Bases de données – cours 2
Éléments d'algèbre relationnelle

Catalin Dima

Qu'est-ce qu'une algèbre ?

- Algèbre : ensemble de domaines et d'opérations.
  - Exemple : les nombres (naturels, réels, complexes).
  - Leurs opérations : $+, -, *, /$, produit vectoriel, etc.
  - Autre exemple : la logique et ses opérateurs.
- Chaque opération a une arité :
  - Binaire pour $+, -, *, /$, unaire pour $\neg$.
- Règles de construction des opérations,
- ... ou *axiomes* définissant les opérations (lorsque celles-ci ne sont pas définissables à partir d'autres opérations plus simples !).
  - Axiomes pour $+, *$ (associativité, commutativité, distributivité, etc.).
  - Axiomes pour $\wedge, \neg$.
  - Définition pour $\vee$ (loi de Morgan).
- Calcul dans l'algèbre :
  - Variables, fonctions, (systèmes de) équations, etc.

Algèbre relationnelle

- Les objets sont des relations.
  - Définies par leur arité : nombre et nom des colonnes/attributs.
  - Chaque colonne étant définié aussi par son domaine.
  - Les variables (relvars) sont des relations aussi.
  - Les lignes d'une table, qui sont les éléments d'une relation, ne sont pas référencés par les relvars !
    - ... mais le plus souvent on définit les relations par les propriétés que leur tuples doivent satisfaire !
    - D'où une des difficultés du calcul relationnel...
- Les opérateurs sont de plusieurs types :
  - Opérateurs d'ensembles (eh oui ! une relation est un ensemble !).
  - Opérateurs sur les tuples (mais définis au niveau relation, et pas au niveau tuple !).
  - Opérateurs d'aggregation.
  - Opérateurs dérivés, utilisant les contraintes.

Domaines (ou types de données)

- Chaque colonne dans un table est accompagnée par le domaine de valeurs auquel appartiennent les éléments de la colonne.
  - Domaine = réservoir de valeurs légales.
  - Exemple : INTEGER, CHAR (chaîne de caractères), BOOLEAN, FLOAT, DOUBLE, DATE, TIME, etc.
  - Ces sont des domaines standard dans chaque LDD/LMD, même si souvent sous des noms différents.
  - Souvent on parle aussi de type = domaine.
- Chaque domaine vient avec son propre ensemble d'opérations :
  - Opérations booléennes pour BOOLEAN, non-disponibles pour les autres types.
  - Opérations arithmétiques pour INTEGER, FLOAT, DOUBLE (avec leurs restrictions).
  - Opérations spécifiques sur les chaînes de caractères.
- Ces informations sont stockées dans le catalogue d'un SGBD.

Arité et tuples (ou *n*-uplets)

- Une relation est définie par :
  - L'ensemble de ses colonnes (arité), chacune ayant son nom et domaine.
  - L'ensemble de ses éléments.
- Tuple = élément d'une relation.
- Si les colonnes d'une relation ont les domaines D_1, D_2, D_3, alors une relation est un sous-ensemble

$$R \subseteq D_1 \times D_2 \times D_3$$

  - Ses éléments sont des tuples de valeurs (v_1, v_2, v_3) où $v_1 \in D_1, v_2 \in D_2, v_3 \in D_3$.
  - Dans cette notation le nom des colonnes est l'indice des domaines (ou l'indice des éléments de chaque tuple).
  - R est le plus souvent strictement inclue dans $D_1 \times D_2 \times D_3$! Une table particulière ne contient que très rarement toutes les combinaisons de valeurs !
- Cardinalité d'une relation = nombre de tuples.
- Clé primaire = attribut qui a une valeur différente pour chaque tuple dans la relation.

Représentation tabulaire

Relation = tableau :

- En-tête donnant le nom de chaque colonne (ou attribut), plus son domaine.
- Une ligne par tuple dans la relation.
  - Plusieurs colonnes peuvent avoir le même type !
  - Certaines valeurs du domaine d'une colonne peuvent ne pas être présentes dans la colonne respective.
- Arité = (grosso modo) l'en-tête du tableau.
  - Mais assez souvent, l'arité ne désigne que le nombre de colonnes, pas leur nom et type.
- Quelques contraintes :
  - Unicité des tuples (mais, à la suite d'application de diverses opérations de mise à jour, on peut obtenir des duplicats... à revoir !).
  - Pas d'ordre dans la liste des tuples d'une relation.
  - Pas d'ordre dans la liste des noms d'attributs.
  - Valeurs atomiques dans chaque colonne (aussi à revoir...).

Exemple de relation

Id INT	Nom CHAR(20)	Prénom CHAR(20)	Dept CHAR(5)	Salaire(kE) INT
1	Dupont	Jean	D1	32
10	Ndiaye	Cyril	D3	40
4	Dupont	Anne	D2	27
67	Mahrzoug	Amine	D3	31
6	Bonnet	Franck	D2	27

- Arité/schéma : ...
- Attributs et leur domaines : ...
- Clé primaire : “Id”.

Opérations ensemblistes sur les relations

- Supposons deux relations R_1, R_2 (ou deux tables).
  - Hypothèse : $R_1, R_2 \subseteq D_1 \times \ldots \times D_n$ (même schéma de relation !).
- Ces sont des *ensembles* de tuples, donc les opérations ensemblistes peuvent s'appliquer sur R_1 et R_2.

$$R_1 \cup R_2 = \{t \mid t \in R_1 \textbf{ ou } t \in R_2\}$$
$$R_1 \cap R_2 = \{t \mid t \in R_1 \textbf{ et } t \in R_2\}$$
$$R_1 \setminus R_2 = \{t \mid t \in R_1 \textbf{ et } t \notin R_2\}$$

- Contraintes :
  - Les deux relations doivent avoir la même arité/schéma et les mêmes attributs (noms et domaines).
  - Avant de calculer les résultats d'opérations, les attributs des deux relations doivent être placés dans le même ordre dans les deux opérations (en prenant une des deux comme "patron").

Opérations non-booléennes sur les relations

Opérations dont le résultat n'a pas la même structure (arité) que les opérandes :

- Opérations enlevant une partie de la relation : sélection (éliminer des tuples selon une règle spécifique), et projection (éliminer des colonnes).
- Opérations augmentant les relations : produit cartésien, jointures (divers types), etc.
- Opération de renommage d'attributs.

Projection

- Projection d'une relation $R \subseteq D_1 \times D_2 \times \ldots \times D_n$ sur un sous-ensemble d'attributs $i_1, \ldots, i_k$:

$$\pi_{i_1,\ldots,i_k}(R) = \left\{(t_{i_1}, \ldots, t_{i_k}) \mid (t_1, t_2, \ldots, t_k) \in R\right\}$$

- Prenons notre table employés
  $R \subseteq \text{INT}_{\text{Id}} \times \text{CHAR(20)}_{\text{Nom}} \times \text{CHAR(20)}_{\text{Pre}} \times \text{CHAR(5)}_{\text{Dept}} \times \text{INT}_{\text{Sal}}$.
- La projection $\pi_{\text{Nom,Dept,Sal}}$ est la suivante :

Nom CHAR(20)	Dept CHAR(5)	Salaire(kE) INT
Dupont	D1	32
Ndiaye	D3	40
Dupont	D2	27
Mahrzoug	D3	31
Bonnet	D2	27

Produit cartésien

- Étant donnés deux relations :

$$R_1 \subseteq D_1 \times \ldots \times D_n$$
$$R_2 \subseteq E_1 \times \ldots \times E_m$$

  Leur produit cartésien est une relation :

$$R_1 \times R_2 \subseteq D_1 \times \ldots \times D_n \times E_1 \times \ldots \times E_m$$

  définie comme suit :

$$R_1 \times R_2 = \left\{ (t_1, \ldots, t_n, u_1, \ldots, u_m) \mid (t_1, \ldots, t_n) \in R_1, (u_1, \ldots, u_m) \in R_2 \right\}$$

- Chaque tuple de R_1 crée, ensemble avec chaque tuple de R_2, un nouveau tuple de $R_1 \times R_2$.
- Le schéma de $R_1 \times R_2$ contient la liste d'attributs de R_1, suivie (duplicats possibles !) de la liste d'attributs de R_2.
  - Désambiguation des attributs provenant des deux relations : préfixer par le nom de la relation – comme pour les champs d'un `struct` en C ou objet en Java !
- Exemple...

Sélection

- Supposons une relation $R \subseteq D_1 \times \ldots \times D_n$,
- ... et une condition C, définie sur les domaines $D_1, \ldots, D_n$ et qui utilise comme variables les noms d'attributs de la relation R.
  - Par exemple, définie par une combinaison booléenne de comparaisons d'expressions arithmétiques...
  - Exemple plus précis pour notre table employés :

$$C = (\text{Salaire} < 35) \wedge (\mathit{startswith}(\text{Nom}, \text{Du}))$$

- La sélection de R contrainte par C est la nouvelle relation :

$$\sigma_(R) = \{ t \in R \mid \text{ les valeurs d'attributs dans } t \text{ satisfont } C \}$$

- Plus précisément, on élimine tout tuple dont les valeurs d'attributs ne satisfont pas C.
- Exemple avec notre table employés et la condition C ci-dessus...

Jointure naturelle

- Étant donnés deux relations :

$$R_1 \subseteq D_1 \times \ldots \times D_n$$
$$R_2 \subseteq E_1 \times \ldots \times E_m$$

- Supposons qu'une partie des attributs de R_1 est identique (nom+domaine) à une partie des attributs de R_2.
  - Pour simplicité, supposons que cette partie est $D_1, \ldots, D_k$ pour R_1, attributs qui sont identiques (tant leur nom que leur domaine) respectivement avec les attributs $E_1, \ldots E_k$ de R_2.
- La jointure naturelle de R_1 avec R_2 est :

$$R_1 \bowtie R_2 = \{(t_1, \ldots, t_n, u_{k+1}, \ldots, u_m) \mid (t_1, \ldots, t_n) \in R_1, (t_1, \ldots, t_k, u_{k+1}, \ldots, u_m) \in R_2\}$$

- On fusionne chaque tuple t de R_1 avec un tuple u de R_2 si, dans la partie commune d'attributs, t a les mêmes valeurs que u.
  - On peut exprimer cette condition en utilisant la projection *sur les tuples* : $\pi_{D_1,\ldots,D_k}(t) = \pi_{E_1,\ldots,E_k}(u)$.

Exemple de jointure naturelle

(Tous les domaines ci-dessous sont le domaine des entiers) :

$R_1 =$

A	B	C
1	2	5
3	4	6
2	5	12
6	7	8

$R_2 =$

B	C	D	E
2	5	6	3
4	6	8	10
5	7	10	3
4	6	7	8
3	4	6	8

$R_1 \bowtie R_2 =$

A	B	C	D	E
1	2	5	6	3
3	4	6	8	10
3	4	6	7	8

- Remarquer qu'il y a, tant dans R_1 que dans R_2, des tuples défaillants – qui ne participent pas dans les tuples résultats.
- Remarquer aussi qu'il y a des tuples dans R_1 qui peuvent se combiner avec plusieurs tuples de R_2 pour produire des tuples résultat.

θ-jointure

- Prenons encore une fois deux relations :

$$R_1 \subseteq D_1 \times \ldots \times D_n$$
$$R_2 \subseteq E_1 \times \ldots \times E_m$$

- Et une condition C sur l'ensemble des attributs (de R_1 et de R_2).
- La θ-jointure dirigée par C de R_1 avec R_2 est :

$$R_1 \overset{\bowtie}{C} R_2 = \{(t_1, \ldots, t_n, u_1, \ldots, u_m) \mid (t_1, \ldots, t_n) \in R_1, \\ (u_1, \ldots, u_m) \in R_2 \text{ et } (t_1, \ldots, u_m) \text{ satisfait } C\}$$

- Il s'agit en fait d'une combinaison d'un produit cartésien de R_1 et R_2, suivi par une sélection dirigée par C du résultat.
- Plus formellement,

$$R_1 \overset{\bowtie}{C} R_2 = \sigma_C(R_1 \times R_2)$$

- Exemple avec $R_1 \overset{\bowtie}{C} R_2$ ou $C = R_1.C < R_2.B$ et nos deux relations entières ci-dessus.

Jointure externe

- Parfois, perdre les tuples défaillants n'est pas convenable...
- Opération de jointure permettant de retrouver aussi les tuples défaillants, en mettant des valeurs "inconnu" (ou $\perp$) pour les attributs qui ne sont définis pas dans les résultats.
- Deux variantes : jointure externe *gauche* et *droite*.
- Formellement, en partant du cadre des jointures naturelles :

$$R_1 \overset{\circ}{\bowtie}_L R_2 = \{(t_1, \ldots, t_n, u_{k+1}, \ldots, u_m) \mid (t_1, \ldots, t_n) \in R_1, \\ \text{si } (u_{k+1}, \ldots, u_m) \notin R_2 \text{ alors } u_{k+1} = \ldots = u_m = \perp\}$$

- Définition similaire pour la jointure externe droite $R_1 \overset{\circ}{\bowtie}_R R_2$, ainsi que pour celle bilaterale.

Exemple de jointure externe gauche et bilaterale

$R_1 =$

A	B	C
1	2	5
3	4	6
2	5	12
6	7	8

$R_2 =$

B	C	D	E
2	5	6	3
4	6	8	10
5	7	10	3
4	6	7	8
3	4	6	8

$R_1 \overset{\circ}{\bowtie}_L R_2 =$

A	B	C	D	E
1	2	5	6	3
3	4	6	8	10
3	4	6	7	8
2	5	12	$\perp$	$\perp$
6	7	8	$\perp$	$\perp$

$R_1 \overset{\circ}{\bowtie} R_2 =$

A	B	C	D	E
1	2	5	6	3
3	4	6	8	10
3	4	6	7	8
2	5	12	$\perp$	$\perp$
6	7	8	$\perp$	$\perp$
$\perp$	5	7	10	3
$\perp$	3	4	6	8

Renommage

- Il s'agit simplement de renommer la relation et/ou un (ou plusieurs) attribut(s) d'une relation.
- Pour une relation dont les attributs sont nommés $A_1, \ldots, A_n$,
  - C'est à dire, dans notre notation, $R \subseteq D_{A_1} \times \ldots \times D_{A_n}$,
- L'opération de renommage de R en S et de tous les attributs en $B_1, \ldots, B_n$ est notée :

$$\rho_{S(B_1,\ldots,B_n)}$$

- Exemple pour notre table employé, le résultat du renommage $\rho_{\text{list-emp(IdEmp,Nom,Pre,Ind,Rev)}}$ est le suivant :

IdEmp INT	Nom CHAR(20)	Pre CHAR(20)	Ind CHAR(5)	Rev INT
1	Dupont	Jean	D1	32
10	Ndiaye	Cyril	D3	40
4	Dupont	Anne	D2	27
67	Mahrzoug	Amine	D3	31
6	Bonnet	Franck	D2	27

Autres opérations

- Semi-jointure de deux relations R_1 et R_2, notée $R_1 \ltimes R_2$: le multi-ensemble de tuples t de R_1 pour lequel il existe au moins un tuple t' dans R_2 qui possède les mêmes valeurs sur tous les attributs communs entre les deux relations.
- Quotient (ou division) :
  - Supposons deux relations $R_1 = (A_1, \ldots, A_n, B_1, \ldots B_m)$, et $R_2 = (B_1, \ldots, B_m)$ (donc attributs de $R_2 \subseteq$ attributs de R_1).
  - $R_1 \div R_2$ composé de tous les tuples t sur les attributs propres à R_1 tels que pour tout tuple s dans R_2, le tuple ts appartient à R_1.
  - ts correspond au “produit cartésien” des deux tuples t et s.

Opérations combinées

- Toutes les opérations présentées peuvent être combinées entre elles.
- Les requêtes de BD sont des combinaisons de ces opérations.
- Exemple :

  Donner les noms des employés dont le salaire est inférieur à 40kE.

- Une solution possible : $\pi_{\text{Nom}}\left(\sigma_{\text{Salaire}<40}(\text{Employés})\right)$
- Une activité essentielle tout au long de ce module : construire des combinaisons d'opérations pour résoudre une requête.
  - Une petite partie en algèbre relationnelle, puis la plupart du temps en SQL (qui est une implémentation de l'algèbre relationnelle) !

Opérations dépendentes

- Est-ce qu'on a défini toutes les opérations dont on va jamais s'en servir pour écrire des requêtes ?
  - D'autres variantes de jointure/auto-jointure, division, etc. etc...
- Heureusement, une petite partie de ces opérations suffira !
  - Comme pour la logique : pas besoin de considérer tous les opérateurs booléens, il suffit d'utiliser deux ("et" et "non") et on peut tout exprimer !
- Opérateurs de base (ou indépendents) :
  - Union, différence, projection, sélection, produit cartésien et renommage.
  - Tous les autres opérateurs peuvent se définir à partir de ces six !
  - Bien connu pour l'intersection.
  - Jointure naturelle : ...
- Une autre possibilité d'ensemble d'opérations de base : sélection, projection, renommage, jointure naturelle, opérateurs booléens.

Multi-ensembles et relations sur les multi-ensembles

- Certaines opérations peuvent produire des relations ayant des *duplicats*.
- Cela sort de notre modèle de relations en tant qu'ensembles.
- Modèle d'ensemble avec duplicats : **multi-ensemble**.
- Formellement, un **multi-ensemble** sur un ensemble A est une *fonction* $f : A \longrightarrow \mathbb{N}$:
  - $f(a) = 0$ indique que a n'apparaît pas dans le multi-ensemble f.
  - $f(a) = 1$ indique que a apparaît une seule fois dans f.
  - $f(a) = 2$ indique que a apparaît *deux fois* dans f, etc.
- Opérations booléennes positives (union, intersection) :

$$(f_1 \cup f_2)(a) = f_1(a) + f_2(a)$$
$$(f_1 \cap f_2)(a) = \min(f_1(a), f_2(a))$$

- Le complément ne peut se définir que si A est fini.

Les autres opérations de l'algèbre relationnelle sur les multi-ensembles

$$(f_1 \times f_2)(a_1, \ldots, a_m, b_1, \ldots, b_n) = f_1(a_1, \ldots, a_m) * f_2(b_1, \ldots, b_n)$$

$$\tilde{\pi}_{i_1,\ldots,i_k}(f)(a_1, \ldots, a_n) = \sum \{ f(t) \mid t \in D_1 \times \ldots \times D_n, \pi_{i_1,\ldots,i_k}(t) = \pi_{i_1,\ldots,i_k}(a_1, \ldots, a_n) \}$$

$$\tilde{\sigma}_C(f)(t) = \begin{cases} f(t) & \text{si } t \text{ satisfait } C \\ 0 & \text{sinon.} \end{cases}$$

- Les autres opérations se généralisent facilement aussi.

Opérations étendues

- *Élimination des duplicats* : pour convertir un multi-ensemble en ensemble.
  - Notation : $\delta(R)$, où R est le multi-ensemble donné.
- Opération aggrégées :
  - *SUM*, *AVG* (moyenne), *MIN*, *MAX*, *COUNT*.
  - Ex. : $SUM(R.Salaire)$ produit la somme des salaires des employés.
- Opérateurs de tri :
  - Étant donnée une relation R et une liste d'attributs L de R, $\tau_L(R)$ trie les tuples de R selon les valeurs des attributs dans L, considérés dans l'ordre de leur apparition dans la liste.
- Projection étendue : certains attributs peuvent changer de nom, d'autres nouveaux peuvent être construits.

Opérateurs de groupement

- L'opérateur de groupement, premier cas : considérons une relation R et une liste d'attributs L.
- $\gamma_L(R)$ représente l'ensemble de tuples construits comme suit :
  - Regrouper d'abord les tuples de R en un ensemble de groupes, chaque groupe ayant une seule combinaison de valeurs pour les attributs dans L.
  - Pour chaque élément de l'ensemble de groupes, produire un seul tuple dans le résultat.
  - Exemple :

$R =$

A	B	C
1	2	3
1	2	4
1	1	3
2	1	3
2	1	2

$\gamma_{A,B}(R) =$

A	B
1	2
1	1
2	1

Opérateur de groupement (2)

- L'opérateur de groupement, 2e cas : considérons une relation R, une liste d'attributs L et un opérateur aggrégé op.
- $\gamma_{L,op->NvAttr}(R)$ se construit comme suit :
  - Regrouper d'abord les tuples de R en un ensemble de groupes, chaque groupe ayant une seule combinaison de valeurs pour les attributs dans L.
  - À l'intérieur de chaque groupe, appliquer l'opérateur d'aggrégation op, et produire, avec les résultats, une nouvelle colonne contenant le résultat.
  - Enfin, produire un tuple pour chaque combinaison de valeurs présente pour les attributs dans la liste L plus $NvAttr$.
- Exemple pour $\gamma_{SUM(B)}(R)$ et la relation ci-dessus :

$\gamma_{A,SUM(B)->SB}(R) =$

a	SB
1	5
2	2

- Noter que l'élimination des duplicats est une opération dérivée de l'opération de groupement.

Construction des requêtes

Il est rare de trouver des expressions résolvant des requêtes sur une seule BD.

- Supposons une BD `films`, de schéma (Titre, Producteur, Réalisateur, Budget, ActeurPP, ActricePP)
- ... et une autre BD `programme` (cinéma) de schéma (Titre, Réalisateur, Date, Salle, NbSpect).
- Quels sont les *acteurs* qui ont été vus (en rôle principal) par au moins 100 spectateurs ?
- Solution :
  - Faire une jointure naturelle entre `films` et `programme` pour croiser les informations.
  - Puis appliquer un opérateur de groupement sur les attributs *acteur* et utilisant un opérateur d'aggrégation de type SUM sur le *NbSpect*, pour construire une nouvelle BD ayant que les attributs *acteur* et la somme des spectateurs visionnant les films avec l'acteur respectif.
  - Enfin, appliquer une sélection pour ne garder que les tuples de cette nouvelle relation dont le 2e attribut est supérieur à 1000.