SHAKE THE FUTURE

CENTRALE NANTES

Bases de Données

Formes Normales

JY Martin

Plan

Construire une base de données

But d'un schéma logique : décrire une base de données qui va effectivement

- Etre utilisée
- Chargée,
- Accédée,
- Mise à jour

Problème : comment s'assurer que lors de la conception, on n'introduit pas des éléments qui poseront des soucis lors de ces opérations ?

D'où peut venir le problème ?

Les mises-à-jour (insertions, suppressions, modifications) doivent conserver la cohérence de la base de données

- intégrité référentielle
- toute contrainte d'intégrité
- en particulier les dépendances entre attributs

Selon le schéma c'est plus ou moins facile
Plus la base de données contient de redondances, plus les mise-à-jour avec maintien de la cohérence est difficile.

Exemple de table

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité

Le fournisseur **NFourn**, qui est actuellement à l'adresse **AdrFourn**, a livré au total une **quantité** du produit **NProduit** de type **typeProduit**.

Exemple de contenu de la table

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité
3	Nantes	52	Meuble	12
22	Paris	10	Ordinateur	6
22	Paris	25	Papier	200
3	Nantes	25	Papier	500
3	Angers	10	Portable	15

- Le fournisseur 3 a changé d'adresse. En cas de requête sur ce fournisseur, quelle adresse fournit-on ?
- La référence de produit 10 a changé. En cas de statistiques sur le nombre de portables, si on utilise le libellé ou le numéro de produit on n'obtient pas le même résultat
- Impossible de créer un fournisseur s'il n'a pas livré quelque chose.

Le problème

Une relation / une base de données n'est pas correcte si :

- elle implique des répétitions au niveau de sa population
- elle pose des problèmes lors des mises-à-jour (insertions, modifications et suppressions)

Les conditions pour qu'une relation / une base de données soit correcte peuvent être définies formellement :
=> règles de normalisation

Ce qui ne va pas ...

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité

- L'adresse du fournisseur ne dépend que du fournisseur et pas du produit.
- Le type du produit ne dépend que du produit et pas du fournisseur

Cette relation n'est pas correcte. Il faut la normaliser.

Plan

la Normalisation

Définition

Processus de transformation d'un schéma pour obtenir un nouveau schéma - qui est équivalent (même contenu) - dont les mises-à-jour assurant la cohérence de la base de données sont simples

Exemple

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité
3	Nantes	52	Meuble	12
22	Paris	10	Ordinateur	6
22	Paris	25	Papier	200
3	Nantes	25	Papier	500
3	Angers	10	Portable	15

Un changement élémentaire dans le monde réel se traduit par une mise à jour d'un unique tuple :

- La quantité totale pour un produit et un fournisseur est mise à jour
  => 1 tuple à mettre à jour. OK.
- Un fournisseur change d'adresse
  => N tuples à mettre à jour. NON

Théorie de la Normalisation

Repose sur l'observation que certaines relations ont de meilleures propriétés, dans un environnement de mise à jour, que d'autres relations équivalentes contenant les mêmes informations.

Fournit un cadre rigoureux pour la définition du schéma relationnel.

Normalisation d'une relation

- Processus de décomposition d'une relation à mises-à-jour complexes en plusieurs relations à mises-à-jour simples
- Processus sur le schéma relationnel formel

Les formes normales

On mesure la qualité d'une relation par son degré de normalisation :

1FN, 2FN, 3FN, FNBC, 4FN, 5FN, FNDC, 6FN...

au moins à 3FN, sur par garder l'information, jusqu'à 5FN

Normalisation par décomposition

Definition

Soit une relation R qui contient des redondances et pose des problèmes lors des mises-à-jour

Si elle n'est pas "normalisée", il faut la décomposer en plusieurs relations mieux "normalisées".

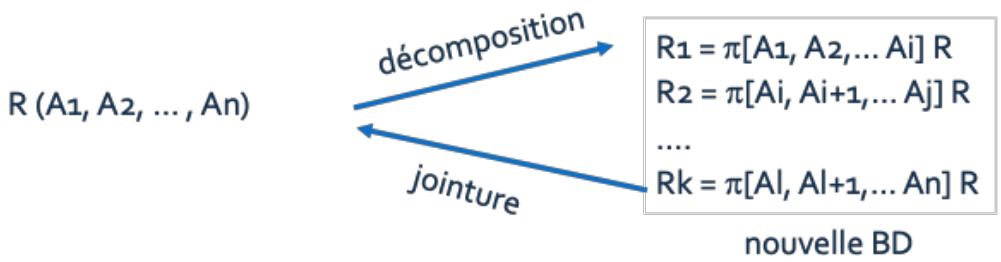
- par projection
- en suivant les Dépendances Fonctionnelles

Contrainte

Il faut s'assurer de conserver le même contenu

=> La jointure des nouvelles relations = R

Normalisation par décomposition



Si R = R1*R2* ...*Rk, la décomposition est sans perte d'information

Les requêtes sur R et celles sur la nouvelle Base de Données donneront toujours le même résultat

Exemple de décomposition sans perte

R			
Nom	Adresse	Poste	Age
Zoë	Nantes	Secrétaire	27
Armand	Paris	Secrétaire	32
Marie	Angers	Directreur	38

Peut se décomposer en : **R=R1*R2**

R1		
Nom	Adresse	Poste
Zoë	Nantes	Secrétaire
Armand	Paris	Secrétaire
Marie	Angers	Directreur

R2	
Nom	Age
Zoë	27
Armand	32
Marie	38

Cette décomposition est sans perte d'information

Exemple de décomposition avec perte

R			
Nom	Adresse	Poste	Age
Zoë	Nantes	Secrétaire	27
Armand	Paris	Secrétaire	32
Marie	Angers	Directreur	38

Peut se décomposer en :

R1		
Nom	Adresse	Poste
Zoë	Nantes	Secrétaire
Armand	Paris	Secrétaire
Marie	Angers	Directreur

R2	
Poste	Age
Secretaire	27
Secretaire	32
Directeur	38

Cette décomposition ne permet pas de retrouver la relation d'origine

Théorème de HEATH

Théorème

R (X, Y, Z) est décomposable sans perte d'information en :

- R1 = π[X,Y]R
- R2 = π[X,Z]R

si la Dépendance Fonctionnelle X->Y existe.

Conséquence

R1 est alors nécessairement normalisée (en 3FN).

Elle décrit le fait élémentaire X->Y.

Les requêtes posées sur R et celles posées sur R1*R2 donnent le même résultat

Application de HEATH

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité

Dépendances fonctionnelles :

- NFourn -> AdrFourn
- NProduit -> typeProduit
- (NFourn, NProduit) -> Quantité

Application de HEATH

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité

- NFourn -> AdrFourn

Fournisseur	
NFourn	AdrFourn

Livraison			
NFourn	NProduit	typeProduit	Quantité

Avec un lien externe entre NFourn dans Livraison et NFourn dans Fournisseur

Application de HEATH

Fournisseur	
NFourn	AdrFourn

Livraison			
NFourn	NProduit	typeProduit	Quantité

Avec un lien externe entre NFourn dans Livraison et NFourn dans Fournisseur

- NProduit -> typeProduit

Produit	
NProduit	typeProduit

Livraison		
NFourn	NProduit	Quantité

Avec un lien externe entre NProduit dans Livraison et NProduit dans Produit

Application de HEATH

Fournisseur	
NFourn	AdrFourn

Produit	
NProduit	typeProduit

Livraison		
NFourn	NProduit	Quantité

Avec un lien externe entre NFourn dans Livraison et NFourn dans Fournisseur
Avec un lien externe entre NProduit dans Livraison et NProduit dans Produit

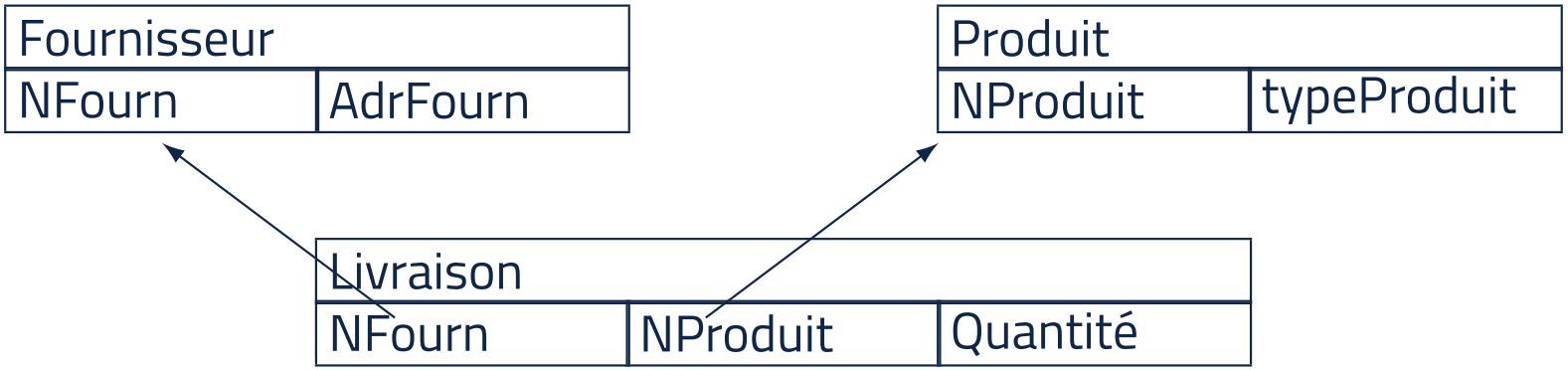
- (NFourn, NProduit) -> Quantité

Formalisé dans Livraison

Application de HEATH

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité

Après décomposition :



Qualité d'une décomposition

Une « bonne » décomposition est une décomposition

- sans perte d'information
- sans perte de Dépendance Fonctionnelle
  - Toute Dépendance Fonctionnelle doit être dans l'une des relations obtenues par décomposition
  - Une Dépendance Fonctionnelle ayant comme source un identifiant sera automatiquement vérifiée par le SGBD
  - Une Dépendance Fonctionnelle perdue => une contrainte d'intégrité implicite => le SGBD ne peut pas la vérifier
- qui produit de meilleures relations (mieux normalisées)

CENTRALE NANTES

Plan

1FN

Définition :

Une relation est en **1FN** si chaque valeur de chaque attribut de chaque tuple est une valeur atomique (tous les attributs sont simples et monovalués) et sont constants dans le temps.

Exemples 1FN

- Exemple 1 :

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité

Est en 1FN.

Exemples 1FN

- Exemple 2 :

Personne			
NPersonne	Nom	Prenom	Age

N'est pas en 1FN : Age n'est pas constant dans le temps.

- Exemple 3 :

Eleve		
NEleve	NPersonne	Diplômes

N'est pas en 1FN : Diplômes est une liste.

1FN : comment procéder

- Vérifiez que tous les attributs sont atomiques (ça devrait être le cas sur un modèle relationnel)
- Vérifiez que vos attributs ne changent pas simplement parce que e temps passe (Age, délai avant expiration, ...)

2FN

Objectif : Permettre d'éliminer les attributs qui ne décrivent pas l'objet représenté par la relation

Exemple :

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité

- L'adresse du fournisseur n'a rien à voir avec la livraison. Cette information devrait être déduite du numéro de fournisseur.
- Le type du produit n'a rien à voir avec la livraison. Cette information devrait être déduite du numéro de produit

2FN

Définition :

Une relation est en **2FN** si

- elle est en 1FN
- chaque attribut qui ne fait pas partie de l'identifiant dépend de l'identifiant entier

2FN : comment procéder

- S'assurer que la relation est en 1FN
- Vérifier que la relation a bien un identifiant
- Prendre successivement chacun des attributs qui ne sont pas dans l'identifiant et vérifier qu'ils dépendent de l'identifiant entier, et pas d'une partie de l'identifiant.

Si la 3e condition n'est pas vérifiée, appliquer Heath pour décomposer la relation en relations plus simples, en suivant les décompositions fonctionnelles.

Exemple 2FN

Livraison				
NFourn	AdrFourn	NProduit	typeProduit	Quantité

Se décompose en :

Fournisseur	
NFourn	AdrFourn

Produit	
NProduit	typeProduit

Livraison		
NFourn	NProduit	Quantité

Toutes les relations sont maintenant en 2FN.

3FN

Objectif : Permettre d'éliminer des sous-relations incluses dans une relation
= s'assurer que l'arbre des dépendances fonctionnelles a une profondeur de 1

Exemple :

Fournisseur			
NFourn	AdrFourn	Ville	Pays

Dépendances fonctionnelles :

- NFourn -> AdrFourn
- NFourn -> Pays
- NFourn -> Ville -> Pays

3FN

Définition :

Une relation est en **3FN** si

- elle est en 2FN
- chaque attribut qui ne fait partie d'aucun identifiant dépend **directement** de l'identifiant entier

3FN : comment procéder

- S'assurer que la relation est en 2FN
- Etablir l'arbre des Dépendances Fonctionnelles (l'identifiant en est la source). Vérifier qu'il n'y a pas de dépendance entre un attribut qui ne fait pas partie de l'identifiant et un autre identifiant.

Si la 2e condition n'est pas vérifiée, décomposer les attributs générant le problème en suivant les dépendances fonctionnelles.

Exemple 3FN

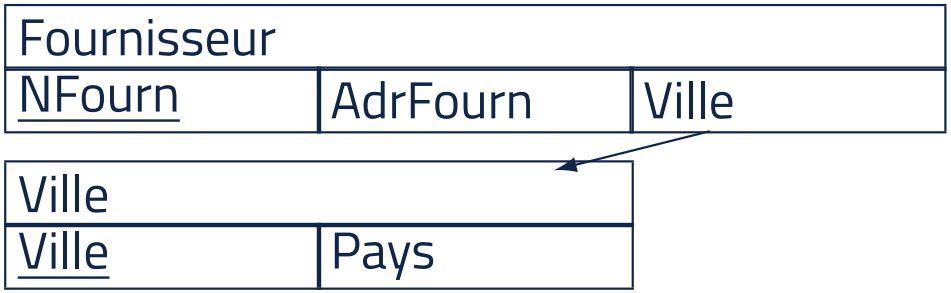
Exemple :

Fournisseur			
NFourn	AdrFourn	Ville	Pays

- NFourn -> AdrFourn : OK
- NFourn -> Pays : OK
- NFourn -> Ville -> Pays : Erreur

Exemple 3FN

Exemple :



Remarque

Toute relation peut toujours être décomposée en relations en 3FN

- Sans perte de dépendance fonctionnelle
- Sans perte d'information

Une Base de Données DOIT donc toujours être au moins en 3FN

FNBC : Forme Normale de Boys-Codd

Objectif : Généraliser la 3FN à toutes les clés candidates

Exemple :

Achat		
NFourn	NomFourn	NProduit

Clés candidates :

- NFourn, NProduit
- NomFourn, NProduit
- NFourn, NomFourn, NProduit (pas minimal)

FNBC

Définition :

Une relation est en **FNBC** si

- elle est en 3FN,
- toute source complète de Dépendance Fonctionnelle est un identifiant entier

Définition 2 :

Une relation est en FNBC si et seulement si les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé détermine un attribut

FNBC : comment procéder

- Vérifier que la relation est en 3FN
- S'assurer qu'il n'y a pas de dépendance fonctionnelle entre certains attributs de l'identifiant.
- Si la 2e condition n'est pas vérifiée, changer l'identifiant et décomposer la relation en plusieurs relations si nécessaire.

NB : le fait de décomposer l'identifiant peut entraîner des pertes de dépendances fonctionnelles.

FNBC : comment procéder

Achat		
NFourn	NomFourn	NProduit

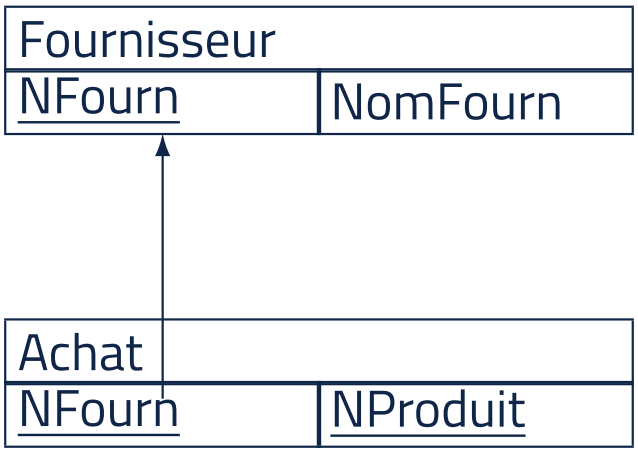
Dépendances fonctionnelles :

- NFourn -> NomFourn

Comme il y a une dépendance fonctionnelle au niveau de l'identifiant, il faut décomposer l'identifiant.

FNBC : comment procéder

On crée une relation qui prend en compte la dépendance fonctionnelle, et on la prend en compte dans la relation d'origine.



FNBC : limitation

La décomposition peut parfois engendrer la perte de certaines dépendances fonctionnelles.

2 solutions :

- Ne pas faire la décomposition
- Effectuer la décomposition et ajouter une contrainte supplémentaire explicite

Un autre type de dépendance

Certaines relations en FNBC peuvent encore contenir des redondances, et poser des problèmes lors des mises-à-jour

Exemple : Cours (nomC, prof, livre)

Les données :

Cours		
nomC	prof	livre
Programmation	Duval Schmidt	Algorithmes Programmation en C++
BD	Magnin Moreau	Ullmann Gardarin

Un autre type de dépendance

... Une fois mis en œuvre

nomC	Prof	livre
Programmation	Duval	Algorithmes
Programmation	Duval	Programmation en C++
Programmation	Schmidt	Algorithmes
Programmation	Schmidt	Programmation en C++
BD	Magnin	Ullmann
BD	Magnin	Gardarin
BD	Moreau	Ullmann
BD	Moreau	Gardarin

Un autre type de dépendance

Cours pose des problèmes de mise-à-jour

- ajouter un nouveau professeur, Alex, au cours de BD
- corriger le nom d'un livre
- ...

Cependant Cours est déjà bien normalisée : Cours est en FNBC

Dépendance Multivaluée

Définition

Soit une relation R (X, Y, Z). Il y a dépendance multivaluée : X ->> Y Si à toute valeur de X correspond un ensemble de valeurs de Y qui est totalement indépendant de Z

Propriétés

S'il y a la DM : X ->> Y
alors il y a aussi X ->> Z cd ters On note : X ->> Y | Z

Dépendance Multivaluée

Cours		
nomC	prof	livre

La relation Cours contient une dépendance Multivaluée

nomC ->> prof | livre

4FN

Objectif :

La 4FN permet de séparer des faits multivalués indépendants qui auraient été réunis dans une même relation

4FN

Définition :

R est en 4FN si :

- elle est en 3FN,
- toute DF ou DM de R a pour source un identifiant entier de R

Autre définition :

R est en 4FN si elle est en FNBC et ne contient pas de DM

Remarque :

4FN implique FNBC

4FN : comment procéder

Une relation est en 4e forme normale si, pour toute relation de dimension n en forme normale de Boyce-Codd, les relations de dimension n-1 construites sur sa collection ont un sens

Il ne doit pas être possible de reconstituer les occurrences de la relation de dimension n par jointure de deux relations de dimension n-1

2e théorème de Heath

Théorème

Si R(X, Y, Z) contient la DM X ->> Y | Z
alors la décomposition en : - R1 = π[X,Y]R - R2 = π[X,Z]R
est sans perte d'information.

4FN : Exemple de décomposition

Cours		
nomC	prof	livre

nomC ->> prof | livre

Se décompose en :

CoursProf	
nomC	prof

CoursLivre	
nomC	livre

Dépendance de Jointure

L'objectif est de décomposer une relation qui regrouperait 3 (ou +) liens indépendants

Définition

Soit R(A1, A2, ... An) une relation et X1, X2, ... Xm des relations composées des attributs A1, A2, ... An.
On dit qu'il existe une **dépendance de jointure** si R peut être reconstruite par jointure des relations X1... Xm.

Dépendance de Jointure

Exemple : Soit la relation Suit(Eleve, Matière, Enseignant)
Suit peut être décomposée en 3 relations

- V1 (eleve, matiere)
- V2 (eleve, enseignant)
- V3 (matiere, enseignant)
- et Suit = V1 * V2 * V3 (jointure des 3 relations)

Suit peut contenir des redondances et peut poser des problèmes lors des mises-à-jour.
Il faut alors décomposer Suit en V1, V2 et V3

Dépendance de jointure

Exemple :

Eleve	Matière	Enseignant
Jacques	Info	Olivier
Hervé	Matériaux	Erwan
Jacques	Matériaux	Erwan
Hervé	Info	Olivier
Pascal	Info	Didier

Si Olivier, l'enseignant d'Info est modifié en Guillaume, il faut mettre à jour toutes les lignes concernées sinon l'information ne sera pas cohérente entre toutes les lignes.

Il faut alors décomposer la relation.

5FN

Définition :

R est en 5FN si toute dépendance de jointure est impliquée par un identifiant (DJ triviale)

5FN : comment procéder

Une relation est en 5e forme normale si, pour toute relation de dimension n en 4e forme normale, il n'est pas possible de retrouver l'ensemble de ses occurrences par jointure sur les occurrences des relations partielles prises deux à deux.

NB : la décomposition n'est utile que si elle posera potentiellement des problèmes de mise-à-jour.

FNDC

Définition :

Une relation est en Forme Normale Domaine Clef si et seulement si toutes les contraintes sont la conséquence logique des contraintes de domaines et des contraintes de clefs qui s'appliquent à la relation.

Plan

Conclusion

Les Formes normales permettent d'assurer que le modèle relationnel mis en œuvre est conforme à un certain nombre de règles.

Ces règles assurent que, pour les éléments qui les concernent, les tables ne poseront pas de problèmes de mise à jour.

A minima, votre schéma doit être en 3e forme normale, et si possible en 5e forme normale.

Recommandation

Si vous êtes passé par un Modèle Conceptuel des données de type Entités-Associations et que vous avez traduit ce modèle en un Modèle Physique des Données de type relationnel, vous devriez avoir avité la plupart des écueils.


Success
NEXT EXIT