

Normalisation de MCD

Normalisation

- Les règles de normalisation permettent de concevoir un schéma de base de données correct :
 - sans redondance d'information.
 - sans anomalie de mise à jour.
- Elles se basent sur
 - les **dépendances fonctionnelles** (DF) qui traduisent les relations entre les données.
 - les **formes normales** qui définissent les relations bien conçues.

Normalisation

L'intérêt de la normalisation

- voyons les problèmes que peuvent poser l'utilisation d'une représentation non normalisée.
- Soit le schéma entité:

FOURNISSEUR (NomFournisseur, AdresseFournisseur, Produits, Prix)

- On peut distinguer les occurrences suivantes des attributs de cette entité :

NomFournisseur	AdresseFournisseur	Produit	Prix
Ali	10 , Rue des Gras - Fès	Chaise	20
		Table	35
Said	86 , Rue neuve - Rabat	Bureau	60
Taher	26 , Rue des Dômes - Méknes	Lit	50
Said	39, Rue des Buttes - Fès	Lampe	18
		Table de chevet	25

Normalisation

1° problème :

Il n'y a pas de clé primaire : on ne sait pas si les deux **Said** sont différents ou pas (si c'est le même **Said**, il y a une des deux adresses qui est fausse).

2° problème :

L'adresse n'est pas décomposée. Si on veut par exemple rechercher tous les fournisseurs qui habitent la même ville, ça ne va pas être possible

3° problème :

Une relation (table) correspondant à ce schéma pourra éventuellement contenir plusieurs produits pour un même fournisseur.

Dans ce cas, il faudra faire face à un certain nombre de problèmes :

- l'adresse du fournisseur sera dupliquée dans chaque occurrence (redondance),
- si on souhaite modifier l'adresse d'un fournisseur, il faudra rechercher et mettre à jour toutes les occurrences correspondant à ce fournisseur,
- si on insère un nouveau produit pour un fournisseur déjà référencé, il faudra vérifier que l'adresse est identique,
- si on veut supprimer un fournisseur, il faudra retrouver et supprimer tous les occurrences correspondantes à ce fournisseur (pour différents produits) dans la table.

Normalisation

La normalisation **élimine les redondances**, ce qui permet:

- une diminution de la **taille de la base de donnée sur le disque**
- une diminution des **risques d'incohérence**
- d'éviter une **mise à jour multiple des mêmes données**

Normalisation

- Il existe plusieurs niveaux de normalisation :
 - Première forme normale (1FN)
 - Deuxième forme normale (2FN)
 - Troisième forme normale (3FN)
- ...
- Un modèle relationnel est dit **normalisé** quand toutes ses tables sont en **3FN**.

Première forme normale (1FN)

Une entité est normalisée en *première forme normale* si :

- 1) elle possède une clé qui identifie formellement chaque occurrence
- 2) chaque attribut dépend fonctionnellement de la clé
- 3) chaque attribut ne peut avoir qu'une seule valeur par enregistrement
- 4) aucun attribut n'est décomposable en plusieurs attributs

Première forme normale (1FN)

Exemple :

- **CLIENT (nom-client, adresse)**
- Cette entité n'est pas en 1FN, car il n'y a pas de clé (plusieurs clients peuvent avoir le même nom).
- D'autre part adresse-client est sans doute la concaténation de rue et ville et ne constitue alors pas une propriété élémentaire.
- Il doit être : **CLIENT (code_client, nom, rue, ville)**

Deuxième forme normale (2FN)

Une entité est en *deuxième forme normale* si et seulement si :

- elle est en 1FN
- toute propriété de l'entité doit dépendre de la clé par **une dépendance fonctionnelle élémentaire**. Autrement dit toute propriété de l'entité doit dépendre de tout l'identifiant.
- *conséquence* : toutes les entités qui n'ont qu'un seul attribut clé, sont en 2FN si elles sont en 1FN.

Deuxième forme normale (2FN)

Exemple :

- **LIGNE(num_commande+num_article,description_article, quantité_commandée)**
- L'identifiant de cette entité est la concaténation de **num_commande** et **num_article**.
- Cette entité n'est pas en 2FN car la dépendance fonctionnelle **num_commande+num article → description_article** n'est pas élémentaire, puisque « description_article » ne dépend que d'une partie de la clé « num_article »
- *Elle doit être :*
 - **LIGNE (num_commande)**
 - **Liée par la relation « Concerne (quantité) »**
 - **à l'entité ARTICLE (num_article, description_article)**

Troisième forme normale (3FN)

Une entité est en *3^o forme normale* si et seulement si :

- elle est en 2^o forme normale
- toute propriété de l'entité doit dépendre de l'identifiant par **une dépendance fonctionnelle élémentaire directe**, c'est-à-dire qu'aucun attribut ne doit dépendre de la clé par transitivité.

Troisième forme normale (3FN)

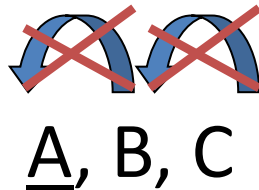
Exemple :

- CLIENT (code_client, nom_client, code_categ, nom_categ)
- Cette relation n'est pas en 3FN car la dépendance fonctionnelle **code_client** \rightarrow **nom_categ** n'est pas directe du fait de la transitivité **code_client** \rightarrow **code_categ** \rightarrow **nom_categ**
- Ça doit être :
 - CLIENT (code_client, nom_client)
 - Lié par la relation « appartient à » à l'entité:
 - CATEGORIE (code_categ, nom_categ)

Forme normale de Boyce-Codd

Définition : Boyce-Codd forme normale:

- Une relation est en **BCFN** si :
 - elle est en 3^{ème} forme normale ;
 - aucun attribut ne dépend d'un attribut non clé.
- C.a.d. toute DF a pour partie gauche (origine de la flèche) une clé candidate ou primaire entière.



Forme normale de Boyce-Codd

Considérons la relation suivante:

Cours (Matiere, Classe, Professeur)

complétée par les règles de gestion suivantes:

un professeur n'enseigne qu'une seule matière,

une classe n'a qu'un seul enseignant par matière

desquelles on déduit les DF suivantes:

Matière, Classe → Professeur

Professeur → Matière

- Cette relation **est en 3NF**, néanmoins il est impossible d'enregistrer un professeur sans classe affectée (puisque classe est une partie de la clé primaire de la table Cours),
- et la disparition d'une classe peut entraîner la disparition de professeur.

Forme normale de Boyce-Codd

On pourrait alors décomposer la table Cours en 2 tables:

Spécialité (Professeur, Matière)

Enseignant (Classe, Professeur)

Mais la première DF serait alors perdue...

Normalisation : 2ième exemple FNBC

Considérons la relation suivante:

Fournisseur (nom, adresse, produit, prix)

avec les DF suivantes :

DF1: nom \rightarrow adresse

DF2: nom, produit \rightarrow prix

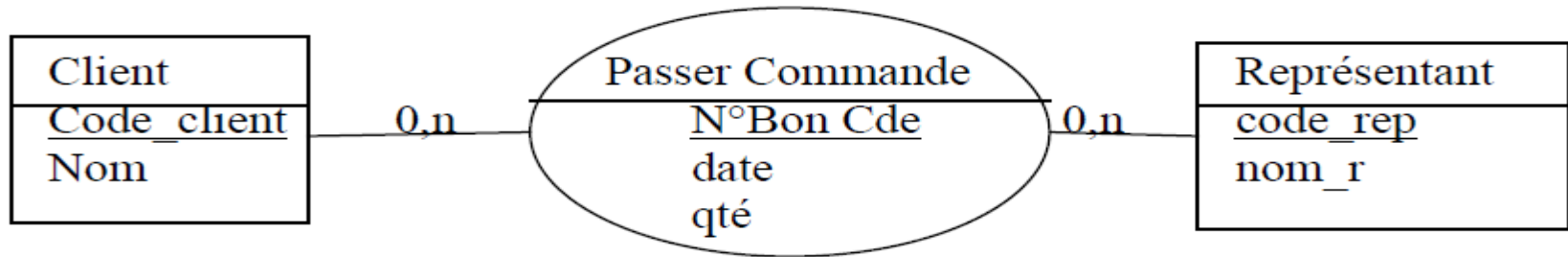
Alors la clé de la relation Fournisseur est [nom, produit].

Cette relation **n'est pas en FNBC**, car dans la DF1, la partie gauche "nom" n'est pas une clé entière.

Exercices

Exercice 4 :

Soit le MCD suivant:



Donner une version normalisée de ce modèle.

Exercices

Exercice 5 :

Critiquer un MCD qui comporterait l'entité suivante :

Stock
<u>Réf-Produit, N°-Magasin</u> Désignation Stock

Sachant qu'un produit peut être en stock dans plusieurs magasins

Proposer un MCD normalisé