

Plan du cours

- Chap. III : Le modèle relationnel
 - Les concepts de base
 - Normalisation d'une BD relationnelle
 - Processus de normalisation

64

Les concepts de base

- Avec le modèle **relationnel**, les données sont organisées sous forme de **relation** (appelée encore **table**) :
 - **Instance** ou **extension** de la relation : *ensemble de lignes et de colonnes*.
Nombre de lignes = *cardinalité*, Nombre de colonnes = *degré*.
 - **Schéma** de la relation : spécifie le nom de la relation, plus le nom et le **domaine** (type) de chaque colonne (attribut).
- **Base de données relationnelles**: un ensemble des **relations** de la BD, chacune ayant un nom distinct.
 - **Schéma d'une BD**: ensemble de schémas des relations dans la BD.
 - **Instance de la BD**: ensemble des instances relationnelles de la BD.

65

Les concepts de base

- ❖ Exemple :
 - ❖ **Schéma** : Etudiant (N_Etud: integer, Prénom : string, Adresse : string, age: integer).
 - ❖ Cardinalité = 3, degré = 4, les lignes sont **distinctes**.
 - ❖ **L'identifiant** : (clé primaire) N_Etud, il permet de caractériser de manière unique chaque tuple de la relation.
 - ❖ **Instance** :

Attributs

	<u>N_Etud</u>	Prénom	Adresse	Age	
Etudiant ↑ Nom de la relation	136	Mohamed	Gabes	20	} Tuples N-uplets Enregistrements
	253	Amine	Tunis	20	
	101	Ali	Gafsa	21	

66

Les concepts de base : clé étrangère

- Une **clé étrangère** est un ensemble d'une ou de plusieurs colonnes d'une relation (précédée par le symbole #) qui fait **référence** à une clé primaire d'une (autre) relation.

Exemples :

```

COMMANDES(NUMCDE,DATCDE,NUMCLI#,ETATCDE)
CLIENTS (NUMCLI, NOMCLI, ADRCLI)
EMPLOYE (EMPNO, ENOM, ... , EMPNO_CHEF#, .....)
    
```

(Note: In the original image, red arrows indicate that NUMCLI# in COMMANDES is a foreign key to NUMCLI in CLIENTS, and EMPNO_CHEF# in EMPLOYE is a foreign key to EMPNO in EMPLOYE.)

67

Les concepts de base : Les contraintes d'intégrité

- Une **Contrainte d'Intégrité (CI)** : condition qui doit être satisfaite dans *toutes* les instances de la base de données.
 - Les CIs sont **spécifiées** lorsque le schéma est défini.
 - Les CIs sont **vérifiées** lorsque les relations sont modifiées.
- Une instance **légale** d'une relation est une instance qui satisfait toutes les CIs spécifiées.
 - Un SGBD ne doit pas permettre des instances illégales.
 - Si le SGBD vérifie les CIs, les données stockées reflètent mieux la signification du monde réel.
 - Évite les erreurs d'entrée de données aussi!

68

Les contraintes d'intégrité

Trois types de C.I. obligatoires

- **Contrainte de clé** : une relation doit posséder une clé primaire
- **Contrainte d'entité** : un attribut d'une clé ne doit pas posséder de valeurs nulles (vides)
- **Contrainte de référence** (pour les clés étrangères)
 - C'est une contrainte exprimée **entre deux tables**.
 - Tout enregistrement d'une relation faisant référence à une autre relation doit se référer à un enregistrement **qui existe**.
 - Les seules valeurs que peut prendre une clé étrangère sont celles qui sont déjà saisies dans **la table qu'elle référence**.

69

Normalisation d'une BD relationnelle

- Un schéma de relation est décrit par la liste de ses **attributs** et de leurs **contraintes d'intégrité**.
- Les **formes normales** des relations et les mécanismes pour les construire permettent d'obtenir des relations **non redondantes**.
- Les **formes normales** sont fondés sur les notions de clés de relations et de **dépendances entre données**.
- Les **formes normales** fournissent les conditions d'application d'un processus dit de **normalisation** qui mène à des formes « correctes » de relations qui sont les formes normales.

70

Normalisation d'une BD relationnelle

PRODUIT(RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

RefProduit	LibelleProduit	PU	Quantité	NumService	Adresse	Capacité
P1	CH7	23.510	300	S1	Sousse	9000
P1	CH7	23.510	500	S2	Tunis	6000
P3	VIS12	0.150	900	S1	Sousse	2000

- Cette relation présente certaines anomalies :
 - **Redondance** : un produit apparaît autant de fois qu'il sera livré par un service
 - Risques d'incohérences lors des mises à jour :
 - si l'on s'aperçoit que le libelle de P1 n'est pas CH7 mais CH1, il faudra veiller à mettre à jour tous les tuples contenant P1.
- Il est nécessaire d'autoriser la présence de valeurs nulles dans une telle relation afin de pouvoir conserver dans la base des services sans produit.
- si nous supprimons un produit qui soit le seul livré à un service donné, nous supprimons avec lui toute trace de ce service

71

Normalisation d'une BD relationnelle

- Une Base de Données relationnelle est 'correcte' ou normalisée si :
 - Chaque relation décrit une **information élémentaire** avec les seuls attributs qui lui sont directement liés
 - Il n'y a pas des **redondances d'informations** qui peuvent produire des problèmes de mise à jour
- La relation Produit peut être décomposée en trois relations non redondantes

72

Normalisation d'une BD relationnelle

PRODUIT(RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

Décomposition 1

PRODUIT1(RefProduit, Libelle, PU)

P1	CH7	23.510
P3	VIS12	0.150

PRODUIT2 (RefProduit, NumService, Quantité, Adresse, Capacité)

Décomposition 2

PRODUIT22 (NumService, Adresse, Capacité)

PRODUIT 21 (RefProduit#, NumService#, Quantité)

P1	S1	300
P1	S2	500
P3	S4	900

S1	Sousse	9000
S2	Tunis	6000
S4	Sousse	2000

73

Normalisation d'une BD relationnelle

- Le résultat final de la décomposition est donc les relations suivantes :
- PRODUIT1 (RefProduit, LibelleProduit, PU) contient les données relatives aux produits.
- PRODUIT21 (RefProduit#, NumService#, Quantité) contient les données relatives aux produits distribués par des services.
- PRODUIT22 (NumService, Adresse, Capacité) contient les données relatives aux services.

74

Dépendance fonctionnelle

- Définition
- Soit une relation R [..., A, B, ...], on dit que A détermine B ou B dépend fonctionnellement de A **si à toute valeur de A il ne lui est associé qu'une seule valeur de B.**
- On note une telle DF :
 $A \rightarrow B$.
- Exemple
- PRODUIT (RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)
- Pour cette relation, les dépendances fonctionnelles suivantes sont :
 - $\text{RefProduit} \rightarrow \text{LibelleProduit}$
 - $\text{RefProduit} \rightarrow \text{Adresse, Capacité}$
 - $\text{RefProduit} \rightarrow \text{PU}$
 - $\text{NumService} \rightarrow \text{Quantité}$

75

Propriétés des dépendances fonctionnelles

- Les **règles d'inférence** nous permettent de découvrir de nouvelles dépendances à partir d'un ensemble initial.
Soit R une relation :
- Propriété 1 : Réflexivité
 $A \rightarrow A$ et si $\exists B \subseteq A$ alors $A \rightarrow B$
Tout ensemble d'attributs détermine **lui-même** ou **une partie** de lui-même.
- Propriété 2 : Augmentation
 $A \rightarrow B$ alors $A, C \rightarrow B, C$
Si A détermine B, les deux ensembles d'attributs peuvent être enrichis par un même troisième.
- Propriété 3 : Transitivité
 $A \rightarrow B$ et $B \rightarrow C$ alors $A \rightarrow C$

76

Propriétés des dépendances fonctionnelles

- Propriété 4 : Union
 $A \rightarrow B$ et $A \rightarrow C$ alors $A \rightarrow B, C$
- Propriété 5 : Pseudo-transitivité
 $A \rightarrow B$ et $C, B \rightarrow D$ alors $A, C \rightarrow D$ ($A, C \rightarrow B, C$ et $C, B \rightarrow D$ alors $A, C \rightarrow D$)
- Propriété 6 : Décomposition
 $A \rightarrow B$ et $C \subseteq B$ alors $A \rightarrow C$
(Ref_P, Numserv, Date \rightarrow Qte, Date
donc Ref_P, Numserv, Date \rightarrow Qte)

77

Dépendance fonctionnelle élémentaire Dépendance fonctionnelle Canonique

- Une DF $A \rightarrow B$ est **élémentaire** si pour tout $A' \subset A$ la dépendance fonctionnelle $A' \rightarrow B$ n'est pas vraie.
- **$\rightarrow B$ ne dépend pas fonctionnellement d'une partie de A** (A est la plus petite quantité d'information donnant B)
 - Exemple : RefProduit, LibelleProduit \rightarrow PU
n'est pas élémentaire car il suffit d'avoir la **référence** du produit pour déterminer le prix unitaire.
- Une dépendance fonctionnelle $A \rightarrow B$ est **canonique** si sa partie droite ne comporte qu'un seul attribut.
- Exemple
 - NumService \rightarrow Adresse, Capacité : n'est pas canonique

78

Clé d'une relation

- La **clé d'une relation** est l'ensemble d'attributs dont les valeurs permettent de caractériser les n-uplets de la relation de manière **unique**.
- Formellement :
- Un attribut ou une liste d'attributs A est **une clé** pour la relation R(A, B, C) si
 - B et C dépendent fonctionnellement de A dans R : $A \rightarrow B, C$
 - et $A \rightarrow B, C$ est élémentaire.
- Une relation peut avoir **plusieurs clés**. Une clé sera choisie et désignée comme **clé primaire**. Les autres seront appelées **clés candidates**.

79

Processus de normalisation

- Les **formes normales** ont été définies pour permettre la **décomposition** des relations sans perte d'informations en utilisant la notion de dépendance fonctionnelle.
- La normalisation des relations est l'application d'un ensemble de **règles prédéfinies**, introduites dans le modèle relationnel afin de garantir la **cohérence** de la base lors des différentes opérations de **manipulation** et de **MAJ**.
- Le processus est basé sur 5 niveaux ou 5 formes normales appelées dans l'ordre 1FN, 2FN, 3FN, 4FN et 5FN.

80

Première Forme Normale (1FN)

- Une relation R est en 1^{ère} FN si
 - Elle possède une clé
 - Tous ses attributs sont **atomiques** (non décomposables)
- Exemples
- LIVRE (No-ISBN, Titre, AuteurS, Editeur)
Cette relation n'est pas en 1FN car l'attribut "Auteurs" est **multivalué**.
- Proposition de correction : →
LIVRE (No-ISBN, Titre, Auteur1, Auteur2, Auteur3, Editeur)
- ETUDIANT (Num, Nom, Prénom, Adresse(Rue, Ville)) n'est pas en 1FN car l'attribut Adresse n'est pas atomique.
- Proposition de correction : →
ETUDIANT (Num, Nom, Prénom, Rue, Ville)

81

Deuxième Forme Normale (2FN)

- Une relation R est en 2FN si et seulement si :
 - Elle est en 1FN,
 - Toutes les DF entre la clé et les autres attributs sont élémentaires, **aucun attribut ne dépend d'une partie de la clé**.
- Exemple
 - Soit la relation CLIENT avec ses DF
CLIENT (NumCl, AdrCl, RefProduit, PU)
NumCl, RefProduit → PU
 - F2 : NumCl, RefProduit → AdrCl n'est pas élémentaire car NumCl → AdrCl
 - La clé de la relation est (NumCl, RefProduit)
 - Suite à F2, une partie de la clé (NumCl) détermine un attribut n'appartenant pas à la clé. Cette relation n'est donc pas en 2FN. Elle pourra être décomposée en :
CLIENT (NumCl, AdrCl)
PRODUIT (RefProduit, PU)

82

Troisième Forme Normale (3FN)

- Une relation R est en troisième forme normale (3FN) si et seulement si :
 - Elle est en 2FN,
 - Tout attribut n'appartenant pas à une clé ne dépendra pas d'un attribut non clé : (Toutes les DF : sont directes pas de transitivité).
- La troisième forme normale permettra d'éliminer les **redondances** dues aux dépendances **transitives**.
- La décomposition en 3FN est sans perte d'informations et préserve les DF.

83

Troisième Forme Normale (3FN)

■ Exemple1 :

CLIENT (NumCl, ChiffreAffaire, Ville, Pays)

Avec les dépendances fonctionnelles suivantes :

□ F1 : NumCl → ChiffreAffaire

□ F2 : NumCl → Ville

□ F3 : Ville → Pays

■ La relation CLIENT n'est pas en 3FN à cause des dépendances fonctionnelles F3.

■ Cette relation doit être décomposée en deux relations :

□ CLIENT (NumCl, ChiffreAffaire, Ville#)

□ ADRESSE (Ville, Pays)

84

Troisième Forme Normale (3FN)

■ Exemple2 :

VOITURE (NumVoiture, Marque, Type, Puissance, Couleur)

□ F1 : NumVoiture → Marque, Type, Puissance, Couleur

□ F2 : Type → Marque

■ N'est pas en 3FN. En effet, l'attribut non clé TYPE détermine MARQUE (F2).

■ Cette relation peut ainsi être décomposée en deux relations :

□ VOITURE (NumVoiture, Type#, Couleur, Puissance)

□ MODELE (Type, Marque)

85