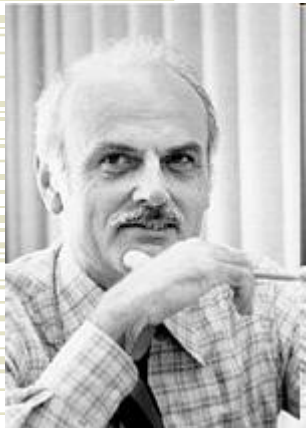


LE MODELE RELATIONNEL



A relational model of Data For Large Shared Data Banks

E.F. Codd, *CACM* 13(6), 1970

1923-2003

1. Concepts pour la description
2. Concepts pour la manipulation
3. Concepts additionnels

1. CONCEPTS DESCRIPTIFS

- ◆ Ensemble de concepts pour formaliser la description d'articles de fichiers plats
- ◆ Modèle standardisé mais extensible (depuis 1986)
 - Introduction de types de données variés (SQL2)
 - Introduction de la dynamique (Triggers), des types non scalaires et objets, requêtes récursives... (SQL3)
 - Introduction du XML (SQL:2003)
 - Amélioration du XML (SQL:2006)
 - Version actuelle : SQL:2011 (fenêtres glissantes, support du temporel)

Approche constructive : Domaine

- ◆ ENSEMBLE DE VALEURS

- ◆ Exemples:

- ENTIER
- REEL
- CHAINES DE CARACTERES
- EUROS
- SALAIRE = {4 000..100 000}
- COULEUR= {BLEU, BLANC, ROUGE}
- POINT = ~~{(X:REEL,Y:REEL)}~~
- TRIANGLE = ~~{(P1:POINT,P2:POINT,P3:POINT)}~~

Produit cartésien de domaines

- ◆ LE PRODUIT CARTESIEN $D_1 \times D_2 \times \dots \times D_n$ EST L'ENSEMBLE DES TUPLES (N-UPLETS) $\langle V_1, V_2, \dots, V_n \rangle$ TELS QUE $V_i \in D_i$

- ◆ Exemple:

- $D_1 = \{\text{Bleu, Blanc, Rouge}\}$
- $D_2 = \{\text{Vrai, Faux}\}$

Bleu	Vrai
Bleu	Faux
Blanc	Vrai
Blanc	Faux
Rouge	Vrai
Rouge	Faux

Relation

- ◆ SOUS-ENSEMBLE DU PRODUIT CARTESIEN D'UNE LISTE DE DOMAINES

- ◆ Une relation est caractérisée par un nom

CoulVins	Coul	Choix
	Bleu	Faux
	Blanc	Vrai
	Rouge	Vrai

- ◆ Exemple:

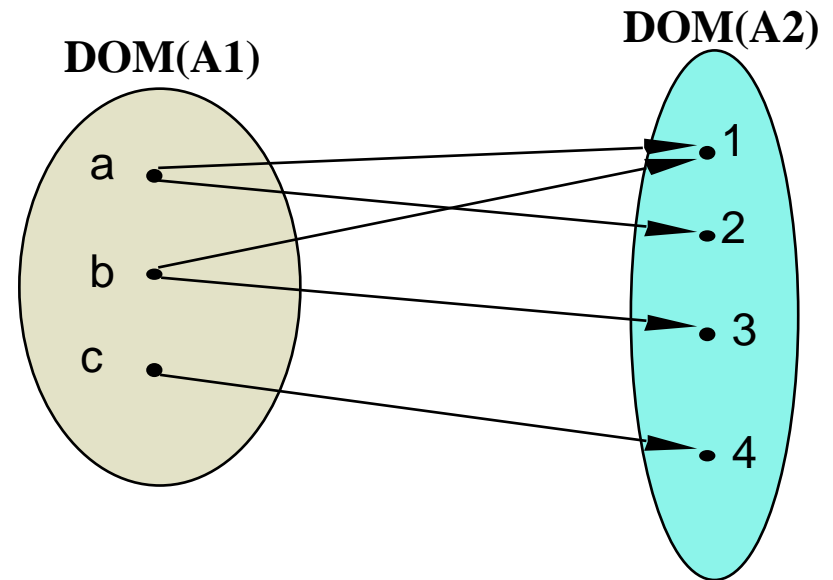
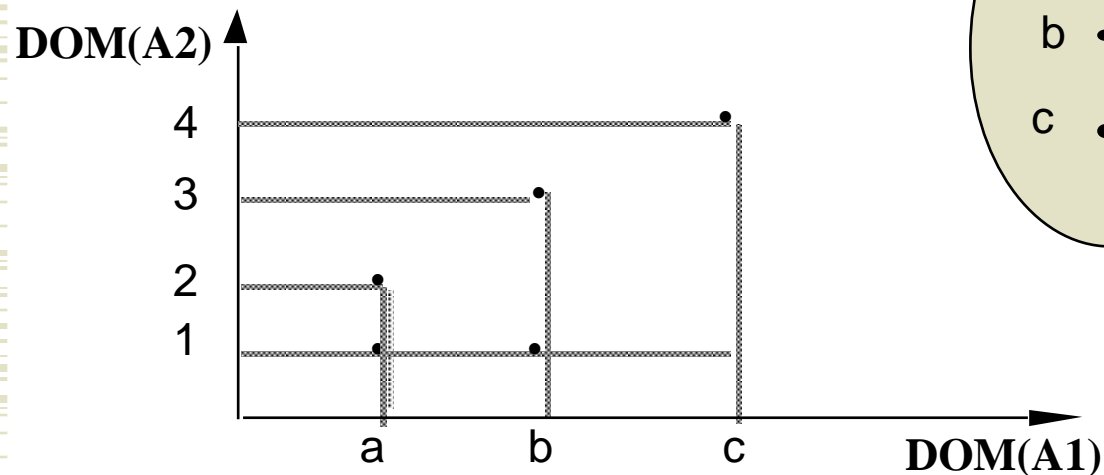
- $D_1 = \text{COULEUR}$
- $D_2 = \text{BOOLEEN}$

Attribut

- ◆ VISION TABULAIRE DU RELATIONNEL
 - Une relation est une table à deux dimensions
 - Une ligne est un tuple
 - Un nom est associé à chaque colonne afin de la repérer indépendamment de son numéro d'ordre
- ◆ ATTRIBUT
 - nom donné à une colonne d'une relation
 - prend ses valeurs dans un domaine

« Théorie » Graphe d'une relation

- ♦ Relation binaire $R(A1, A2)$
- ♦ Une relation n-aire est une généralisation à n dimensions



Exemple de relation

ETUDIANTS	NOM	DATENAISS	VILLE	SECTION
	ANNE	1991	VERSAILLES	INFO
	BERNARD	1993	PARIS	MASS
	CELINE	1993	VERSAILLES	INFO
	DAVID	1981	VERSAILLES	MATH
	EMILIE	1992	VELIZY	MASS

CLE ?

NORMALISATION?

Clé

- ◆ GROUPE D'ATTRIBUTS MINIMUM QUI DETERMINE UN TUPLE UNIQUE DANS UNE RELATION
- ◆ Exemples:
 - Ajouter NUMETU dans ETUDIANTS
 - ...ou bien un ensemble d'attributs dont la valeur est unique!
- ◆ CONTRAINTE D'ENTITE
 - Toute relation doit posséder au moins une clé

Schéma

- ◆ NOM DE LA RELATION, LISTE DES ATTRIBUTS AVEC DOMAINES, ET LISTE DES CLES D'UNE RELATION
- ◆ Exemple:
 - ETUDIANTS(NE: Int, NOM:texte, DATENAISS:entier, VILLE:texte, SECTION:texte)
 - Par convention, la clé primaire est soulignée
- ◆ INTENTION ET EXTENSION
 - Un schéma de relation définit l'intention de la relation
 - Une instance de table représente une extension de la relation
- ◆ SCHEMA D'UNE BD RELATIONNELLE
 - C'est l'ensemble des schémas des relations composantes

Clé Etrangère

- ◆ GROUPE D'ATTRIBUTS DEVANT APPARAÎTRE COMME CLE DANS UNE AUTRE RELATION
- ◆ Les clés étrangères définissent les contraintes d'intégrité référentielles
 - Lors d'une insertion, la valeur des attributs doit exister dans la relation référencée
 - Lors d'une suppression dans la relation référencée les tuples référençant doivent disparaître
 - Elles correspondent aux liens entité-association obligatoires

Exemple de Schéma

◆ EXEMPLE

ETU(NE, NOM, DATENAISS, VILLE)

SECTION (NS, NOMSEC, DEPARTEMENT)

INSCRIPTION(NE, NS, ANNEE, FRAIS)

◆ CLES ETRANGERES

INSCRIPTION.NE REFERENCES ETU.NE

INSCRIPTION.NS REFERENCES SECTION.NS

Diagramme des Liens

ETU	<u>NE</u>	NOM	DATENAIS	VILLE

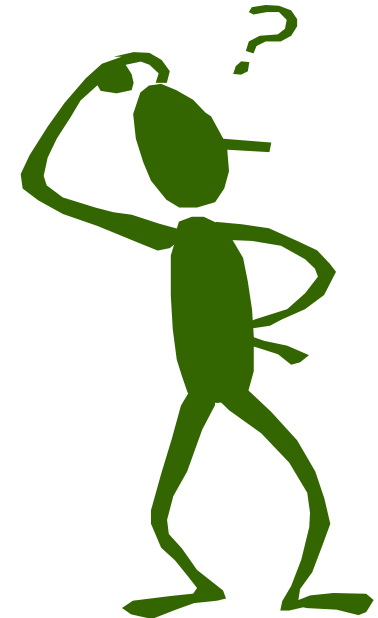
SECTION	<u>NS</u>	NOM	DPT

INSCR.	<u>NE</u>	<u>NS</u>	<u>ANNEE</u>	FRAIS

Concepts Descriptifs : Bilan

- ◆ RELATION ou TABLE
- ◆ ATTRIBUT ou COLONNE
- ◆ DOMAINE ou TYPE
- ◆ CLE
- ◆ CLE ETRANGERE

Questions ?



Synthèse : Create Table

- ♦ CREATION DES TABLES EN SQL

CREATE TABLE <relation name>

(<attribute definition>+)

[{PRIMARY KEY | UNIQUE} (<attribute name>+)]

- ♦ Avec :

<attribute definition> ::= <attribute name> <data type>

[NOT NULL [{UNIQUE | PRIMARY KEY}]]

- ♦ Exemple :

CREATE TABLE ETU

(NE INTEGER PRIMARY KEY,

NOM VARCHAR (32),

DATENAISS INTEGER NOT NULL,

VILLE VARCHAR(64))

Passage de l'E/R au relationnel

Implémentations des entités et associations sous forme de tables :
Méthodologie

- ♦ Transformer toutes les associations n-aires en associations binaires
- ♦ **Chaque entité devient une table**
- ♦ Les attributs correspondent aux colonnes des tables
 - Nom attribut → Nom colonne
 - Ensemble de valeurs → Domaine
 - Clé primaire E/A → Clé primaire de la table
 - Clé candidate E/A → Contrainte UNIQUE de la table

Traduction des associations

◆ Règle de base

- Une association est représentée par une table dont le schéma est le nom de l'association et la liste des clés primaires des entités participantes suivie des attributs de l'association. Ces clés primaires deviennent des clés étrangères de cette nouvelle table.
- Exemples :
 - POSSEDE (N° Ss, N° Veh, Date , Prix)
 - ABUS (Nv, Nb, Date, Quantité)

◆ Amélioration possible

- Regrouper les associations 1:1 --> 1:N avec la classe cible
- Exemple :
 - VOITURE (N°VEH, MARQUE, TYPE, PUISSANCE, COULEUR)
 - POSSEDE (N° SS, N° VEH, DATE, PRIX)
 - regroupées si toute voiture a un et un seul propriétaire

2. CONCEPTS MANIPULATOIRES

- ◆ Un ensemble d'opérations formelles
 - Algèbre relationnelle
- ◆ Ces opérations permettent d'exprimer toutes les requêtes sous forme d'expressions algébriques
- ◆ Elles sont la base du langage SQL
 - Paraphrasage en anglais des expressions relationnelles
 - Origine SEQUEL
- ◆ Ces opérations se généralisent à l'objet
 - Algèbre d'objets complexes

Projection

- ◆ Elimination des attributs non désirés et suppression des tuples en double

ETU	NOM	DN	VILLE
	ANNE	1991	VERSAILLES
	BERNARD	1993	PARIS
	CELINE	1993	PARIS
	DAVID	1991	VERSAILLES
	EMILIE	1993	VELIZY

- ◆ Relation \rightarrow Relation notée:

$$\pi_{A_1, A_2, \dots, A_p}(R)$$

$\pi_{DN, VILLE}(ETU)$

$\pi_{DN, VILLE}(ETU)$	DN	VILLE
	1991	VERSAILLES
	1993	PARIS
	1993	VELIZY

Restriction

- ◆ Obtention des tuples de R satisfaisant un critère Q
- ◆ Relation \rightarrow Relation, notée $\sigma_Q(R)$
- ◆ Q est le critère de qualification de la forme :
 - $A_i \theta \text{ Valeur}$
 - $\theta = \{ = , < , > , \leq , \geq , \neq \}$
- ◆ Il est possible de réaliser des "ou" (union) et des "et" (intersection) de critères simples

Exemple de Restriction

ETU	NOM	DN	VILLE
	ANNE	1991	VERSAILLES
	BERNARD	1993	PARIS
	CELINE	1993	PARIS
	DAVID	1991	VERSAILLES
	EMILIE	1993	VELIZY

$\sigma_{DN>1992}(\text{ETU})$

ETU	NOM	DN	VILLE
	BERNARD	1993	PARIS
	CELINE	1993	PARIS
	EMILIE	1993	VELIZY

Opérations Ensemblistes

- ◆ Opérations pour des relations de même schéma

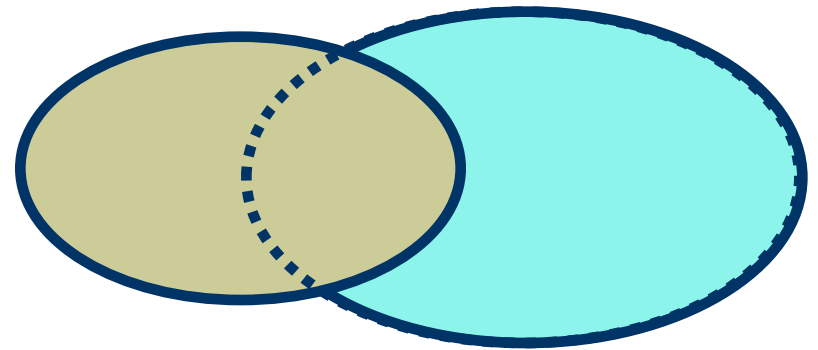
- UNION notée \cup
- INTERSECTION notée \cap
- DIFFERENCE notée $-$

- ◆ Opérations binaires

- Relation X Relation \rightarrow Relation

- ◆ Extension

- Union externe pour des relations de schémas différents
- Ramener au même schéma avec des valeurs nulles



Jointure

- ♦ Composition des deux relations sur un domaine commun
- ♦ Relation X Relation \rightarrow Relation
 - notée \bowtie
- ♦ Critère de jointure
 - Attributs de même nom égaux :
 - Attribut = Attribut (Equi-jointure)
 - Jointure naturelle
 - ♦ Se fait en principe en utilisant une clé étrangère !!!
 - Comparaison d'attributs :
 - Attribut1 θ Attribut2
 - Théta-jointure
- ♦ La jointure peut se voir comme un produit cartésien, combiné à une restriction sur l'attribut de jointure.

Exemple de Jointure

ETU	NOM	DN	VILLE
	ANNE	1991	VERSAILLES
	BERNARD	1993	PARIS
	CELINE	1993	PARIS
	DAVID	1991	VERSAILLES

INFOVILLE	NOMV	DPT
	VERSAILLES	78
	PARIS	75



ETU.VILLE = INFOVILLE.NOMV

R=ETU ⋈ INFOVILLE

ETU.VILLE = INFOVILLE.NOMV



R	NOM	DN	VILLE	NOMV	DPT
	ANNE	1991	VERSAILLES	VERSAILLES	78
	BERNARD	1993	PARIS	PARIS	75
	CELINE	1993	PARIS	PARIS	75
	DAVID	1991	VERSAILLES	VERSAILLES	78

Exemple de Produit Cartésien

$$R = \text{ETU} \times \text{INFOVILLE}$$

ETU	NOM	DN	VILLE
	ANNE	1991	VERSAILLES
	BERNARD	1993	PARIS
	CELINE	1993	PARIS
	DAVID	1991	VERSAILLES

INFOVILLE	NOMV	DPT
	VERSAILLES	78
	PARIS	75



R	NOM	DN	VILLE	NOMV	DPT
	ANNE	1991	VERSAILLES	VERSAILLES	78
	ANNE	1991	VERSAILLES	PARIS	75
	BERNARD	1993	PARIS	VERSAILLES	78
	BERNARD	1993	PARIS	PARIS	75
	CELINE	1993	PARIS	VERSAILLES	78
	CELINE	1993	PARIS	PARIS	75
	DAVID	1991	VERSAILLES	VERSAILLES	78
	DAVID	1991	VERSAILLES	PARIS	75

Jointure et Produit Cartésien

$$R_1 \bowtie_{A=B} R_2 \text{ Équivaut à : } \sigma_{A=B}(R_1 \times R_2)$$

R	NOM	DN	VILLE	NOMV	DPT
	ANNE	1991	VERSAILLES	VERSAILLES	78
	ANNE	1991	VERSAILLES	PARIS	75
	BERNARD	1993	PARIS	VERSAILLES	78
	BERNARD	1993	PARIS	PARIS	75
	CELINE	1993	PARIS	VERSAILLES	78
	CELINE	1993	PARIS	PARIS	75
	DAVID	1991	VERSAILLES	VERSAILLES	78
	DAVID	1991	VERSAILLES	PARIS	75

Complétude

- ♦ **Théorème de Codd** : l'algèbre relationnelle a un pouvoir expressif équivalent à celui du calcul relationnel, ainsi que de la logique du premier ordre.
 - Les cinq (sept) opérations de base permettent de formaliser sous forme d'expressions toutes les questions que l'on peut poser avec la logique du premier ordre (sans fonction).
- ♦ Exemple :
 - Nom et Section des étudiants de Versailles nés en 1991 ?

Algèbre Relationnelle :

$\Pi_{\text{NOM}, \text{NOMSEC}} (\sigma_{\text{DATEN}=1991 \text{ ET VILLE}=\ll \text{VERSAILLES} \gg} (\text{ETU} \bowtie \text{INSCR} \bowtie \text{SECTION}))$
ETU.NE=INSCR.NE INSCR.NS = SECTION.NS

Notation Fonctionnelle :

PROJECT (NOM, NOMSEC,
RESTRICT(DATEN=« 1991 » AND VILLE=« VERSAILLES »,
JOIN(ETU, INSCRIPTION, SECTION)))

Si vous avez compris l'algèbre alors vous comprendrez SQL...

- ◆ Une requête SQL est un paraphrasage d'une expression de l'algèbre relationnelle en anglais

- ◆ Requête élémentaire :

SELECT A_1, A_2, \dots, A_p
FROM R_1, R_2, \dots, R_k
WHERE $Q \ [\{ \text{UNION} \mid \text{INTERSECT} \mid \text{EXCEPT} \} \dots]$

- ◆ Sémantique du bloc select :

$\Pi_{A_1, A_2, \dots, A_p} (\sigma_Q (R_1 \times R_2 \times \dots \times R_k))$

!/ UN PRODUIT CARTESIEN N'EST PAS UNE JOINTURE !!!

3. CONCEPTS ADDITIONNELS

- ◆ Ensemble de concepts pour :
 - Etendre les fonctionnalités de manipulation
 - Décrire les règles d'évolution des données
 - Supporter des objets complexes (SQL3)
- ◆ Introduits progressivement dans le modèle :
 - Complique parfois le modèle
 - Standardisés au niveau de SQL3 (1999)
 - Des extensions multiples ...

Renommage

- ◆ Pour changer le nom d'une colonne.
- ◆ Notation simple en algèbre relationnelle:

$$\rho_{A \rightarrow B} (R)$$

- ◆ Exemple : $ETU2 = \rho_{DN \rightarrow DATEN} (ETU1)$

ETU1	NOM	DN	VILLE
	BERNARD	1993	PARIS
	CELINE	1993	PARIS
	EMILIE	1993	VELIZY



ETU2	NOM	DATEN	VILLE
	BERNARD	1993	PARIS
	CELINE	1993	PARIS
	EMILIE	1993	VELIZY

Fonction et Agrégat

◆ FONCTION

- Fonction de calcul en ligne appliquée sur un ou plusieurs attributs
- Exemple : $MUTUELLE = FRAIS * 15 / 100$

◆ AGREGAT

Partitionnement horizontal d'une relation selon les valeurs d'un groupe d'attributs (B_i), suivi d'un regroupement par une (ou plusieurs) fonction(s) F_i de calcul en colonne (SUM, MIN, MAX, AVG, COUNT, ...) sur les attributs C_i respectifs

◆ NOTATION : gamma minuscule

$$\underbrace{(B_1, B_2, \dots, B_N \gamma F_1(C_1), F_2(C_2), \dots, F_N(C_N))}_{\text{Ensemble des colonnes}} (R)$$

Exemples d'agrégats

ETU	NOM	AGE	VILLE
	ANNE	21	VERSAILLES
	BERNARD	19	PARIS
	CELINE	19	PARIS
	DAVID	20	VERSAILLES

$\gamma_{AVG(AGE)}(ETU)$

SQL :

**SELECT AVG(AGE)
FROM ETU;**

AVG(AGE)
19.75

$VILLE \gamma_{MAX(AGE)}(ETU)$

SQL :

**SELECT VILLE, MAX(AGE)
FROM ETU
GROUP BY VILLE;**

VILE	MAX(AGE)
VERSAILLES	21
PARIS	19

**!/ le HAVING se fait
tout simplement avec un σ**

Vue

- ◆ Relation d'un schéma externe déduite des relations de la base par une question

- ◆ Exemple : Etudiants Versaillais

```
CREATE VIEW ETUVERSAILLAIS AS  
SELECT NE, NOM, NOMSECTION  
FROM ETU E, INSCRIPTION I, SECTION S  
WHERE E.NE = I.NE AND I.NS=S.NS  
AND E.VILLE = « VERSAILLES »
```

- ◆ Calcul de la vue

- Une vue est une fenêtre dynamique sur la BD et est recalculée à chaque accès.
- Une vue peut être matérialisée (vue concrète) pour accélérer les calculs l'utilisant. Dans ce cas, le SGBD doit être capable de savoir quand recalculer la vue.

Déclencheur (Trigger)

- ◆ Action base de données déclenchée suite à l'apparition d'un événement particulier
- ◆ Forme :
 - {BEFORE | AFTER} <événement> THEN <action>
 - Un événement peut être :
 - une opération sur une table (début ou fin)
 - un événement externe (heure, appel, etc.)
 - Une action peut être :
 - une requête BD (mise à jour)
 - une annulation (abort) de transaction
 - l'appel à une procédure cataloguée

Déclencheur avec condition (Règle)

- ◆ Il est possible d'ajouter une condition afin de déclencher l'action seulement quand la condition est vérifiée
 - Une condition est une qualification portant sur la base.
- ◆ Exemples :

```
BEFORE      UPDATE EMPLOYE
IF          SALAIRE > 100.000
THEN  ABORT TRANSACTION
```

4. CONCLUSION

- ◆ Un ensemble de concepts bien compris et bien formalisés
- ◆ Un modèle unique, riche et standardisé
 - intégration des BD actives
 - intégration des BD objets
 - intégration des BD XML
- ◆ Un formalisme qui s'étend plutôt bien
 - algèbre d'objets
- ◆ Un langage associé défini à plusieurs niveaux
 - SQL1, 2, 3, 2003, etc.