

Bases de données

Claudine Chaouiya
claudine.chaouiya@univ-amu.fr

Polytech Marseille - Biomédical 4ème année
COURS 1



Good news, when we transferred your medical history to our digital system, it became coherent

Février 2019

Objectifs du cours

- Conception d'une base de données
- Introduction à l'algèbre relationnelle
- Création, manipulation et interrogation d'une base de données à l'aide du langage SQL

2 / 32

Introduction aux bases de données

Qu'est-ce qu'une base de données ?

Assemblage et stockage de données sur différents support (papiers, fichiers, etc.), de façon organisée dans l'objectif de pouvoir y accéder de façon efficace et appropriée

Une base de données (en anglais *database*), permet de stocker et de retrouver l'intégralité de données brutes ou d'informations en rapport avec un thème ou une activité ; celles-ci peuvent être de natures différentes et plus ou moins reliées entre elles. Dans la très grande majorité des cas, ces informations sont très structurées, et la base est localisée dans un même lieu et sur un même support. Ce dernier est généralement informatisé. (Wikipedia)

3 / 32

Introduction aux bases de données

Qu'est-ce qu'une base de données ?

Assemblage et stockage de données sur différents support (papiers, fichiers, etc.), de façon organisée dans l'objectif de pouvoir y accéder de façon efficace et appropriée

Une BD est un ensemble de données structuré selon un modèle. Les données sont enregistrées sur des supports accessibles par ordinateur. Elles représentent des informations du monde réel. qui peuvent être interrogées et mises à jour par des utilisateurs.

- Bibliothèque : livres, emprunts, etc.
- Université : étudiants, enseignants, cours, etc.
- Sécurité Sociale : dossiers, remboursements, etc.
- Hôpital : dossiers PATIENTS, traitements, personnel, services, etc.
- Banques : clients, comptes, etc.
- Entreprises : RH, stocks, etc.
- Sites Web personnels
- etc.

3 / 32

Introduction aux bases de données

Qu'est-ce qu'une base de données ?

Quelques grandes bases de données*

World Data Centre for Climate, **220 terabytes** of web data, **6 petabytes** of additional data

Amazon, 59 million active customers, More than **42 terabytes** of data

YouTube, 100 million videos watched per day, 65,000 videos added each day, At least **45 terabytes** of videos



1 gigaoctets (Go) = 1024 mégaoctets (Mo)
1 téraoctet (To) = 1024 gigaoctets (Go)
1 pétaoctet (Po) = 1024 téraoctets (To)
1 exaocet (Eo) = 1024 pétaoctets (Po)

* volumes en 2010 <https://www.comparebusinessproducts.com/fyi/10-largest-databases-in-the-world>

4 / 32

Introduction aux bases de données

Qu'est-ce qu'une base de données ?

A quoi ça sert ?

- **Stockage** de données de façon **persistante**
- **Accès** rapide et facile aux données
- **Partage** des données avec de multiples utilisateurs

5 / 32

Introduction aux bases de données

Qu'est-ce qu'une base de données ?

A quoi ça ressemble ?

Exemple d'un centre médical

MEDECIN
matricule
nom
prenom
specialite

PATIENT
num_secu_soc
nom
prenom
date_naissance
telephone

CONSULTE
#matricule
#num_secu_soc
date
heure

6 / 32

Introduction aux bases de données

Qu'est-ce qu'une base de données ?

A quoi ça ressemble ?

Exemple d'un centre médical

MEDECIN
matricule
nom
prenom
specialite

PATIENT
num_secu_soc
nom
prenom
date_naissance
telephone

CONSULTE
#matricule
#num_secu_soc
date
heure

MEDECIN	matricule	nom	prenom	specialite
	135455	Durand	Michèle	generaliste
	236456	Martin	Jean	dermatologiste
	43526	Martin	Virginie	pediatre
	134252	Weber	Eric	generaliste

PATIENT	num_secu_soc	nom	prenom	date_naissance	telephone
	1820113156148	Birzeit	Hamid	25/01/1982	0670908700
	2900626008155	Dupond	Marie	26/06/90	0689775478
	1761013022765	Rei	Jules	02/10/76	

CONSULTE	matricule	num_secu_soc	date	heure
	135455	1820113156148	3/02/19	11:00
	236456	2900626008155	4/03/19	14:00
	134252	1761013022765	21/02/19	9:00

Quel est le nom des generalistes ? Quels PATIENTs ont consulté un dermatologue ?

6 / 32

Introduction aux bases de données

Qu'est-ce qu'une base de données ?

A quoi ça ressemble ?

MEDECIN			
matricule	nom	prénom	specialite
135455	Durand	Michèle	generaliste
236456	Martin	Jean	dermatologiste
435526	Martin	Virginie	pediatre
134252	Weber	Eric	generaliste

PATIENT

num_secu_soc	nom	prenom	date_naissance	telephone
1820113156148	Birzeit	Hamid	25/01/1982	0670908700
2900626008155	Dupond	Marie	26/06/90	0689775478
1761013022765	Rei	Jules	02/10/76	

CONSULTE

matricule	num_secu_soc	date	heure
135455	1820113156148	3/02/19	11:00
236456	2900626008155	4/03/19	14:00
134252	1761013022765	21/02/19	9:00

Une BD relationnelle est une collection de données structurées en relations

- Relations/Tables : un ensemble de données ex : **MEDECIN**
- Attributs/Champs/Colonnes d'une relation ex : **matricule, nom, adresse, specialite**
- n-uplets/Tuples/Enregistrements/Lignes : liste de valeurs d'une ligne d'une relation
ex : **<134252, Weber, Eric, generaliste>**

7/32

Introduction aux bases de données

Comment ça marche?

SGBD, Système de Gestion de Bases de Données → outil logiciel spécialisé pour la gestion de BD

- description des données et relations
- interrogation des données
- mise à jour des données
- partage des données
- protection des données

8/32

Introduction aux bases de données

Comment ça marche?

SGBD, Système de Gestion de Bases de Données → outil logiciel spécialisé pour la gestion de BD

- description des données et relations
- interrogation des données
- mise à jour des données
- partage des données
- protection des données

Il existe divers SGBD : **MySQL, PostgreSQL, ORACLE, SQLServer, DB2, etc.**

8/32

Introduction aux bases de données

Comment ça marche?

SGBD, Système de Gestion de Bases de Données → outil logiciel spécialisé pour la gestion de BD

- description des données et relations
- interrogation des données
- mise à jour des données
- partage des données
- protection des données

Il existe divers SGBD : **MySQL, PostgreSQL, ORACLE, SQLServer, DB2, etc.**

Objectifs

- **Indépendance physique** → modification de l'organisation physique (accès) sans modification des programmes.
- **Indépendance logique** → modification du schéma conceptuel sans modification des programmes.
- **Persistence** des données → mécanismes de sauvegarde et restauration.
- **Non redondance** des données → la présence de mêmes données dans différents champs peut entraîner des incohérences lors de mises à jour
- **Intégrité** des données → la cohérence des données est maintenue lors des mises à jour.
- **Sécurité** des données → e.g., contrôle des droits d'accès.
- Partageabilité des données → les données sont simultanément consultables et modifiables.
- Manipulation des données → mécanismes de manipulation des données sans connaissance de l'organisation de la base.
- Efficacité des accès
- Administration centralisée → pour les structures de données, de stockage et de contrôle.

8/32

Introduction aux bases de données

Comment ça marche?

SGBD, Système de Gestion de Bases de Données → outil logiciel spécialisé pour la gestion de BD

Langage d'interrogation et manipulation de bases de données

- Définition et manipulation d'une BD *via* un langage **déclaratif**
- Le plus connu et le plus utilisé des langages : **SQL** (Structured Query Language)
- Modes d'utilisation autonome / intégré dans un langage hôte (C, Java, etc.)

9/32

Introduction aux bases de données

Comment ça marche?

SGBD, Système de Gestion de Bases de Données → outil logiciel spécialisé pour la gestion de BD

Langage d'interrogation et manipulation de bases de données

- Définition et manipulation d'une BD *via* un langage **déclaratif**
- Le plus connu et le plus utilisé des langages : **SQL** (Structured Query Language)
- Modes d'utilisation autonome / intégré dans un langage hôte (C, Java, etc.)

Les instructions de SQL sont regroupées en 5 catégories :

- 1 **Data définition Language, DDL**, langage de définition de données → manipulations de la structure de la BD
- 2 **Data Manipulation Language, DML**, langage de manipulation des données → extraction, ajout, suppression et modification des données
- 3 **Data Control Language, DCL**, langage de protection d'accès → gestion des privilèges accordés aux utilisateurs, e.g., droits d'accès aux tables
- 4 **Transaction Control Language, TCL**, langage de contrôle de transactions → gestion des modifications faites *via* le DDL et le DML, e.g., validation, annulation des modifications
- 5 **Embedded SQL**, SQL intégré permet d'intégrer les commandes SQL dans un langage hôte

9/32

Introduction aux bases de données

Comment ça marche?

Exemple SQL 1

MEDECIN			
matricule	nom	prénom	specialite
135455	Durand	Michèle	generaliste
236456	Martin	Jean	dermatologiste
435526	Martin	Virginie	pediatre
134252	Weber	Eric	generaliste

Nom et matricule des généralistes ?

```
SELECT nom, matricule
FROM MEDECIN
WHERE specialite='generaliste';

<Durand,135455>, <Weber, 134252>
```

10/32

Introduction aux bases de données

Comment ça marche?

Exemple SQL 2

PATIENT				
num_secu_soc	nom	prenom	date_naissance	telephone
1820113156148	Birzeit	Hamid	25/01/1982	0670908700
2900626008155	Dupond	Marie	26/06/90	0689775478
1761013022765	Rei	Jules	02/10/76	

MEDECIN			
matricule	nom	prénom	specialite
135455	Durand	Michèle	generaliste
236456	Martin	Jean	dermatologiste
435526	Martin	Virginie	pediatre
134252	Weber	Eric	generaliste

CONSULTE			
matricule	num_secu_soc	date	heure
135455	1820113156148	3/02/19	11:00
236456	2900626008155	4/03/19	14:00
134252	1761013022765	21/02/19	9:00

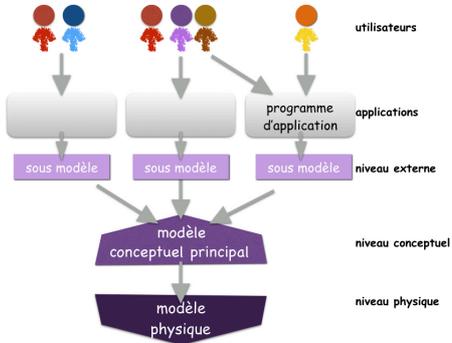
Nombre de PATIENTs n'ayant jamais consulté un pédiatre ?

```
SELECT COUNT (*)
FROM PATIENT
WHERE num_secu_soc NOT IN(
SELECT num_secu_soc
FROM PATIENT, CONSULTE, MEDECIN
WHERE PATIENT.num_secu_soc=CONSULTE.num_secu_soc
AND MEDECIN.matricule=CONSULTE.matricule
AND specialite='pediatre');
```

11/32

Introduction aux bases de données

Architecture d'un SGBD



Objectif : indépendance entre données et traitements

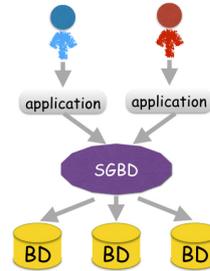
Norme ANSI / SPARC (1975)

12/32

Introduction aux bases de données

Architecture opérationnelle d'un SGBD

Architecture centralisée

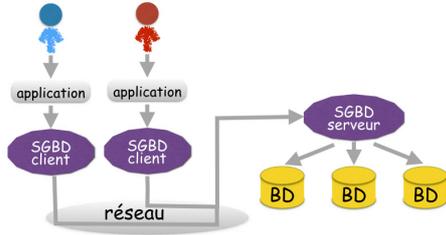


13/32

Introduction aux bases de données

Architecture opérationnelle d'un SGBD

Architecture client-serveur (simple)



Côté client
logiciel sur un ordinateur qui se connecte via le réseau au serveur → effectue des requêtes

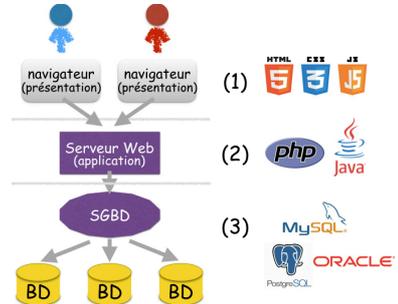
Côté serveur
logiciel sur l'ordinateur où sont stockées les données
→ attend des requêtes des clients, accède à la BD pour exécuter les requêtes, renvoie les réponses
tâches de gestion, stockage et de traitement de données

14/32

Introduction aux bases de données

Architecture opérationnelle d'un SGBD

Architecture trois-tiers



meilleure dissociation des parties serveur applicatif et base de données

Objectifs : minimiser le trafic de données sur le réseau, traiter de gros volumes, nombreux utilisateurs, assurer les contraintes d'intégrité et la tolérance aux pannes

15/32

Introduction aux bases de données

Architecture opérationnelle d'un SGBD



LAMP Suite de logiciels libres, comprenant le système d'exploitation (Linux), un serveur WEB (Apache), un SGBD (MySQL) et un langage de programmation interprété (PHP), utiles au développement et à l'hébergement d'un site Web.

⚠ de nouvelles alternatives, plus performantes, ont vu le jour ces dernières années

16 / 32

Introduction aux bases de données

Intervenants

Administrateur (une personne ou une équipe) :

- définition & évolution du schéma conceptuel de la BD
- définition de l'organisation physique pour des performances optimisées
- gestion des droits d'accès et des mécanismes de sécurité

17 / 32

Introduction aux bases de données

Intervenants

Administrateur (une personne ou une équipe) :

- définition & évolution du schéma conceptuel de la BD
- définition de l'organisation physique pour des performances optimisées
- gestion des droits d'accès et des mécanismes de sécurité

Développeurs/Programmeurs d'application :

- définition des schémas externes & construction des programmes d'alimentation et exploitation de la BD en vue d'applications spécifiques
- utilisation du langage du SGBD, éventuellement couplé avec un langage de programmation classique

17 / 32

Introduction aux bases de données

Intervenants

Administrateur (une personne ou une équipe) :

- définition & évolution du schéma conceptuel de la BD
- définition de l'organisation physique pour des performances optimisées
- gestion des droits d'accès et des mécanismes de sécurité

Développeurs/Programmeurs d'application :

- définition des schémas externes & construction des programmes d'alimentation et exploitation de la BD en vue d'applications spécifiques
- utilisation du langage du SGBD, éventuellement couplé avec un langage de programmation classique

Utilisateurs finaux : accès à la BD au travers

- des outils construits par les développeurs ou
- du langage de requêtes

17 / 32

Introduction aux bases de données

Modèles de description des données : petit historique

Organisation sous forme de systèmes de fichiers, jusqu'aux années 60

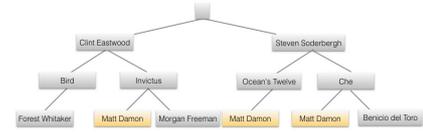
Introduction aux bases de données

Modèles de description des données : petit historique

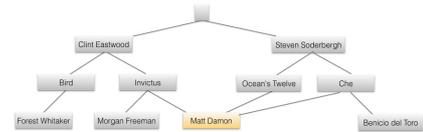
Organisation sous forme de systèmes de fichiers, jusqu'aux années 60

Modèle hiérarchique, fin des années 60

Le schéma est exprimé au moyen d'un arbre, ou d'une arborescence



Extension du schéma sous forme de réseau



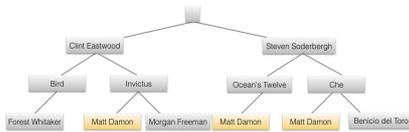
Introduction aux bases de données

Modèles de description des données : petit historique

Organisation sous forme de systèmes de fichiers, jusqu'aux années 60

Modèle hiérarchique, fin des années 60

Le schéma est exprimé au moyen d'un arbre, ou d'une arborescence



Extension du schéma sous forme de réseau

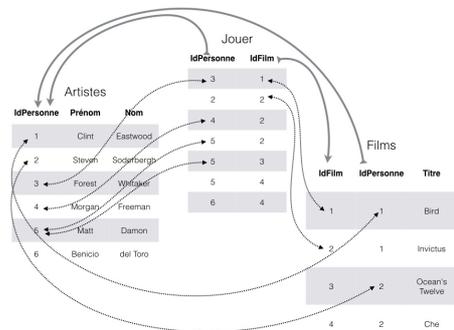


- pour retrouver une donnée, il faut connaître le chemin d'accès → dépendance logique
- pour insérer ou supprimer une donnée, il faut modifier la structure → anomalies lors des opérations de stockage

Introduction aux bases de données

Modèles de description des données : petit historique

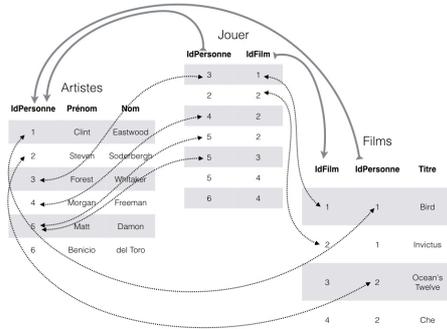
Modèle relationnel, années 70-80 → Edgar Franck Codd



Introduction aux bases de données

Modèles de description des données : petit historique

Modèle relationnel, années 70-80 → Edgar Franck Codd



Modèles orientés objet, début 90 → 3ème génération de SGBD, puis

- relationnel-objet,
- semi-structuré,
- multimédia, etc.

19 / 32

Introduction aux bases de données

Quel SGBD choisir ?

Différents facteurs/critères :

- la politique sécuritaire
- le budget à disposition
- la politique de l'entreprise concernant ses fournisseurs
- les compétences déjà acquises en terme de développement et d'administration, et au besoin le prix de la formation
- l'attente vis-à-vis du support technique
- la richesse fonctionnelle du SGBD
- le système d'exploitation hébergeant
- les ressources (disques, mémoire, CPU(s), transactions par secondes, nombre de connexions simultanées)
- ...

20 / 32

Conception des bases de données

Objectifs de l'étape de conception

- Gérer les cas complexes
- Eviter la redondance des données
- Optimiser les requêtes
- Faciliter les modifications et la maintenance
- Faciliter la ré-exploitation, les extensions, etc.

→ Méthodologie **Merise** pour la conception de BD
→ Langage **UML** pour la conception orientée-objet

21 / 32

Conception des bases de données

Détail des étapes de conception

- Analyse du problème et modélisation conceptuelle de la BD
Représentation informationnelle de l'organisation étudiée par un modèle conceptuel
- Implantation logique de la BD
Transformation du modèle conceptuel des données en objets propres au type de SGBD utilisé
- Implantation physique de la BD
Phase totalement prise en charge par le SGBD adopté



22 / 32

Conception des bases de données

Phase d'analyse

- Quelles sont les informations à stocker ?
- Comment représenter ces informations ?
- Quelles sont les relations entre ces informations ?
- Qui aura accès aux informations ou à certaines informations ?
- ...

23 / 32

Conception des bases de données

Phase d'analyse

- Quelles sont les informations à stocker ?
- Comment représenter ces informations ?
- Quelles sont les relations entre ces informations ?
- Qui aura accès aux informations ou à certaines informations ?
- ...

→ Langage de modélisation pour faciliter la communication entre les informaticiens, les clients, experts du domaine, futurs utilisateurs

23 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

- Description de la *sémantique* du domaine
- Vue d'ensemble des données et de leurs liens
- 1ère étape de conception complexe et essentielle
- Structure permettant les futures requêtes, complétude, pas d'incohérences ou redondances

Formalisme le plus répandu : **modèle entités-associations** introduit par Peter Chen en 1976

Basé sur la théorie des ensembles et des relations

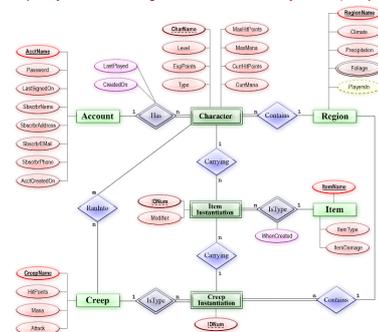
24 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Formalisme le plus répandu : **modèle entités-associations** introduit par Peter Chen en 1976
Basé sur la théorie des ensembles et des relations

Exemple : jeu de rôle en ligne massivement multi-joueurs (Wikipedia)



24 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Description statique de la future BD, à l'aide de :

- Entités
- Associations, Cardinalités
- Attributs, Domaines

Pas de modélisation des traitements
Langage Graphique

25 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Entité

Objet du monde réel qui a une existence propre

client N° 10, mon compte bancaire, ma voiture, ton médecin, le livre que je lis

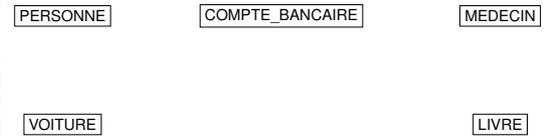
Une entité n'est pas représentée sur un modèle E/A qui représente le **type-entité** correspondant

Type-Entité

Un ensemble d'occurrence d'entités possédant une sémantique et des propriétés communes

CLIENT, COMPTE_BANCAIRE, VOITURE, MEDECIN, LIVRE

Représentation graphique → rectangle



26 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Association (ou relation)

Lien entre plusieurs entités

Une association n'est pas représentée sur un modèle E/A qui représente le **type-association** correspondant

Type-Association

Un ensemble d'associations qui possèdent les mêmes caractéristiques. Un type-association décrit un lien entre types-entités. Les occurrences d'associations lient les entités de ces types-entités.

27 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Association (ou relation)

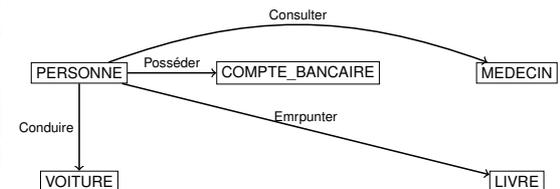
Lien entre plusieurs entités

Une association n'est pas représentée sur un modèle E/A qui représente le **type-association** correspondant

Type-Association

Un ensemble d'associations qui possèdent les mêmes caractéristiques. Un type-association décrit un lien entre types-entités. Les occurrences d'associations lient les entités de ces types-entités.

Représentation graphique → ligne



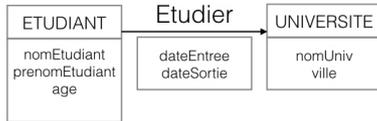
27 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Attributs (ou propriétés)

Caractéristiques des entités et associations



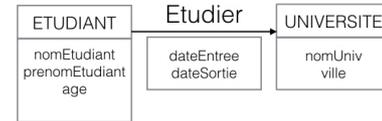
28 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Attributs (ou propriétés)

Caractéristiques des entités et associations



Les attributs ont un type

- Entier
- Flottant
- Chaîne de caractères
- Date
- Énumération (ex: {janvier, février, mars, avril, mai, juin, ..., décembre})

28 / 32

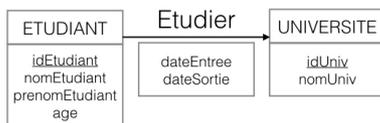
Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Identifiant (clé) d'un type-entité

Ensemble minimal d'attributs permettant d'identifier chaque entité de manière unique

- impossible que 2 entités aient le même identifiant
- tout type-entité possède au moins un identifiant



29 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Cardinalité Qualification (quantitative) du type d'association entre deux types-entités

- **Cardinalité minimale** : nombre minimal d'occurrences d'un type-entité participant aux occurrences de l'association (0,1,...)
- **Cardinalité maximale** : nombre maximal d'occurrences d'un type-entité participant aux occurrences de l'association (1,...,*)

Représentation graphique : couple de chiffres 0..1, 1..1, 0..*, ou 1..* ou n,m à gauche → cardinalité minimale, à droite → cardinalité maximale

30 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Cardinalité Qualification (quantitative) du type d'association entre deux types-entités

- **Cardinalité minimale** : nombre minimal d'occurrences d'un type-entité participant aux occurrences de l'association (0,1,...)
- **Cardinalité maximale** : nombre maximal d'occurrences d'un type-entité participant aux occurrences de l'association (1,..., *)

Représentation graphique : couple de chiffres 0..1, 1..1, 0..*, ou 1..* ou n,m à gauche → cardinalité minimale, à droite → cardinalité maximale

Une université a plusieurs étudiants

Un étudiant étudie dans une unique université



NB : on adoptera la convention UML où la cardinalité d'un côté spécifie combien d'entités du type-entité du côté considéré sont associées à une entité du type-entité de l'autre côté.

30 / 32

Conception des bases de données

Modèle conceptuel des données (MCD) - modèle Entités/Associations

Cardinalité Qualification (quantitative) du type d'association entre deux types-entités

- **Cardinalité minimale** : nombre minimal d'occurrences d'un type-entité participant aux occurrences de l'association (0,1,...)
- **Cardinalité maximale** : nombre maximal d'occurrences d'un type-entité participant aux occurrences de l'association (1,..., *)

Représentation graphique : couple de chiffres 0..1, 1..1, 0..*, ou 1..* ou n,m à gauche → cardinalité minimale, à droite → cardinalité maximale

Relationship	Example	left	right
one-to-one	person ↔ birth certificate	1	1
one-to-one (optional on one side)	person ↔ driving license	1	0..1 or ?
many-to-one	person ↔ birth place	1..* or +	1
many-to-many (optional on both sides)	person ↔ book	0..* or *	0..* or *
one-to-many	order ↔ line item	1	1..* or +
many-to-one	line item ↔ order	1..* or +	1
many-to-many	course ↔ student	1..* or *	1..* or *

30 / 32

Références

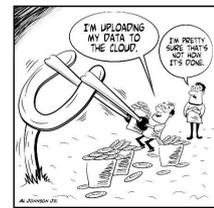
- AUDIBERT L. Bases de données, de la modélisation au SQL, Ellipses, 2009
- GARDARIN G., Bases de données, Eyrolles, 2003
- HAINAUT, J-L. Bases de données - Concepts, utilisation et développement, Dunod
- LAROUSSE N., Création de bases de données, Collection Informatique, Pearson Education, 2006
- Support de cours de Nicolas DURAND (Polytech Marseille, 2016)
- Support de cours de Henri Méloni & Thierry Spriet (Université d'Avignon et des Pays de Vaucluse, 2009)

31 / 32

Bases de données

Claudine Chaouiya
claudine.chaouiya@univ-amu.fr

Polytech Marseille - Biomédical 4ème année COURS 2



Février 2019

Résumé du cours précédent

1ère étape de conception : **Modèle conceptuel des données - MCD**

→ modèle Entités/Associations

- Type-Entité et ses occurrences (les entités)
 - Attributs (propriétés) avec un type associé
 - Identifiant(s) (clé)
- Type-Association (ou relation) entre type-entités, ses occurrences relient les entités de ces type-entités
 - Attributs
 - Cardinalités minimales et maximales pour chaque type-entité participant



2/30

Modèle conceptuel des données

Règles essentielles

Sur les attributs

- 1 chaque attribut doit avoir une valeur (si ce n'est pas le cas, cela doit être provisoire)
- 2 un attribut est une donnée élémentaire (pas calculée)
- 3 un attribut d'un type association doit dépendre de toutes les entités liées par le type-association

Sur les identifiants

- 1 Ne peuvent pas prendre la même valeur pour 2 occurrences
- 2 Tout type-entité a un identifiant composé d'au moins un attribut
- 3 Un type-association peut avoir seulement un identifiant implicite

Sur les cardinalités

- 1 Une cardinalité minimale est toujours de 0 ou 1, maximale de 1 ou n (*)
- 2 Les cardinalités admises sont donc :

- 0,1 une entité peut ne pas être impliquée dans une association ou dans une au maximum
- 0,n une entité peut ne pas être impliquée dans une association ou dans un nombre illimité d'associations
- 1,1 une entité ne peut exister que si elle est impliquée dans exactement une association
- 1,n une entité ne peut exister que si elle est impliquée dans au moins une association

3/30

Modèle conceptuel des données

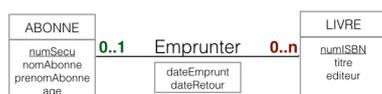
Règles essentielles

Catégories de type-associations selon leur cardinalités

toutes les card. max=1 1 vers 1
 une card. max=1 et une card. max=n 1 vers n (type-association *fonctionnel*)
 toutes les card. max=n n vers n (type-association *maillé*)

Sur les identifiants des type-associations

- 1 maillé → composé des identifiants de ses participants (implicite)
- 2 fonctionnel → c'est l'identifiant du participant de cardinalité max=1 (implicite)
- 3 1 vers 1 → deux identifiants candidats issus des 2 participants, il suffit d'en choisir un (implicite)



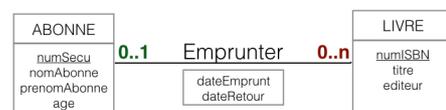
Exercice : comment modifier le schéma précédent pour qu'un livre puisse être emprunté plusieurs fois, y compris par un même abonné (dans ce cas à des dates différentes)

4/30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Type-associations pluriels



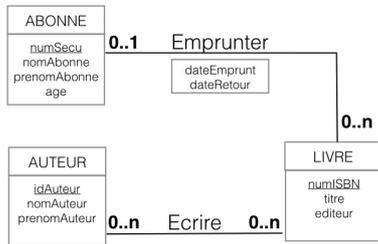
→ ajouter les auteurs de livres

5/30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Type-associations pluriels



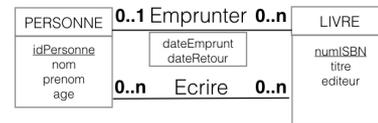
→ ▲AUTEUR et ABONNE représentent le même type d'entité

5/30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Type-associations pluriels



→ type-association pluriel où 2 type-entités sont liés par 2 relations. La nature des personnes est identifiable par le type des associations auxquelles elles participent

Nb : Une personne peut être à la fois un auteur et un abonné

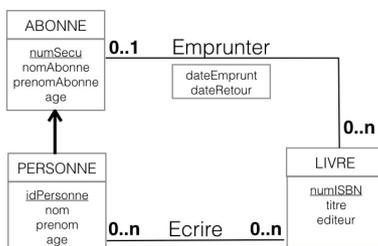
▲Comment ajouter aux abonnés des informations spécifiques qui ne concernent pas les auteurs (et vice versa) ?

5/30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Spécialisation associant un type-entité *générique* à un type-entité spécialisé de **cardinalité fixée à 0..1** côté générique et **1..1** côté spécialisée :



"est_un" est un **type-association spécialisé** qui impose la valeur des cardinalités entre les type-entités → notation simplifiée

6/30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Agrégation précisant une relation tout/partie (ou ensemble/élément) entre 2 type-entités



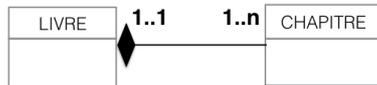
L'agrégation garde toutes les propriétés d'une association classique, elle ajoute simplement une terminologie plus précise *via* la notion de tout/partie.

7/30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Composition précisant une relation d'agrégation forte



- La composition est une association binaire de cardinalité contrainte
- La composition n'est pas symétrique, une classe joue le rôle de conteneur pour les classes liées
- La composition est une agrégation avec des contraintes supplémentaires (non partageabilité et cycle de vie lié)

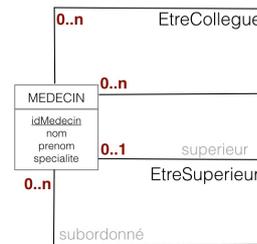
8 / 30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Type-associations réflexifs associant un type-entité à lui-même

- une occurrence relie deux entités différentes (du même type)
- peut être symétrique ou pas (l'ambiguïté est levée en précisant les rôles si nécessaire)



9 / 30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Liens identifiants



Un bâtiment peut avoir plusieurs salles, qui en retour appartiennent à un unique bâtiment
▲ l'identifiant de SALLE est absolu → 2 salles de 2 bâtiments différents ne peuvent avoir le même numéro

- un identifiant est **absolu** s'il est constitué d'attributs intrinsèques d'un même type-entité
- un identifiant est **relatif** s'il comporte au moins un attribut lié à un autre type-entité (e.g. identifiant implicite d'un type-association)

10 / 30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Liens identifiants



Un bâtiment peut avoir plusieurs salles, qui en retour appartiennent à un unique bâtiment
▲ l'identifiant de SALLE est absolu → 2 salles de 2 bâtiments différents ne peuvent avoir le même numéro

on introduit un **lien identifiant** (de cardinalité nécessairement 1..1), indiqué entre parenthèses

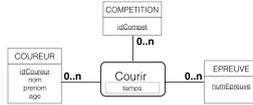
- un identifiant est **absolu** s'il est constitué d'attributs intrinsèques d'un même type-entité
- un identifiant est **relatif** s'il comporte au moins un attribut lié à un autre type-entité (e.g. identifiant implicite d'un type-association)
- l'identifiant du type-entité SALLE est maintenant le couple (idBatiment, numSalle)
- numSalle est un identifiant relatif, l'association Inclure est un lien identifiant
- BATIMENT est un type **agrégéant**, SALLE est un type **agrégé**

10 / 30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Type-associations n-aires



- Un coureur peut courir pour une même compétition plusieurs épreuves
- Un coureur peut courir une même épreuve pour différentes compétitions
- Une compétition et une épreuve sont disputées par plusieurs coureurs

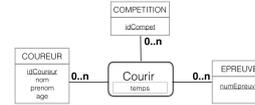
Les cardinalités maximales valent toutes obligatoirement n

11 / 30

Modèle conceptuel des données

Encore plus loin sur les type-associations

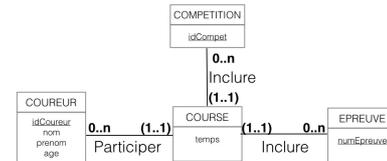
Type-associations n-aires



- Un coureur peut courir pour une même compétition plusieurs épreuves
- Un coureur peut courir une même épreuve pour différentes compétitions
- Une compétition et une épreuve sont disputées par plusieurs coureurs

Les cardinalités maximales valent toutes obligatoirement n

Décomposition en type-associations binaires en remplaçant le type-association n-aire par un type-entité



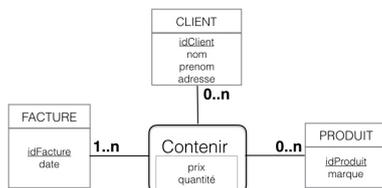
11 / 30

Modèle conceptuel des données

Encore plus loin sur les type-associations

Type-associations n-aires

Exercice : pourquoi ce schéma est-il inapproprié ? Comment le corriger ?



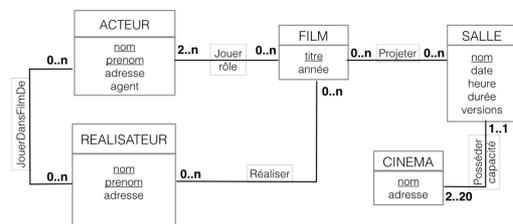
emprunté à L. Audibert

12 / 30

Modèle conceptuel des données

Exemple d'un modèle mal défini

Exercice : Lister les erreurs de modélisation et incohérences de ce modèle, et proposer un modèle bien formé



emprunté à L. Audibert

13 / 30

Modèle conceptuel des données

Exemple de gestion des hôpitaux d'un département

La direction départementale des Bouches du Rhône souhaite informatiser la gestion des hôpitaux du département. Chaque hôpital a des activités de soin dans les services médicaux et des activités de recherche dans les laboratoires.

Les médecins employés sont obligatoirement rattachés à un seul hôpital. Ils ont le choix entre trois types de fonctions : consultant indépendant, médecin praticien, médecin chercheur. Les consultants indépendants ne sont rattachés à aucune structure interne de laboratoire ou de service. Les praticiens sont rattachés à un seul service. Les chercheurs sont rattachés à un laboratoire unique. Les fonctions de praticien et de chercheur peuvent être cumulées. Lorsqu'un médecin est consultant indépendant ou médecin praticien, il se voit confier la responsabilité d'un ou plusieurs patients. Dans certains cas nécessitant des traitements complexes, un même patient peut être suivi par plusieurs médecins.

Les hôpitaux, en plus de leurs nom et adresse, sont identifiés au moyen d'un code. De même, chaque service hospitalier et chaque laboratoire sont décrits par un code d'identification et un nom. On connaît, pour chaque médecin, son numéro matricule, ses coordonnées (nom, prénom, sexe, adresse, date de naissance) et sa spécialité (urologie, gynécologie, gériatrie, pédiatrie, etc.).

Enfin, chaque patient possède un numéro de sécurité sociale, un nom, un prénom, une adresse, une date de naissance.

Les fonctions que le système devra effectuer sont : la gestion d'activité des médecins, la gestion du suivi des patients, les statistiques départementales (nombre de laboratoires par domaine, nombre de patients par hôpital, etc.)

emprunté à N. Durand

14 / 30

Modèle logique des données

Modèle relationnel

Le MCD ne peut pas être implémenté directement une BD → il faut le transformer en un modèle logique de données (MLD)

- Modèle relationnel proposé par Edgar Frank Codd en 1970
- Garantie de l'indépendance physique et logique
- Base solide pour traiter cohérence et redondance des données
- Modèle le plus utilisé

15 / 30

Modèle logique des données

Modèle relationnel

Le MCD ne peut pas être implémenté directement une BD → il faut le transformer en un modèle logique de données (MLD)

- Modèle relationnel proposé par Edgar Frank Codd en 1970
- Garantie de l'indépendance physique et logique
- Base solide pour traiter cohérence et redondance des données
- Modèle le plus utilisé

- 1 Introduction au modèle relationnel
- 2 Passage du modèle entités-associations au modèle relationnel
- 3 Normalisation
- 4 Introduction à l'algèbre relationnelle

15 / 30

Modèle logique des données

Introduction au modèle relationnel

Données organisées sous forme de tables à 2 dimensions, appelées **relations**
Données manipulées par des opérateurs de l'**algèbre relationnelle**
La cohérence des données est définie par des **contraintes d'intégrité**

16 / 30

Modèle logique des données

Introduction au modèle relationnel

Données organisées sous forme de tables à 2 dimensions, appelées **relations**
Données manipulées par des opérateurs de l'**algèbre relationnelle**
La cohérence des données est définie par des **contraintes d'intégrité**

Constituants du modèle relationnel :

Attribut Identifiant décrivant une information
Domaine Ensemble des valeurs possibles pour un attribut
Relation Sous-ensemble du produit cartésien de n domaines ($n > 0$)
Schéma de relation Nom de la relation et liste des attributs et de leur domaine

PERSONNE		
numSecu	nom	prenom
286099587607866	Dupond	Caroline
190117381214351	Delpech	Jean
191748827163849	Ramid	Ahmed
288372979400058	M'bassé	Augustine

PERSONNE(numSecu : entier, nom : chaîne, prenom : chaîne)

16 / 30

Modèle logique des données

Introduction au modèle relationnel

PATIENT		
numSecu	nomPatient	prenomPatient
209112944987389	Durand	Marie
194074728948266	Boulangier	Pierre
201129808798665	Changeur	Sylvie
102038573551749	Cali	Thierry

PATIENT(numSecu : entier, nomPatient : chaîne, prenomPatient : chaîne)

Définitions complémentaires :

Degré nombre d'attributs
Occurrence, enregistrement ou n-uplet Ligne de la table représentant la relation
Cardinalité Nombre d'occurrences
Clé candidate Ensemble minimal d'attributs identifiant une occurrence de la relation
Clé primaire Une des clés candidates
Clé étrangère Un ou plusieurs attributs constituant une clé candidate dans une autre relation

17 / 30

Modèle logique des données

Introduction au modèle relationnel

MEDECIN					
matricule	numSecu	nomMedecin	prenomMedecin	ville	specialite
12345	286099587607866	Dupond	Caroline	Marseille	pédiatrie
29817	190117381214351	Delpech	Jean	Marseille	dermatologie
73628	190117381214351	Ramid	Ahmed	Marseille	dermatologie
84627	288372979400058	M'bassé	Augustine	Marseille	cardiologie

Exercice : Donner les clés candidates de la relation MEDECIN, proposer un schéma pour cette relation

18 / 30

Modèle logique des données

Introduction au modèle relationnel

MEDECIN					
matricule	numSecu	nomMedecin	prenomMedecin	ville	specialite
12345	286099587607866	Dupond	Caroline	Marseille	pédiatrie
29817	190117381214351	Delpech	Jean	Marseille	dermatologie
73628	190117381214351	Ramid	Ahmed	Marseille	dermatologie
84627	288372979400058	M'bassé	Augustine	Marseille	cardiologie

Exercice : Donner les clés candidates de la relation MEDECIN, proposer un schéma pour cette relation

PATIENT(numSecu, nomPatient, prenomPatient)

MEDECIN(Matricule, numSecu, nomMedecin, prenomMedecin, ville, specialite)

CONSULTE(#Matricule, #numSecu)

La clé primaire de la relation CONSULTE est constituée de deux clés étrangères

18 / 30

Modèle logique des données

Introduction au modèle relationnel

Contraintes d'intégrité

- Contrainte s'appliquant à toutes les données d' une BD
- Doit être vérifiée à chaque instant
- De différents types
 - 1 de domaine
 - 2 de non-nullité
 - 3 de vérification
 - 4 d'unicité de clé
 - 5 de référence

Exemples :

Matricule de la relation MEDECIN est un entier (contrainte de domaine)

#Matricule, #numSecu, clés étrangères de la relation CONSULTE, sont liées à une contrainte de référence

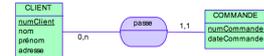
Nous reviendrons sur les contraintes d'intégrité avec leur spécification en SQL

19 / 30

Modèle logique des données

Passage du MCD au MLD

- 1 Une entité du MCD devient une relation du MLD
- 2 Une association de type 1 : n (cardinalités maximales à d'un côté, et à n de l'autre) se traduit par une clé étrangère dans la relation correspondante à l'entité côté 1. Cette clé étrangère fait référence à la clé primaire de la relation correspondante à l'autre entité.



CLIENT(numClient, nom, prenom, adresse)
COMMANDE(numCommande, dateCommande, #numClient)

- 3 Une association de type n : n (cardinalités maximales à n des 2 côtés) se traduit par une relation dont la clé primaire est composée des clés étrangères référençant les relations correspondant aux entités liées par l'association. Les éventuels attributs de l'association deviennent des attributs de la relation.



COMMANDE(numCommande, dateCommande)
PRODUIT(idProduit, libelleProduit)
CONCERNE(#numCommande, #dateProduit, quantite)

on pourra renommer la relation CONCERNE par un nom plus significatif e.g. LIGNE_DE_COMMANDE

emprunté à P. Chochois

20 / 30

Modèle logique des données

Passage du MCD au MLD

Exemples d'associations binaires :

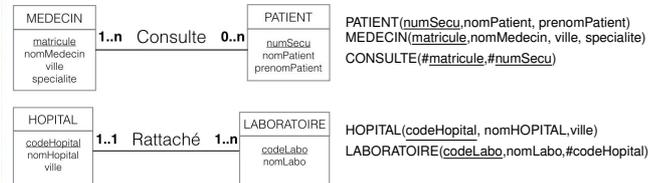


21 / 30

Modèle logique des données

Passage du MCD au MLD

Exemples d'associations binaires :

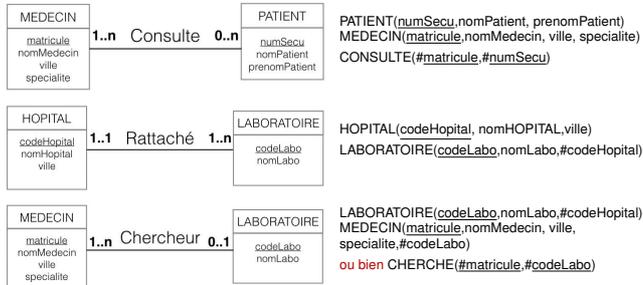


21 / 30

Modèle logique des données

Passage du MCD au MLD

Exemples d'associations binaires :

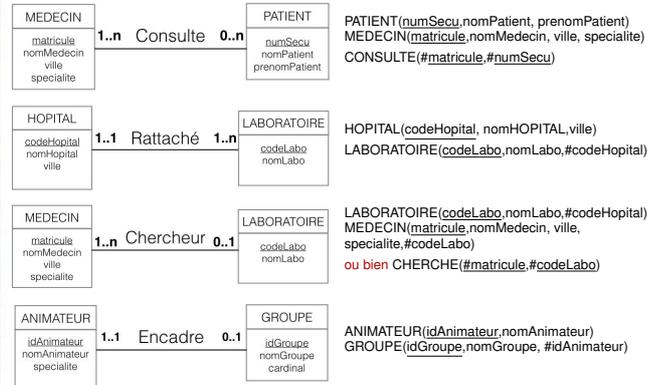


21 / 30

Modèle logique des données

Passage du MCD au MLD

Exemples d'associations binaires :



exemples empruntés à N. Durand 21 / 30

Modèle logique des données

Passage du MCD au MLD

Exemples d'association n-aire :



22 / 30

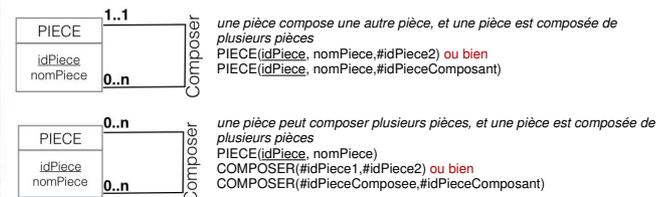
Modèle logique des données

Passage du MCD au MLD

Exemples d'association n-aire :



Exemples d'associations réflexives :

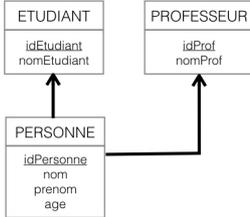


22 / 30

Modèle logique des données

Passage du MCD au MLD

Exemple d'une association de spécialisation :



PERSONNE(idPersonne, nom, prenom, age)
ETUDIANT(idEtudiant, nomEtudiant, #idPersonne)
PROFESSEUR(idProf, nomProf, #idPersonne)
ou bien
ETUDIANT(idEtudiant, nomEtudiant, prenom, age)
PROFESSEUR(idProf, nomProf, prenom, age)

23 / 30

Modèle logique des données

Normalisation

On veut éviter

- la redondance des données
- les incohérences lors des mises-à-jour
- les anomalies lors des suppressions ou insertions

La normalisation repose sur l'analyse des dépendances entre attributs
Elle consiste à décomposer les relations sujettes à redondance

Exemple :

ISBN	Biblio	Titre	Editeur	Nbr
2-100	Luminy	C++	Eyrolles	2
2-101	St Charles	Java	Vuibert	4
2-102	Luminy	PHP	Dunod	3
2-100	Aix	C++	Eyrolles	1

- **Insertion** : un livre (ISBN) ne peut être ajouté que s'il est présent dans une bibliothèque
- **Modification** : si l'éditeur d'un livre est modifié, il faut le modifier dans tous les enregistrements contenant ce livre
- **Suppression** : si le seul enregistrement contenant un livre est supprimé, les informations sur ce livre sont perdues

exemple emprunté à N. Durand

24 / 30

Modèle logique des données

Normalisation – Dépendances fonctionnelles

- 1 **Dépendance fonctionnelle (DF)** : $R(a_1, a_2, \dots, a_n)$ schéma de relation, et X, Y sous-ensembles de $\{a_1, a_2, \dots, a_n\}$
 Y dépend fonctionnellement de X ssi des valeurs identiques de X impliquent des valeurs identiques de Y . **Notation** : $X \rightarrow Y$
une occurrence de X peut être associée à une unique occurrence de Y , i.e., X détermine Y

25 / 30

Modèle logique des données

Normalisation – Dépendances fonctionnelles

- 1 **Dépendance fonctionnelle (DF)** : $R(a_1, a_2, \dots, a_n)$ schéma de relation, et X, Y sous-ensembles de $\{a_1, a_2, \dots, a_n\}$
 Y dépend fonctionnellement de X ssi des valeurs identiques de X impliquent des valeurs identiques de Y . **Notation** : $X \rightarrow Y$
une occurrence de X peut être associée à une unique occurrence de Y , i.e., X détermine Y
- 2 **DF triviale** : $X \rightarrow Y$ est triviale si $Y = \emptyset$ ou $Y \subseteq X$

25 / 30

Modèle logique des données

Normalisation – Dépendances fonctionnelles

- 1 **Dépendance fonctionnelle (DF)** : $R(a_1, a_2, \dots, a_n)$ schéma de relation, et X, Y sous-ensembles de $\{a_1, a_2, \dots, a_n\}$
 Y dépend fonctionnellement de X ssi des valeurs identiques de X impliquent des valeurs identiques de Y . Notation : $X \rightarrow Y$
une occurrence de X peut être associée à une unique occurrence de Y , i.e., X détermine Y
- 2 **DF triviale** : $X \rightarrow Y$ est triviale si $Y = \emptyset$ ou $Y \subseteq X$
- 3 **DF élémentaire** : $X \rightarrow \{a\}$ avec $a \notin X$ est élémentaire si $\exists Y \subset X$ tel que $Y \rightarrow \{a\}$
la cible est un attribut unique, et la source ne contient pas d'attributs superflus

25 / 30

Modèle logique des données

Normalisation – Dépendances fonctionnelles

- 1 **Dépendance fonctionnelle (DF)** : $R(a_1, a_2, \dots, a_n)$ schéma de relation, et X, Y sous-ensembles de $\{a_1, a_2, \dots, a_n\}$
 Y dépend fonctionnellement de X ssi des valeurs identiques de X impliquent des valeurs identiques de Y . Notation : $X \rightarrow Y$
une occurrence de X peut être associée à une unique occurrence de Y , i.e., X détermine Y
- 2 **DF triviale** : $X \rightarrow Y$ est triviale si $Y = \emptyset$ ou $Y \subseteq X$
- 3 **DF élémentaire** : $X \rightarrow \{a\}$ avec $a \notin X$ est élémentaire si $\exists Y \subset X$ tel que $Y \rightarrow \{a\}$
la cible est un attribut unique, et la source ne contient pas d'attributs superflus
- 4 **DF directe** : $X \rightarrow Y$ est directe si $\exists W$ tel que $X \rightarrow W$ et $W \rightarrow Y$ la dépendance entre X et Y ne peut être obtenue par transitivité

Restent les 4^{ème} et 5^{ème} formes normales et les notions d'algèbre relationnelle → dans un cours prochain !

25 / 30

Modèle logique des données

Normalisation – Dépendances fonctionnelles

Exemples de DF

VOITURE(numImmatr, marque, type, puissance, annee)

numImmatr → {marque, type, puissance, annee} V/F??

{marque, type, puissance, annee} → numImmatr V/F??

type → marque V/F??

PROFESSEUR(idProf, nom, prenom, discipline, tel, email)

idProf → {nom, prenom, discipline, tel, email} V/F??

{prenom, discipline} → idProf V/F??

tel → idProf V/F??

26 / 30

Modèle logique des données

Normalisation – Dépendances fonctionnelles

Propriétés des DF

- **Réflexivité** si $Y \subseteq X$ alors $X \rightarrow Y$
- **Augmentation** si $X \rightarrow Y$ alors $X \cup Z \rightarrow Y \cup Z$
- **Transitivité** si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$
- **Pseudo-transitivité** si $X \rightarrow Y$ et $W \cup Y \rightarrow Z$ alors $W \cup X \rightarrow Z$
- **Décomposition** si $X \rightarrow Y \cup Z$ alors $X \rightarrow Y$ et $X \rightarrow Z$

27 / 30

Modèle logique des données

Normalisation - formes normales traitant de redondances expliquées par des DF

- **1^{ère} forme normale 1FN** : une relation est en 1FN ssi tout attribut contient une valeur atomique

```
PERSONNE (idPerson, nomPerson, prenomPerson, adresse)
⇒ PERSONNE (idPerson, nomPerson, prenomPerson, rue, numRue, codePostal, ville)
```

28 / 30

Modèle logique des données

Normalisation - formes normales traitant de redondances expliquées par des DF

- **1^{ère} forme normale 1FN** : une relation est en 1FN ssi tout attribut contient une valeur atomique

```
PERSONNE (idPerson, nomPerson, prenomPerson, adresse)
⇒ PERSONNE (idPerson, nomPerson, prenomPerson, rue, numRue, codePostal, ville)
```

- **2^{ème} forme normale 2FN** : une relation est en 2FN ssi elle est en 1FN et toutes les DF entre sa clé et les autres attributs sont élémentaires (chaque attribut (non-clé) dépend totalement de la clé primaire)

Une relation avec une clé primaire réduite à un attribut, ou contenant tous les attributs est en 2FN

```
AFFECTATION (idPerson, idEtabl, nomPerso, prenomPerso, nomEtabl)
⇒ PERSONNE (idPerson, nomPerson, prenomPerson)
ETABLISSEMENT (idEtabl, nomEtabl)
AFFECTATION (idPerson, idEtabl)
```

28 / 30

Modèle logique des données

Normalisation - formes normales traitant de redondances expliquées par des DF

- **1^{ère} forme normale 1FN** : une relation est en 1FN ssi tout attribut contient une valeur atomique

```
PERSONNE (idPerson, nomPerson, prenomPerson, adresse)
⇒ PERSONNE (idPerson, nomPerson, prenomPerson, rue, numRue, codePostal, ville)
```

- **2^{ème} forme normale 2FN** : une relation est en 2FN ssi elle est en 1FN et toutes les DF entre sa clé et les autres attributs sont élémentaires (chaque attribut (non-clé) dépend totalement de la clé primaire)

Une relation avec une clé primaire réduite à un attribut, ou contenant tous les attributs est en 2FN

```
AFFECTATION (idPerson, idEtabl, nomPerso, prenomPerso, nomEtabl)
⇒ PERSONNE (idPerson, nomPerson, prenomPerson)
ETABLISSEMENT (idEtabl, nomEtabl)
AFFECTATION (idPerson, idEtabl)
```

- **3^{ème} forme normale 3FN** : une relation est en 3FN ssi elle est en 2FN et tout attribut non-clé n'est pas en DF directe avec un ensemble d'attributs non-clé

```
ETUDIANT (idEtudiant, nomEtudiant, prenomEtudiant,
etablissement, villeEtablissement)
⇒ ETUDIANT (idEtudiant, nomEtudiant, prenomEtudiant, #idEtablissement)
où idEtablissement est une clé étrangère faisant ref. à la relation ETABLISSEMENT
ETABLISSEMENT (idEtablissement, villeEtablissement)
```

28 / 30

Modèle logique des données

Normalisation - formes normales traitant de redondances expliquées par des DF

- **Forme normale de Boyce-Codd BC** : une relation est en BC ssi les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé détermine un attribut non-clé. La BC tient compte des dépendances entre parties de clé et entre attribut non-clé et partie de clé.

```
LYCEE (nomLycee, department, ville, capacite)
l'attribut non-clé ville d'etermine une partie de la clé (department)
⇒ LYCEE (nomLycee, #ville, capacite)
VILLE (ville, departement)
```

29 / 30

Références

- AUDIBERT L. Bases de données, de la modélisation au SQL, Ellipses, 2009
- GARDARIN G., Bases de données, Eyrolles, 2003
- HAINAUT, J-L. Bases de données - Concepts, utilisation et développement, Dunod
- LAROUSSE N., Création de bases de données, Collection Informatique, Pearson Education, 2006
- Support de cours de Nicolas DURAND (Polytech Marseille, 2016)
- Support de cours de Henri Méloni & Thierry Spriet (Université d'Avignon et des Pays de Vaucluse, 2009)