

M1104 - Introduction au bases de données

Partie - Modélisation de données

Analyse et Conception des Systèmes d'Information
Le modèle relationnel de Codd

J. Christian Attiogbé

Septembre 2008, maj 09/2016



Plan de ce cours

Plan du chapitre

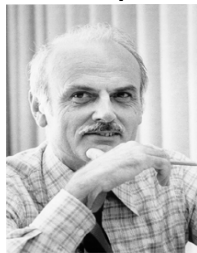
- 1 Modèle relationnel de Codd
 - Introduction
 - Méthodologie
 - Notions de base du modèle relationnel
 - Exemples de modèles
 - Opérations sur les relations
 - Transformation des modèles EAP en modèles de CODD
 - Qualité et normalisation

Modèle relationnel de Edgar Codd (Prix Turing, 1981)

Modèle de données relationnel

- Basé sur l'**algèbre relationnelle**
- Fondements mathématiques: **théorie des ensembles**
- Modèle très utilisé pour la mise en œuvre des systèmes d'information : la **construction des bases de données** (relationnelles)

(dans ce cours nous nous limiterons à la partie modélisation et n'allons pas aborder les bases de données qui font l'objet d'un autre cours)



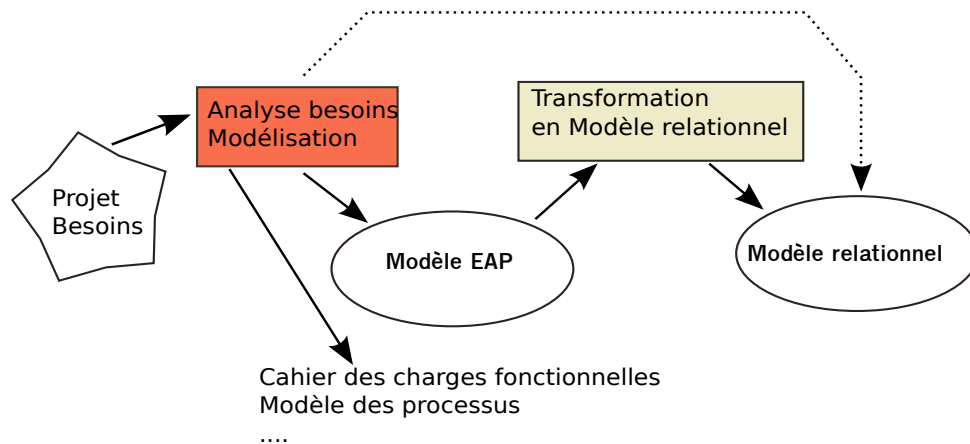
Du modèle EA au modèle relationnel

- Modèle relationnel (**n-aire**) : introduit par Codd pour modéliser et construire les bases de données.
- Les **entités** de base sont des **ensembles**,
- Des **relations** lient les ensembles
- Codd représente une relation par une table dont les colonnes sont nommées.
- Passage du modèle EAP au modèle de Codd :

Modèle Entité-Association	Modèle relationnel
Entité	Entité
Association	Relation ou Table
Propriété	Attribut
Identifiant	Clé

☛ il y a une méthode pour la transformation (à voir plus tard)

Processus de modélisation



Notions de base

Domaines : ensemble de valeurs

Attribut : donne une valeur (ou propriété), c'est une fonction dans un domaine de valeurs.

Relation entre domaines : sous-ensemble du produit cartésien entre les domaines des attributs

Exemple : Une relation *Etudiant(Nom, Prenom, ddn, serieBac, dateBac)*

$$NOM \times PRENOM \times DDN \times SERIEBAC \times DATEBAC$$

Notez qu'on a fait le produit sur les **domaines** (avec leurs noms introduits en majuscule, mais il faut les définir !)

Notions de base : table

Pratiquement, une relation est représentée par une table

Nom	Prénom	ddn	serieBac	dateBac
Abraca	Dabra	03071988	S	2006

- les colonnes de la table représentent les attributs.
- Les lignes correspondent aux éléments (n-uplets) de la relation ; Elles sont appelées aussi **occurrences** de la relation.
(Abraca, Dabra, 03071988, S, 2006)

On parle de **schéma relationnel** ou **table relationnelle**

Notions de base : table

- les colonnes de la table peuvent occuper n'importe quelle position

Nom	Prénom	ddn	serieBac	dateBac
Abraca	Dabra	03071988	S	2006

est équivalent à

Nom	serieBac	Prénom	ddn	dateBac
Abraca	S	Dabra	03071988	2006

Chaque ligne (l_i) de la table dénote **un énoncé vrai**.

La relation R a un sens : c'est le **prédicat** $P(l)$ qui lie ses attributs.

Ce sens est donné par le concepteur de la relation et on a

$$\forall l. l \in R \Rightarrow P(l)$$

Notions de base : table, type des attributs

- Un **attribut peut être élémentaire ou non** (ie décomposable).
On préférera des attributs élémentaires (en normalisant).
- Un attribut peut être **non calculable** ou calculable.
Lorsqu'un attribut peut être calculé à partir d'autres attributs, on n'en fait pas un attribut de la relation.
Par exemple à partir d'un prix U, d'un taux, et d'une quantité on peut calculer un (sous-)total, HT, TTC, etc
- **A chaque attribut est associé son domaine** (de valeurs) ;
Nom : NOM, serieBac : SERIEBAC, etc
plusieurs attributs peuvent avoir le même domaine.
(par convention, on notera en majuscule les domaines ou types)

Notions de base : table

Attention à l'usage des tables relationnelles.

- Une relation est un sous-ensemble d'un produit cartésien
Etudiant(Nom, Prenom, ddn, serieBac, dateBac)

$$Etudiant \subseteq NOM \times PRENOM \times DDN \times SERIEBAC \times DATEBAC$$

- Dans un produit cartésien, il y a un ordre pour les ensembles.
- Or dans le modèle de Codd, les colonnes peuvent occuper différentes positions !
donc le **nom de la colonne est absolument nécessaire**.

En effet

$$A \times B \times C \neq C \times A \times B$$

Notions de base : cardinalité, degré

cardinalité

La cardinalité (d'une relation) est le nombre de lignes dans la table (de la relation).

...comme le nombre d'éléments dans un ensemble.

La table donne la relation en extension, donc on a tous ses éléments.

degré

Le degré d'une relation est le nombre de colonnes dans la table (de la relation).

Notions de base : clé

Motivation : **implantation**

Considérons maintenant que chaque étudiant a un numéro qui n'identifie que lui.

La relation devient

Etudiant(NumEtudiant, Nom, Prenom, ddn, serieBac, dateBac)

NumEtudiant	Nom	Prénom	ddn	serieBac	dateBac
2007SleO487	Abraca	Dabra	03071988	S	2006

- Le numéro **2007SleO487** n'identifie qu'**un et un seul étudiant** ;
- On ne peut donc avoir dans la table deux lignes ayant le même numéro

Notions de base : clé

Dans le formalisme de Codd, pour exprimer qu'un numéro d'étudiant ne correspond qu'à un seul étudiant, avec nom, prénom, ddn, serieBac, dateBac on utilise la notion de **clé** ;

La **clé primaire** c'est un attribut (ou un groupe d'attributs, ou de constituants) auquel n'est associé qu'un seul n-uplet (des autres attributs).

Vous auriez noté, il s'agit d'une **fonction** entre la clé et les autres attributs.

De plus, la **fonction est totale** : on ne peut avoir de numéro auquel ne corresponde un n-uplet des autres attributs.

On parle de **dépendance fonctionnelle** entre la clé et les autres attributs.

Etudiant(NumEtudiant, Nom, Prenom, ddn, serieBac, dateBac)

Dépendance fonctionnelle = contraintes d'intégrité fonctionnelle (CIF)

Notions de base : clé

Etudiant(NumEtudiant, Nom, Prenom, ddn, serieBac, dateBac)

La clé est soulignée.

Fonction

NumEtudiant → *NOM* × *PRENOM* × *DDN* × *SERIEBAC* × *DATEBAC*

Il faut la nommer pour identifier la dépendance fonctionnelle.

Une **clé étrangère** est un attribut qui est une clé primaire d'une autre relation.

Les clés étrangères permettent de relier les tables entre elles.

Notions de base : propriétés des dépend. fonct.

Couverture minimale

La couverture minimale des dépendances fonctionnelles pour une relation, c'est l'ensemble des dépendances fonctionnelles obtenues en éliminant toutes les redondances dans la relation et toutes les dépendances qu'on peut déduire (du fait des propriétés des fonctions).

Dépendance fonctionnelle élémentaire

On a une dépendance fonctionnelle élémentaire entre les constituants X, et les constituants Y d'une relation R(X,Y) s'il n'existe pas un sous-ensemble de X dont Y dépend fonctionnellement.

Autrement dit, une partie de la clé n'identifie pas les autres attributs.

Si $A_1, A_2, A_3 \rightarrow A_4, A_5$

aucun sous ensemble de $\{A_1, A_2, A_3\} \rightarrow A_4, A_5$

Exemples de modèles relationnels

Ville(codePostal, nomVille, nomHabitant)

une **instance** de la relation

codePostal	nomVille	nomHabitant
64000	Pau	pallois
34500	Béziers	bitterois
12000	Rodez	ruthénois
85800	St-gilles croix de vie	gillocruciens
64200	Biarritz	biarrots
05000	Gap	gapençais
56000	Vannes	vannetais
56400	Auray	alréens

Chaque ligne représente une **proposition vraie** !

64000 est le code postal de Pau et les habitants sont les pallois.

Exemples de modèles relationnels (suite)

Ville(codePostal, nomVille, nomHabitant)

une **instance** de la relation

codePostal	nomVille	nomHabitant
56400	Lebono	bonoviste
01260	Lochieu	Chicaneur
01260	Songieu	Songiolan
05700	Sigottier	...

Est-ce une bonne modélisation ? (choix de la clé) ?

Non ! le code postal n'est pas unique ; c'est le **code communal** !!

👉 ne pas se fier à ses connaissances, mais à celles des experts du domaine

Exemples de modèles relationnels

Mais que veulent dire les noms des colonnes *codePostal*, *nomVille*, *nomHabitant* ?

Ce sont des constituants de la relation, mais notez que :

nomVille(34500) = Beziers *nomHabitant*(05000) = gapençais

- attributs ? constituants ? ensembles ? codomaines ? ...
- le nom de colonnes désigne bien un **ensemble d'éléments** !
- donc **codomaine de fonction**, ou le nom de la fonction qui donne la valeur qui est dans le codomaine.

Exemples de modèles relationnels

commande(numCommande, numClient, raisonSociale, adresse,
numPprod, prixU, designation, qtte)

- Que veulent dire les noms des colonnes numCommande, numClient, raisonSociale, adresse, numPprod, prixU, designation, qtte ?
- Peut-on structurer la relation autrement ? (en plusieurs parties ?)

Exemples de modèles relationnels

Etudiant(numEtudiant)
Controle(numControle, numMatiere, coeffControle)
Matiere(numMatiere, nomMatiere, coeffMatiere)
Notes(numControle, numEtudiant, Note)

- Dans un modèle, élaborer **une relation ou plusieurs ?**
- **Comment concevoir les modèles ?** (1 relation, plusieurs ?)
- Comment faire pour **décomposer une relation en plusieurs ?**
- Faire dès le début plusieurs relations ?
- Quelles sont les propriétés d'**un bon modèle ?**

Méthodes de normalisation

Exemples de modèles relationnels

Controle

numControle	numMatiere	coeffControle
1	1.1	1
2	1.2	1
3	1.3	2

Etudiant

numEtudiant
2007SleO487
2007SleO512
2007SSeO177

Matiere

numMatiere	nomMatiere	coeffMatiere
1.1	Archi	2
1.2	MoDyn	2
1.3	SD1	2

Notes

numControle	numEtudiant	Note
1	2007SleO487	13
1	2007SSeO177	10
1	2007SleO512	15
2	2007SSeO177	12



Opérateurs sur les relations

Codd a proposé un **langage** comprenant :

- des opérateurs ensemblistes : **union**, **intersection**, **différence entre relations** de même degré
- des opérateurs algébriques
 - des opérateurs unaires (qui agissent sur une seule relation) : **sélection/suppression d'une ligne** selon des conditions, projection de certaines colonnes
 - des opérateurs binaires (qui agissent sur 2 relations) : la **jointure de relations**, la **division**
 -

On l'aura compris, une relation étant un ensemble, on bénéficie des opérations ensemblistes



Transformation EAP -> Relationnel

Règle de base 1 : Toute ENTITE est transformée en une relation dont la clé primaire et les attributs viennent de l'ENTITE.

Règle de base 2 : Toute ASSOCIATION est transformée en une relation dont la clé primaire est formée par la juxtaposition de tous les identifiants (appelés alors clés étrangères) des ENTITE qui constituent l'association.

Lorsque l'ASSOCIATION possède des attributs, ils sont transformés en attributs de la relation.

Regles auxilliaires :

Règle d'optimisation : Une association binaire Aab (entre les entités Ea, Eb) avec une cardinalité 0,1 ou 1,1 pour l'entité Ea et une cardinalité quelconque pour Eb, ne donne pas lieu à une traduction en relation ;

L'identifiant de Eb (devient clé étrangère) ainsi que les attributs de l'association Aab sont ajoutés comme attributs de la relation qui traduit l'entité Ea.



Qualité d'un bon modèle (relationnel)

- facilement lisible et compréhensible,
- simple,
- permet de répondre aisément aux questions que l'on se pose,
- dépourvu d'ambigüité,
- dépourvu de redondance,
- ...



Normalisation

Normalisation

Normaliser c'est **décomposer** une relation en des relations plus simples dépourvues de redondances.

C'est le principe **Diviser pour mieux régner** présent dans toute démarche pour contrer la complexité.

Relation normalisée - 1ere forme normale

On appelle **relation normalisée (ou en première forme normale)** une relation dont les constituants sont tous des valeurs simples (d'un domaine) et non des noms d'autres relations.

Dans la pratique, c'est comme si on esquissait un brouillon du modèle, puis on l'améliorait jusqu'à le rendre propre.



Normalisation

Etudiant(numEtudiant, Note)

Note(nomMatiere, numControle, noteSur20)

Cette relation *Etudiant* **n'est pas normalisée** ; en effet *Note* qui est un des constituants de *Etudiant* est une relation et non un constituant simple.

Etudiant(numEtudiant, nomMatiere, numControle, noteSur20)

est normalisée et **est en première forme normale**



Normalisation : les formes normales des relations

Vous étudierez aussi dans le cours de Base de Données...
Ici on en fait une introduction.

- Plusieurs formes normales
- 1ere, 2eme, 3eme, 4eme, Boyce-Codd, 5eme, ... formes normales
- On va de la moins précise (1ere) à la plus précise (5ème)

1ere forme normale

On appelle **relation normalisée (ou en première forme normale)** une relation dont les constituants sont tous des domaines simples (ou ensemble) et non des noms d'autres relations.

1ere, 2eme, 3eme, 4eme, 5eme.... formes normales

Deuxième forme normale

La relation est en première forme normale et les constituants qui ne sont pas clé, dépendent fonctionnellement entièrement de la clé (et nom d'une partie).

Troisième forme normale

La relation est en deuxième forme normale et les constituants non clé ne dépendent pas fonctionnellement de façon transitive de la clé.

Ce serait redondant (car la composée des fonctions permet de retrouver la transitivité entre les constituants)

2eme et 3eme formes normales : exemples

Soit la relation :

$R_{Fact}(\underline{NumComm}, NumCli, RaiSociale, Adresse, NumProd, PrixUnit, Design, QtteComm)$

décomposée en plusieurs relations en **troisième forme normale** :

$R_{Prod}(\underline{NumProd}, Design, PrixUni)$

$R_{Client}(\underline{NumCli}, RaiSociale, Adresse)$

$R_{Comm}(\underline{NumComm}, NumCli)$

$L_{Prod}(\underline{NumProd}, NumComm, QtteComm)$

4eme, 5eme formes normales

Quatrième forme normale

La relation est en troisième forme normale et il n'y a pas de dépendances multivaluées.

Les dépendances multivaluées : Soit $R(A, B, C)$ une relation, on a une dépendance multivaluée dans R si et seulement si, chaque fois que (a, b, c) et (a, b', c') apparaissent dans R , alors (a, b', c) et (a, b, c') apparaissent dans R .

Exemple

NumEmpl	Competence	Langue
4	Marketing	français
4	Marketing	allemand
4	Marketing	mandarin
4	Economie	français
4	Economie	allemand
4	Economie	mandarin

Dépendance multivaluée et 4eme forme normale

Dans l'exemple, les compétences sont indépendantes des langues parlées, Il y a une dépendance multivaluée,

notée **NumEmpl** → **Compétence/Langue**

Pour une occurrence de NumEmpl, on a dans la relation un ensemble de valeurs de compétence qui est indépendant des valeurs que peut prendre Langue.

il y a 2 relations indépendantes :

- un employé est compétent en ... ;
- un employé parle les langues

NumEmpl	Compétence
4	Marketing
4	Economie

NumEmpl	Langue
4	français
4	allemand
4	mandarin

Compléments : forme normale de Boyce-Codd

Forme normale de Boyce-Codd

Une relation est en forme normale de Boyce-Codd si et seulement si : elle est en troisième forme normale, et les dépendances fonctionnelles qu'elle comporte, et elles seules, sont celles dans lesquelles une clé détermine un attribut.

Lorsque la clé est composée de plusieurs attributs, les attributs composant la clé, ne doivent pas dépendre d'autres attributs. On définit aussi la quatrième forme normale à partir de la forme normale de Boyce-Codd.

Cinquième forme normale

Cinquième forme normale

Une relation de degré n ($n > 2$) est en cinquième forme normale si et seulement si : elle est en quatrième forme normale, et il n'est pas possible de retrouver l'ensemble de ses occurrences par jointure sur les occurrences des relations partielles prises deux à deux.

(On ne peut pas la décomposer en deux sans perte d'information)

La normalisation en cinquième forme normale **conduit à décomposer une relation complexe en plusieurs (> 2) relations plus simples.**

Dépendance mutuelle et 5eme forme normale

Cinquième forme normale

Une relation de degré n ($n > 2$) est en cinquième forme normale si elle est en quatrième forme normale et il y a des dépendances mutuelles.

Dépendance mutuelle : des attributs dépendent les uns des autres et réciproquement.

Soit une relation $R(X, Y, Z)$

On note $R[X, Z]$ la projection de R sur les constituants X et Z

Dépendance mutuelle (notée $\langle \sim \rangle$)

Soit une relation $R(X, Y, Z)$, $X \langle \sim \rangle Y$ dans la relation R ,

si $(x, y, z) \in R$ et $(x, y', z') \in R$ alors

$(y, z') \in R[Y, Z] \Rightarrow (x, y, z') \in R$ et

$(y', z) \in R[Y, Z] \Rightarrow (x, y', z) \in R$

De plus on a $X \langle \sim \rangle Y$ et $Y \langle \sim \rangle X$

Exemple : relation en 5ème forme normale

Soit la relation *Représente*(*REPRESENTANT*, *PRODUIT*, *SOCIETE*)

$(r, p, s) \in \text{Représente}$ exprime que le représentant r représente le produit p pour la société s .

Soit la table relationnelle suivante

REPRESENTANT	PRODUIT	SOCIETE
r1	p1	s1
r1	p2	s1
r1	p3	s2
r2	p1	s1
r2	p4	s3
r3	p4	s4

On constate qu'on ne peut savoir si une société commercialise un produit, sans connaître au moins un représentant de la société qui représente ce produit.

Un représentant ne travaille pas pour deux sociétés qui commercialisent certains produits identiques.



Exemple : relation en 5ème forme normale

REPRESENTANT	PRODUIT	SOCIETE
r1	p1	s1
r1	p2	s1
r1	p3	s2
r2	p1	s1
r2	p4	s3
r3	p4	s4

Un représentant ne travaille pas pour deux sociétés qui commercialisent certains produits identiques.

La **dépendance mutuelle entre PRODUIT et SOCIETE** peut s'exprimer ainsi :

Lorsqu'un représentant représente un produit pour une société alors il représente ce produit pour toute société pour laquelle il travaille et qui commercialise le produit.



Exemple : relation en 5ème forme normale

REPRESENTANT	PRODUIT	SOCIETE
r1	p1	s1
r1	p2	s1
r1	p3	s2
r2	p1	s1
r2	p4	s3
r3	p4	s4

Il y a une dépendance mutuelle entre PRODUIT et SOCIETE.

$(r1, p1, s1) \in \text{Représente}$

$(r1, p3, s2) \in \text{Représente}$

$(p1, s2) \in \text{Représente}[\text{PRODUIT}, \text{SOCIETE}] \Rightarrow (r1, p1, s2) \in \text{Représente (Vrai)}$ et

$(p3, s1) \in \text{Représente}[\text{PRODUIT}, \text{SOCIETE}] \Rightarrow (r1, p3, s1) \in \text{Représente (Vrai)}$

Un représentant ne travaille pas pour deux sociétés qui commercialisent certains produits identiques.

Exemple : relation en 5ème forme normale

L'existence de la dépendance mutuelle permet de **décomposer la relation en 3 de ses projections** :

ReprésenteProduit[REPRESENTANT, PRODUIT],

ReprésenteSociete[REPRESENTANT, SOCIETE],

CommercialiseProduit[PRODUIT, SOCIETE]

REPR.	PROD.
r1	p1
r1	p2
r1	p3
r2	p1
r2	p4
r3	p4

REPR.	SOC.
r1	s1
r1	s2
r2	s1
r2	s3
r3	s4

PROD.	SOC.
p1	s1
p2	s1
p3	s2
p4	s3
p4	s4

Exemple : relation en 5ème forme normale

Si on n'avait pas la troisième relation on n'aurait pas pu reconstituer la relation initiale

Exercice :

- Faire la jointure des relations
RepresenteProduit[REPRESENTANT, PRODUIT],
RepresenteSociete[REPRESENTANT, SOCIETE],
- Utilisez ensuite **CommercialiseProduit[PRODUIT, SOCIETE]** pour retrouver la relation initiale

Sur la qualité des modèles relationnels de données

Leçons

- Une relation qui **n'est pas en première forme normale**, entraîne éventuellement des **lourdeurs dans l'exploration des données**, car il faut analyser les attributs pour distinguer simple/relation.
- Une relation qui **n'est pas en deuxième forme normale** entraîne éventuellement la **redondance des données** (c'est inutile et pénalisant quand on construit les bases de données).
- Une relation qui **n'est pas en troisième forme normale** (et aussi en forme normale de Boyce-Codd, ou en quatrième forme normale FNBC,) entraîne la **redondance dans les données**.
- Une relation qui **n'est pas en cinquième forme normale** entraîne éventuellement la **perte de données et les informations manquent de précision**.

Etudes de cas

A volonté . . .

Synthèse

- Le modèle relationnel est un des outils rigoureux pour la modélisation des données en vue de l'élaboration d'un système d'information, et la construction de bases de données.
- Les fondamentaux à la base de la méthode sont la théorie des ensembles ; cela lui confère les propriétés de simplicité et de rigueur.
- La normalisation des modèles de données permet d'obtenir des modèles très précis.
- Il y a des algorithmes pour effectuer la normalisation.

Transformation du modèle EA au modèle relationnel

Modèle relationnel (n-aire) : introduit par Codd pour construire les bases de données.

Codd représente une relation par une table (ou tableaux) dont les colonnes sont nommées.

Modèle Entité-Association	Modèle relationnel
Entité	Entité
Association	Table ou relation
Propriété	Attribut
Identifiant	Clé

Bibliographie

H. HABRIAS, *Le modèle relationnel binaire*, Eyrolles, 1988

P. ANDRE, A. VAILLY, *Conception des systèmes d'information ; Panorama des méthodes et des techniques*, Editions Ellipses, janvier 2001, ISBN 2-7298-0479-X