



# Algèbre relationnelle

L'**algèbre relationnelle** est un langage de requêtes dans des bases de données relationnelles. L'algèbre relationnelle a été inventée en 1970 par Edgar Frank Codd, le directeur de recherche du centre IBM de San José.

Il s'agit de la théorie sous-jacente aux langages de requête des SGBD, comme SQL. Le théorème de Codd dit que l'algèbre relationnelle est équivalente au calcul relationnel (logique du premier ordre sans symbole de fonction). Elle est aussi équivalente à Datalog<sup>1</sup> (Datalog avec la négation) non récursif<sup>1</sup>. **Datalog** est un langage de requête et de règles pour les bases de données déductives.

Selon Abiteboul et al.<sup>2</sup>, l'algèbre relationnelle est conceptuellement un langage "procédural" : en effet, les requêtes sont des suites d'opérations qui construisent la réponse. Cela s'oppose aux langages conceptuellement "déclaratifs" comme le calcul relationnel et Datalog.

## Modèle relationnel

Dans le modèle relationnel, les données sont stockées dans des tables, aussi appelées relations. Voici un exemple de relation :

Clé	Nom	Email
1	Edgar	edgar@wikipedia.fr
2	Frank	frank@wikipedia.fr
3	Bob	bob@wikimedia.fr

Plus précisément<sup>3</sup>, une relation (au sens du modèle de Codd) est constituée :

1. d'un schéma, c'est-à-dire l'ensemble des noms des champs (ici Clé, Nom, Email), et des types correspondants (dans l'exemple respectivement, un nombre entier, puis deux chaînes de caractères).
2. d'une extension, c'est-à-dire le contenu de la table, qui est un ensemble de n-uplets dont l'ordre n'a pas d'importance.

## Définition

Le langage procédural contient les opérations ensemblistes de la théorie des ensembles<sup>4</sup> sur les relations ainsi que des opérations pour fusionner/projeter des relations.

### Opérateurs ensemblistes

Les opérateurs ensemblistes sont l'union, l'intersection, la différence et le produit cartésien.

## Union

L'union de deux relations sur le même schéma est la relation sur ce schéma contenant exactement l'union des enregistrements de ces deux relations. Formellement,  $R \cup S = \{t : t \in R \text{ ou } t \in S\}$ .

Employés de Renault			Employés de Citroën			Employés de Renault U Employés de Citroën		
Nom	ID	Département	Nom	ID	Département	Nom	ID	Département
Harry	3415	Finance	Bertrand	0808	Vente	Harry	3415	Finance
Sally	2241	Vente	Donald	0007	Vente	Sally	2241	Vente
George	3401	Finance				George	3401	Finance
						Bertrand	0808	Vente
						Donald	0007	Vente

## Intersection

L'intersection de deux relations sur le même schéma est la relation sur ce schéma contenant exactement les enregistrements qui apparaissent dans les deux relations. Formellement,  $R \cap S = \{t : t \in R \text{ et } t \in S\}$ .

Personnes inscrits en football		Personnes inscrits en cours de piano		Personnes inscrits en football $\cap$ Personnes inscrits en cours de piano	
Nom	ID	Nom	ID	Nom	ID
Harry	3415	Harry	3415	Harry	3415
Sally	2241	Bertrand	2	George	3401
George	3401	George	3401	George	3401
		Yoda	1000		

## Différence

La différence de deux relations sur le même schéma est la relation sur ce schéma contenant exactement les enregistrements qui apparaissent dans la première relation mais pas dans la deuxième. Formellement,  $R - S = \{t : t \in R \text{ et } t \notin S\}$ . Par exemple, on peut calculer les personnes inscrits en football mais qui ne sont pas inscrits en cours de piano :

<i>Personnes inscrits en football</i>		<i>Personnes inscrits en cours de piano</i>		<i>Personnes inscrits en football - Personnes inscrits en cours de piano</i>	
Nom	ID	Nom	ID	Nom	ID
Harry	3415	Harry	3415		
Sally	2241	Bertrand	2	Sally	2241
George	3401	George	3401		
		Yoda	1000		

## Produit cartésien

Avec le produit cartésien de deux relations, on recopie chaque tuple de la première relation pour chacun des tuples de la deuxième. Formellement,  $R \times S = \{(r, s) : r \in R \text{ et } s \in S\}$ .

<i>Personnes</i>		<i>Cadeaux</i>		<i>Personnes - Cadeaux</i>			
Nom	ID	Type	Prix	Nom	ID	Type	Prix
Harry	3415	livre	10	Harry	3415	livre	10
Sally	2241	gâteau	20	Harry	3415	gâteau	20
		ordinateur	300	Harry	3415	ordinateur	300
				Sally	2241	livre	10
				Sally	2241	gâteau	20
				Sally	2241	ordinateur	300

## Opérateurs relationnels

### Sélection

La *sélection* (ou restriction) consiste à ne retenir que les enregistrements vérifiant une condition donnée. Dans l'exemple plus bas, on ne garde que les touristes dont la ville est Paris.

- Données : Une relation  $R$  et une formule  $F$  formée d'une combinaison de comparaisons et de connecteurs logiques.
- Résultat :  $\sigma_F(R) = \{r \in R : r \text{ satisfait la condition donnée par } F\}$ , dans l'exemple  $\sigma_{\text{Ville}='Paris'}$  (Touristes)
- Équivalent SQL : **WHERE**

Touristes

idTouriste	NomT	Ville	Sport
1	Marc	Paris	Ski
2	Jean	Toulouse	Tennis
3	Franc	Marseille	Football
4	Thomas	Lyon	Voile
5	Max	Paris	Golf

Sélection des touristes où Ville vaut Paris

idTouriste	NomT	Ville	Sport
1	Marc	Paris	Ski
5	Max	Paris	Golf

## Projection

La *projection* permet de ne garder que certains attributs. Dans l'exemple ci-dessous, on ne garde que les attributs NomT et Ville de la relation Touristes.

- Données : Une relation  $R$  et un ensemble d'attributs  $A$  de  $R$
- Résultat :  $\pi_A(R)$ , qui est la Relation  $R$  où on ne considère que les attributs de  $A$ , par exemple  $\pi_{NomT, Ville}(Touristes)$
- Équivalent SQL : **SELECT**

Touristes

idTouriste	NomT	Ville	Sport
1	Marc	Paris	Ski
2	Jean	Toulouse	Tennis
3	Franc	Marseille	Football
4	Thomas	Lyon	Voile
5	Max	Paris	Golf

Projection de la relation Touristes sur les attributs NomT et Ville

NomT	Ville
Marc	Paris
Jean	Toulouse
Franc	Marseille
Thomas	Lyon
Max	Paris

## Renommage

*Renommage* :

- Données : Une relation  $R$  et un attribut  $b$  de  $R$
- Résultat :  $\rho_{a/b}(R)$ , qui est la Relation  $R$  avec  $b$  renommé  $a$
- Équivalent SQL : **AS**

## Jointure

*Jointure* :

- Données : deux relations  $R$  et  $S$
- Résultat :  $R \bowtie S = \{(a,b,c) : (a,b) \in R \text{ et } (b,c) \in S\}$
- Équivalent SQL : **JOIN**

Touristes				Destinations		Touristes ⋈ Destinations				
idTouriste	NomT	Ville	Sport	idTouriste	VilleD	idTouriste	NomT	Ville	Sport	VilleD
1	Marc	Paris	Ski	1	Cannes	1	Marc	Paris	Ski	Cannes
2	Jean	Toulouse	Tennis	2	Ibiza	2	Jean	Toulouse	Tennis	Ibiza
3	Franc	Marseille	Football	4	Tokyo	4	Thomas	Lyon	Voile	Tokyo
4	Thomas	Lyon	Voile							
5	Max	Paris	Golf							

Division

La *division* prend en entrée deux relations  $R(x_1, \dots, x_m, y_1, \dots, y_p)$  et  $S(y_1, \dots, y_p)$ . Ainsi, tout n-uplet  $r \in R$  se décompose en deux n-uplets  $r = (t, s)$ , avec  $t = (t_1, \dots, t_m)$  de schéma  $X = \{x_1, \dots, x_m\}$  et  $s = (s_1, \dots, s_p)$  de schéma  $y = \{y_1, \dots, y_p\}$ . et retourne la table de schéma  $X$  tel que  $R/S = \{t : \forall s \in S, (t, s) \in R\}$  . La division revient à donner “tous les x tels que pour tout y...”

Expressivité

L'algèbre SPC (sélection, projection et produit cartésien) correspond au calcul conjonctif (calcul relationnel sans disjonction et sans négation) : c'est une des versions du théorème de Codd. L'algèbre SPCU- (l'algèbre SPC avec en plus l'union et la différence) correspond au calcul relationnel en entier : c'est une autre version du théorème de Codd. L'équijointure<sup>[Quoi ?]</sup> peut être exprimée avec un produit cartésien, suivi d'une sélection, puis une projection.

Optimisation

Voici des règles de réécriture pour transformer une expression de l'algèbre relationnelle en une autre expression équivalente.

Implémentation

Cependant, les bases de données relationnelles ne fonctionnent pas tout à fait selon les règles ensemblistes de l'algèbre relationnelle. En effet, si l'on ne définit pas de clé primaire, il est possible d'insérer plusieurs lignes identiques dans une table, ce qui d'un point de vue ensembliste n'a pas de sens : un élément fait partie ou ne fait pas partie d'un ensemble. Si l'on veut appliquer strictement les règles des ensembles, il faut vérifier à chaque ajout dans une table que les lignes introduites ne sont pas déjà présentes.

Objets précis du modèle

Il s'agit ici de déterminer des Domaines (i.e., type atomique) :

- Numérique : entier ou réel (SQL : Int, Float, etc.) ;
- Chaîne de caractères (SQL : Char(20), VarChar(32), etc.) ;
- Date (SQL : DATE, TIME, YEAR, etc.) ;
- Type énuméré.

## Notes et références

- (en) *Foundations of Databases : The Logical Level*, Addison-Wesley Longman Publishing Co., Inc., 1995, 685 p. (ISBN 978-0-201-53771-0, lire en ligne (https://dl.acm.org/citation.cfm?id=551350)), p. 10
- (en) *Foundations of Databases : The Logical Level*, Addison-Wesley Longman Publishing Co., Inc., 1995, 685 p. (ISBN 978-0-201-53771-0, lire en ligne (https://dl.acm.org/citation.cfm?id=551350)), Part B - Basics: Relational QueryLanguages - p. 35
- « Apprendre les bases de données et SQL » (http://sgbd.developpez.com/tutoriels/cours-complet-bdd-sql/), sur *Developpez.com* (consulté le 19 mai 2019)
- http://www.scribube.com/limba/franceza/Aide-mmoire-sur-les-bases-de-d21481108.php

## Voir aussi

- Système de gestion de base de données
- Algèbre
- Théorie des ensembles
- Base de données
- Base de données relationnelle

Sur les autres projets Wikimedia : *algèbre relationnelle*, sur le Wiktionnaire  
*Algèbre relationnelle*, sur Wikiversity

## Liens externes

- RAT, Software Rational Algebra Translator to SQL (http://www.sinfo.una.ac.cr/rat/rat.html)
- (fr) Laurent Audibert, « Bases de données relationnelles » (http://laurent-audibert.developpez.com/Cours-BD/), sur *developpez.com*

Ce document provient de « https://fr.wikipedia.org/w/index.php?title=Algèbre\_relationnelle&oldid=197058049 ».

