

Chapitre 5: les langages de manipulation des données relationnelles

Partie I: L'algèbre relationnelle

- L'algèbre relationnelle a été proposée par CODD comme une collection d'opérateurs permettant de réaliser des opérations sur des relations.
- Elle permet par exemple de sélectionner certains enregistrements d'une relation satisfaisant une condition ou encore de regrouper des enregistrements de relations différentes.
- Le résultat de l'application d'un opérateur sur une ou deux relations est une nouvelle relation.

1. Opérations ensemblistes

Soient 2 relations R et S de même schéma.

- **Union:** Opération binaire portant sur R et S produisant une relation T de même schéma ayant pour tuples ceux appartenant à R et S.

Notation: $T = R \cup S$ $T = \text{UNION}(R, S)$

Exemple: R

A	B
a	b
b	b
y	z

S

A	B
u	v
y	z

$R \cup S$

A	B
a	b
b	b
y	z
u	v
y	z

- **Intersection:** Opération binaire portant sur R et S ayant le même schéma qui produit une relation T de même schéma contenant les tuples appartenant à la fois à R et S.

Notation: $T = R \cap S$ $T = \text{INTERSECT}(R, S)$

Exemple: **R**

A	B
a	b
b	b
y	z

S

A	B
u	v
y	z

$R \cap S$

A	B
y	z

- **Différence:** Opération binaire portant sur R et S produisant une relation T de même schéma ayant pour tuples ceux appartenant à R mais pas à S.

Notation: $T = R - S$

$T = \text{MINUS}(R, S)$

Exemple: **R**

A	B
a	b
b	b
y	z

S

A	B
u	v
y	z

$R - S$

A	B
a	b
b	b

Soient 2 relations R et S de schémas quelconques.

- **Produit cartésien:** Opération binaire portant sur R et S produisant une relation T ayant pour schéma la concaténation des attributs de R et S et pour tuples les combinaisons des relations opérandes.

Notation: $T = R \times S$

$T = \text{PRODUCT}(R, S)$

Exemple1:

R

A	B
a1	b1
a2	b2

S

C	D
c1	d1
c2	d2

R x S

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a2	b2	c1	d1
a2	b2	c2	d2

Exemple2:

NSS	Nom	Prénom	Grade	Dept
12345	Manouvrier	Maude	MCF	1
45678	Toto	Titi	Prof	2

La relation Enseignant

Dept ID	Nom Dept
1	Info
2	Math

La relation Département

NSS	Nom	Prénom	Grade	Dept	Dept ID	Nom Dept
12345	Manouvrier	Maude	MCF	1	1	Info
45678	Toto	Titi	Prof	2	1	Info
12345	Manouvrier	Maude	MCF	1	2	Math
45678	Toto	Titi	Prof	2	2	Math

La relation Enseignant x Département

2. Opérations unaires

- **Projection:** Opération unaire portant sur R qui produit une relation T en enlevant du schéma de R tous les attributs non mentionnés en opérande et en éliminant les tuples en double.

Notation: $T = \Pi_{A_i, A_j, A_k}(R)$ $T = \text{PROJECT}(R / A_i, A_j, A_k)$.

Exemple:

Resultat

NSS	CodeMod	Note
001	BDD	15
001	SE	10
001	DAW	16

$\text{PROJECT}(R / \text{CodeMod})$

CodeMod
BDD
SE
DAW

- **Sélection(Restriction):** Opération unaire portant sur R qui produit une relation T de même schéma ayant pour tuples ceux vérifiant la condition donnée en opérande.

Notation: $T = \sigma_{\text{cond}}(R)$ $T = \text{RESTRICT}(R / \text{cond})$

Exemple: $\text{RESTRICT}(\text{Resultat} / \text{CodeMod} = \text{'BDD'})$

NSS	CodeMod	Note
001	BDD	15

- **Renommage:** Opération unaire permet de renommer les attributs d'une relation

Notation: $\rho_{A_1 \rightarrow B_1, \dots, A_n \rightarrow B_n}(R)$

Exemple: Renommer l'attribut *Note* par l'attribut *Moyenne* dans la table T

$T = \text{RESTRICT}(\text{Resultat} / \text{CodeMod} = \text{'BDD'})$

$\rho_{\text{Note} \rightarrow \text{Moyenne}}(T)$

NSS	CodeMod	Moyenne
001	BDD	15

3. Opérations additionnelles

- Ces opérations se déduisent des opérations précédentes.
- Elles sont essentielles pour la réalisation des systèmes relationnels.
- **Jointure:** Opération binaire portant sur R et S consistant à rapprocher les tuples des 2 relations selon un critère C pour former une relation T qui contient l'ensemble des tuples du produit cartésien $R \times S$ vérifiant le critère C.

Notation: $T = R \bowtie S$ $T = \text{JOIN}(R,S)$

Exemple:

Client

NumCl	NomCl	AdrCl
Cl01	Batam	Constantine
Cl02	AMS	Sétif
Cl03	ProdElec	Batna
Cl04	Electron	Elelma

Commande

NCmde	DateCmde	NumCl
C001	10/12/2009	CL02
C002	12/03/2010	CL04
C003	15/01/2011	CL03
C004	20/06/2009	CL02

Client ⚡ **Commande** (commande.NumCl=Client.NumCl)

NumCl	NomCl	AdrCl	NCmde	DateCmde
Cl02	AMS	Sétif	C001	10/12/2009
Cl02	AMS	Sétif	C004	20/06/2009
Cl03	ProdElec	Batna	C003	15/01/2011
Cl04	Electron	ELelma	C002	12/03/2010

Supposons que la relation **R** soit définie sur l'ensemble d'attributs **A** et que la relation **S** soit définie sur l'ensemble d'attributs **B**, de telle sorte que $\mathbf{B} \subseteq \mathbf{A}$. Soit $\mathbf{C} = \mathbf{A} - \mathbf{B}$.

➤ **Division**: Opération binaire portant sur R et S, elle définit une relation T sur les attributs **C**, tel que un tuple t est dans T si et seulement si pour tout tuple s de S, le tuple $\langle t, s \rangle$ est dans R

Notation: $\mathbf{T} = \mathbf{R} \div \mathbf{S}$ $\mathbf{Q} = \text{DIVISION}(\mathbf{R}, \mathbf{S})$

- Le schéma de **T** est constitué de tous les attributs de **R** n'appartenant pas à **S**.
- La division est utilisée pour répondre aux requêtes de type «**tous les** »

Exemple:

Quelles sont les commandes qui portent sur tous les produits ?

Cmde

NCmd	NumP	Qte
C001	P001	250
C001	P002	180
C001	P004	200
C002	P001	120
C002	P002	150
C003	P002	200
C004	P004	250

Produit

NumP
P001
P002
P004

Cmde ÷ produit

NCmd
C001

Résumé: sémantique et notation des opérateurs

Opérateur	Sémantique	Notation
Renommage	Renommer des attributs d'une relation	$\rho_{A1 \rightarrow B1, \dots, An \rightarrow Bn}(R)$
Selection	Sélectionner des tuples vérifiant une condition	$T = \sigma_{\text{cond}}(R)$
Projection	Sélectionner des attributs d'une relation	$T = \Pi_{A_i, A_j, A_k}(R)$
Union	Fusionner les extensions de 2 relations	$T = R \cup S$
Intersection	Obtenir l'ensemble des tuples communs à deux relations	$T = R \cap S$
Différence	Tuples d'une relation qui ne figurent pas dans l'autre	$T = R - S$
Produit cartésien	Concaténer chaque tuple de R avec chaque tuple de S	$T = R \times S$
Jointure	Etablir le lien sémantique entre les relations	$T = R \bowtie S$
Division	répondre aux requêtes de type «tous les »	$T = R \div S$

Exercice:

Soient les relations suivantes :

PILOTE (NUMPIL, NOMPIL, ADR, SA L)

AVION (NUMAV, NOMAV, CAPACITE, LOC)

VOL (NUMVOL, #NUMPIL, #NUMAV, VILLE_DEP, VILLE_ARR, H_DEP, H_ARR)

Exprimez les requêtes suivantes à l'aide des opérateurs de l'algèbre relationnelle :

- a. Donnez la liste des avions dont la capacité est supérieure à 350 passagers.
- b. Quels sont les numéros et noms des avions localisés à Alger ?
- c. Quel est le nom des pilotes domiciliés à Oran dont le salaire est supérieur à 200000 DA ?.
- d. Quels sont les numéros des pilotes qui ne sont pas en service ?
- e. Donnez le numéro des vols effectués au départ de Alger par des pilotes Algérois.

Partie II: Le langage SQL

Objectifs de SQL

- Créer la structure de la base de données et de ses tables
- Exécuter les tâches de base de la gestion des données, telle que l'insertion, la modification et la suppression de données des tables
- Effectuer des requêtes simples ou complexes

Caractéristiques de SQL

Fonctionnalités:

Extraction des données

- SELECT

Langage de définition des données LDD

- CREATE; ALTER; DROP; RENAME

Langage de manipulation des données LMD

- INSERT; UPDATE;
DELETE; MERGE

Langage des transactions

- SAVEPOINT; COMMIT;
ROLLABCK

Langage de contrôle de données

- GRANT; REVOKE

Utilisé par : DBA, développeurs, quelques utilisateurs

On s'intéresse dans ce cours à la fonctionnalité: Extraction des données

Extraction des données

- La recherche d'information dans une base de données s'effectue à l'aide de la commande SQL SELECT.

Syntaxe:

```
SELECT <liste des attributs du résultat>  
FROM <liste des tables impliquées dans la requête>  
WHERE <formule de sélection> ;
```

- La formule de sélection utilise les colonnes des tables présentes dans la clause FROM ;
- La formule de sélection peut contenir :
 - Les opérateurs de comparaison : >, <, >=, <=, <>
 - Les opérateurs logiques : AND, OR, NOT
 - Les prédicats : ALL, ANY, EXISTS, BETWEEN, LIKE, IN (à combiner éventuellement avec NOT)
 - Des sous-requêtes

On se basera sur cet exemple de schéma pour illustrer les commandes SQL:

PRODUIT(NP, LIB, QTE ,PRIX)

ACHAT(NP, NCLI, QTEA)

CLENT(NCLI, NOM, TEL, ADR)

CLIENT

NCL	NOM	TEL	ADR
CL001	ALI	06612526	ORAN
CL002	MED	07715280	BATNA
CL005	SAMIR	06634132	ANNABA
CL010	RADI	05526263	ALGER
CL020	KADER	07758081	BATNA

PRODUIT

NP	LIB	QTE	PRIX
P001	Robinet	1200	1000
P002	Prise	1000	300
P003	Cable	1500	200
P004	Peinture	900	1500

ACHAT

NP	NCL	QTEA
P001	CL010	250
P003	CL020	500
P004	CL001	200

Projection

SELECT liste d'attributs
FROM nom_relation;

Exemple:

SELECT LIB, QTE
FROM PRODUIT;




LIB	QTE
P001	1200
P002	1000
P003	1500
P004	900

Pour éliminer les doubles il faut rajouter le mot clé DISTINCT
SELECT DISTINCT LIB, QTE
FROM PRODUIT;

'*' : permet de désigner dans la clause SELECT tous les attributs d'une table

SELECT *
FROM ACHAT;



NP	NCL	QTEA
P001	CL010	250
P003	CL020	500
P004	CL001	200

Sélection

- Spécification d'un Prédicat dans la clause « WHERE »

Notion de Prédicat :

- Expression logique à laquelle on peut répondre par VRAI ou par FAUX pour chaque ligne de la table opérande.
- Une restriction ne garde que les lignes pour lesquelles le prédicat est vrai.

➤ *Prédicat simple :*

- NomRelation.NomAttribut θ Valeur ou
- NomAttribut θ Valeur
- θ est un comparateur : =, <, <=, >, >=, <>

Exemple: liste des clients habitant à BATNA

```
SELECT *
```

```
PROM CLIENT
```

```
WHERE ADR = 'BATNA';
```

➤ *Prédictat composé :*

- Combinaison de prédicats reliés par les connecteur AND (ET), OR (OU)
- L'utilisation de parenthèses peut être utile pour préciser comment combiner des prédicats reliés par des AND et des OR

Exemple: liste des numéros et libellés des produits dont la quantité >50 et le prix <1000

```
SELECT NP, LIBP
```

```
PROM PRODUIT
```

```
WHERE QTE >50 AND PRIX <1000;
```

➤ **NOT** : inverse le résultat d'un prédicat

Exemple: liste des numéros et libellés des produits dont la quantité >50 et le prix <1000

```
SELECT NP, LIBP  
FROM PRODUIT  
WHERE NOT(QTE <=50 OR PRIX >=1000);
```

➤ **BETWEEN** : Pour spécifier un prédicat comportant un intervalle

Exemple: les numéros de clients avec les numéros de produits achetés en quantité entre 500 et 1000

```
SELECT NP, NCL  
FROM ACHAT  
WHERE QTEA BETWEEN 500 AND 1000;
```

➤ **Prédicat avec IN** : teste si la valeur d'une colonne est incluse dans un ensemble

Exemple: les noms et téléphones des clients d'adresse 'Alger' ou 'Oran' ou 'Batna'

```
SELECT NOM, TEL  
FROM CLIENT  
WHERE ADR IN( 'Alger', 'Oran', 'Batna' );
```

➤ **Prédicat avec LIKE** : Comparer la chaîne de caractère d'une colonne à un Modèle

• **Modèle** : chaîne de caractère avec des jokers :

'_' (souligné) → remplace tout caractère isolé

'%' (pourcentage) → remplace Zéro, un ou plusieurs caractères

Exemple: nom des clients dont l'adresse commence par A.

```
SELECT NOM
FROM CLIENT
WHERE ADR LIKE 'A%';
```

➤ **UPPER, LOWER** : transformer la chaîne de caractère d'une colonne majuscules ou en minuscules

Exemple: les clients habitant à Annaba.

```
SELECT NCL, NOM
FROM CLIENT
WHERE UPPER(ADR)='ANNABA'
```

Jointure:

➤ **Produit cartésien**(jointure sans qualification)

```
SELECT *
```

```
FROM Relation1, Relation2;
```

Exemple: produit cartésien des tables Produit et achat

```
SELECT *
```

```
FROM PRODUIT, ACHAT;
```

➤ **Jointure avec qualification:**

Exprimée par un prédicat dans la clause WHERE

```
NomRel1.NomAtt1  $\theta$  NomRel2.NomAtt2
```

θ est un comparateur : =, <, <=, >, >=, <>

Exemple: Les libellés des produits achetés
avec leurs quantités d'achat

```
SELECT LIB, QTEA  
FROM PRODUIT, ACHAT  
WHERE PRODUIT.NP= ACHAT.NP;
```



LIB	QTEA
Robinet	250
Cable	500
Peinture	200

ACHAT

NP	NCL	QTEA
P001	CL010	250
P003	CL020	500
P004	CL001	200

PRODUIT

NP	LIB	QTE	PRIX
P001	Robinet	1200	1000
P002	Prise	1000	300
P003	Cable	1500	200
P004	Peinture	900	1500

➤ **Jointure par requête imbriquées avec l'opérateur IN:**

Une sous-requête produit un ensemble de valeurs compatibles avec l'attribut de jointure de la relation dite externe

Exemple: Les libellés des produits qui ont été achetés

```
SELECT LIB
FROM PRODUIT
WHERE NP IN (SELECT NP
             FROM ACHAT);
```

➤ **EXISTS** : tester l'existence d'un lien via une sous-requête

Le prédicats EXISTS est VRAI si la sous-requête associée ramène au moins une ligne résultat.

Exemple: Les libellés des produits qui ont été achetés

```
SELECT LIB
FROM PRODUIT
WHERE EXISTS (SELECT *
             FROM ACHAT
             WHERE QTEA >=150
             AND PRODUIT.NP = ACHAT.NP);
```


Opérations ensemblistes: union, intersection, différence

```
SELECT Nom,Prénom
FROM Enseignant
WHERE Département_ID = ' INFO '
INTERSECT
SELECT Nom,Prénom
FROM Enseignant
WHERE Département_ID = ' MATH '
```

```
SELECT Nom,Prénom
FROM Enseignant
WHERE Département_ID = ' INFO '
UNION
SELECT Nom,Prénom
FROM Enseignant
WHERE Département_ID = ' MATH '
ORDER BY Nom,Prénom
```

```
SELECT Nom,Prénom
FROM Enseignant
WHERE Département_ID = ' INFO '
MINUS
SELECT Nom,Prénom
FROM Enseignant
WHERE Département_ID = ' MATH '
```

➤ Tri du résultat

ORDER BY : permet de trier le résultat

- Dernière clause d'une requête SQL.
- Les colonnes sur lesquelles le résultat est trié sont désignées par leur nom ou leur n° d'apparition dans la clause SELECT
- **ASC** : tri croissant, **DESC** : tri décroissant

Exemple : afficher les pays dans l'ordre alphabétique

```
SELECT NomP
```

```
FROM Pays
```

```
ORDER BY NomP ASC ;
```

Exemple : afficher pour chaque pays le ratio superficie/nombre d'habitants.
Trier les

pays par ordre décroissant de ce ratio.

```
SELECT NomP AS Pays, Superficie/NbHab AS Occupation
```

```
FROM Pays
```

```
ORDER BY 2 DESC, 1 ASC ;
```

Remarque : Le tri sur la colonne 1 indique que les pays ayant le même ration sont affichés dans l'ordre alphabétique

Fonctions de calcul

SUM(Att) effectue la somme des valeurs de la colonne Att

MIN(Att) restitue la plus petite valeur de la colonne Att

MAX(Att) restitue la plus grande valeur de la colonne Att

AVG(Att) effectue la moyenne des valeurs de la colonne Att

COUNT(*) compte le nombre de lignes du résultat.

COUNT(DISTINCT (Att)) compte le nombre de valeurs différentes dans la colonne Att

Exemple1 : compter le nombre de produits dans la table PRODUIT

```
SELECT Count(*) AS NbProd  
FROM PRODUIT ;
```

Exemple2 : afficher le numéro de produit de quantité minimale et de quantité maximale

```
SELECT NP , MIN( QTE), MAX(QTE)  
FROM PRODUIT ;
```

GROUP BY : La clause GROUP BY permet de partitionner une table en plusieurs groupes.

- Toutes les lignes d'un même groupe ont la même valeur pour la liste des attributs de partitionnement spécifiés après GROUP BY.
- Les fonctions de calcul opèrent sur chaque groupe de valeurs.

Exemple: afficher la quantité achetée pour chaque numéro de produit

```
SELECT NP, SUM (QTEA)
```

```
FROM ACHAT
```

```
GROUP BY NP;
```

HAVING : La clause HAVING permet de spécifier une condition de restriction des groupes

- Elle sert à éliminer certains groupes, comme WHERE sert à éliminer des lignes
- Cette clause se place derrière la clause GROUP BY
- Lorsqu'un prédicats est évalué à FAUX, le sous-ensemble est éliminé

Exemple: afficher le numéro des produits achetés par plus de 100 clients.

```
SELECT NP  
FROM ACHAT  
GROUP BY NP HAVING COUNT(*)>100;
```