



REPUBLIQUE DU BENIN
UNIVERSITE D'ABOMEY CALAVI(UAC)
INSTITUT DE FORMATION ET DE RECHERCHE EN
INFORMATIQUE(IFRI)



Titre du cours:
ALGEBRE RELATIONNELLE

"Le résultat de toute opération est une relation"

Partie I

Chargé d'enseignement

Ir. Pierre Jérôme ZOHOU
PhD Student

Année académique 2019-2020

Algèbre Relationnelle

Introduction

Rappel: Modèle Relationnel

Les opérations de base

- Opération PROJECTION

- Opération SELECTION (ou RESTRICTION)

Les opérations ensemblistes

- Opération UNION

- Opération INTERSECTION

- Opération DIFFERENCE

- Opération PRODUIT CARTESIEN

- Opération DIVISION

- Opération JOINTURE

- Opération JOINTURE (équi-jointure)

Introduction

"La conception et l'utilisation de **bases de données relationnelles sur micro-ordinateurs** n'est pas un domaine réservé aux informaticiens". C'est en tout cas ce que pensent beaucoup d'utilisateurs en voyant ce type de logiciel intégré aux suites bureautiques les plus connues.

Cependant la maîtrise d'un **SGBDR** micro (Système de Gestion de Bases de Données Relationnelles) est loin d'être aussi facile à acquérir que celle d'un logiciel de traitement de texte ou d'un tableur.

Plusieurs étapes sont nécessaires à la mise en place d'une base de données, dès lors que l'on a précisément défini ses besoins (ce qui n'est déjà pas chose facile !) : **la création de la structure de la base** sous forme de tables (tableaux de données) reliées entre elles par des données clés, **la conception des requêtes** qui permettront d'extraire ou de mettre à jour les informations qu'elle contient, **la conception de l'interface** homme-machine (écrans et états) qui rendra plus conviviale la saisie et la restitution des informations.

Le degré de difficulté dans la conception de l'interface varie beaucoup selon le logiciel utilisé qui est d'ailleurs le plus souvent différent du SGBDR.

La conception de la structure de la base de données, si elle est un peu complexe à appréhender, peut nécessiter, en amont, l'utilisation d'**outils de modélisation conceptuels** de type **entités-associations** (Modèle Conceptuel des Données de la méthode MERISE ou diagramme de classes du langage UML). Mais, même dans les cas les plus simples il faut obligatoirement connaître les concepts du **Modèle Relationnel**, sans quoi un utilisateur non averti pourra toujours arriver à créer une structure inadaptée et sera vite bloqué dans la conception des requêtes.

Il s'agit ici, d'étudier les principaux opérateurs de l'algèbre relationnelle servant de base à l'élaboration et à l'analyse (plan d'exécution) des requêtes.

Bon nombre d'utilisateurs qui voient les matériels informatiques et les logiciels changer tous les trois mois, seraient surpris d'apprendre que l'algèbre relationnelle a été définie par Codd en 1970.

Elle est à l'origine du langage **SQL** (Structured Query Language) d'IBM, langage d'interrogation et de manipulation de tous les SGBDR actuels (Oracle, PostgreSQL, MySQL, MS SQLServer, MS Access et tous les autres).

Une bonne maîtrise de l'algèbre relationnelle permet de concevoir n'importe quelle requête aussi complexe soit elle avant de la mettre en œuvre à l'aide du langage SQL.

Parmi les opérations de l'algèbre relationnelle, on dispose **d'opérations classiques sur les ensembles** (union, intersection, différence, produit cartésien) puis d'**opérations propres** (projection, sélection, jointure, division).

Nous allons également exposer ici des **opérations de calcul, de regroupement**, de comptage et de tri, non définies à l'origine par Codd mais très utiles.

Tous les opérateurs sont présentés à l'aide d'exemples clairs. Pris séparément, ils sont faciles à appréhender. La rédaction de requêtes (combinaison d'opérateurs) est illustrée par des exercices concrets.

Le langage SQL n'est abordé que dans le cadre des opérations évoquées ci-dessus. Seule l'instruction SELECT et ses multiples aspects sont donc présentés.

Rappel: Modèle Relationnel

L'exemple suivant, relatif à la gestion simplifiée des étapes du Tour de France 97, va nous servir à introduire le vocabulaire lié au modèle relationnel.

CodeEquipe	NomEquipe	DirecteurSportif
BAN	BANESTO	Eusebio UNZUE
COF	COFIDIS	Cyrille GUIMARD
CSO	CASINO	Vincent LAVENU
FDJ	LA FRANCAISE DES JEUX	Marc MADIOT
FES	FESTINA	Bruno ROUSSEL
GAN	GAN	Roger LEGEAY
ONC	O.N.C.E.	Manolo SAIZ
TEL	TELEKOM	Walter GODEFROOT
...

NuméroCoureur	NomCoureur	CodeEquipe	CodePays
8	ULLRICH Jan	TEL	ALL
31	JALABERT Laurent	ONC	FRA
61	ROMINGER Tony	COF	SUI
91	BOARDMAN Chris	GAN	G-B
114	CIPOLLINI Mario	SAE	ITA
151	OLANO Abraham	BAN	ESP
...

NuméroEtape	DateEtape	VilleDépart	VilleArrivée	NbKm
1	06-jul-97	ROUEN	FORGES-LES-EAUX	192
2	07-jul-97	ST-VALERY-EN-CAUX	VIRE	262
3	08-jul-97	VIRE	PLUMELEC	224
...

NuméroCoureur	NuméroEtape	TempsRéalisé
8	3	04:54:33
8	1	04:48:21
8	2	06:27:47
31	3	04:54:33
31	1	04:48:37
31	2	06:27:47
61	1	04:48:24
61	2	06:27:47
91	3	04:54:33
91	1	04:48:19
91	2	06:27:47
114	3	04:54:44
114	1	04:48:09
114	2	06:27:47
151	3	04:54:33
151	1	04:48:29
151	2	06:27:47
...

CodePays	NomPays
ALL	ALLEMAGNE
AUT	AUTRICHE
BEL	BELGIQUE
DAN	DANEMARK
ESP	ESPAGNE
FRA	FRANCE
G-B	GRANDE BRETAGNE
ITA	ITALIE
P-B	PAYS-BAS
RUS	RUSSIE
SUI	SUISSE
...	...

- Comme nous pouvons le constater, le modèle relationnel est un modèle d'organisation des données sous forme de Tables (Tableaux de valeurs) ou chaque **Table** représente une **Relation**, au sens mathématique d'**Ensemble**.

C'est ainsi que dans l'exemple présenté, figurent l'ensemble des Equipes, des Coureurs, des Etapes, des Temps réalisés par les coureurs à chacune des étapes, et enfin l'ensemble des pays.

- Les colonnes des tables s'appellent des **attributs** et les lignes des **n-uplets** (où n est le degré de la relation, c'est à dire le nombre d'attributs de la relation). Un attribut ne prend qu'une seule valeur pour chaque n-uplet. L'ordre des lignes et des colonnes n'a pas d'importance.
- Chaque table doit avoir une **clé primaire** constituée par un ensemble minimum d'attributs permettant de distinguer chaque n-uplet de la Relation par rapport à tous les autres. Chaque ensemble de valeurs formant la clé primaire d'un n-uplet est donc **unique** au sein d'une table.

C'est ainsi que dans la table COUREURS, chaque coureur a un NuméroCoureur différent.

Dans certains cas, plusieurs clés primaires sont possibles pour une seule table. On parle alors de **clés candidates**. Il faut alors en choisir une comme clé primaire.

- Les liens sémantiques (ou règles de gestion sur les données) existants entre les ensembles sont réalisés par l'intermédiaire de **clés étrangères** faisant elles-mêmes référence à des clés primaires d'autres tables.

C'est ainsi que dans la table COUREURS, la clé étrangère CodeEquipe (faisant référence à la clé primaire de même nom dans la table EQUIPES) traduit les deux règles de gestion suivantes :

Un COUREUR appartient à une EQUIPE
 Une EQUIPE est composée de plusieurs COUREURS

- Il existe deux grands types de liens : **Un - Plusieurs** (comme le précédent) et **Plusieurs - Plusieurs**. La réalisation de ce dernier type de liens, un peu plus complexe, passe par l'utilisation d'une table intermédiaire dont la clé primaire est formée des clés étrangères des tables qu'elle relie.

C'est ainsi que la table des TEMPS réalisés à chaque étape par chacun des coureurs exprime les deux règles de gestion suivantes :

Un COUREUR participe à plusieurs ETAPES
 Une ETAPE fait participer plusieurs COUREURS

- Le modèle relationnel est le plus souvent décrit sous la forme suivante, les clés primaires étant soulignées et les clés étrangères marquées par un signe distinctif (ici * ou précédées du signe #).

EQUIPES (CodeEquipe, NomEquipe, DirecteurSportif)

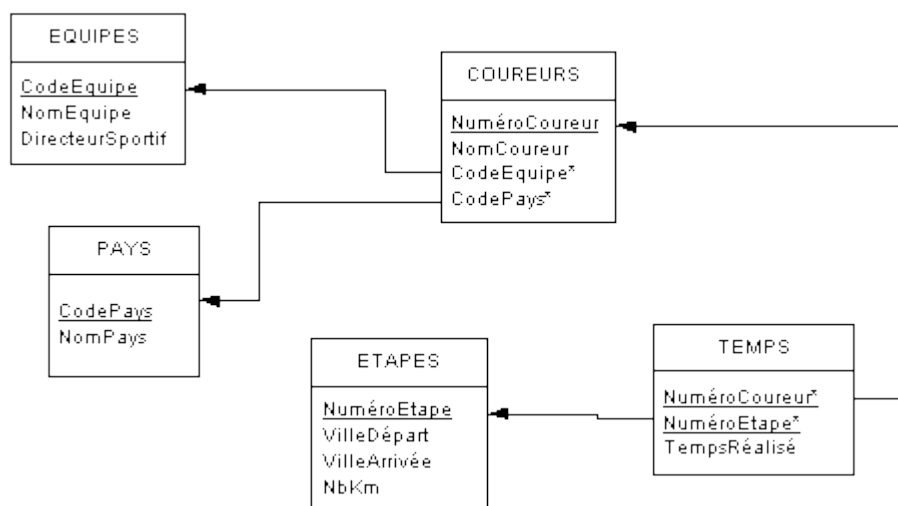
COUREURS (NuméroCoureur, NomCoureur, CodeEquipe*, CodePays*)

ETAPES (NuméroEtape, VilleDépart, VilleArrivée, NbKm)

TEMPS (NuméroCoureur*, NuméroEtape*, TempsRéalisé)

PAYS (CodePays, NomPays)

- On peut aussi le représenter sous forme graphique, de manière à mieux visualiser et interpréter les liens :



Un COUREUR appartient à une EQUIPE
Une EQUIPE est composée de plusieurs COUREURS

Un COUREUR est originaire d'un PAYS
Un PAYS est représenté par plusieurs COUREURS

Un COUREUR participe à plusieurs ETAPES
Une ETAPE fait participer plusieurs COUREURS

- Dans le cadre d'un projet d'informatisation, la conception d'une base de données relationnelle passe d'abord par l'identification des objets de gestion (Coureurs, Etapes, ...) et des règles de gestion du domaine modélisé (interviews des utilisateurs, étude des documents manipulés, des fichiers existants, ...). Une fois énoncées et validées, ces règles nous conduisent automatiquement à la structure du modèle relationnel correspondant.

Les opérations de base

Les opérations de sélection et de projection sont des opérations unaires, car elles s'appliquent à une seule relation.

Opération PROJECTION

L'opération projection travaille sur une seule relation R et définit une relation qui contient un sous-ensemble vertical de R, en extrayant les valeurs des attributs spécifiés et en supprimant les doublons.

Formalisme : $R = \text{PROJECTION}(R1, \text{liste des attributs})$

Exemples

CHAMPIGNONS

Espèce	Catégorie	Conditionnement
Rosé des prés	Conserve	Bocal
Rosé des prés	Sec	Verrine
Coulemelle	Frais	Boîte
Rosé des prés	Sec	Sachet plastique

$R1 = \text{PROJECTION}(\text{CHAMPIGNONS}, \text{Espèce})$

Espèce
Rosé des prés
Coulemelle

R2 = PROJECTION (CHAMPIGNONS, Espèce, Catégorie)

Espèce	Catégorie
Rosés des prés	Conserve
Rosé des prés	Sec
Coulemelle	Frais

Cet opérateur ne porte que sur 1 relation.

- Il permet de ne retenir que certains attributs spécifiés d'une relation.
- On obtient tous les n-uplets de la relation **à l'exception des doublons**

Opération SELECTION (ou RESTRICTION)

L'opération sélection travaille sur une relation simple R et définit une relation qui ne contient que les tuples de R qui satisfont à la condition (ou prédicat) spécifié.

Formalisme : R = **SELECTION** (R1, condition)

Exemple :

CHAMPIGNONS

Espèce	Catégorie	Conditionnement
Rosé des prés	Conserve	Bocal
Rosé des prés	Sec	Verrine
Coulemelle	Frais	Boîte
Rosé des prés	Sec	Sachet plastique

R3 = SELECTION (CHAMPIGNONS, Catégorie = "Sec")

Espèce	Catégorie	Conditionnement
Rosé des prés	Sec	Verrine
Rosé des prés	Sec	Sachet plastique

Cet opérateur porte sur 1 relation.

- Il permet de ne retenir que les n-uplets répondant à une condition exprimée à l'aide des opérateurs arithmétiques (=, >, <, >=, <=, <>) ou logiques de base (ET, OU, NON).
- Tous les attributs de la relation sont conservés.
- Un attribut peut ne pas avoir été renseigné pour certains n-uplets. Si une condition de sélection doit en tenir compte, on indiquera simplement : nomattribut "non renseigné".

Les opérations ensemblistes

Les opérations de sélection et de projection extraient des informations d'une seule relation. Nous souhaitons évidemment combiner dans certains cas des informations de plusieurs relations. Dans le reste de cette section, nous examinons les opérations binaires de l'algèbre relationnelle, en commençant par l'union, l'intersection, la différence d'ensemble, le produit cartésien et la jointure.

Opération UNION

L'union de deux relations R et S définit une relation qui contient tous les tuples de R, de S ou à la fois R et S, les tuples en double étant éliminés. R et S doivent être compatibles envers l'union.

Formalisme : $R = \text{UNION} (R1, R2)$

Exemple :

E1 : Enseignants élus au CA

n°enseignant	nom_enseignant
1	DUPONT
3	DURAND
4	MARTIN
5	BERTRAND

E2 : Enseignants représentants syndicaux

n°enseignant	nom_enseignant
1	DUPONT
4	MARTIN
6	MICHEL

On désire obtenir l'ensemble des enseignants élus au CA ou représentants syndicaux.

$R1 = \text{UNION} (E1, E2)$

n°enseignant	nom_enseignant
1	DUPONT
3	DURAND
4	MARTIN
5	BERTRAND
6	MICHEL

Cet opérateur porte sur deux relations qui doivent avoir le même nombre d'attributs définis dans le même domaine (ensemble des valeurs permises pour un attribut). On parle de relations ayant le même schéma.

- La relation résultat possède les attributs des relations d'origine et les n-uplets de chacune, avec élimination des doublons éventuels.

Opération INTERSECTION

L'opération d'intersection définit une relation constituée de l'ensemble de tous les tuples présents à la fois dans R et dans S. R et S doivent être compatibles envers l'union.

Formalisme : $R = \text{INTERSECTION}(R1, R2)$

Exemple :

E1 : Enseignants élus au CA

E2 : Enseignants représentants syndicaux

n°enseignant	nom_enseignant
1	DUPONT
3	DURAND
4	MARTIN
5	BERTRAND

n°enseignant	nom_enseignant
1	DUPONT
4	MARTIN
6	MICHEL

On désire connaître les enseignants du CA qui sont des représentants syndicaux.

$R2 = \text{INTERSECTION}(E1, E2)$

n°enseignant	nom_enseignant
1	DUPONT
4	MARTIN

Cet opérateur porte sur deux relations de même schéma.

- La relation résultat possède les attributs des relations d'origine et les n-uplets communs à chacune.

Opération DIFFERENCE

L'opération de différence d'ensemble définit une relation qui comporte les tuples qui existent dans la relation R et non dans la relation S. Les relations R et S doivent être compatibles envers l'union.

Formalisme : $R = \text{DIFFERENCE}(R1, R2)$

Exemple :

E1 : Enseignants élus au CA

E2 : Enseignants représentants syndicaux

n°enseignant	nom_enseignant
1	DUPONT
3	DURAND
4	MARTIN
5	BERTRAND

n°enseignant	nom_enseignant
1	DUPONT
4	MARTIN
6	MICHEL

On désire obtenir la liste des enseignants du CA qui ne sont pas des représentants syndicaux.

$R3 = \text{DIFFERENCE}(E1, E2)$

n°enseignant	nom_enseignant
3	DURAND
5	BERTRAND

Cet opérateur porte sur deux relations de même schéma.

- La relation résultat possède les attributs des relations d'origine et les n-uplets de la première relation qui n'appartiennent pas à la deuxième.
- **Attention !** $\text{DIFFERENCE}(R1, R2)$ ne donne pas le même résultat que $\text{DIFFERENCE}(R2, R1)$

Opération PRODUIT CARTESIEN

L'opération de produit cartésien définit une relation constituée de la concaténation de tous les tuples de la relation R avec tous ceux de la relation S.

Formalisme : $R = \text{PRODUIT}(R1, R2)$

Exemple :

Etudiants

n°étudiant	nom
101	DUPONT
102	MARTIN

Epreuves

libellé épreuve	coefficient
Informatique	2
Mathématiques	3
Gestion financière	5

Examen = $\text{PRODUIT}(\text{Etudiants}, \text{Epreuves})$

n°étudiant	nom	libellé épreuve	coefficient
101	DUPONT	Informatique	2
101	DUPONT	Mathématiques	3
101	DUPONT	Gestion financière	5
102	MARTIN	Informatique	2
102	MARTIN	Mathématiques	3
102	MARTIN	Gestion financière	5

Cet opérateur porte sur deux relations.

- La relation résultat possède les attributs de chacune des relations d'origine et ses n-uplets sont formés par la concaténation de chaque n-uplet de la première relation avec l'ensemble des n-uplets de la deuxième.

Opération DIVISION

L'opération de division définit une relation sur les attributs C, constituée de l'ensemble des tuples de R qui correspondent à la combinaison de tous les tuples de S.

Formalisme : $R = \text{DIVISION}(R1, R2)$

Exemple :

PARTICIPER

Athlète	Epreuve
Dupont	200 m
Durand	400 m
Dupont	400 m
Martin	110 m H
Dupont	110 m H
Martin	200 m

EPREUVE

Epreuve
200 m
400 m
110 m H

DIVISION (PARTICIPER, EPREUVE)

Athlète
Dupont

"L'athlète Dupont participe à toutes les épreuves"

- Cet opérateur porte sur 2 relations qui doivent avoir au moins un attribut défini dans le même domaine.
- Tous les attributs du diviseur (ici EPREUVE) doivent être des attributs du dividende (ici PARTICIPER).
- La relation dividende doit avoir au moins une colonne de plus que la relation diviseur.
- La relation résultat, le quotient, possède les attributs non communs aux deux relations initiales et est formée de tous les n-uplets qui, concaténés à chacun des n-uplets du diviseur (ici EPREUVE) donne toujours un n-uplet du dividende (ici PARTICIPER).

Opération JOINTURE

Nous souhaitons généralement et de façon type qu'un produit cartésien satisfasse à certaines conditions, de sorte que nous utilisons normalement une **opération de jointure** (ou conjonction) au lieu du produit cartésien. L'opération de jointure qui combine deux relations de manière à former une nouvelle relation, est l'une des opérations les plus essentielles de l'algèbre relationnelle. La jointure est un dérivé du produit cartésien et équivaut à une opération de sélection, utilisant un prédicat de jointure comme formule de sélection, sur le produit cartésien des deux avec efficacité dans un SGBDR; c'est

une des raisons qui expliquent que les systèmes relationnels éprouvent des problèmes intrinsèques de performances.

Les opérations de jointure possède des variantes de formes, qui ne diffèrent que par des différences subtiles, et dont certaines sont plus utiles que d'autres:

- Jointure thêta
- Equijointure(un type particulier de jointure thêta)
- Jointure naturelle
- Jointure externe
- Sémi-jointure

Jointure thêta

L'opération de jointure thêta (ou THETA jointure) définit une relation qui contient des tuples satisfaisant au prédicat F du produit cartésien de R et S . Le prédicat F est de la forme $R.a_i \text{ thêta } S.b_i$ où *thêta* est peut être l'un des opérateurs de comparaison (=, >, <, >=, <=, <>). le degré d'une jointure thêta est la somme des degrés des relations opérandes R et S .

Opération JOINTURE (équi-jointure)

Nous parlons **d'équi-jointure** ou d'équijointure, en lieu et place de jointure thêta dans le cas où le prédicat F ne contient que l'égalité (=). Ce qui fera l'objet de notre étude en ce qui concerne la jointure.

Formalisme : $R = \text{JOINTURE}(R1, R2, \text{condition d'égalité entre attributs})$

Exemple :

PRODUIT

CodePrd	Libellé	Prix unitaire
590A	HD 1,6 Go	1615
588J	Scanner HP	1700
515J	LBP 660	1820

DETAIL_COMMANDE

N°cde	CodePrd	quantité
97001	590A	2
97002	515J	1
97003	515J	3

$R = \text{JOINTURE}(\text{PRODUIT}, \text{DETAIL_COMMANDE}, \text{Produit.CodePrd} = \text{Détail_Commande.CodePrd})$

A.CodePrd	Libellé	Prix unitaire	N°cde	B.CodePrd	quantité
590A	HD 1,6 Go	1615	97001	590A	2
515J	LBP 660	1820	97002	515J	1
515J	LBP 660	1820	97003	515J	3

Cet opérateur porte sur 2 relations qui doivent avoir au moins un attribut défini dans le même domaine (ensemble des valeurs permises pour un attribut).

- La condition de jointure peut porter sur l'égalité d'un ou de plusieurs attributs définis dans le même domaine (mais n'ayant pas forcément le même nom).
- Les n-uplets de la relation résultat sont formés par la concaténation des n-uplets des relations d'origine qui vérifient la condition de jointure.

Remarque: Des jointures plus complexes que l'équijointure peuvent être réalisées en généralisant l'usage de la condition de jointure à d'autres critères de comparaison que l'égalité (<,>, <=,>=, <>).