



## Histoire du modèle relationnel

# Relation

Fabien Coelho, Claire Medrala

Mines Paris – PSL

Décembre 2023

Relation

Intro

Concept

Opérations

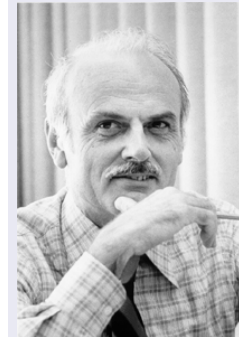
Props

Règles

Conclusion

**Inventeur : Edgar (Ted) Frank Codd**

**UK (1923-2003)**



- pilote RAF WWII
- IBM 1948-1984
- Canada 1953
- PhD UMICH 1965
- CACM, June 1970  
*A Relational Model of Data for Large Shared Data Banks*
- Turing Award 1981
- ACM Follow 1994

1 / 22

2 / 22



## Histoire du modèle relationnel

## Le modèle relationnel

**Précurseur : David L. Childs**

**US**



- UMICH 1968
- *Description of a Set-Theoretic Data Structure*
- *Feasibility of a Set-Theoretic Data Structure: A General Structure Based on a reconstituted Definition of Relation*
- <http://xsp.xegeesis.org/>

Relation

Intro

Concept

Opérations

Props

Règles

Conclusion

**bases** mathématiques

- théorie des ensembles
- algèbre relationnelle
- calcul relationnel, complétude
- théorie dépendances fonctionnelles, normalisation

**pratique** implémentations initiales 1970s

Berkeley Ingres, langage QUEL

IBM System R, SEQUEL (*Structured English QUery Language*)

**actuel** Oracle, MS SQL Server, MS Acces, PostgreSQL, MySQL...

3 / 22

4 / 22



## Domaine

ensemble de valeurs



## Définitions

*Relation, Attribut, Tuple*

Relation

Intro

Concept

Opérations

Props

Règles

Conclusion

Type des données élémentaires stockées, manipulées dans des relations. . .

- entier et réel de différentes précisions
- booléen, caractère (ASCII ou unicode)
- chaîne de caractère ASCII ou unicode, taille fixe ou variable
- date, heure, interval de temps
- géométrie : point, segment, rectangle, cercle. . .
- misc : numéro ou réseau IP, Ethernet, audio, vidéo, image. . .

PAS DE STRUCTURE À CE NIVEAU ! listes, tableaux, . . .

5 / 22

Relation

Intro

Concept

Opérations

Props

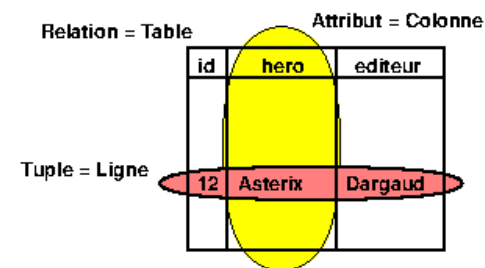
Règles

Conclusion

**relation** une **table** contenant un **ensemble** de données *classe*

**attribut** une colonne **nommée** et **typée** d'une relation *attribut*  
**degré** = nombre d'attributs

**tuple** une donnée (ligne, *row*, n-uplet) d'une relation *instance*  
**cardinalité** = nombre de tuples



6 / 22



## Schéma Relationnel



## Notion de clef

Unique et clef primaire

Relation

Intro

Concept

Opérations

Props

Règles

Conclusion

- description d'un ensemble de données
- nom des relations, noms des attributs et leurs domaines éventuel

DesFilms(titre, auteur, durée)

**schéma**

**données**

```
DesFilms
(titre TEXT,
 auteur TEXT,
 duree INTERVAL);
```

DesFilms
•titre
•auteur
•durée

titre	auteur	duree
City Lights	Chaplin	01:27:00
Citizen Kane	Wells	01:59:00
The Dictator	Chaplin	02:07:00

7 / 22

Relation

Intro

Concept

Opérations

Props

Règles

Conclusion

- **clef** : groupe **minimum** d'attributs définissant un tuple **unique**  
il peut y avoir **plusieurs** clefs, et plusieurs attributs par clef

- **clef primaire** clef principale, souvent un **numéro**

Film(**fid**, titre, auteur, durée),  
Cinéma(**cid**, nsalles, nom, adresse)

```
Film(
  fid INTEGER,
  titre TEXT,
  auteur TEXT,
  durée INTERVAL,
  PRIMARY KEY(fid),
  UNIQUE (titre, auteur)
);
```

```
Cinéma(
  cid INTEGER,
  nsalles INTEGER,
  nom TEXT,
  adresse TEXT,
  PRIMARY KEY(cid),
  UNIQUE (adresse)
);
```

8 / 22



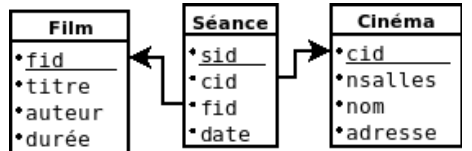
## Notion de clef

## Clef étrangère



## Opérations relationnelles

- groupe d'attribut clef dans une **autre** relation
- **référence**, liens entre tuples...



Séance(sid, tid, mid, date)

```

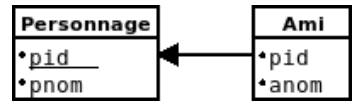
Séance(
  sid INTEGER PRIMARY KEY,
  cid INTEGER,
  fid INTEGER,
  date TIMESTAMP,
  FOREIGN KEY (cid) REFERENCES Cinéma(cid),
  FOREIGN KEY (fid) REFERENCES Film(fid)
);

```

Relation

Intro  
 Concept  
 Opérations  
 Props  
 Règles  
 Conclusion

- opérations **internes** : tables vers tables
- basées en partie sur la théorie des ensembles
- opérations basiques : union, différence, produit, projection, restriction
- opérations complémentaires : intersection, jointure, division, coupure, fermeture transitive...



Personnage	Ami
pid	pnom
1	Tintin
2	Calvin
3	Astérix
4	Méluéline

Ami	pid	anom
1	Haddock	
2	Hobbes	
3	Obélix	
3	Idéfix	



## projection de colonnes d'une table



## Restriction de lignes

pid	pnom
1	Tintin
2	Calvin
3	Astérix
4	Méluéline

pid	anom
1	Haddock
2	Hobbes
3	Obélix
3	Idéfix



Proj(pnom, Personnage)

pnom
Tintin
Calvin
Astérix
Méluéline

Proj(pid, Ami)

pid
1
2
3
3

Relation

Intro  
 Concept  
 Opérations  
 Props  
 Règles  
 Conclusion

pid	anom
1	Haddock
2	Hobbes
3	Obélix
3	Idéfix



Rest(pid=3, Ami)

pid	anom
3	Obélix
3	Idéfix

Rest(anom finit par ix, Ami)...

pid	anom
3	Obélix
3	Idéfix



# Produit Cartésien de tables

combinaisons

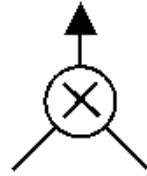


# Union ensembliste

- Relation
- Intro
- Concept
- Opérations
- Props
- Règles
- Conclusion

<i>pid</i>	<i>pnom</i>
1	Tintin
2	Calvin
3	Astérix
4	Mélusine

<i>pid</i>	<i>anom</i>
1	Haddock
2	Hobbes
3	Obélix
3	Idéfix



Personnage  $\times$  Ami

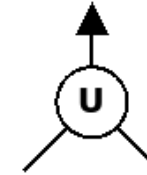
<i>pid</i>	<i>pnom</i>	<i>aid</i>	<i>anom</i>
1	Tintin	1	Haddock
1	Tintin	2	Hobbes
1	Tintin	3	Obélix
1	Tintin	3	Idéfix
2	Calvin	1	Haddock
2	Calvin	2	Hobbes
2	Calvin	3	Obélix

...

- Relation
- Intro
- Concept
- Opérations
- Props
- Règles
- Conclusion

<i>pid</i>	<i>pnom</i>
1	Tintin
2	Calvin
3	Astérix
4	Mélusine

<i>pid</i>	<i>anom</i>
1	Haddock
2	Hobbes
3	Obélix
3	Idéfix



Personnage  $\cup$  Ami

<i>id</i>	<i>nom</i>
3	Idéfix
3	Astérix
1	Haddock
4	Mélusine
2	Hobbes
3	Obélix
2	Calvin
1	Tintin



# Différence ensembliste

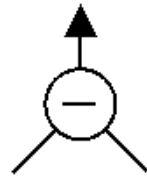


# Intersection ensembliste

- Relation
- Intro
- Concept
- Opérations
- Props
- Règles
- Conclusion

<i>pid</i>	<i>pnom</i>
1	Tintin
2	Calvin
3	Astérix
4	Mélusine

<i>pid</i>	<i>anom</i>
1	Haddock
2	Hobbes
3	Obélix
3	Idéfix



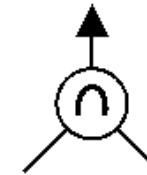
Proj(*pid*, Personnage)  $-$  Proj(*pid*, Ami)

<i>id</i>
3
4

- Relation
- Intro
- Concept
- Opérations
- Props
- Règles
- Conclusion

<i>pid</i>	<i>pnom</i>
1	Tintin
2	Calvin
3	Astérix
4	Mélusine

<i>pid</i>	<i>anom</i>
1	Haddock
2	Hobbes
3	Obélix
3	Idéfix



Proj(*pid*, Personnage)  $\cap$  Proj(*pid*, Ami)

<i>id</i>
3
2
1



# Jointure de tables sur un critère

- Relation
- Intro
- Concept
- Opérations
- Props
- Règles
- Conclusion

pid	pnom
1	Tintin
2	Calvin
3	Astérix
4	Mélusine

pid	anom
1	Haddock
2	Hobbes
3	Obélix
3	Idéfix



Join(même pid, Personnage, Ami)

pid	pnom	aid	anom
1	Tintin	1	Haddock
2	Calvin	2	Hobbes
3	Astérix	3	Obélix
3	Astérix	3	Idéfix



# Pas d'unicité...

- Relation
- Intro
- Concept
- Opérations
- Props
- Règles
- Conclusion

## les paires avec des amis en ix

- Proj(pnom anom, Join(pid, Personnage, Rest(pnom en ix, Ami)))
- Rest(anom finit en ix, Proj(pnom anom, Join(pid, Personnage, Ami)))

## les noms des personnages sans amis

- Proj(pnom, Personnage) - Proj(pnom, Join(pid, Personnage, Ami))
- Proj(pnom, Personnage - Proj(pid pnom, Join(pid, Personnage, Ami)))



# Puissance du modèle relationnel

- Relation
- Intro
- Concept
- Opérations
- Props
- Règles
- Conclusion

**interne** résultat d'une requête est une relation  
combinaisons naturelles des relations

**complet** théorème d'équivalence des questions et réponses  
**Q – relational/domain calculus** expressions logiques sur tuples  
 atomes :  $t.a = v, t_1.a_1 = t_2.a_2, t \in R$   
 formules :  $\wedge, \vee, \neg, \exists t \in R, \forall t \in R$   
 $\exists p \in P (\exists a \in A (p.pid = a.pid \wedge p.pnom = 'Tintin'))$   
**R – relational algebra** avec opérateurs relationnels  
 union, différence, projection, produit cartésien, restriction  
 $Rest(pnom='Tintin', Join(pid, Personnage, Ami))$

Est-ce que Tintin a des amis ?



# Codd's 12 rules

- Relation
- Intro
- Concept
- Opérations
- Props
- Règles
- Conclusion

- **13** règles pour identifier un système **relationnel**
  - publiées en 1985
  - très peu de systèmes passent le test !
- il existe deux versions
  - OLTP (OnLine Transaction Processing)
  - OLAP (OnLine Analytical Processing, décisionnel)



## Liste des 13 règles

Relation

Intro

Concept

Opérations

Props

Règles

Conclusion

- 0 base de donnée **et** gestion du système relationnel
- 1 informations : valeurs en colonnes dans les lignes d'une table
- 2 accès garanti (table + colonne + clef primaire)
- 3 traitement systématique des NULL (valeur spéciale distincte)
- 4 catalogue sous forme de relations
- 5 langage relationnel (linéaire, interactif/appli, définitions)
- 6 vues modifiables si possible
- 7 opérateurs ponctuels pour insertion, modification et effacement
- 8 indépendance physique (application/stockage)
- 9 indépendance logique (application/tables)
- 10 contraintes d'intégrité dans le catalogue (pas application)
- 11 indépendance de distribution (localisation données invisible)
- 12 non subversion (pas de contournement)

21 / 22



## Conclusion

Relation

Intro

Concept

Opérations

Props

Règles

Conclusion

**relation** permet de supprimer de la redondance

- théorie des ensembles
- liens entre concepts (tuples)
- liens entre tables (clefs étrangères)

**algèbre** opérations **internes** entre relations

- opérateurs ensemblistes plus relationnels

**complétude** répond à toutes les questions !

- en fait presque... pas de passage au complémentaire

22 / 22