

Institut Préparatoire aux Études d'Ingénieurs de Tunis



Leila Ben Othman

Deuxième année de classes préparatoires

SM - SP

Année universitaire: 2018 - 2019

Algèbre relationnelle

Plan

- ① Introduction
- ② Modèle relationnel
- ③ Algèbre relationnelle

Introduction

Définition - (Base de données et langage SQL, Laurent AUDIBERT)

Une base de données informatisée est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur, représentant des informations du monde réel et pouvant être interrogées et mises à jour par une communauté d'utilisateurs.

- Une **base de données relationnelle** est une base de données structurée suivant les principes de **l'algèbre relationnelle**.
- En 1970, **Codd** (chercheur chez IBM) proposait de stocker des données hétérogènes dans des **tables**, permettant d'établir des relations entre elles.
- Le premier prototype de Système de gestion de bases de données relationnelles (**SGBDR**) a été construit dans les laboratoires d'IBM.

Modèle relationnel: Définition et concepts

Définition - Wikipédia

Le modèle relationnel est une manière de modéliser les relations existantes entre plusieurs informations. Cette modélisation qui repose sur des principes mathématiques mis en avant par **Codd** est souvent retranscrite physiquement (implémentée) dans une base de données.

Concepts

- Attribut (Champ), domaine d'attribut
- Relation, table
- Ligne, enregistrement, n-uplet
- Clé primaire, étrangère, contrainte d'intégrité
- Schéma relationnel
- Opérateurs algébriques: union, intersection, différence, projection, sélection, produit cartésien, division, jointure.

Attribut

- Un attribut est un identificateur (un nom) décrivant une information pertinente stockée dans une base.
- On parle d'**attribut** ou de **champ**. On notera $A_1, A_2, \dots, A_n$ les différents attributs d'une relation donnée (n est le nombre d'attributs de la relation).
- **Exemple** : Personne (Id, Nom, Prénom, Sexe, EtatCivil)
- Id, Nom, Prénom, Sexe et EtatCivil sont les attributs de la relation Personne.

Domaine de valeurs

- À chaque attribut A_i est associé un **domaine de valeurs**, noté $dom(A_i)$ qui est l'ensemble des valeurs possibles que peut prendre l'attribut A_i .
- Exemples:
 - $dom(EtatCivil) = \{\text{Célibataire, Marié(e), Divorcé(e), Veuf(ve)}\}$
 - $dom(Sexe) = \{\text{Féminin, Masculin}\}$
 - $dom(Id) : entier$

Relation - Table

- Une **relation**, noté $\mathcal{R}$ est un sous-ensemble du produit cartésien de n domaines d'attributs : $\mathcal{R} \subset \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \text{dom}(A_n)$.
- Relation Personne:

id	Nom	Prénom	Sexe	EtatCiv
1	Ben Salem	Salem	Masculin	Célibataire
2	Mestiri	Farah	Féminin	Mariée
3	Rebai	Salwa	Féminin	Mariée
4	Ben Youssef	Nader	Masculin	Divorcé
5	Gammoudi	Mariam	Féminin	Veuve

- Une relation est représentée sous la forme d'un tableau à deux dimensions (qu'on appellera **Table**) dans lequel les attributs correspondent aux noms des colonnes.

Ligne, enregistrement, tuple ou n-uplet

- Chaque **ligne** de la table est un **n-uplet**, appelé aussi **enregistrement** ou **tuple**.

id	Nom	Prénom	Sexe	EtatCivil
1	Ben Salem	Salem	Masculin	Célibataire
2	Mestiri	Farah	Féminin	Mariée
3	Rebai	Salwa	Féminin	Mariée
4	Ben Youssef	Nader	Masculin	Divorcé

(3,Rebai,Salwa,Féminin,Mariée) est une ligne ou enregistrement de la relation Personne (n-uplet avec $n=5$).

Clés primaires

- Dans le but d'identifier un tuple de manière unique, le modèle relationnel repose sur la notion de clé primaire.
- Une clé primaire est un ou plusieurs attributs qui permettent d'identifier une ligne d'une relation de manière unique. Ainsi, il ne peut jamais y avoir deux lignes identiques au sein d'une relation.
- Dans la suite, nous soulignerons l'attribut pour indiquer qu'il constitue une clé primaire.
⇒ Personne (Id, Nom, Prénom, Sexe, EtatCivil)

Schéma relationnel

- À chaque relation, on associe un **schéma relationnel** noté S .
- Un **schéma relationnel** S précise le nom de la relation ainsi que la liste des attributs.
- La notion $\mathcal{R}(S)$ désigne qu'une relation $\mathcal{R}$ est de schéma S .
NB: On peut associer aux attributs leurs domaines de valeurs.
- Personne (Id, Nom, Prénom, Sexe, EtatCivil)
- Personne (Id:entier positif, Nom:chaîne, Prénom:chaîne, Sexe:{Féminin, Masculin}, EtatCivil:{Célibataire, Marié(e), Divorcé(e), Veuf(ve)})

Schéma d'une base de données relationnelle

- Une base de données relationnelle est constituée par un ensemble de relations.
- Le schéma relationnel d'une base de données est composé des schémas relationnels des différentes relations de la base.

Schéma d'une base de données relationnelle

Exemple (1) de schéma relationnel d'une bibliothèque:

- Emprunteur(idEmprunteur, Nom, Prénom, Adresse, tel)
- Livre(idLivre, Titre, datePublication, idAuteur, idEditeur, Thème)
- Auteur(idAuteur, Nom, Prénom)
- Éditeur(idEditeur, Nom, Adresse)
- Emprunt(idEmprunt, idEmprunteur, idLivre, dateEmprunt)

Questions:

- Que modélise ce schéma?
- Est-ce que ce schéma permet de modéliser le fait qu'un livre est écrit par un auteur ou par plusieurs auteurs?
- Que représente l'attribut idAuteur dans la relation Livre?

Schéma d'une base de données relationnelle

- L'exemple (1) du schéma relationnel permet de modéliser:
 - Un emprunteur possède un id, un nom, un prénom, un numéro de téléphone ainsi qu'une adresse.
 - Un livre possède un id, un titre, une date de publication et est écrit par un auteur. Chaque livre est édité chez un éditeur et est classé sous un thème donné.
 - Un auteur est caractérisé par son id, son nom et son prénom
 - Un éditeur est caractérisé par son id, son nom et son adresse.
 - Un emprunt indique que l'emprunteur dont l'id est idEmprunteur a emprunté un livre dont l'id est idLivre, à une date donnée. Un emprunt est identifié par un idEmprunt.
- Ce schéma relationnel permet de modéliser le fait qu'un livre est écrit par un seul et unique auteur.
- L'attribut idAuteur représente une **clé étrangère** dans la relation Livre.

Clés étrangères

- Une clé étrangère dans une relation est formée d'un ou plusieurs attributs qui constituent une clé primaire dans une autre relation.
- C'est le cas des attributs idAuteur et idEditeur dans la relation Livre. Ainsi que les attributs idEmprunteur et idLivre dans la relation Emprunt.

Schéma d'une base de données relationnelle

Exemple (2) de schéma relationnel d'une bibliothèque:

- Emprunteur(idEmprunteur, Nom, Prénom, Adresse, tel)
- Livre(idLivre, Titre, datePublication, idEditeur, Thème)
- Auteur(idAuteur, Nom, Prénom)
- LivreAuteur(idLivre, idAuteur)
- Éditeur(idEditeur, Nom, Adresse, tel)
- Emprunt(idEmprunt, idEmprunteur, idLivre, dateEmprunt)

Est-ce que ce deuxième schéma permet de modéliser le fait qu'un livre est écrit par un auteur ou par plusieurs auteurs?

Schéma d'une base de données relationnelle

Exemple (2) de schéma relationnel d'une bibliothèque:

- L'exemple (2) du schéma relationnel permet de modéliser le fait qu'un livre est écrit par un ou plusieurs auteurs. Remarquons la nouvelle table LivreAuteur ajoutée.
- Les deux attributs idLivre et idAuteur constituent la clé primaire de la table LivreAuteur.
- La définition d'un schéma relationnel d'une base de données est appelée étape de Conception/Modélisation. La Conception des bases de données sort du cadre de notre programme.

⇒ Mais, c'est important de comprendre la logique derrière un schéma relationnel.

Contraintes d'intégrité

Définition - Wikipédia

Une contrainte d'intégrité permet de garantir la cohérence des données lors des mises à jour de la base. En effet, les données d'une base ne sont pas indépendantes, mais obéissant à des règles sémantiques, après chaque mise à jour, le SGBD contrôle qu'aucune contrainte d'intégrité n'est violée.

Citons trois types de contraintes:

- Contrainte d'intégrité de domaine.
- Contrainte de clé.
- Contrainte de référence.

Contraintes d'intégrité

- **Contrainte de domaine:** Les valeurs d'un attribut doivent appartenir au domaine de valeurs de l'attribut en question.
- **Contrainte de clé:** Chaque clé primaire doit respecter les deux contraintes suivantes:
 - doit être unique;
 - ne doit pas être définie comme valeur nulle (*NULL*).
NB: *NULL* est une valeur spéciale en base de données qui indique qu'une valeur est manquante (non spécifiée).

Contraintes d'intégrité

- **Contrainte de référence (intégrité référentielle):** L'intégrité référentielle est une condition qui doit être vérifiée lors de situation dans laquelle une clé étrangère d'une table donnée fait référence à une clé primaire d'une autre table.
- La définition d'une intégrité référentielle entraîne un certain nombre de vérifications. Par exemple dans le cas du schéma relatif à la bibliothèque, la contrainte d'intégrité référentielle interdira :
 - L'ajout d'un livre dont l'auteur n'existe pas dans la base de données (dans la table Auteur).
 - La suppression d'un auteur tel qu'il existe un livre se référant à cet auteur.

⇒ Nous allons définir des contraintes lors de la création des tables avec SQL.

Algèbre relationnelle

- L'algèbre relationnelle est un formalisme mathématique sur lequel repose le modèle relationnel.
- Notre objectif est d'aborder l'algèbre relationnelle dans le but de décrire les opérations qu'on peut appliquer sur des relations pour produire de nouvelles relations.

Algèbre relationnelle

Nous distinguons trois catégories d'opérateurs relationnels :

- Les opérateurs **binaires** ensemblistes (**Union**, **Intersection** et **Différence**) : ces opérateurs permettent de produire une nouvelle relation à partir de deux relations de même schéma.
- Les opérateurs **unaires** (**Sélection**, **Projection**) : ces opérateurs permettent de produire une nouvelle relation à partir d'une autre relation.
- Les opérateurs **binaires** ou **n-aires** (**Produit cartésien**, **Jointure**, **Division**) : ces opérateurs permettent de produire une nouvelle relation à partir de deux ou plusieurs autres relations.

Opérateurs binaires

- Trois opérateurs binaires ensemblistes peuvent être effectuées sur des relations **de même schéma relationnel** S .
- Le résultat est une relation de schéma S .
- L'intersection: $\mathcal{R}_1(S) \cap \mathcal{R}_2(S)$
- L'union : $\mathcal{R}_1(S) \cup \mathcal{R}_2(S)$
- La différence: $\mathcal{R}_1(S) - \mathcal{R}_2(S)$

Opérateurs binaires

A_1	A_2
a	c
b	e
a	d
b	c

Table: $\mathcal{R}_1$

A_1	A_2
a	c
a	d
b	d
b	e

Table: $\mathcal{R}_2$

A_1	A_2
a	c
b	e
a	d

Table: $\mathcal{R}_1(S) \cap \mathcal{R}_2(S)$

A_1	A_2
a	c
b	e
a	d
b	c
b	d

Table: $\mathcal{R}_1(S) \cup \mathcal{R}_2(S)$

A_1	A_2
b	c

Table: $\mathcal{R}_1(S) - \mathcal{R}_2(S)$

Sélection

La **sélection** qu'on appelle aussi **restriction** permet de générer une nouvelle relation à partir d'une relation $\mathcal{R}$ contenant les enregistrements qui satisfont **une expression logique E**. On la note $\sigma_{(E)}\mathcal{R}$.

Exemple: $\sigma_{(A_1=a)}\mathcal{R}_1$.

A_1	A_2
a	c
b	e
a	d
b	c

Table: $\mathcal{R}_1$

A_1	A_2
a	c
a	d

Table: $\sigma_{(A_1=a)}\mathcal{R}_1$

Projection

On appelle **projection** de $\mathcal{R}(S)$ selon X ($X \subset S$), la relation qui consiste à générer une nouvelle relation à partir de $\mathcal{R}$ contenant les enregistrements de $\mathcal{R}$ en ne considérant que les attributs de X . On la note $\pi_{(X)}\mathcal{R}$.

Exemple: $\pi_{(A_1)}\mathcal{R}_1$.

A_1	A_2
a	c
b	e
a	d
b	c

Table: $\mathcal{R}_1$

A_1
a
b

Table: $\pi_{(A_1)}\mathcal{R}_1$

NB: Notez l'élimination des doublons lorsqu'on réalise une projection.

Produit cartésien

Le **produit cartésien** de deux relations $\mathcal{R}_1$ et $\mathcal{R}_2$ est une troisième relation regroupant exclusivement toutes les combinaisons possibles construites à partir des enregistrements des relations $\mathcal{R}_1$ et $\mathcal{R}_2$. On la note $\mathcal{R}_1 \times \mathcal{R}_2$.

A_1	A_2
a	c
b	e
a	d

Table: $\mathcal{R}_1$

A_3	A_4
f	h
g	i

Table: $\mathcal{R}_2$

A_1	A_2	A_3	A_4
a	c	f	h
a	c	g	i
b	e	f	h
b	e	g	i
a	d	f	h
a	d	g	i

Table: $\mathcal{R}_1 \times \mathcal{R}_2$

On a donc $\text{card}(\mathcal{R}_1 \times \mathcal{R}_2) = \text{card}(\mathcal{R}_1) \times \text{card}(\mathcal{R}_2)$

Jointure

- L'intérêt d'une base de données réside dans la possibilité de croiser des informations provenant de plusieurs tables par l'intermédiaire d'une jointure.
- La **jointure** de deux relations $\mathcal{R}_1$ et $\mathcal{R}_2$ est une troisième relation regroupant exclusivement toutes les combinaisons possibles construites à partir des enregistrements des relations $\mathcal{R}_1$ et $\mathcal{R}_2$ qui satisfont l'expression logique E . On la note $\mathcal{R}_1 \bowtie_{(E)} \mathcal{R}_2$.
- La jointure est un produit cartésien auquel on ajoute une sélection. Nous avons donc $\mathcal{R}_1 \bowtie_{(E)} \mathcal{R}_2 = \sigma_{(E)} (\mathcal{R}_1 \times \mathcal{R}_2)$.

Jointure

- La jointure la plus fréquente est la jointure naturelle est celle où l'expression logique porte sur l'égalité entre deux attributs A et A' tels que:
 $A \in S, A' \in S'$ et $dom(A) = dom(A')$ avec S et S' les schémas respectifs de $\mathcal{R}_1$ et $\mathcal{R}_2$.
- Nous pouvons donc écrire: $\mathcal{R}_1 \bowtie_{(A=A')} \mathcal{R}_2 = \sigma_{(A=A')} (\mathcal{R}_1 \times \mathcal{R}_2)$
- En général, la jointure est effectuée lorsque les deux tables possèdent un attribut en commun (ne portant pas nécessairement le même nom). Cet attribut va servir de jointure entre elles.
- Le plus souvent, cet attribut commun constitue d'une part une clé primaire et d'autre part une clé étrangère dans les relations de la jointure.

Jointure - Exemple

A_1	A_2
a	c
b	e
a	d

Table: $\mathcal{R}_1$

A_3	A_4	A_5
f	h	c
g	i	e

Table: $\mathcal{R}_2$

A_1	A_2	A_3	A_4	A_5
a	c	f	h	c
a	c	g	i	e
b	e	f	h	c
b	e	g	i	e
a	d	f	h	c
a	d	g	i	e

Table: $\mathcal{R}_1 \times \mathcal{R}_2$

A_1	A_2	A_3	A_4	A_5
a	c	f	h	c
b	e	g	i	e

Table: $\mathcal{R}_1 \bowtie_{(A_2=A_5)} \mathcal{R}_2$

Jointure

- En théorie, la table obtenue après jointure contient les attributs A et A' : les attributs A et A' sont alors redondants (dans notre exemple, il s'agit des attributs A_2 et A_5).
- Les deux attributs sur lesquels portent la jointure peuvent avoir le même nom. Dans ce cas, deux solutions existent:
 - procéder à une opération de renommage des attributs
 - préfixer chaque attribut par le nom de la relation: on écrira alors $\mathcal{R}_1 \bowtie_{(\mathcal{R}_1.A=\mathcal{R}_2.A)} \mathcal{R}_2$

Renommage

- Le renommage permet la modification du nom d'un ou plusieurs attributs d'une relation.
- Pour renommer l'attribut A par A' de la relation $\mathcal{R}$, on notera $\rho_{A \leftarrow A'}(\mathcal{R})$.

A_1	A_2
a	c
b	e
a	d

Table: $\mathcal{R}_1$

X_1	X_2
a	c
b	e
a	d

Table: $\rho_{A_1, A_2 \leftarrow X_1, X_2}(\mathcal{R}_1)$

Division

La division produit une relation à partir de deux relations $\mathcal{R}_1$ et $\mathcal{R}_2$ vérifiant $S_2 \subset S_1$ (S_1 et S_2 étant les schémas respectivement de $\mathcal{R}_1$ et $\mathcal{R}_2$). La relation obtenue:

- regroupe toutes les lignes de la relation $\mathcal{R}_1$ qui sont associées à toutes les lignes de la relation $\mathcal{R}_2$.
- possède tous les attributs de $\mathcal{R}_1$ que ne possède pas $\mathcal{R}_2$.

On la note $\mathcal{R}_1 \div \mathcal{R}_2$.

$\implies$ La division de $\mathcal{R}_1$ par $\mathcal{R}_2$ génère une relation qui regroupe toutes les lignes qui concaténées à chacune des lignes de $\mathcal{R}_2$ donne toujours une ligne de $\mathcal{R}_1$.

Division - Exemple

A_1	A_2	A_3
a	c	f
b	e	g
a	d	h
a	d	f

Table: $\mathcal{R}_1$

A_2
c
d

Table: $\mathcal{R}_2$

A_1	A_3
a	f

Table: $\mathcal{R}_1 \div \mathcal{R}_2$

Fonctions d'agrégation

- Les fonctions d'agrégation :
 - SOMME: permet de calculer la somme des éléments.
 - MOYENNE: permet de calculer la moyenne des éléments.
 - MINIMUM: permet de sélectionner l'élément minimum.
 - MAXIMUM: permet de sélectionner l'élément maximum.
 - COMPTE: permet de compter les éléments.
- On la notera $\gamma_{nomFonction}(nomAttribut)$

X_1	X_2
1	c
2	c
3	d

Table: $\mathcal{R}_1$

MOYENNE
2

Table: $\gamma_{Moyenne}(X_1)\mathcal{R}_1$

Fonctions d'agrégation

- Ces fonctions sont généralement employées dans le cadre d'agrégation, d'où le nom fonction d'agrégation.
- Un agrégat est un regroupement, c'est-à-dire un sous-ensemble des lignes d'une table ayant la même valeur pour un attribut donné.
- On notera cette fois $AttributAggregat_{\gamma nomFonction}(nomAttribut)$

X_1	X_2
1	c
2	c
3	d

Table: $\mathcal{R}_1$

MoyenneParGroupe
1.5
3

Table: $X_2 \gamma_{Moyenne}(X_1) \mathcal{R}_1$

⇒ En pratique, ces fonctions sont utilisées avec les clauses **group by** (agrégation) et **having** (condition de groupe) (détails en SQL): On reviendra sur ces fonctions avec les détails.

Application

En considérant le schéma relationnel de la base bibliothèque (Exemple 1), exprimer les requêtes suivantes en utilisation la notation de l'algèbre relationnelle:

- ① Le titre, ainsi que le thème de tous les livres de la bibliothèque.
- ② Toutes les informations relatives aux livres dont le thème est 'Big Data'.
- ③ Les titres ainsi que la date de publication des livres dont le thème est 'Big Data'.
- ④ Les titres des livres dont le thème est 'Big Data' ou 'Base de données'.
- ⑤ Les titres des livres écrits par 'Georges Gardarin'.
- ⑥ Tous les emprunts effectuées à la date '09-01-17'.
- ⑦ Tous les emprunts effectués par l'emprunteur d'id=2.
- ⑧ Tous les titres des livres empruntés par l'emprunteur d'id=2.
- ⑨ Tous les titres des livres empruntés par l'emprunteur 'Salem Ben Salem'.