



Todd Codd

Entité - Association modèle

Entity - Relationship Model

Peter Chen en 1976

- entité = ensemble d'objets ou d'instances (table)
- attribut = propriété ou caractéristique d'entité
  - ↳ domaine = ensemble de valeur attribut peut prendre

- association

↳ lien entre entités ex:

acteur - <sup>association</sup> <joue> film

peut être représenté par entité ou par attribut.

cardinalité = nombre de lien sortant d'une instance

- clé d'une entité (e)  $e \neq e' \Rightarrow K(e) \neq K(e')$   
composé par un ou plusieurs attributs

"un sous-ensemble d'attributs qui garantit l'unicité des instances"

- Entité faible (ou identification relative)  
clé d'une entité dépend d'attribut d'autre entité.

- Héritage:



Schema = description d'une table (quels champs, quels types)  
Instance = une des données (ligne)

Différents types de contraintes:

- ↳ structurelles (les instances respectent les domaines)
- ↳ d'intégrité
- ↳ dynamiques

Dépendance fonctionnelle:

$$X \rightarrow Y$$

Clé étrangère et bi-relation:

ex: Film

Nom	Année
Louis	2

Programme

Film Nom	Nom Cinema
Louis 1	

$$\text{Programme [Film.Nom]} \subseteq \text{Film [Nom]}$$

$$\text{Louis 1} \subseteq \text{Louis 2}$$

SQL: IBM, 1970, Donald Chamberlain & Raymond Boyce

UNIQUE - allows NULL

# Algèbre relationnel:

## requets simples:

projection  $\pi$   
 returns only  
 specified columns  
 (removes duplicates)

sélection  
 select particular  
 lines under  
 criterias  
 select operator

opérateurs d'extraction  
 $\sigma_{cond}(table)$  returns whole table  
 [table name]

schéma source = les tables sur lesquelles on effectue les opérations

schéma cible = résultat

## projection:

$\pi_{column}(table) = \text{SELECT column FROM table};$

if  $\pi_{\emptyset}(table) = \begin{cases} \{\emptyset\} & \text{if table is non-empty} \\ \{\} & \text{if table is empty} \end{cases}$

## requets plus complexes:

•  $\bowtie$  •

jointure  
 combine columns  
 from different  
 tables  
 acts as inner join

R			S		
A	B	C	B	C	D
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>
a <sub>1</sub>	b <sub>2</sub>	c <sub>3</sub>	b <sub>2</sub>	c <sub>4</sub>	d <sub>2</sub>
a <sub>2</sub>	b <sub>3</sub>	c <sub>4</sub>	b <sub>3</sub>	c <sub>4</sub>	d <sub>3</sub>
a <sub>3</sub>	b <sub>4</sub>	c <sub>4</sub>	b <sub>4</sub>	c <sub>5</sub>	d <sub>4</sub>

$[R] \bowtie [S] =$

A	B	C	D
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>
a <sub>2</sub>	b <sub>3</sub>	c <sub>4</sub>	d <sub>3</sub>

si R et S ont les mêmes colonnes

$$[R] \bowtie [S] = [R] \cap [S]$$

si R et S sont les colonnes complètement différentes, alors  $[R] \bowtie [S] = [R] \times [S]$

chaque instance de R avec chacune de S.

union  
seulement si les colonnes sont les mêmes

$$R \quad A \mid B \\ a_1 \mid b_1 \\ a_2 \mid b_2$$

$$S \quad A \mid B \\ a_1 \mid b_1 \\ a_2 \mid b_2$$

$$[R] \cup [S] = \begin{array}{c|c} A & B \\ \hline a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_2 \end{array}$$

différence  
seulement si les colonnes sont les mêmes

$$[R] - [S]$$

(même comme R \setminus S en maths)

$$R \quad A \mid B \\ a_1 \mid b_1 \\ a_2 \mid b_2$$

$$S \quad A \mid B \\ a_1 \mid b_1 \\ a_2 \mid b_2$$

$$[R] - [S] = \begin{array}{c|c} A & B \\ \hline a_2 & b_2 \end{array}$$

renommage

$$R \quad A \mid B \mid C \mid D \\ a_1 \mid b_1 \mid c_1 \mid d_1$$

$$P_B \rightarrow R, C \rightarrow L([R]) = \begin{array}{c|c|c|c} A & R & C & D \\ \hline a_1 & b_1 & c_1 & d_1 \end{array}$$

comme (AS) dans SQL