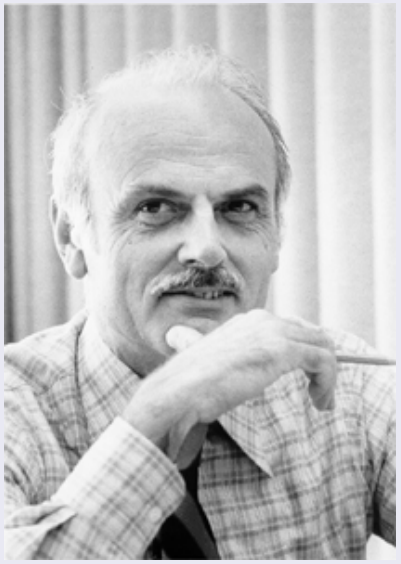
Histoire



- Publié par E.F. CODD (laboratoire de recherche IBM San Jose) dans ACM 1970 : “A Relational Model of Data for Large Shared Data Banks”

- Le modèle relationnel vise à se libérer de la dépendance avec la représentation physique des modèles Hiérarchique et Réseau
- Le modèle relationnel ne vise pas à être l’interface entre le client et l’informaticien, il est l’interface entre l’informaticien et l’ordinateur.

Domaine

➜ Ensemble de valeurs autorisées

- Elémentaire
  - Types de base : Entier, Réel, Texte, Booléen
  - Intervalle : Salaire = [1500, ..., 10 000]
  - Enuméré : Couleur = (Rose, Blanc, Rouge)
- Structuré
  - Point = [X : Réel, Y : Réel]
  - Segment = [origine : Point, extremite : Point]
  - Figure = {Segment}

- Le modèle relationnel est un modèle “à plat”
- Fondement théorique : théorie des ensembles
- Principe : atomicité de valeur dans le modèle relationnel

Produit cartésien - noté X

Le produit cartésien de D1, ..., Dn est l'ensemble des n-uplets (tuples)

$$< V1, .., Vn > / Vi \in Di \quad (1)$$

- D1 = {Rose, Blanc, Rouge} : Couleur
- D2 = {Bordeaux, Alsace} : Region

D1	D2
Rose	Bordeaux
Rose	Alsace
Blanc	Bordeaux
Blanc	Alsace
Rouge	Bordeaux
Rouge	Alsace

Relation

Définition Relation

➜ Sous ensemble du produit cartésien d'une liste de domaines.

- Une **relation** est caractérisée par un nom.
- Une **relation** est un tableau à deux dimensions (table).
- Une ligne de ce tableau est un n-uplet. L'ordre des lignes n'est pas important.

Exemple

D1= Couleur, D2 = Region

Couleur_Vins	D1	D2
	Blanc	Bordeaux
	Blanc	Alsace
	Rouge	Bordeaux
	Rose	Alsace

Attribut

Définition Attribut

➜ Un **attribut** est le nom donné à une colonne d'un tableau modélisant une relation.

- L'ordre des colonnes dans le tableau n'est pas important.
- La valeur d'un **attribut** pour un n-uplet de la relation (i.e., l'élément matriciel colonne / ligne du tableau) prend sa valeur dans le domaine associé à cet attribut.

Exemple

TEINTE : Couleur

Exemple de relation

- 1 nom de relation : VINS.
- 4 attributs auto-définis : CRU, MILLESIME, REGION, TEINTE.

VINS	CRU	MILLESIME	REGION	TEINTE
	Chenas	1983	Beaujolais	Rouge
	Tokay	1980	Alsace	Blanc
	Tokay	1981	Alsace	Blanc
	Tavel	1986	Rhone	Rose
	Chablis	1986	Bourgogne	Blanc
	St-Emilion	1987	Bordeaux	Rouge

Clé

Définition Clé

Attribut ou groupe d'attributs (minimum) qui détermine un n-uplet unique dans une relation (à tout instant).

- Se définit a priori et non pas en regardant la relation.
- Toute relation doit posséder au moins une clé documentée (sans valeur inconnue).
- Exemple : {CRU, MILLESIME} dans VINS.

Schéma de relation

Définition Schéma de relation

- Nom de la relation
- Liste des attributs avec leurs domaines
- Liste des clés : L'administrateur en choisit une, elle devient la clé primaire. Elle est soulignée.

Exemple

VINS (<u>CRU, MILLESIME</u>, REGION, TEINTE)

Un schéma de relation décrit l'**intention** de la relation alors qu'une relation (table) est une **extension**.
Le schéma d'une base de données relationnelle est l'ensemble des schémas des relations de la base.

Clé étrangère

Définition Clé étrangère

- Attribut ou groupe d'attributs qui n'est pas clé dans une relation mais qui apparaît comme clé dans une autre relation.
- Définit les liens Entités-Associations
- Exprime une contrainte d'intégrité référentielle
  - ➜ Insertion de valeurs de n-uplets pour cette clé étrangère dans la relation référençant : doivent exister dans la relation référencée
  - ➜ Suppression dans la relation référencée : les n-uplets référençant doivent disparaître.

Exemple de clé étrangère

BUVEURS (NUMERO_BUVEUR, NOM, PRENOM, TYPE)

VINS (NUMERO_VIN, CRU, MILLESIME, REGION, TEINTE)

ABUS (NUMERO_BUVEUR, NUMERO_VIN, DATE, QUANTITE)

- **ABUS**.NUMERO_VIN référence **VINS**.NUMERO_VIN
- **ABUS**.NUMERO_BUVEUR référence **BUVEURS**.NUMERO_BUVEUR

Méta-base

Définition MÉTA-BASE

Dictionnaire de données, organisé sous forme relationnelle, contenant la description des relations, attributs, domaines, clés, ...

Avantages

- Se manipule comme une relation utilisateur
- Le système la gère comme une relation
- Outil de communication entre concepteurs d'applications

Quelques conseils

- Choisissez des noms d'attributs significatifs
- Appliquez une règle de nommage (organisation logique)
- Acceptez de définir des domaines spécifiques
- Prenez le temps de définir les règles de gestion

Eléments complémentaires

- **Contraintes d'intégrité** (intra ou inter) : le degré d'un vin est toujours positif, intégrité référentielle, ...
- Règles de passage d'un **modèle conceptuel** (UML, Merise, ...) à un **modèle relationnel**
- **Indépendance physique / logique** et le contact de la réalité (dé-normalisation).
- **Fondements mathématiques rigoureux** permettant une analyse des langages de manipulation (algèbre, calculs)
- **Concepts ensemblistes** alors que les langages de programmation sont élément par élément

Objet-Relationnel ? 

Tirer profit des apports des modèles/langages orientés objet

- Extensions :
  - Spécifiques à l'utilisateur (éventuellement normalisées - SQL/MM)
  - Généralement conçues comme des "verrues" au dessus des SGBD relationnels
  - Problèmes des "liaisons dangereuses" (i.e., couplage du SGBD avec du code externe).
- Dynamique : triggers (déclencheurs)