

IV. Algèbre relationnelle

- 1) Introduction
- 2) Union
- 3) Intersection
- 4) Différence
- 5) Produit cartésien
- 6) Projection
- 7) Sélection de lignes
- 8) Jointures
- 9) Division
- 10) Agrégation

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

1/63

IV.1) Introduction

- Les SGBD relationnels intègrent un noyau dans lequel est implanté un ensemble d'opérateurs algébriques qui manipulent les relations de la base de données.
- Le résultat de l'application d'un opérateur sur une ou deux relations restitue toujours une relation.
- Ces opérateurs sont donc internes à l'ensemble des relations et à ce titre constituent une algèbre.

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

4/63

IV.1) Introduction

- Il y a 9 opérateurs que l'on peut rencontrer dans un SGBD :

Classiques :

- Union
- Intersection
- Différence
- Produit cartésien

Spécifiques :

- Projection
- Sélection
- Jointure
- Division

Autre :

- Agrégation

- Chaque opérateur sera présenté en utilisant trois notations :

- symbolique
- graphique
- langage SQL

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

9/63

IV.2) Union

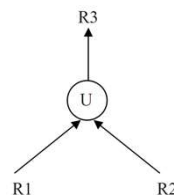
a) Notation symbolique : $R3 = R1 \cup R2$

Les relations $R1[A_1, \dots, A_n]$ et $R2[B_1, \dots, B_n]$ doivent être compatibles, c'est-à-dire comporter le même nombre d'attributs et avoir chaque paire d'attributs $\langle A_i, B_i \rangle$ de même domaine.

$R3 = R1 \cup R2$ est une relation :

- Formée avec les attributs de $R1 (A_1, \dots, A_n)$
- Contenant les n-uplets de $R1$ **ou** de $R2$

b) Notation graphique :



BUT 2022/2023

C4

R1.05

Modélisation de bases de données

12/63

IV.2) Union

c) Notation langage SQL :

```
01| ( select A_1 , A_2 , ... , A_n from R1)
02| UNION
03| ( select B_1 , B_2 , ... , B_n from R2)
```

Exemple :

R1	#EMP	NOM
	e3	xxx
	e1	zzz
	e7	www

R2	#EMP	NOMAD
	e1	zzz
	e2	fff

R3	#EMP	NOM
	e3	xxx
	e1	zzz
	e7	www
	e2	fff

BUT 2022/2023 C4 R1.05

Modélisation de bases de données

14/63

IV.3) Intersection

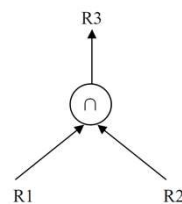
a) Notation symbolique : $R3 = R1 \cap R2$

Les relations $R1[A_1, \dots, A_n]$ et $R2[B_1, \dots, B_n]$ doivent être compatibles

$R3 = R1 \cap R2$ est une relation :

- Formée avec les attributs de $R1$ (A_1, \dots, A_n)
- Contenant les n-uplets **communs** à $R1$ et $R2$

b) Notation graphique :



BUT 2022/2023 C4 R1.05

Modélisation de bases de données

16/63

IV.3) Intersection

c) Notation langage SQL :

```

01 | ( select A_1 , A_2 , ... , A_n from R1 )
02 | INTERSECT
03 | ( select B_1 , B_2 , ... , B_n from R2 )

```

Exemple :

R1	#EMP	NOM
	e3	xxx
	e1	zzz
	e7	www

R2	#EMP	NOMAD
	e1	zzz
	e2	fff

R3	#EMP	NOM
	e1	zzz

BUT 2022/2023 C4 R1.05

Modélisation de bases de données

18/63

IV.4) Différence

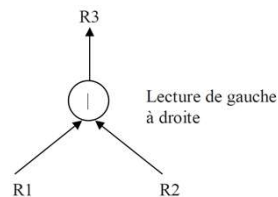
a) Notation symbolique : $R3 = R1 \setminus R2$

Les relations $R1[A_1, \dots, A_n]$ et $R2[B_1, \dots, B_n]$ doivent être compatibles

$R3 = R1 \setminus R2$ est une relation :

- Formée avec les attributs de $R1$ (A_1, \dots, A_n)
- Contenant les n-uplets de $R1$ **qui n'apparaissent pas dans $R2$**

b) Notation graphique :



BUT 2022/2023 C4 R1.05

Modélisation de bases de données

20/63

IV.4) Différence

c) Notation langage SQL :

```
01| ( select A_1 , A_2 , ... , A_n from R1)
02| MINUS
03| ( select B_1 , B_2 , ... , B_n from R2)
```

Exemple :

R1	#EMP	NOM
	e3	xxx
	e1	zzz
	e7	www

R2	#EMP	NOMAD
	e1	zzz
	e2	fff

R3=R1\R2	#EMP	NOM
	e3	xxx
	e7	www

R'3=R2\R1	#EMP	NOMAD
	e2	fff

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

22/63

IV.5) Produit cartésien

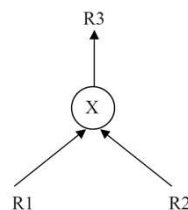
a) Notation symbolique : $R3 = R1 \times R2$

Soient les relations $R1[A_1, \dots, A_n]$ et $R2[B_1, \dots, B_k]$.

$R3 = R1 \times R2$ est une relation :

- Formée avec les attributs de R1 et de R2 ($A_1, \dots, A_n, B_1, \dots, B_k$)
- Contenant les $n+k$ uplets obtenus en associant chaque n -uplet de R1 avec chaque k -uplet de R2

b) Notation graphique :



BUT 2022/2023

C4

R1.05

Modélisation de bases de données

24/63

IV.5) Produit cartésien

c) Notation langage SQL :

01 | **select** A_1,A_2,...,A_n,B_1,B_2,...,B_k **from** R1, R2;

Exemple :

R1	A	B	C
n1 lignes	x	u	a
	y	v	d
	z	v	b

R2	D	E
n2 lignes	a	u
	x	u

R3	A	B	C	D	E
n1*n2 lignes	x	u	a	a	u
	y	v	d	a	u
	z	v	b	a	u
	x	u	a	x	u
	y	v	d	x	u
	z	v	b	x	u

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

26/63

IV.6) Projection

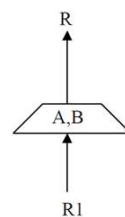
a) Notation symbolique : $R = \Pi_X R1$

X est une sous-liste de la liste des attributs de R1.

$R = \Pi_X R1$ est la relation :

- Formée avec la liste d'attributs de X
- Contenant les n-uplets de R1 restreints aux valeurs des attributs de la liste X
- Les doublons n'apparaissent pas dans le résultat

b) Notation graphique :



BUT 2022/2023

C4

R1.05

Modélisation de bases de données

28/63

IV.6) Projection

c) Notation langage SQL :

Si $X = A, B$

01 | **select distinct** A, B **from** R1 ;

Exemple :

R1	A	B	C	D
	a	x	u	a
	b	y	v	c
	a	z	v	d
	a	x	v	e
	b	z	v	d
	c	z	v	e
	c	y	u	u

Remarque :

Cet opérateur réduit le nombre de colonnes de la relation R1

$R = \Pi_{AC} R1$	A	C
	a	u
	b	v
	a	v
	c	v
	c	u

$R' = \Pi_B R1$	B
	x
	y
	z

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

31/63

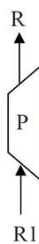
IV.7) Sélection de lignes

a) Notation symbolique : $R = \sigma_P R1$

P est un prédicat, c'est-à-dire une condition logique qui s'applique à certains attributs. R est la relation :

- Formée avec les attributs de R1
- Contenant les lignes de R1 vérifiant le prédicat P

b) Notation graphique :



BUT 2022/2023

C4

R1.05

Modélisation de bases de données

33/63

IV.7) Sélection de lignes

c) Notation langage SQL :

01| **select** A_1 , A_2 , ... , A_n **from** R1 ;
 02| **where** P;

Exemple :

R1	A	B	C	D
	a	G	x	31
	a	D	y	31
	b	D	x	42
	a	G	a	34
	b	G	x	31
	c	O	y	34

$R = \sigma_{(D=31) \text{ and } (B='D')} R1$	A	B	C	D
	a	D	y	31

SELECT A,B,C,D FROM R1
WHERE D=31 AND B='D';

$R = \sigma_{(D=31)} R1$	A	B	C	D
	a	G	x	31
	a	D	y	31
	b	G	x	31

SELECT A,B,C,D FROM R1
WHERE D=31;

IV.7) Sélection de lignes

d) Règles syntaxiques pour l'évaluation d'un prédicat :

Prédicat simple : on utilise les opérateurs de comparaison suivants :

= (égal), != (différent), >= (supérieur ou égal), <= (inférieur ou égal),
 > (strictement supérieur), < (strictement inférieur)

Syntaxe : Attribut **opérateur** Attribut, Valeur **opérateur** Attribut, ou
 Attribut **opérateur** Valeur

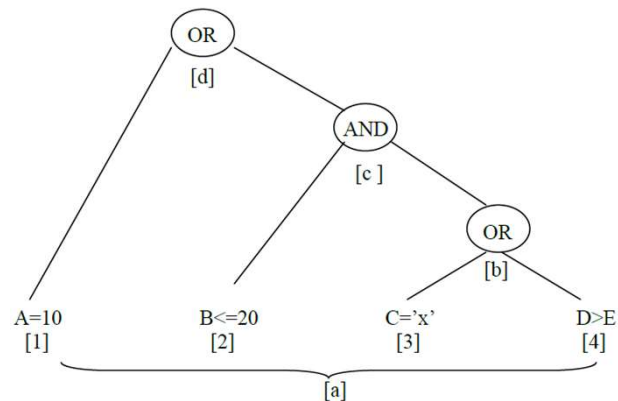
Prédicat composé : on utilise des prédicats simples interconnectés avec
 les opérateurs logiques **not**, **and**, **or** et de parenthèses éventuelles.

Priorité d'évaluation : D'abord entre les parenthèses les plus
 intérieures, puis du plus prioritaire au moins prioritaire **not** puis **and**
 puis **or**, et enfin de gauche à droite.

IV.7) Sélection de lignes

d) Règles syntaxiques pour l'évaluation d'un prédicat :

Exemple : $A = 10$ or $B \leq 20$ and ($C = 'x'$ or $D > E$)



BUT 2022/2023

C4

R1.05

Modélisation de bases de données

41/63

IV.8) Jointures

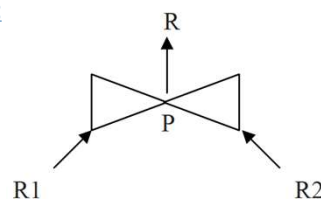
a) Notation symbolique : $R = R1 \bowtie_P R2$

Soient les relations $R1[A_1, \dots, A_n]$ et $R2[B_1, \dots, B_k]$ et le prédicat P .

R est la relation :

- Formée avec les attributs de $R1$ et $R2$ ($A_1, \dots, A_n, B_1, \dots, B_k$)
- Contenant les $n+k$ uplets formés des n -uplets de $R1$ et des k -uplets de $R2$ vérifiant le prédicat P .

b) Notation graphique :



BUT 2022/2023

C4

R1.05

Modélisation de bases de données

43/63

IV.8) Jointures

c) Notation langage SQL :

Deux notations sont possibles.

Syntaxe ANSI :

```
01| select A_1, ..., A_n, B_1, ..., B_k
02| from R1 inner join R2 on P ;
```

Syntaxe non-ANSI :

```
01| select A_1, ..., A_n, B_1, ..., B_k
02| from R1, R2
03| where P ;
```

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

44/63

IV.8) Jointures

Exemple :

R1	A	B	C
	x	b	a
	y	v	b
	y	u	a
	x	v	c

R2	X	Y	C
	a	x	w
	c	y	z

R = R1 ⋈ _(R1.C=R2.X) R2	A	B	C	X	Y	C
	x	b	a	a	x	w
	y	u	a	a	x	w
	x	v	c	c	y	z

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

45/63

IV.8) Jointures

d) Types de jointures possibles :

Equijointure :

Si P est de la forme $\text{AttributR1} = \text{AttributR2}$

Jointure Naturelle :

Equijointure dans laquelle AttributR1 et AttributR2 ont le même nom.
Dans ce cas, un seul des deux attributs (celui de gauche) est conservé.

Theta-Jointure (θ -jointure, Θ -jointure):

Si P est de la forme AttributR1 **opérateur** AttributR2 , où opérateur est un opérateur de comparaison

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

48/63

IV.8) Jointures

d) Types de jointures possibles :

Semi-jointure (gauche ou droite) :

S'obtient en ne gardant que la partie de R correspondant soit à R1 (semi-jointure gauche), soit à R2 (semi-jointure droite)

Exemple :

$R' = R1 \triangleright_{R1.C=R2.X} R2$	A	B	C
	x	b	a
	y	u	a
	x	v	c

Semi-jointure gauche

$R' = R1 \triangleleft_{R1.C=R2.X} R2$	X	Y	C
	a	x	w
	c	y	z

Semi-jointure droite

$$R' = R1 \triangleleft_P R2 = \prod_X (\sigma_P(R1 \times R2))$$

X = liste des attributs de R1.

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

50/63

IV.8) Jointures

d) Types de jointures possibles :

Jointures externes :

Les jointures précédentes sont dites internes (`inner join`).

Dans une jointure externe, les tuples ne satisfaisant pas le prédicat sont également conservés.

Selon si on ajoute les tuples de gauche ou de droite, il s'agira alors d'une jointure externe gauche (`left outer join` \bowtie) ou droite (`right outer join` \bowtie).

Si on conserve les deux, ils s'agit d'une jointure externe entière (`full outer join` \bowtie).

BUT 2022/2023

C4

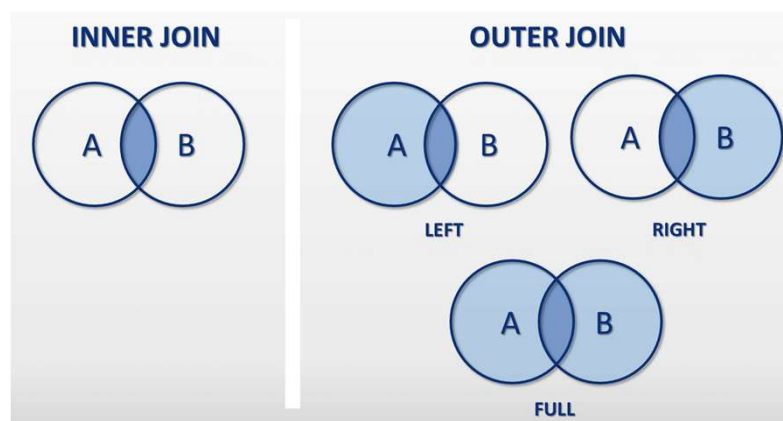
R1.05

Modélisation de bases de données

54/63

IV.8) Jointures

d) Types de jointures possibles :



BUT 2022/2023

C4

R1.05

Modélisation de bases de données

55/63

IV.9) Division

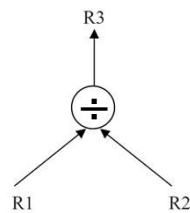
a) Notation symbolique : $R3 = R1 \div R2$

R1 et R2 doivent avoir au moins un attribut commun (même nom et même domaine).

R3 est la relation qui :

- Comporte les attributs appartenant à R1 mais n'appartenant pas à R2
- L'ensemble des tuples qui concaténés à ceux de R2 donnent toujours un tuple de R1.

b) Notation graphique :



Syntaxe SQL complexe.
Nécessite de combiner la
projection, le produit cartésien
et la différence

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

57/63

IV.9) Division

Exemple :

R1	A	B
	a	u
	a	x
	a	y
	b	y
	b	v
	c	u
	c	x
	c	t
	d	x
	d	y

R2	B
	u
	x

R	A
	a
	c

<a> → u,x,y: <a> apparaît dans R car a est associé à u et x.

 → y,v: n'apparaît pas dans R car b n'est pas associé à u et x.

<c> → u,x,t: <c> apparaît dans R car c est associé à u et x.

<d> → x,y: <d> n'apparaît pas dans R car d n'est pas associé à u.

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

58/63

IV.10) Agrégation

a) Notation symbolique : $R2 = AGG(R1; X; B)$

- R1 : relation sur laquelle s'applique la fonction d'agrégation AGG
- X : liste des attributs de R1 servant à définir un critère de regroupement.
On regroupe dans R1 les lignes possédant la même valeur par rapport à X. X est optionnel : s'il n'est pas spécifié la fonction d'agrégation s'appliquera à la totalité de la relation R (1 seul groupe)
- B : attribut agrégé
- AGG : fonction d'agrégation :
 - COUNT : Comptage,
 - SUM : Cumul,
 - MIN : Plus petite valeur,
 - MAX : Plus grande valeur,
 - AVG : Moyenne.

BUT 2022/2023

C4

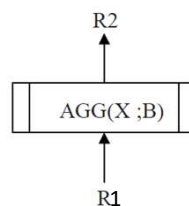
R1.05

Modélisation de bases de données

59/63

IV.10) Agrégation

b) Notation graphique :



c) Notation langage SQL :

```

01 | select X, agg (B) from R1
02 | group by X ;
  
```

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

60/63

IV.10) Agrégation

Exemple :

Soit la relation EMP décrivant des caractéristiques d'employés où E# identifie l'employé, NOM correspond à son nom, SAL à son salaire et SERV l'identifiant du service auquel il est affecté :

EMP	E#	NOM	SAL	SERV
	e3	Jean	8000	s1
	e2	Jacques	12000	s3
	e4	Pierre	10000	s1
	e6	Paul	12000	s2
	e1	Emile	9000	s3
	e7	Claude	6000	s1
	e5	Bernard	8000	s2

1) Pour compter le nombre d'employés par service :

$R = \text{COUNT}(\text{EMP}; \text{SERV}; \text{E\#})$

En SQL :

```
01 | select SERV, COUNT(E#) from EMP
02 | group by SERV ;
```

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

62/63

IV.10) Agrégation

Exemple :

Soit la relation EMP décrivant des caractéristiques d'employés où E# identifie l'employé, NOM correspond à son nom, SAL à son salaire et SERV l'identifiant du service auquel il est affecté :

EMP	E#	NOM	SAL	SERV
	e3	Jean	8000	s1
	e2	Jacques	12000	s3
	e4	Pierre	10000	s1
	e6	Paul	12000	s2
	e1	Emile	9000	s3
	e7	Claude	6000	s1
	e5	Bernard	8000	s2

2) Pour calculer le salaire moyen par service :

$R' = \text{AVG}(\text{EMP}; \text{SERV}; \text{SAL})$

En SQL :

```
01 | select SERV, AVG(SAL) from EMP
02 | group by SERV ;
```

BUT 2022/2023

C4

R1.05

Modélisation de bases de données

63/63