

Introduction aux bases de données

Concepts, utilisation et développement

Damien Pellier
damien.pellier@imag.fr
<http://membres-liglab.imag.fr/pellier/>



1/369

Objectif du cours

- Former les étudiants à l'utilisation et à la conception de Bases de Données au sein d'un système d'information

2/369

Compétences visées

1. Connaître les principes des bases de données relationnelles
2. Maîtriser les principes de l'algèbre relationnelle
3. Savoir écrire des requêtes de manipulation de données en SQL
4. Savoir concevoir et normaliser un modèle conceptuel de données à partir d'un cahier des charges
5. Savoir dériver un schéma relationnel de bases de données à partir d'un modèle de données

3/369

Plan du cours

- **Partie I : Introduction aux bases de données relationnelles**
 - Cours 1 : Concepts des bases de données relationnelles
 - Cours 2 : L'algèbre relationnelle
- **Partie II : Utilisation des bases de données relationnelles**
 - Cours 3 : Le langage SQL DML (1)
 - Cours 4 : Le langage SQL DML (2)
 - Cours 5 : Le langage SQL DDL
- **Partie III : Développement des bases de données relationnelles**
 - Cours 6 : Le modèle entité-association
 - Cours 7 : Élaboration d'un schéma conceptuel
 - Cours 8 : Production du schéma de la base de données

4/369

Organisation et modalités de contrôle des connaissances

- Organisation du cours
 - 8 séances de cours
 - 4 séances de TD
 - 4 séances de TP
- Modalités de contrôle de connaissances
 - Note finale = TP ou Projet
- Adresse du cours en ligne
 - <http://membres-liglab.imag.fr/pellier/teaching/bd>

5/369

Organisation et modalités de contrôle des connaissances

- Semaine 1 : 18/09 CM1 TD1 Projet
- Semaine 2 : 25/09 CM2 TD2 Projet
- Semaine 3 : 02/10 CM3 TD3 TP1-DCISS
- Semaine 4 : 09/10 CM4 TD4 TP1-SSD-C2ES
- Semaine 5 : 16/10 CM5 TD5 TP2-DCISS
- Semaine 6 : 23/10 CM6 TD6 Projet
- Semaine 7 : 06/11 CM7 TD7 TP2-SSD-C2ES
- Semaine 8 : 13/11 CM8 TD8 TP3-DCISS
- Semaine 9 : 20/11 TP3-SSD-C2ES TP4-DCISS
- Semaine 10 : 27/11 TP4-SSD-C2ES Projet

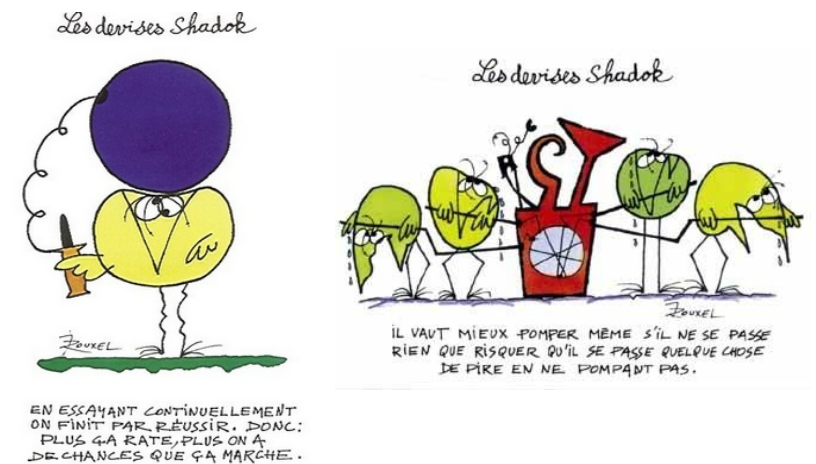
6/369

Références bibliographiques

- Introduction aux Bases de Données, C. Date, Vuibert, 2004
- Bases de Données, G. Gardarin, Eyrolles, 2003
- SQL 2 De la théorie à l'application, P. Delmal, De Boeck Université, 1998
- Bases de données, J-L. Hainaut, 2009

7/369

Comment réussir ce module ?



8/369

Comment réussir ce module ?



Règle 1 : Assistez à tous les cours, TD et TP

- En cas d'absence à un cours rattrapez rapidement
- Une absence non rattrapée \Rightarrow un décrochage

Règle 2 : Lisez le sujet de TP ou TD avant la séance encadrée

- Vous gagnerez beaucoup de temps

Règle 3 : Posez des questions !

- Les enseignants sont là pour y répondre

Règle 4 : Ne laissez pas passer un mot ou un concept sans le comprendre

- Demandez systématiquement à un enseignant

Règle 5 : Relisez votre TD ou TP le soir après l'avoir fait

- Profitez de la séance suivante pour poser les questions que vous aurez préalablement préparées

9/369

Première partie I

Concepts des bases de données relationnelles

10/369

Plan du cours

- **Partie I : Introduction aux bases de données relationnelles**
 - Cours 1 : Concepts des bases de données relationnelles
 - Cours 2 : L'algèbre relationnelle
- **Partie II : Utilisation des bases de données relationnelles**
 - Cours 3 : Le langage SQL DML (1)
 - Cours 4 : Le langage SQL DML (2)
 - Cours 5 : Le langage SQL DDL
- **Partie III : Développement des bases de données relationnelles**
 - Cours 6 : Le modèle entité-association
 - Cours 7 : Élaboration d'un schéma conceptuel
 - Cours 8 : Production du schéma de la base de données

11/369

Qu'est ce qu'une base de données ?

- Une collection de données cohérentes entre elles, généralement de taille importante
- Modélise des informations du monde réel
 - Entités, e.g., étudiants, cours, notes, etc.
 - Associations, e.g., Bob a choisi le cours de BD
- Exemples de bases de données
 - Achats en ligne
 - Abonnement à un club de sport
 - Compte en banque
 - Réservation de billets de train
 - Étude à l'université
 - etc.

12/369

Les défis des bases de données aujourd'hui

- Multicité des types de données
 - Exemples :
 - Données multimédias (images, vidéo, musique)
 - Données géographiques
- L'information incorrecte et incomplète
 - Comment produire de l'information correcte à partir de données imprécises ou erronées ?
- Accès aisé par des non-informaticiens
- Maintenance et évolution
- Des données distribuées et nomades
 - Les données sont nomades, intermittentes, dupliquées
- Base de données et le Web
- Les données décisionnelles
- Vers des normes d'accès aux données pour les applications
 - Par exemples : Hibernate, EJB

13/369

Qu'est ce qu'un SGBD ?

- Un Système de Gestion de Bases de Données (SGBD) est un logiciel destiné au stockage et à la manipulation de bases de données
- Tous les traitements des données se font obligatoirement par son intermédiaire :
 - Interrogation
 - Ajout
 - Suppression
 - Modification
- Exemples de fournisseurs de SGBD
 - Oracle
 - MySQL
 - PostgreSQL
 - etc.

14/369

Pourquoi un SGBD ?

- Indépendance des données/applications et sûreté d'accès aux données
- Temps de développement d'applications réduit
 - ⇒ réduction des coûts de développement
- Intégrité des données et sécurité des accès
- Administration des données uniformes et cohérentes
- Concurrence des accès et reprise sur panne

15/369

Indépendance des données

- Indépendance des applications par rapport au modèle physique autorise :
 - de modifier l'organisation physique sans modifier le schéma logique ni les applications
 - d'ajout d'index
 - de changer de méthode d'accès, e.g., chaînage vs. hachage
- Indépendance logique des données
 - Chaque application ou groupe d'utilisateurs peut :
 - assembler différemment les données
 - ne voir qu'une partie des données (Schémas externes)
 - Permet de modifier le schéma logique sans modifier les applications
 - Ajout/suppression d'une entité

16/369

- **Intégrité**
 - les données contenues dans une BD forment un tout cohérent répondant à des contraintes d'intégrités vérifiées à tout moment par le SGBD
 - Exemples :
 - Tout électeur a un âge > 18 ans
 - A tout instant la somme des crédits = la somme des débits
- **Unicité des données (non redondance)**
 - Cohérence des données
 - Si une donnée est dupliquée, laquelle est la bonne ?
 - Faciliter de mise à jour et de recherche
 - Est-on sûr d'avoir mis à jour toutes les copies d'une même donnée ?

17/369

- **Confidentialité**
 - Gestion des autorisations d'accès à tout ou partie de la BD aux différentes personnes de l'organisation
- **Concurrence**
 - Les données sont accédées simultanément par plusieurs utilisateurs
 - ⇒ Éviter les conflits qui dégraderaient la cohérence de la BD
 - Chaque utilisateur à l'impression d'être le seul à travailler sur la BD
 - ⇒ verrouillage, transaction
- **Sécurité**
 - Remise de la BD dans l'état cohérent le plus récent après une panne
 - Idée : Garder un journal ou log (historique) de toutes les actions élémentaires de mise à jour et de validation réalisées par le SGBD :
 - Avant qu'un changement ne soit réalisé, l'action est tracée dans un fichier de log
 - Après un crash, l'effet des transactions non abouties est annulé à l'aide du fichier de log

18/369

Description des données (1/3)

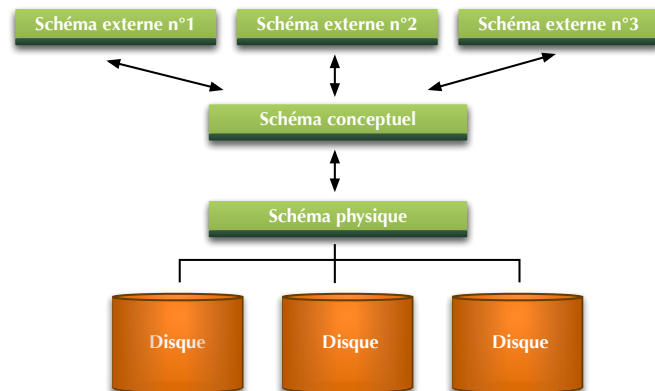


Figure 1 – Modèle ANSI – SPARC

19/369

Description des données (2/3)

- **Schéma externe** aussi appelé vue
 - Structure des données telle que perçue par chaque groupe de travail de l'organisation
 - tous les utilisateurs n'ont pas à connaître le schéma logique dans son ensemble
 - ⇒ données inutiles à l'utilisateur
 - ⇒ sécurité, confidentialité
- **Schéma conceptuel** aussi appelé schéma logique
 - Structure et sémantique des données d'une organisation sans souci d'implémentation. Par exemple :
 - Type de données élémentaires des attributs des objets
 - Règles régissant les données
- **Schéma interne**
 - Structure de stockage des données en machine
 - fichiers (nom, organisation, localisation)
 - article des fichiers (longueur, champs, placement)
 - chemin d'accès aux données (index, chaînage, hachage)

20/369

Description des données (3/3)

- Schéma conceptuel

Clients(idl : integer, Nom : string, Prénom : string, Profession : string)
Comptes(idc : string, type : string, solde : string, idl : integer)
Opérations(idc : string, dte : date, type : string, montant : integer)

- Schéma externe

Cumul_mensuel(mois : string, idc :string, Nom : String, Prénom : string,
total_crédits :integer, total_débits : integer)

- Schéma interne

- Les relations sont stockées dans des fichiers non séquentiels

21/369

Modèle de description de données

- Modèle de description de données

- Ensemble de concepts et de règles de composition de ces concepts permettant de décrire des données

- Exemples :

⇒ Modèle hiérarchique
⇒ Modèle réseau
⇒ Modèle relationnel
⇒ Modèle objets

- Langage de description de données (LDD)

- Langage supportant un modèle et permettant de décrire les données d'une BD d'une manière assimilable par une machine

- Exemples :

⇒ SQL (modèle relationnel)

22/369

Place aux concepts au travers d'un exemple (1/7)

Commande n°: 30188		Date : 2/1/2009		
Numéro de client : B512				
Nom : GILLET				
Adresse : 14, r. de l'Eté				
Localité : Toulouse				
N° PRODUIT	LIBELLE PRODUIT	PRIX	QUANTITE	SOUS-TOTAL
CS464	CHEV. SAPIN 400x6x4	220	180	39600
PA45	POINTE ACIER 45 (20K)	105	22	2310
PA60	POINTE ACIER 60 (10K)	95	70	6650
PH22	PL. HETRE 200x20x2	230	92	21160
...
TOTAL COMMANDE				97200

23/369

Place aux concepts au travers d'un exemple (2/7)

Commande n°: 30188

Date : 2/1/2009

Numéro de client : B512

Nom : GILLET

Adresse : 14, r. de l'Eté

Localité : Toulouse

Données du client

Données de la commande

N° PRODUIT	LIBELLE PRODUIT	PRIX	QUANTITE	SOUS-TOTAL
CS464	CHEV. SAPIN 400x6x4	220	180	39600
PA45	POINTE ACIER 45 (20K)	105	22	2310
PA60	POINTE ACIER 60 (10K)	95	70	6650
PH22	PL. HETRE 200x20x2	230	92	21160
...	
TOTAL COMMANDE				97200

Données d'un détail

24/369

Place aux concepts au travers d'un exemple (3/7)

Données de la commande

NCOM	DATECOM	TOTAL-COMMAND
30188	2/1/2009	97200

Données du client

NCLI	NOM	ADRESSE	LOCALITE
B512	GILLET	14, r. de l'Eté	Toulouse

Données des détails

NPRO	LIBELLE	PRIX	QCOM	SOUS-TOTAL
CS464	CHEV. SAPIN 400x6x4	220	180	39600
PA45	POINTE ACIER 45 (20k)	105	22	2310
PA60	POINTE ACIER 60 (10k)	95	70	6650
PH222	PL. HETRE 200x20x2	230	92	21160

25/369

Place aux concepts au travers d'un exemple (4/7)

- On avance mais le résultat n'est pas satisfaisant

- Certaines données sont calculées, e.g., total de la commande et sous-totale
⇒ il n'est pas nécessaire de les stocker
- En l'état il est impossible de reconstruire le document initial
 - Comment récupérer le client d'une commande puisque nous avons extrait et rangé ailleurs le fragment décrivant ce client ?
 - Comment identifier la commande de laquelle nous avons extrait un détail ?
 ⇒ il manque des données de références pour effectuer des liaisons entre les données

26/369

Place aux concepts au travers d'un exemple (5/7)

Données des commandes

NCOM	NCLI	DATECOM
30188	B512	2/1/2009
30179	C400	22/12/2008

Données des clients

NCLI	NOM	ADRESSE	LOCALITE
B512	GILLET	14, r. de l'Eté	Toulouse
C400	FERARD	63, r. de Tertre	Poitier

Données des détails

NCOM	NPRO	LIBELLE	PRIX	QCOM
30188	CS464	CHEV. SAPIN 400x6x4	220	180
30188	PA45	POINTE ACIER 45 (20k)	105	22
30188	PA60	POINTE ACIER 60 (10k)	95	70
30188	PH222	PL. HETRE 200x20x2	230	92
30179	CS262	CHEV. SAPIN 400x6x4	75	60
30179	PA60	POINTE ACIER 60 (10k)	95	20

27/369

Place aux concepts au travers d'un exemple (6/7)

- Un résultat prometteur, mais

- Certaines données sont stockées de manière redondantes dans les données de détails, e.g., libellé et prix
 - sources d'erreurs
 - inutiles et dangereux

⇒ il serait plus pertinent de construire une 4ème table pour stocker de manière unique les informations relatives aux produits

28/369

Place aux concepts au travers d'un exemple (7/7)

Données des commandes

NCOM	NCLI	DATECOM
30188	B512	2/1/2009
30179	C400	22/12/2008

Données des clients

NCLI	NOM	ADRESSE	LOCALITE
B512	GILLET	14, r. de l'Eté	Toulouse
C400	FERARD	63, r. de Tertre	Poitier

Données des détails

NCOM	NPRO	QCOM
30188	CS464	180
30188	PA45	22
30188	PA60	70
30188	PH222	92
30179	CS262	60
30179	PA60	20

Données des produits

NPRO	LIBELLE	PRIX
CS464	CHEV. SAPIN 400x6x4	220
PA45	POINTE ACIER 45 (20k)	105
PA60	POINTE ACIER 60 (10k)	95
PH222	PL. HETRE 200x20x2	230
CS262	CHEV. SAPIN 400x6x4	75

29/369

Premières conclusions

1. Une **base de données** est constituée d'un ensemble de **tables**
2. Chaque **table** contient les données relatives à des **entités** de même nature
3. Chaque **ligne** d'une table reprend les données relatives à une entité
4. Chaque **colonne** d'une table décrit une propriété commune des entités
5. Les lignes d'une table sont distinctes
6. Le jeu de colonnes dont les valeurs sont uniques constitue un **identifiant** ou clé primaire de la table
7. Les lignes d'une table peuvent faire **référence** chacune à une ligne d'une autre table. On parle alors de clé étrangère
8. On évite de stocker les informations qui peuvent être calculées
9. On ne conserve pas dans une même table des informations relatives à plusieurs entités
10. Il est nécessaire d'avoir un langage d'interrogation de la base de données

30/369

Tables, lignes et colonnes

- Les données d'une base de données sont organisées sous la forme de tables
- Une table contient une collection de lignes stockées sur un support physique, e.g., un disque
- Une ligne est une suite de valeurs d'un type déterminé, e.g., integer, string
 - Une ligne regroupe des informations concernant un objet, un individu ou un événement, etc.
 - Une ligne représente un concept du monde réel, appelé entité ou fait
 - Les lignes d'une même table ont le même format ou structure
- L'ensemble des valeurs de même type correspondant à une même propriété s'appelle colonne

CLIENT			
NOM	ADRESSE	LOCALITE	COMPTE
HANSENNE	23, A. Dumont	Poitier	1.250,00
MERCIER	25, r. Lemaître	Namur	-2.300,00
TOUSSAINT	5, r. Godefroid	Poitier	0,00
VANBIST	180, r. Forimont	Lille	720,00

31/369

Tables, lignes et colonnes

FOURNISSEUR		
NUMF	NOMF	VILLEF
46	GERCIN	Paris
81	DUMONT	Paris
152	MERCIER	Tours
174	CHARLES	Nevers
259	CHARLES	Liège
376	RENIER	Nevers

OFFRE		
NUMF	NUMP	PRIX
46	15	46
46	57	32
81	14	65
81	15	48
152	14	62
152	15	46
152	57	34
174	57	32

PIECE	
NUMP	TYPE
14	Boulon
15	Boulon
57	Ecrou

32/369

Les identifiants

- Toutes les colonnes ne jouent pas le même rôle dans une table
- Il faut être en mesure de désigner de manière univoque une ligne décrivant une entité
- On appelle **identifiant** de la table la colonne qui l'identifie de manière unique

Remarques

- Un identifiant peut être composé de plusieurs colonnes
- Rien n'interdit d'imposer plus d'un identifiant par table

33/369

Les identifiants

FOURNISSEUR		
NUMF	NOMF	VILLEF
46	GERCIN	Paris
81	DUMONT	Paris
152	MERCIER	Tours
174	CHARLES	Nevers
259	CHARLES	Liège
376	RENIER	Nevers

OFFRE		
NUMF	NUMP	PRIX
46	15	46
46	57	32
81	14	65
81	15	48
152	14	62
152	15	46
152	57	34
174	57	32

PIECE	
NUMP	TYPE
14	Boulon
15	Boulon
57	Ecrou

34/369

Les identifiants

- L'identifiant le plus représentatif est appelé **identifiant primaire**
 - Le terme anglais est *primary key*
- Les autres sont appelés **identifiants secondaires**
- Toutes les tables possèdent un identifiant primaire qui peut être composite (plusieurs colonnes) et un nombre quelconque d'identifiants secondaires
 - Le nombre d'identifiants secondaires peut être nul
- Toutes les entités d'une table possèdent au moins un identifiant
 - ⇒ les identifiants d'une table constituent un ensemble et sont donc distincts

35/369

Les identifiants

- Pour jouer son rôle d'identification, l'identifiant doit être unique
- Cette propriété s'appelle **contrainte d'unicité**
- Elle est garantie par le SGBD au moment des ajouts dans la table
- Exemple :

OFFRE		
NUMF	NUMP	PRIX
46	15	46
46	57	32
81	14	65
81	15	48
152	14	62
152	15	46
152	57	34
174	57	32

36/369

Les identifiants

- Un identifiant primaire ne peut être constitué que de colonnes dont les valeurs doivent être renseignées
 - C'est ce qu'on appelle l'**intégrité d'entité**
- L'idée est que l'essence même d'une entité se concentre dans son identifiant

Remarque

- Cette contrainte n'est pas exigée pour les identifiants secondaires

37/369

Les identifiants

- Tout ensemble de colonnes qui comprend un identifiant est un identifiant
 - Par exemple :
(NUMF, NOMF) est un identifiant de la table FOURNISSEUR
- Un identifiant dont on ne peut retirer aucun élément sans qu'il perde sa qualité d'identifiant est appelé **identifiant minimal**
- Il est évident qu'il est important de définir que des identifiants minimaux

38/369

Les clés étrangères

- Une colonne, ou un ensemble de colonnes, dont le rôle est de référencer une ligne dans une autre table (dite table cible) est dénommée **clé étrangère** (*foreign key*)
- Exemple :

FOURNISSEUR		
NUMF	NOMF	VILLEF
46	GERCIN	Paris
81	DUMONT	Paris
152	MERCIER	Tours
174	CHARLES	Nevers
259	CHARLES	Liège
376	RENIER	Nevers

OFFRE		
NUMF	NUMP	PRIX
46	15	46
46	57	32
81	14	65
81	15	48
152	14	62
152	15	46
152	57	34
174	57	32

PIECE	
NUMP	TYPE
14	Boulon
15	Boulon
57	Ecrou

39/369

Les clés étrangères

- Il est nécessaire que l'ensemble des valeurs d'une clé étrangère soit un sous-ensemble des valeurs de l'identifiant cible
- Cette contrainte s'appelle **contrainte référentielle**
- Exemple :
 - L'ajout de la ligne (174, 97, 125) dans la table OFFRE serait rejeté puisque 97 n'est pas un identifiant de pièce dans la table PIECE

FOURNISSEUR		
NUMF	NOMF	VILLEF
46	GERCIN	Paris
81	DUMONT	Paris
152	MERCIER	Tours
174	CHARLES	Nevers
259	CHARLES	Liège
376	RENIER	Nevers

OFFRE		
NUMF	NUMP	PRIX
46	15	46
46	57	32
81	14	65
81	15	48
152	14	62
152	15	46
152	57	34
174	57	32

PIECE	
NUMP	TYPE
14	Boulon
15	Boulon
57	Ecrou

40/369

Schéma et contenu d'une base de données

- **Schéma d'une base**

- Description au moyen d'un langage déterminé d'un ensemble de données particulier
 - Change peu souvent
- Exemple :

FOURNISSEUR			OFFRE			PIECE	
NUMF	NOMF	VILLEF	NUMF	NUMP	PRIX	NUMP	TYPE

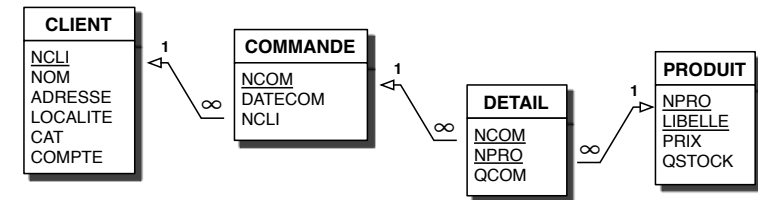
- **Instance d'une base**

- Données effectivement contenues dans la base à un instant précis
 - Change à chaque mise à jour : ajout, suppression ou modification

FOURNISSEUR			OFFRE			PIECE	
NUMF	NOMF	VILLEF	NUMF	NUMP	PRIX	NUMP	TYPE
46	GERCIN	Paris	46	15	46	14	Boulon
81	DUMONT	Paris	46	57	32	15	Boulon
152	MERCIER	Tours	81	14	65	57	Ecrou
174	CHARLES	Nevers	81	15	48		
259	CHARLES	Liège	152	14	62		
376	RENIER	Nevers	152	15	46		
			152	57	34		
			174	57	32		

41/369

Représentation graphique d'un schéma



42/369

Le phénomène de redondance interne

LIVRE					
NUMERO	TITRE	AUTEUR	ISBN	DATE.ACHAT	RAYONNAGE
1029	Les misérables	Hugo V.	2 02 033300 7	17/10/2008	F3
1030	Les misérables	Hugo V.	2 02 033300 7	17/10/2008	F3
1032	Le Postier	Bukowski C.	2 253 14911 X	17/10/2008	G5
1045	Le K	Buzzati D.	2 253 05354 6	22/02/2009	F3
1067	Le Postier	Bukowski C.	2 253 14911 X	24/02/2009	G5
1022	Le Postier	Bukowski C.	2 253 14911 X	03/10/2008	G6

- **Problème** : Lorsqu'un livre existe en plusieurs exemplaires, les informations (TITRE, AUTEUR, ISBN) sont dupliquées
- Cette situation viole le principe fondateur des bases de données
 - tout fait pertinent du domaine d'application doit être enregistré **une et une seule fois**

43/369

Le phénomène de redondance interne

- **Inconvénients de la redondance**

1. La table occupe un espace excessif et inutile
2. Les modifications sont coûteuses puisqu'il faut mettre à jour toutes les données dupliquées
3. Comment garantir que les données dupliquées restent identiques et cohérentes ?
 - Exemples :
 - Si le premier ajout d'un livre se fait librement, les ajouts d'un autre exemplaire doivent se faire conformément aux informations déjà saisies
 - L'effacement du seul exemplaire d'un livre supprimerait définitivement les informations concernant son titre et son auteur

- **Dépendance fonctionnelle**

- Il existe une contrainte entre les colonnes de la table LIVRE
 - ISBN → TITRE, AUTEUR
 - ISBN est le déterminant et (TITRE, AUTEUR) les déterminés
- Les dépendances fonctionnelles doivent être évitées au sein d'une même table

44/369

Le phénomène de redondance interne

OUVRAGE		
ISBN	TITRE	AUTEUR
2 02 033300 7	Les misérables	Hugo V.
2 253 14911 X	Le Postier	Bukowski C.
2 253 05354 6	Le K	Buzzati D.

EXEMPLAIRE			
NUMERO	ISBN	DATE.ACHAT	RAYONNAGE
1029	2 02 033300 7	17/10/2008	F3
1030	2 02 033300 7	17/10/2008	F3
1032	2 253 14911 X	17/10/2008	G5
1045	2 253 05354 6	22/02/2009	F3
1067	2 253 14911 X	24/02/2009	G5
1022	2 253 14911 X	03/10/2008	G6

45/369

La normalisation

- **Objectif**
 - Définir des règles pour décomposer les relations tout en préservant les DF sans perdre d'informations afin de représenter les objets et les associations canoniques du monde réel
- **Eviter les anomalies de mises à jour**
- **Eviter les réponses érronées**
- **Outils :**
 - Dépendances fonctionnelles
 - Décompositions
 - Formes normales

46/369

La normalisation

- La **décomposition** d'une relation est le remplacement d'une relation R par un ensemble de relations R_1, \dots, R_n tel que
 - Attributs de $R =$ l'union de tous les attributs des R_i
 - R_i est obtenu à partir de R par projection sur les attributs de R_i
- **Décomposition sans perte**
 - de sorte que l'on puisse retrouver exactement l'information de départ
 - une décomposition est sans perte ssi elle préserve les dépendances fonctionnelles

47/369

La normalisation

- **Définition**
 - Une relation est dite de première forme normale, si elle admet une clé et que tous ses attributs contiennent une valeur atomique
- **Exemple**
 - PILOT (ID, NOM, AVIONS)

PILOT		
ID	NOM	AVIONS
2	BERLIOZ	{Albastros, AEG}
3	De St-Exupéry	{AGO, Fokker}
5	Garros	{Gotha, Zeppelin}

- La relation PILOT n'est pas en première forme normale
 - elle possède une clé
 - mais l'attribut AVIONS n'est pas atomique

48/369

- Définition

- Une relation est en deuxième forme normale ssi :
 1. Elle est en première forme normale
 2. Tout attribut non clé ne dépend pas d'une partie de clé

- Exemple

- PILOT (ID, NOM, LICENCE, DATE_OBTENTION)
 - n'est pas en deuxième forme normale car
 - $ID \rightarrow NOM$
 - $LICENCE \rightarrow DATE_OBTENTION$
- Une telle relation doit être décomposée en
- PILOT (ID, NOM, LICENCE)
 - LICENCE_PILOT (LICENCE, DATE_OBTENTION)

49/369

- Définition

- Une relation est en troisième forme ssi :
 1. Elle est en deuxième forme normale
 2. Tout attribut n'appartenant pas à une clé ne dépend pas d'un autre attribut non clé

- Exemple

- AVION (ID, CONSTRUCTEUR, TYPE, PUISSANCE, AUTONOMIE)
 - n'est pas en troisième forme normale car
 - $ID \rightarrow CONSTRUCTEUR, TYPE$
 - $TYPE \rightarrow PUISSANCE, AUTONOMIE$
- Une telle relation doit être décomposée en
- AVION (ID, CONSTRUCTEUR, TYPE)
 - MODÈLE (TYPE, PUISSANCE, AUTONOMIE)

50/369

- La troisième forme normale :

- enlève les redondances dues aux dépendances transitives
- permet de ne pas perdre d'information
- permet de ne pas perdre des dépendances

- Remarques :

- Un modèle relationnel doit être de troisième forme normale
- La troisième forme normale est la plus utilisée du fait de son équilibre entre redondance et performance
- Cependant pour certains systèmes (surtout distribués), lorsque la redondance n'est pas un problème, une *dénormalisation* est effectuée pour augmenter les performances

51/369

- C'est quand qu'on s'arrête ?

- Les quatrièmes, cinquièmes et sixièmes formes normales évitent principalement la redondance d'information, elles sont plus précises
- En pratique la FN 3 est suffisante
 - En effet, les projections et les jointures sont coûteuses pour le système, ainsi une trop forte normalisation diminue fortement les performances

52/369

La normalisation

- Tant que l'ensemble des attributs n'est pas vide, on doit successivement :
 1. Choisir un attribut X qui n'est le but d'aucune dépendance fonctionnelle
 2. Construire une relation dont X est la clé et qui comporte tous les attributs qui sont déterminés fonctionnellement par X
 3. Renouveler cette opération pour toutes les dépendances complexes auxquelles l'attribut X participe
 4. Supprimer toutes les dépendances qui ont été prises en compte, ainsi que tous les attributs figurant dans ces dépendances et qui ne sont eux-mêmes sources d'aucune nouvelle dépendance (attributs isolés).

53/369

La normalisation

- **Exercice 1** : Vérifier si le schéma suivant est normalisé. Si nécessaire, le décomposer en tables normalisées :

VENTE (NPRO, CLIENT, DATE, QUANTITE, ADRESSE, DELEGUE, REGION)

54/369

La normalisation

- **Exercice 1** : Vérifier si le schéma suivant est normalisé. Si nécessaire, le décomposer en tables normalisées :

VENTE (NPRO, CLIENT, DATE, QUANTITE, ADRESSE, DELEGUE, REGION)

CLIENT \rightarrow ADRESSE, DELEGUE

DELEGUE \rightarrow REGION

Correction

VENTE (NPRO, CLIENT, DATE, QUANTITÉ)

CLI (CLIENT, ADRESSE, DELEGUE)

REP (DELEGUE, REGION)

55/369

La normalisation

- **Exercice 2** : Décomposer si nécessaire le schéma ci-dessous :

COMMANDE (NCOM, NCLI, NOM, DATE, NPRO, LIBELLE)

56/369

- **Exercice 2** : Décomposer si nécessaire le schéma ci-dessous :

COMMANDE (NCOM, NCLI, NOM, DATE, NPRO, LIBELLE)

NCLI → NOM

NPRO → LIBELLE



Correction

COMMANDE (NCOM, NCLI, DATE, NPRO)

CLIENT (NCLI, NOM)

PRODUIT (NPRO, LIBELLE)

- Les concepts de tables, d'identifiants, de lignes, de colonnes, d'entités de faits
- Les notions de clés primaires, clés étrangères
- La différence entre schéma conceptuel, instance d'une base de données et SGBD
- Le problème de la redondance interne
- Les différents types de contraintes : contrainte d'unicité, dépendances fonctionnelles
- Le principe de normalisation et de décomposition