

2019

SQL

PARTIE 1

JOACHIM ZADI

TABLE DES MATIERES

1	introduction	2
1.1	Historique.....	2
1.2	Les normes SQL.....	3
1.3	Description brève du modèle relationnel.....	4
1.3.1	Principaux concepts du modèle relationnel	4
1.3.2	Principales règles	6
1.4	Les opérateurs en algèbre relationnelle	8
1.4.1	Union.....	8
1.4.2	Intersection	9
1.4.3	Différence.....	10
1.4.4	Division.....	10
1.4.5	Restriction	12
1.4.6	Projection.....	12
1.4.7	Produit cartésien.....	13
1.4.8	Jointure	13
1.4.9	Calculs élémentaires	14
1.4.10	Calcul d'agrégats	15

INTRODUCTION

HISTORIQUE

Les bases de données sont indispensables dans tous développements informatiques. Toutes les données sont stockées dans la majorité des cas dans une structure de base de données.

On parle de **BDD** pour désigner le stockage des données et de **SGBD** pour désigner les éléments qui sont mis à disposition du développeur pour manipuler ces données.

Il existe plusieurs types de base de données, mais celles que nous étudions dans ce module sont les bases de données relationnelles.

Les bases de données de type relationnel sont apparues dans les années 1980. Elles s'appuient sur les travaux développés par un chercheur, **Mr Edgard Codd**, travaillant chez IBM sur le modèle relationnel au début des années 1970. Les données sont organisées en tables distinctes sans niveau de hiérarchie. Il n'y a plus de pointeurs mais les données contenues dans les tables permettent de réaliser des liens entre les tables.

Le langage **SQL** - pour *Structured Query Language* - signifie langage d'interrogation structuré. Il a été créé au début des années 1970 par IBM. C'est une start-up nommée RELATIONAL SOFTWARE qui produira la première version commercialisable en 1979. Cette start-up est depuis devenue Oracle Corp.

Le langage SQL se décompose en plusieurs sous-ensembles :

1. Le **DDL** pour *Data Definition Language*, qui regroupe les ordres utilisés pour créer, modifier ou supprimer les structures de la base (tables, index, vues, etc.). Il s'agit principalement des ordres **CREATE**, **ALTER** et **DROP**.
2. Le **DML** pour *Data Manipulation Language*, qui regroupe les ordres utilisés pour manipuler les données contenues dans la base. Il s'agit principalement des ordres **SELECT**, **INSERT**, **DELETE** et **UPDATE**.
3. Le **DCL** pour *Data Control Language*, qui regroupe les ordres utilisés pour gérer la sécurité des accès aux données. Il s'agit principalement des ordres **GRANT** et **REVOKE**.
4. Le **TCL** pour *Transaction Control Language*, qui regroupe les ordres utilisés pour gérer la validation ou non des mises à jour effectuées sur la base. Il s'agit principalement des ordres **COMMIT** et **ROLLBACK**.

LES NORMES SQL

La première version de SQL normalisée par l'**ANSI** date de 1986. Elle sera suivie de plusieurs versions plus ou moins importantes.

La norme **SQL2** ou **SQL92** est la plus importante. La majorité des SGBDR existants implémentent cette version.

Puis suivront plusieurs autres évolutions **SQL-3**, **SQL:2003** et **SQL:2008** qui apportent chacune quelques fonctions complémentaires mais que tous les SGBDR n'ont pas forcément repris, par exemple des fonctions orientées objet dans SQL-3 ou des fonctions pour manipuler des types XML dans SQL-2003.

Chaque fournisseur de **SGBDR** a implémenté à sa façon le langage SQL et a ajouté ses propres extensions.

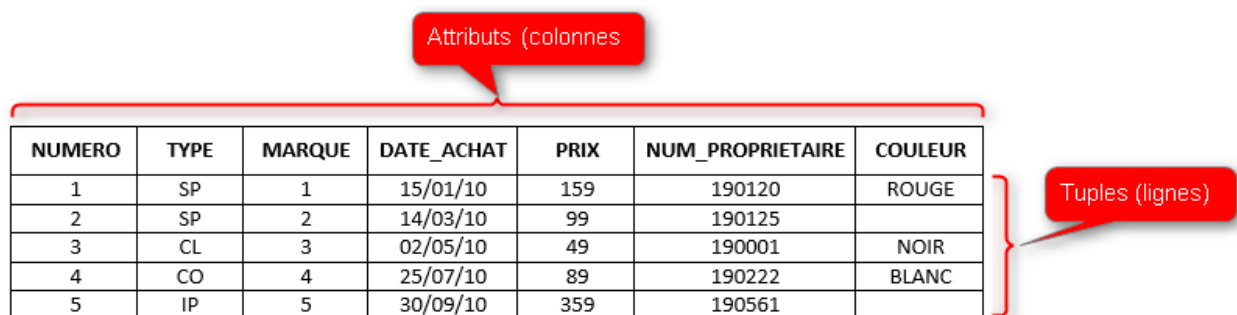
Norme	Nom courant	Explications
ISO/CEI 9075:1986	SQL-86 ou SQL-87	Édité par l'ANSI puis adopté par l'ISO en 1987.
ISO/CEI 9075:1989	SQL-89 ou SQL-1	Révision mineure.
ISO/CEI 9075:1992	SQL-92 ou SQL2	Révision majeure.
ISO/CEI 9075:1999	SQL-99 ou SQL3	Expressions rationnelles, requêtes récursives, déclencheurs, types non scalaires et quelques fonctions orientées objets.
ISO/CEI 9075:2003	SQL:2003	Introduction de fonctions pour la manipulation XML, « Windows Functions », ordres standardisés et colonnes avec valeurs autoproduites (y compris colonnes d'identité).
ISO/CEI 9075:2008	SQL:2008	Ajout de quelques fonctions de fenêtrage (ntile, lead, lag, first value, last value, nth value), limitation du nombre de ligne (OFFSET / FETCH), amélioration mineure sur les types distincts, curseurs et mécanismes d'auto-incréments.

DESCRIPTION BREVE DU MODELE RELATIONNEL

Le modèle relationnel a été créé, comme indiqué au-dessus, par un chercheur, **Edgard Codd**, travaillant chez IBM au début des années 1970. Il est parti des principes mathématiques simples, « *la théorie des ensembles et la logique de prédicats* ».

Le modèle relationnel repose sur la notion d'ensemble. Schématiquement un ensemble peut être représenté par une table, on peut également appeler une table, une **relation**.

Cet ensemble a des **attributs** « *les colonnes* » et des **lignes** contenant des valeurs (les *tuples*). La forme la plus couramment utilisée pour représenter une table la suivante :



NUMERO	TYPE	MARQUE	DATE_ACHAT	PRIX	NUM_PROPRIETAIRE	COULEUR
1	SP	1	15/01/10	159	190120	ROUGE
2	SP	2	14/03/10	99	190125	
3	CL	3	02/05/10	49	190001	NOIR
4	CO	4	25/07/10	89	190222	BLANC
5	IP	5	30/09/10	359	190561	

Table TELEPHONE = Ensemble

Le modèle relationnel présente les données sous forme logique, il est totalement indépendant du modèle physique. C'est le fournisseur qui décide du mode de stockage physique des tables. C'est l'avantage majeur des bases de données relationnelles, l'indépendance entre le modèle logique et le modèle physique.

Une fois les tables définies, il faut disposer d'un langage pour les manipuler, il s'agit de « *l'algèbre relationnelle* » ; Celui-ci a également été inventé par **Edgard Codd**. À l'aide de ces opérateurs, on peut interroger les relations existantes et créer de nouvelles relations. On parle d'opérateurs **ensemblistes** : **union, intersection, différence, produit cartésien, division et jointure**.

L'algèbre relationnelle est mise en œuvre par le SQL et les systèmes de gestion de bases de données relationnelles (SGBDR) implémentent le modèle relationnel.

PRINCIPAUX CONCEPTS DU MODELE RELATIONNEL

Les trois principaux concepts du modèle relationnel sont le **domaine**, le **produit cartésien** et les **relations**.

DOMAINE

C'est un ensemble de valeurs caractérisé par un nom.

Par exemple :

- Le type de téléphone est un domaine qui comprend les valeurs CL, CO, SP et IP (pour CLapet, COulissant, SmartPhone et IPhone).
- La couleur est un autre domaine (ROUGE, NOIR, VERT...).

Le nombre d'occurrences de chacun des domaines donne la CARDINALITÉ. Pour les types de téléphone, la cardinalité est de 4.

PRODUIT CARTESIEN

Celui-ci représente la jonction entre deux domaines. Si par exemple l'on effectue le produit cartésien des types de téléphone et des couleurs, on obtient des tuples D1, D2.

Dans notre exemple, le produit cartésien du domaine 1 : type de téléphone et du domaine 2 : couleur de téléphone donne :

(CL, VERT), (CL, NOIR), (CL, ROUGE), (CO, VERT), (CO, NOIR), etc.

RELATION

La notion de relation est le fondement du modèle relationnel. La relation permet de mettre en relation les domaines selon certains critères.

Par exemple si l'on veut créer une relation nommée TELROUGE, on indiquera que l'on veut associer tous les téléphones du domaine « Type Téléphone » avec le seul élément « ROUGE » du domaine « Couleur ».

La représentation de cette relation se fait sous forme de tableau à deux dimensions.

TYPE	COULEUR
SP	ROUGE
CL	ROUGE
CO	ROUGE
IP	ROUGE

RELATION : TELROUGE

Chaque ligne est unique et représente un objet de la relation. Le **degré** est le nombre d'attributs d'une relation (ici = 2).

PRINCIPALES REGLES

Le modèle relationnel gère donc un objet principal, la relation, associée aux concepts de domaine et d'attribut. Des règles s'appliquent à cette relation afin de respecter les contraintes liées à l'analyse. Voici quelques-unes de ces règles :

1. **Cohérence** : Toute valeur prise par un attribut doit appartenir au domaine sur lequel il est défini.
2. **Unicité** : Tous les éléments d'une relation doivent être distincts.
3. **Identifiant** : Attribut ou ensemble d'attributs permettant de caractériser de manière unique chaque élément de la relation.
4. **Clé primaire** : Identifiant minimum d'une relation.
5. **Clés secondaires** : Autres identifiants de la relation.
6. **Intégrité référentielle** : Cette règle impose qu'un attribut ou ensemble d'attributs d'une relation apparaisse comme clé primaire dans une autre relation.
7. **Clé étrangère** : Attribut ou ensemble d'attributs vérifiant la règle d'intégrité référentielle.

Exemples :

NUMERO	TYPE	MARQUE	DATE_ACHAT	PRIX	NUM_PROPRIETAIRE	COULEUR
1	SP	1	15/01/10	159	190120	ROUGE
2	SP	2	14/03/10	99	190215	
3	CL	3	02/05/10	49	190001	NOIR
4	CO	4	25/07/10	89	190222	BLANC
5	IP	5	30/09/10	359	190561	

RELATION TELEPHONE

- ❖ **NUMERO** est l'identifiant primaire.
- ❖ **TYPE** et **MARQUE** sont des clés étrangères référençant les clés primaires des relations de **MARQUE_TEL** et **TYPE_TEL**.

MARQUE	LIB_MARQUE	PAYS
1	SAMSUNG	COREE
2	SONY	JAPON
3	PHILIPS	PAYS-BAS
4	MOTOROLA	USA
5	APPLE	USA

RELATION TELEPHONE

❖ **MARQUE** est l'identifiant primaire.

8. **Valeur nulle** : Dans le modèle relationnel, la notion de nullité est admise. C'est une valeur représentant une information inconnue ou inapplicable dans une colonne. Elle est notée **NULL**.
9. **Contrainte d'entité** : Toute valeur participant à une clé primaire doit être non NULL.

Exemple

NUMERO	TYPE	MARQUE	DATE_ACHAT	PRIX	NUM_PROPRIETAIRE	COULEUR
1	SP	1	15/01/10	159	190120	ROUGE
2	SP	2	14/03/10	99	190215	
3	CL	3	02/05/10	49	190001	NOIR
4	CO	4	25/07/10	89	190222	BLANC
5	IP	5	30/09/10	359	190561	

RELATION TELEPHONE

La couleur peut être non renseignée alors que le numéro du téléphone, qui est la clé primaire, est **obligatoire**.

LES OPERATEURS EN ALGEBRE RELATIONNELLE

L'algèbre relationnelle a conduit à la mise au point du SQL qui est devenu le standard en ce qui concerne la gestion des données.

C'est une méthode d'extraction permettant la manipulation des tables et des colonnes. Son principe repose sur la création de nouvelles tables (tables résultantes) à partir des tables existantes, ces nouvelles tables devenant des objets utilisables immédiatement.

Les opérateurs de l'algèbre relationnelle permettant de créer les tables résultantes sont basés sur la théorie des ensembles.

La syntaxe et les éléments de notations retenus ici sont les plus couramment utilisés.

UNION

L'union entre deux relations de **même** structure « *degré et domaines* » donne une table résultante de **même** structure ayant comme éléments l'ensemble des éléments distincts des deux relations initiales.

Notation : $R_x = R_1 \cup R_2$

Exemples :

Soit les relations TELEPHONE_SMARTPHONE et TELEPHONE_COULISSANT

NUMERO	TYPE	MARQUE	DATE_ACHAT	PRIX	NUM_PROPRIETAIRE	COULEUR
1	SP	1	15/01/10	159	190120	ROUGE
2	SP	1	14/03/10	99	190215	

RELATION TELEPHONE_SMARTPHONE

NUMERO	TYPE	MARQUE	DATE_ACHAT	PRIX	NUM_PROPRIETAIRE	COULEUR
4	CO	4	25/07/10	89	190222	BLANC

RELATION TELEPHONE_COULISSANT

NUMERO	TYPE	MARQUE	DATE_ACHAT	PRIX	NUM_PROPRIETAIRE	COULEUR
1	SP	1	15/01/10	159	190120	ROUGE
2	SP	1	14/03/10	99	190215	
4	CO	4	25/07/10	89	190222	BLANC

UNION DES DEUX RELATIONS

INTERSECTION

L'intersection entre deux relations de **même** structure « *degré et domaines* » donne une table résultante de **même** structure ayant comme éléments l'ensemble des éléments communs aux deux relations initiales.

Notation : $R_x = R1 \cap R2$

Exemple :

CLE	LIB_MARQUE	PAYS
1	SAMSUNG	COREE
2	SONY	JAPON
3	PHILIPS	PAYS-BAS
4	MOTOROLA	USA
5	APPLE	USA

RELATION MARQUE_TEL

CLE	LIB_MARQUE	PAYS
1	SAMSUNG	COREE
2	SONY	JAPON
3	DELL	USA
4	HP	USA
5	APPLE	USA

RELATION MARQUE_ORDINATEUR

CLE	LIB_MARQUE	PAYS
1	SAMSUNG	COREE
2	SONY	JAPON
5	APPLE	USA

MARQUE communes aux deux relations

DIFFERENCE

La différence entre deux relations de **même** structure « *degré et domaines* » donne une table résultante de **même** structure ayant comme éléments l'ensemble des éléments de la première relation qui ne sont pas dans la deuxième.

Notation : $R_x = R_1 - R_2$

Exemple : Considérant les deux relations précédentes, si nous appliquons la différence, on aura :

CLE	LIB_MARQUE	PAYS
3	PHILIPS	PAYS-BAS
4	MOTOROLA	USA

MARQUE présente dans la relation 1 et pas dans la relation 2

DIVISION

La division entre deux relations est possible à condition que la relation diviseur *soit totalement incluse* dans la relation dividende. Le quotient de la division correspond à l'information qui, présente dans le dividende, n'est pas présente dans le diviseur.

Il est également possible de définir la division de la façon suivante : soit R_1 et R_2 des relations telles que R_2 soit totalement inclus dans R_1 . Le quotient $R_1 \div R_2$ est constitué des tuples t tels que pour tous tuples t' définis sur R_2 , il existe le tuple $t.t'$ défini sur R_1 .

Notation : $R_x = R_1 \div R_2$

Exemple

CLE	LIB_MARQUE	PAYS	VENTES
1	SAMSUNG	COREE	123456
2	SONY	JAPON	789012
3	PHILIPS	PAYS-BAS	3456789
4	MOTOROLA	USA	12356
5	APPLE	USA	1111356

RELATION MARQUE_TEL

CLE	LIB_MARQUE	PAYS
1	SAMSUNG	COREE
2	SONY	JAPON
3	DELL	USA
4	HP	USA
5	APPLE	USA

RELATION MARQUE_ORDINATEUR

La division entre les deux relations permet d'isoler l'information complémentaire à la relation MARQUE_ORDINATEUR et présente dans la relation MARQUE_TEL.

VENTES
123456
789012
1111356

RESTRICTION

La restriction repose sur une condition. Elle produit, à partir d'une relation, une relation de **même** schéma n'ayant que les éléments de la relation initiale qui répondent à la condition.

Notation : $R_x = \sigma (\text{condition})_{R_1}$

La condition s'exprime sous la forme : [NON] [(] attribut opérateur valeur [)] [{ET/OU} condition]

Opérateur : Un opérateur de comparaison : =, <>, >, <, >=, <=

Valeur : Une constante ou un autre attribut.

Exemple :

Marque d'ordinateur des USA

$MARQU_{USA} = \sigma (\text{PAYS} = \text{"USA"})_{MARQUE_ORDINATEUR}$

CLE	LIB_MARQUE	PAYS
3	DELL	USA
4	HP	USA
5	APPLE	USA

PROJECTION

La projection d'une relation sur un groupe d'attributs donne une relation résultante ayant comme schéma uniquement ces attributs, et comme éléments les N-UPLETS distincts composés par les valeurs associées de ces attributs.

Notation: $R_x = \pi R_{(A_1, A_2, \dots, A_n)}$

Exemple sur LIB_MARQUE et PAYS de la table MARQUE_ORDINATEUR :

$LISTMARQ = \pi MARQUE_ORDINATEUR_{(LIB_MARQUE, PAYS)}$

LIB_MARQUE	PAYS
SAMSUNG	COREE
SONY	JAPON
DELL	USA
HP	USA
APPLE	USA

S'il y avait eu des doublons, ils auraient été supprimés du résultat.

PRODUIT CARTESIEN

Le produit cartésien entre deux relations produit une relation ayant comme schéma tous les attributs des deux relations existantes et comme éléments l'association de chaque ligne de la première table avec chaque ligne de la deuxième.

Notation : $R_x = S1 \times S2$

JOINTURE

La jointure entre deux relations est produite par la restriction sur le produit cartésien.

Notation: $R_x = S1 \text{ JOIN}_{(\text{condition})} S2$.

Exemple :

CLE	LIB_MARQUE	PAYS
1	SAMSUNG	COREE
2	SONY	JAPON
3	PHILIPS	PAYS-BAS
4	MOTOROLA	USA
5	APPLE	USA

RELATION MARQUE_TEL

CLE	LIB_MARQUE	PAYS
1	SAMSUNG	COREE
2	SONY	JAPON
3	DELL	USA
4	HP	USA
5	APPLE	USA

RELATION MARQUE_ORDINATEUR

JOINTURE entre ces deux relations

TELMARQ = MARQUE_TEL **JOIN** (MARQUE_TEL.CLE=MARQUE_ORDINATEUR.CLE) MARQUE_ORDINATEUR.

CLE	LIB_MARQUE	PAYS
1	SAMSUNG	COREE
2	SONY	JAPON
3	DELL	USA
4	HP	USA
5	APPLE	USA
1	SAMSUNG	COREE
2	SONY	JAPON
3	DELL	USA
4	HP	USA
5	APPLE	USA

Les différents types de jointure sont:

- ❖ **Thêta-jointure**: La condition est une **comparaison** entre deux attributs.
- ❖ **Équi-jointure**: La condition porte sur **l'égalité** entre deux attributs.
- ❖ **Jointure naturelle**: Équi-jointure entre les attributs portant le même nom.

CALCULS ELEMENTAIRES

Projection sur une relation associée à un calcul portant sur chaque ligne pour créer un ou plusieurs nouveaux attributs.

Notation : $R_x = \pi S (A_1, \dots, N_1 = \text{expression calculée}, \dots)$

L'expression calculée peut être:

- ✚ Une opération arithmétique,
- ✚ Une fonction mathématique,
- ✚ Une fonction portant sur une chaîne.

CALCUL D'AGREGATS

Projection sur une relation associée à un ou des calculs statistiques portant sur un attribut pour tous les éléments de la relation ou du regroupement lié à la projection afin de créer un ou plusieurs nouveaux attributs.

Notation : $R_x = \pi S (A_1, \dots, N_1 = \text{fonction statistique } (A_x), \dots)$

Les principales fonctions statistiques sont :

- ✚ **COUNT (*)**: nombre de lignes.
- ✚ **COUNT (attribut)**: nombre de valeurs non nulles.
- ✚ **SUM (attribut)**: somme des valeurs non nulles.
- ✚ **AVG (attribut)**: moyenne des valeurs non nulles.
- ✚ **MAX (attribut)**: valeur maximum (non nulle).
- ✚ **MIN (attribut)**: valeur minimum (non nulle).