

3ème Partie :

La normalisation des bases de données

Objectifs de la partie :

Définition de la normalisation, son intérêt ainsi que les 3 formes normales

Normalisation. Principe et Objectifs

- i. Normaliser une base = imposer des règles de construction à sa structure afin de respecter la cohérence des données et éviter toute redondance d'informations.
- ii. L'objectif de la normalisation = construire un schéma de bases de données cohérent. Pour qu'un modèle relationnel soit normalisé, il faut respecter certaines contraintes appelées **Formes Normales**. Ces FN s'appuient sur des Dépendances Fonctionnelles (DF).

Exemple. Enoncé

Soit le schéma de relation

FOURNISSEUR (NomFournisseur, AdresseFournisseur, Produits, Prix)

Modèle en extension

NomFournisseur	AdresseFournisseur	Produit	Prix
Lebras	10, Rue des Gras - Clermont	Chaise	20
		Table	35
Dupont	86, Rue de la République - Moulins	Bureau	60
Lajoie	26, Rue des Dômes - Vichy	Lit	50
Dupont	39, Rue des Buttes - Moulins	Lampe	18
		Table de chevet	25

Pour vous montrer l'intérêt de la normalisation d'une BDR, commencez par détecter les problèmes que peuvent poser l'utilisation d'une BD basée sur ce modèle relationnel non normalisé.

Exemple. Réponse

(1/2)

1^o problème :

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

2^o 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

Exemple. Réponse

(2/2)

3^e 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 n-uplet (redondance),
- ✓ si on souhaite modifier l'adresse d'un fournisseur, il faudra rechercher et mettre à jour tous les n-uplets 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 n-uplets correspondant à ce fournisseur (pour différents produits) dans la table.

Exemple. Conclusion

La normalisation élimine les redondances, ce qui permet :

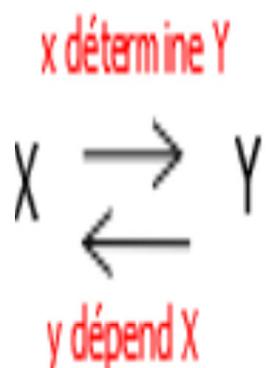
- ✓ Une diminution de la taille de la base de donnée sur le disque
- ✓ Une diminution des risques d'incohérence
- ✓ D'Éviter une mise à jour multiple des mêmes données

Normalisation. Dépendance Fonctionnelle : Définition

Définition :

B est en dépendance fonctionnelle de a si à une valeur quelconque de la propriété a on ne peut faire correspondre qu'une seule valeur au plus de la propriété b.

Si on connaît la valeur de a, on peut en déduire une seule valeur de b (le contraire n'est pas vrai)



Normalisation. Dépendance Fonctionnelle : Exemple

Exemple de DF: $\text{Num_client} \rightarrow \text{Nom_client}$

- i. Il existe une DF entre le numéro client et son nom.
- ii. Si on connaît une valeur de la propriété « num_client » (ex: 127), on ne peut lui correspondre qu'une seule valeur de la propriété « nom_client » (ex: Bensalah ali)

Normalisation. Les formes normales (1^{ère} FN)

(1/3)

1^{ère} FN: Il s'agit de vérifier si tous les attributs sont élémentaires + mono valués

Exemple1. Attributs composés :

1^{er} Cas : Si la personne a une seule adresse

Personnes(Matricule, nom, prenom, adresse)

Rendre les attributs composés -> élémentaires

→ Personnes(Matricule, nom, prenom, NumRue, Rue, Ville, CP)

Normalisation. Les formes normales (1^{ère} FN)

(2/3)

2^{ème} Cas: Si la personne peut avoir plusieurs Adresses

PERSONNES (MATRICULE, nom, prenom, Adresses)

↳ PERSONNES (MATRICULE, nom, prenom)

ADRESSES (IdAdresse, Numero, Rue, codePostal, Ville, Matricule#)

Normalisation. Les formes normales (1^{ère} FN)

(3/3)

Exemple2. Attributs multivalués :

PERSONNES (MATRICULE, nom, prenom, Diplomes)

↳ PERSONNES (MATRICULE, nom, prenom)

DIPLOMES (IdDiplome, description, Matricule# ,

Normalisation. Les formes normales (2^{ème} FN) (1/2)

2^{ème} FN: Une relation est en 2FN si elle est en 1FN et de plus, si les attributs n'appartenant pas à la clé primaire ne dépendent pas fonctionnellement d'une partie de la clé.

Exemple : Gestion de compte bancaire

OPERATION (N°Compte, CodeOpe, DateOpe, Nom, Prenom, LibelOpe, Somme)

On note que :

- Nom et Prénom dépendent fonctionnellement de N°Compte
- Libellé d'opération dépend fonctionnellement de Code opération

Normalisation. Les formes normales (2^{ème} FN) (2/2)

Correction - Exemple : Gestion de compte bancaire

On va obtenir les relations suivantes :

COMPTE(N°Compte, Nom, Prénom)

LIBELLE(CodeOpe, LibelOpe)

OPERATION(#N°Compte, DateOpe, #CodeOpe, somme)

Normalisation. Les formes normales (3^{ème} FN) (1/4)

3^{ème} FN: Une relation est en 3FN si elle est en 2FN et de plus, tout attribut non clé ne dépend pas fonctionnellement d'un autre attribut non clé.

Exemple 1 : l'organisme de formation

Soit la relation:

ADHERENT(CodeAdh,NomAdh,AdressAdh,TypAdh,CotisTyp,NbSaladh)

→ On impose que la cotisation de l'adhérent dépend fonctionnellement du type de l'adhérent. ($\text{TypAdh} \rightarrow \text{CotisTyp}$)

Normalisation. Les formes normales (3^{ème} FN) (2/4)

Correction – Exemple 1 : l'organisme de formation

TYPE(TypAadh, CotisAadh)

ADHERENT(CodeAadh, NomAadh, AdressAadh, #TypAadh, NbSaladh)

Normalisation. Les formes normales (3^{ème} FN) (3/4)

Exemple 2 : l'organisation des salariés

Soit la relation:

SALARIES(num_salarie, nom, date_naissance, num_service, nom_service)

Et la dépendance suivante:

num_service → nom_service

Normalisation. Les formes normales (3^{ème} FN) (4/4)

Correction - Exemple 2 : l'organisation des salariés

SALARIES (Num_salarié, nom, date_naissance, #num_service)

SERVICES (Num_service, nom_service)

Exercice. Enoncé

Soit la relation suivante:

SOIGNER (code medecin, code malade, nom_medecin,
nom_malade, date , adresse)

On note que:

code medecin → nom_medecin

code malade → nom_malade

Chaque médecin a un seul cabinet

Question : Mettre la relation en 3FN.

Exercice. Solution

Medecin (code_medecin, nom_medecin)

Malade (code_malade, nom_malade)

Soigner (code_malade#, code_medecin#, date , num,
nom_rue, ville, CP)