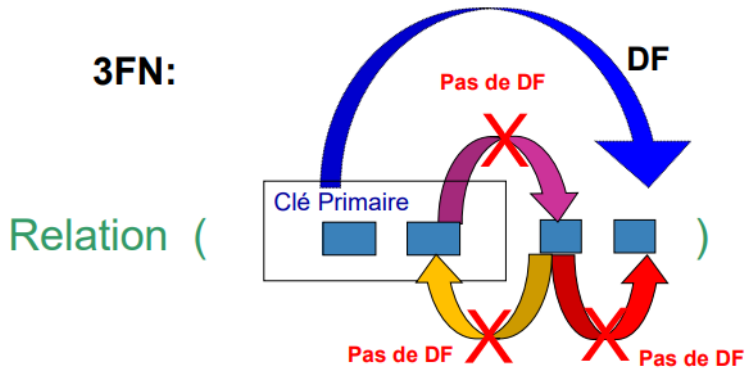
Normalisation des Bases de Données

Forme normale de Boyce Codd (FNBC)

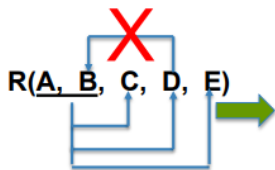
Une relation est en **FNBC** si :

1. Elle est en troisième forme normale (3FN)
2. Aucun attribut membre de la clé ne dépend fonctionnellement d'un attribut non membre de la clé

Normalisation des Bases de Données



Forme normale de Boyce Codd (FNBC)



Une telle relation n'est **pas en FNBC**.

- $D \rightarrow B$: B dépend fonctionnellement de D

Elle peut être décomposée en deux relations :

- $R_1(A, D, C, E)$
- $R_2(D, B)$

Forme normale de Boyce Codd (FNBC)

Comment normaliser en FNBC ?

- Isoler la dépendance fonctionnelle (DF) problématique dans une nouvelle relation.
- Éliminer la **cible** de la DF problématique dans la relation initiale et la remplacer par sa **source**.

Forme normale de Boyce Codd (FNBC)

Soit la relation `Commande(NomF, Produit, AdresseF, Prix)`.

Ses dépendances fonctionnelles (DF) sont :

- $\text{NomF} \rightarrow \text{AdresseF}$
- $\text{Produit} \rightarrow \text{Prix}$

Forme normale de Boyce Codd (FNBC)

La relation Commande n'est **pas en FNBC** car on a la DF $\text{NomF} \rightarrow \text{AdresseF}$

Forme normale de Boyce Codd (FNBC)

Soit la relation Adresse(Ville, Rue, CodePostal).

Ses dépendances fonctionnelles sont :

- $Ville, Rue \rightarrow CodePostal$
- $CodePostal \rightarrow Ville$

Forme normale de Boyce Codd (FNBC)

Soit la relation Adresse(Ville, Rue, CodePostal).

Ses dépendances fonctionnelles sont :

- $Ville, Rue \rightarrow CodePostal$
- $CodePostal \rightarrow Ville$

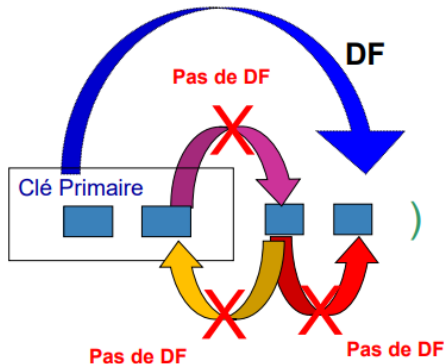
Cette relation n'est **pas en FNBC** car l'attribut Ville (qui fait partie de la clé) dépend fonctionnellement de CodePostal,

Normalisation des Bases de Données

Format d'une relation « bien conçue » :

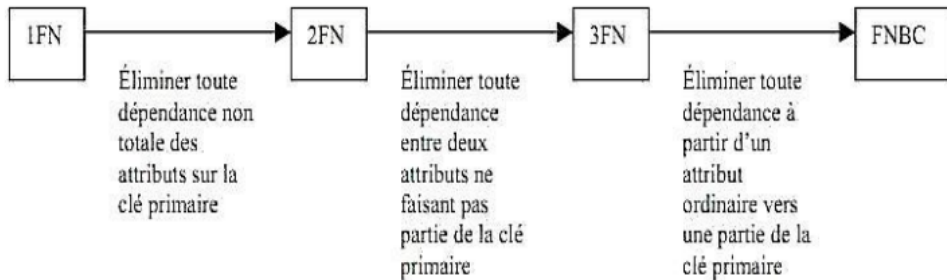


Relation (



Normalisation des Bases de Données

Résumé



Exercices

Exercice1

En quelle forme normale est la relation suivante , qui concerne les employés d'une société implantée sur plusieurs bâtiments?

EMPLOYES (NumE, Nom, Salaire, Département, Bâtiment)

Sachant qu'un employé travaille dans un département donné, et qu'aucun département ne possède des locaux dans plusieurs bâtiments.

Mettre en 3FN cette relation.

Exercices

Les dépendances fonctionnelles (DF) ?

NumE \rightarrow Nom, Salaire, Département

\rightarrow Chaque employé (identifié par son NumE) a un nom, un salaire et est affecté à un département.

Département \rightarrow Bâtiment

\rightarrow Chaque département est localisé dans un seul bâtiment. Il n'existe pas de département réparti sur plusieurs bâtiments

Exercices

Exercice 1: La clé ?

L'ensemble des attributs C est une clé d'une relation R ssi :

- C détermine fonctionnellement tous les attributs de R .
- et il n'existe pas un sous-ensemble propre de C qui détermine tous les attributs de R .

Exercices

Exercice1: 1FN ?

EMPLOYES (NumE, Nom, Salaire, Département, Bâtiment)

Tous les attributs sont simples et non décomposables, donc cette relation est en 1FN

Exercices

Exercice1: **2NF ?**

Oui, car la clé est simple (NumE) et toutes les dépendances sont sur des attributs complets.

Est-elle en 3NF ?

Exercices

Exercice1: **2NF ?**

Oui, car la clé est simple (NumE) et toutes les dépendances sont sur des attributs complets.

Est-elle en 3NF ?

Non, car il existe une dépendance transitive :

NumE \rightarrow Département

Département \rightarrow Bâtiment

donc NumE \rightarrow Bâtiment de façon transitive,
Bâtiment dépend d'un non-clé (Département)

Exercices

Décomposition en 3FN

EMPLOYES (NumE, Nom, Salaire, Département, Bâtiment)

1. EMPLOYES (NumE, Nom, Salaire, Département)
2. DEPARTEMENTS (Département, Bâtiment)

Exercices

Exercice2: Soit la relation "Exemplaire" suivante:

Exemplaire (*code_ouvrage*, *code_édition*, *code_auteur*, *titre*, *prix*, *nbre_pages*, *état*, *date_achat*, *nationalité_auteur*)

Avec les dépendances fonctionnelles suivantes :

- $\text{code_auteur} \rightarrow \text{nationalité_auteur}$
- $\text{code_ouvrage} \rightarrow \text{code_auteur}$
- $\text{code_ouvrage} \rightarrow \text{titre}$
- $(\text{code_ouvrage}, \text{code_édition}) \rightarrow \text{prix}$
- $(\text{code_ouvrage}, \text{code_édition}) \rightarrow \text{nbre_pages}$
- $(\text{code_ouvrage}, \text{code_édition}) \rightarrow \text{état}$
- $(\text{code_ouvrage}, \text{code_édition}) \rightarrow \text{date_achat}$

1. Quelle est la clé possible de cette relation ? Expliquer pourquoi?
2. En quelle forme normale de cette relation ? Expliquer pourquoi?
3. Normaliser cette relation?

Exercices

Exerice2: La clé ?

Exercices

Exerice2: 1FN ?

Exemple (code_ouvrage, code_édition, *code_auteur*, *titre*, *prix*, *nbre_pages*, *état*, *date_achat*, *nationalité_auteur*)

Tous les attributs sont simples et non décomposables, donc cette relation est en 1FN

Exercices

Exerice1: la relation Exempleire est-elle en 2FN ?

Exempleire (code_ouvrage, code_édition, *code_auteur, titre, prix, nbre_pages, état, date_achat, nationalité_auteur*)

Exercices

Exercice2: la relation Cours est-elle en 2FN ?

Exemple (code_ouvrage, code_édition, *code_auteur*, *titre*, *prix*, *nbre_pages*, *état*, *date_achat*, *nationalité_auteur*)

Il existe une dépendance fonctionnelle entre :

- $\text{code_ouvrage} \rightarrow \text{code_auteur}$
- $\text{code_ouvrage} \rightarrow \text{titre}$
- $\text{code_auteur} \rightarrow \text{nationalité_auteur}$

Exercices

Exercice2: la relation Cours est-elle en 2FN ?

Exemple (code_ouvrage, code_édition, *code_auteur*, *titre*, *prix*, *nbre_pages*, *état*, *date_achat*, *nationalité_auteur*)

Il faut décomposer la relation suivante :

- **R1**(code_ouvrage, *titre*, *code_auteur*, *nationalité_auteur*)
- **R2**(code_ouvrage, code_édition, *prix*, *nbre_pages*, *état*, *date_achat*)

Exercices

La relation R1 est-elle en 3FN ?

- Vérifier la normalisation des relations ci-dessus.

Exercices

R1(code_ouvrage, titre, code_auteur, nationalité_auteur)

La relation R1 **n'est pas en 3ème forme normale (3FN)** car :

- $\text{code_ouvrage} \rightarrow \{ \text{nationalité_auteur} \}$ n'est pas une dépendance fonctionnelle directe.

La relation présente des anomalies de mise à jour.

Exercices

Décomposition pour obtenir la 3FN

R1(code_ouvrage, titre, code_auteur, nationalité_auteur)

doit être décomposée en deux relations R1_1 et R1_2 afin d'éliminer les redondances :

- *R1_1*(code_ouvrage, *titre*, *code_auteur*)
- *R1_2*(code_auteur, *nationalité_auteur*)

Les dépendances multivalées

Insuffisances des DFs

- La notion de DF associée à 3NF et BCNF est insuffisante pour éliminer les redondances et les anomalies de mise à jour

Exemple : informations sur les cours d'un étudiant

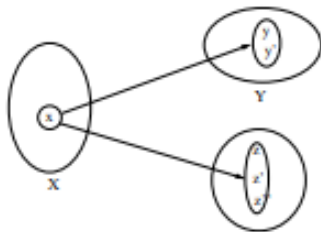
n_étudiant	cours	sport
100	BD	tennis
100	BD	football
102	BD	velo
102	RS	velo

- Pas de DF :
n_étudiant \nrightarrow cours
n_étudiant \nrightarrow sport

Les dépendances multivaluées

Définition

- Soit $R(A_1, A_2, \dots, A_n)$ et X et Y deux sous-ensembles de A_1, A_2, \dots, A_n .
- On dit que $X \twoheadrightarrow Y$ (X **multidétermine** Y , ou il y a une **dépendance multivaluée** de Y sur X) si, étant donné une valeur de X , il y a un ensemble de valeurs de Y associées et cet ensemble est indépendant de $Z = R - (X, Y)$.



Les dépendances multivaluées

Une dépendance multivaluée est une généralisation de la **dépendance fonctionnelle**: pour un même X , les valeurs de Y sont **indépendantes** des autres attributs non contenus dans X ou Y

- **Dépendance multivaluée triviale**

1. $X \twoheadrightarrow Y$ avec $X \cup Y = R$
2. $X \twoheadrightarrow Y$ avec $Y \subseteq X$

La Quatrième Forme Normale: Exemple

Une relation R est dite en 4 forme normale (4NF) si :

1. Elle est déjà en **BCNF (Boyce-Codd)**,
2. et toute **dépendance multivaluée non triviale** dans cette relation est basée sur **une clé**.

La Quatrième Forme Normale: Exemple

Exemple : informations sur les cours d'un étudiant

n_étudiant	cours	sport
100	BD	tennis
100	BD	football
102	BD	velo
102	RS	velo

Dépendances fonctionnelles ?

Il n'y a pas de dépendance fonctionnelle entre *cours* et *sports*.

Clé primaire de cette relation :

La clé primaire est num_étudiant, cours, sports

car il faut connaître ensemble l'étudiant, le cours et le sport pour identifier une ligne de la table.

La Quatrième Forme Normale: Exemple

Dépendances multivaluées ?

Il existe deux dépendances multivaluées non triviales :

- $\text{num_etudiant} \twoheadrightarrow \text{cours}$
(un étudiant peut suivre plusieurs cours, indépendamment des sports)
- $\text{num_etudiant} \twoheadrightarrow \text{sports}$
(un étudiant peut pratiquer plusieurs sports, indépendamment des cours)

Et comme :

- $\text{cours} \neq \text{sports}$
- $\text{cours} \not\subseteq \text{num_etudiant}$
- $\text{sports} \not\subseteq \text{num_etudiant}$
- $\text{num_etudiant} \cup \text{cours} \neq R$

⇒ Ces dépendances ne sont pas triviales.

La Quatrième Forme Normale: Exemple

La relation n'est pas en 4^{ème} forme normale

Car elle contient des dépendances multivaluées non triviales, où le déterminant *num_etudiant* n'est pas une clé candidate de la relation complète.

Décomposition en 4FN :

On décompose la relation $R(\text{num_etudiant}, \text{cours}, \text{sports})$ en deux relations sans perte d'information :

- $R_1(\text{num_etudiant}, \text{cours})$
- $R_2(\text{num_etudiant}, \text{sports})$

La Quatrième Forme Normale: Exemple

Département	Projet	Tâche
D1	P1	T1
D1	P1	T2
D1	P2	T1
D1	P2	T2
D2	P3	T2
D2	P3	T4
D2	P4	T2
D2	P4	T4
D2	P5	T2
D2	P5	T4
D2	P5	T5

Département $\rightarrow\rightarrow$ projets

Département $\rightarrow\rightarrow$ tâches

La Quatrième Forme Normale: Exemple

Département	Projet
D1	P1
D1	P2
D2	P3
D2	P4
D2	P5
D3	P2



Département	Tâche
D1	T1
D1	T2
D2	T2
D2	T4
D3	T5
D3	T6

La cinquième Forme Normale

Définition

La cinquième forme normale interdit de décomposer une relation si elle implique une perte d'information.

Imaginons qu'une marque fabrique différents produits. Et qu'une marque possède différents sites, chaque site fabriquant certains produits de la marque.

La cinquième Forme Normale

Table Production

Brand Id#	Product Id#
1	1
1	2
2	2
2	4
3	5

Table Site

Brand Id#	Site#
1	Dijon
1	Nantes
2	Fontainebleau
3	Orléans
3	Toulouse

La cinquième Forme Normale

Chaque site fabrique certains produits.

La table Site possède une dépendance avec Product Id. Avec cette conception, la relation entre l'entité Site et l'entité Product disparaît.

Il serait préférable de créer une même table identifiée par la marque, le produit et le site.

Table Production

Brand Id#	Product Id#	Site#
1	1	Dijon
1	2	Nantes
2	2	Fontainebleau
2	4	Fontainebleau
3	5	Orléans
3	5	Toulouse

Cette forme normale permet d'éviter une perte de données.

La normalisation-Conclusion

- La normalisation n'est pas une finalité, c'est un outil.
- **Objectif:** Eliminer les redondances d'informations et les risques d'anomalies en mise à jour
- **Approche générale:**
 - Algorithme de synthèse: décomposition en 3NF sans perte et préservant les DF
 - En cas d'insuffisance (le résultat n'est pasq en BCNF ou 4NF), appliquer des décompositions supplémentaires vers BCNF ou 4NF (risque de perte de DF et on garde une décomposition sans perte d'information)

LE MODÈLE RELATIONNEL ET SON ALGÈBRE (partie 2)

Algèbre relationnelle

- L'algèbre relationnelle est un support mathématique cohérent sur lequel repose le modèle relationnel.
- Le but de décrire les opérations qu'il est possible d'appliquer sur des relations pour produire de nouvelles relations.

Algèbre relationnelle

L'algèbre relationnelle est composée de plusieurs opérations sur des tables.

Les plus connues sont :

- la projection (garder des colonnes à partir d'une table) ;
- la sélection (restriction) (garder des lignes à partir d'une table) ;
- l'union de 2 tables ;
- l'intersection entre 2 tables ;
- la différence entre 2 tables ;
- le produit cartésien entre 2 tables (croiser 2 tables) ;
- la jointure entre 2 tables (rapprocher 2 tables sur des critères) ;
- Etc....

Des opérateurs :

- le calcul (minimum, maximum, moyenne, somme, comptage) dans une table ;
- le tri des données etc....

Le langage SQL (prochainement) : est le langage qui permet d'exprimer ces opérations.

Projection

Définition

*La projection de la relation R1 sur l'ensemble des attributs A:
est une relation R dont les tuples sont obtenus:*

- 1. par élimination des valeurs des attributs de R1 n'appartenant pas à A*
- 2. et par suppression des tuples en double.*

L'objectif de cette opération est de supprimer des attributs de la relation R1.

Notation:

$$R = \text{Projection} (R1, \{ A \})$$

$$\Pi_{[A]} (R1)$$

Projection

1.1. Exemple de la projection

$\text{CODES} = \text{Projection} (\text{CLIENTS}, \{ \text{CodeClient} \})$

$\text{CODES} = \Pi_{[\text{CodeClient}]} (\text{CLIENTS})$

CLIENTS

CodeClient	NomClient	AdresseClient
007	Boufarès	TUNIS
005	Le Bon	KAIROUAN
003	Faouzi	KAIROUAN



CODES

CodeClient
007
005
003

En SQL : `SELECT CodeClient FROM Clients ;`

Sélection

Définition:

La sélection dans la relation R1, selon un critère C:

- *donne une relation R, de même schéma:*
- *Les tuples de R sont ceux de R1 qui vérifient la condition logique C (le critère de sélection).*

Notation:

$R = \text{Sélection} (R1, \{ C \})$

$\sigma_{[C]} (R1)$

Exemple de la sélection

LESKAIROUANAIS = Sélection (CLIENTS, { AdresseClient = 'KAIROUAN'})

$\sigma_{[AdresseClient = 'KAIROUAN']}$ (CLIENTS)

CLIENTS

CodeClient	NomClient	AdresseClient
007	Boufarès	TUNIS
005	Le Bon	KAIROUAN
003	Faouzi	KAIROUAN



LESKAIROUANAIS

CodeClient	NomClient	AdresseClient
005	Le Bon	KAIROUAN
003	Faouzi	KAIROUAN

En SQL : *SELECT CodeClient, Nomclient, AdresseClient*
FROM Clients WHERE AdresseClient = 'KAIROUAN' ;

Le critère de sélection

Attribut	Opérateur	Valeur
Un attribut de la relation	<, <=, >, >=, =, <>	Une constante selon le type de donnée : Numérique, Caractère, Date, Logique ...
VILLE	=	"KAIROUAN" 'KAIROUAN'
DATENAISSANCE	>	#01/01/2004# '01-JAN-2004'
COUT	<=	270,00

Attribut	Opérateur	Attribut
Un attribut de la relation	<, <=, >, >=, =, <>	Un (autre) attribut de la relation
VILLEDEPART	<>	VILLEARRIVEE
DATELIVRAISON	>=	DATEDECOMMANDE

L'Union

Définition

L'union de deux relations $R1$ et $R2$ compatibles:

- est une relation R de même schéma*
- contenant l'ensemble des tuples appartenant à $R1$ ou $R2$ ou aux deux relations.*

Notation:

$$R = \text{Union} (R1, R2)$$

$$R1 \cup R2$$

Exemple de l'Union

ROUGEouSEP = Union (ART_ROUGE, ART_SEPTEMBRE)

ART_ROUGE \cup ART_SEPTEMBRE

ART_ROUGE	ART_SEPTEMBRE		ROUGEouSE
CodeArticle	CodeArticle •		^P CodeArticle
A100	FB50		A100
A200	FB51		A200
FB50			FB50
			FB51

En SQL : *SELECT CodeArticle FROM ART_ROUGE*

UNION



SELECT CodeArticle FROM ART_SEPTEMBRE ;

L'Intersection

L'intersection de deux relations $R1$ et $R2$ compatibles:

- est une relation R de même schéma*
- contenant les tuples appartenant à la fois à $R1$ et $R2$.*

Notation

$R = \text{Intersection} (R1, R2)$

$R1 \cap R2$

Exemple de l'Intersection

ROUGEetSEP = Intersection (ART_ROUGE, ART_SEPTEMBRE)

$\text{ART_ROUGE} \cap \text{ART_SEPTEMBRE}$

ART_ROUGE

CodeArticle
A100
A200
FB50

ART_SEPTEMBRE

CodeArticle
FB50
FB51



ROUGEetSEP

CodeArticle
FB50

En SQL : *SELECT CodeArticle FROM ART_ROUGE*
INTERSECT
SELECT CodeArticle FROM ART_SEPTEMBRE ;

Le produit Cartésien

Définition

Le produit cartésien de deux relations $R1$ et $R2$ (de schémas quelconques):

- est une relation R définie sur l'union (la concaténation) des attributs de $R1$ et de $R2$.*
- Les tuples de R sont toutes les concaténations d'un tuple de $R1$ à un tuple de $R2$.*

Notation:

$$R = \text{Produit} (R1, R2)$$

Exemple du produit Cartésien

$R = \text{Produit} (R1, R2)$

$R1 \times R2$

R1

A1	A2
a	17
b	10
c	58

R2

B1	B2	B3
16	10	96
22	11	69



R

A1	A2	B1	B2	B3
a	17	16	10	96
a	17	22	11	69
b	10	16	10	96
b	10	22	11	69
c	58	16	10	96
c	58	22	11	69

*En SQL : `SELECT * FROM R1, R2 ;`*

La jointure

Définition

*La jointure de deux relations $R1$ et $R2$ selon **une condition Q** est l'ensemble des tuples du produit cartésien $R1 \times R2$ satisfaisant à cette **condition Q** .*

Notation:

$$R = \text{Jointure} (R1, R2, \{ Q \})$$

$$R1 \times_{[Q]} R2$$

Exemple de la Jointure

$R = \text{Jointure} (R1, R2, \{ R1.A2 = R2.B2 \})$

$R1 \bowtie_{[R1.A2=R2.B2]} R2$

R1

A1	A2
a	17
b	10
c	58

R2

B1	B2	B3
16	10	96
22	11	69



R

A1	A2	B1	B2	B3
a	17	16	10	96
b	10	22	11	69

*En SQL : `SELECT * FROM R1, R2 WHERE R1.A2 = R2.B2;`*

References

- H. Garcia Molina, J.D. Ullman et J. Widom: Database Systems - The Complete Book, Prentice Hall, 2002
- R.A. El Masri et S.B. Navathe : Fundamentals of Database Systems, Fourth Edition; Prentice Hall C.J. Date : An introduction to Database Systems; Pearson Education 2004
- Ala Eddine Ben Salem, Bases de données (BD), Institut Supérieur d'Informatique et de Mathématique de Monastir, Licence Informatique