

Bases de données

Cours II

Lycée Thiers

Opérateurs binaires ensemblistes

Produit cartésien, jointure et différence

Opérateurs binaires ensemblistes

- On considère deux relations r_1 et r_2 ayant le même schéma $(A_1, \dots, A_p)$.
- La **Réunion** des relations r_1 et r_2 est la relation :

$$r_1 \cup r_2 = \{e; e \in r_1 \text{ ou } e \in r_2\}$$

- L'**Intersection** des relations r_1 et r_2 est la relation :

$$r_1 \cap r_2 = \{e; e \in r_1 \text{ et } e \in r_2\}$$

- La **Différence** des relations r_1 et r_2 est la relation :

$$r_1 \setminus r_2 = \{e; e \in r_1 \text{ et } e \notin r_2\}$$

- Si r_1 et r_2 sont les tables ayant pour attributs nom, prénom :

r_1	
Dupond	Gérard
Allais	Jules
Leroy	Amandine
Zanola	Julie

r_2	
Dupond	Gérard
Descours	Antoine
Zanola	Julie

- Alors les tables réunion $r_1 \cup r_2$, intersection $r_1 \cap r_2$ et différence $r_1 \setminus r_2$ sont :

$r_1 \cup r_2$	
Dupond	Gérard
Allais	Jules
Leroy	Amandine
Zanola	Julie
Descours	Antoine

$r_1 \cap r_2$	
Dupond	Gérard
Zanola	Julie

$r_1 \setminus r_2$	
Allais	Jules
Leroy	Amandine

- Soient deux tables `table1` et `table2` vérifiant le même schéma relationnel.
- Leur réunion s'obtient par l'instruction `UNION` :

```
SELECT * FROM table1 UNION SELECT * FROM table2
```

- On obtient une intersection par l'instruction `INTERSECT` :

```
SELECT * FROM table1 INTERSECT SELECT * FROM table2
```

- On obtient une différence par l'instruction `EXCEPT` :

```
SELECT * FROM table1 EXCEPT SELECT * FROM table2
```

Produit cartésien, jointure et différence

- On considère deux relations r_1 et r_2 (n'ayant pas forcément le même schéma) ayant pour ensembles d'attributs E_1 et E_2 .
- Le **Produit cartésien** des relations r_1 et r_2 est la relation ayant pour ensemble d'attributs $E_1 \cup E_2$:

$$r_1 \times r_2 = \{e; e[E_1] \in r_1 \text{ et } e[E_2] \in r_2\}$$

- La **Jointure** des relations r_1 et r_2 par une condition P est la relation, sous-ensemble de leur produit cartésien :

$$r_1 \bowtie_P r_2 = \{e; e[E_1] \in r_1 \text{ et } e[E_2] \in r_2 \text{ et } e \text{ vérifie } P\}$$

C'est le produit cartésien suivie de la sélection : $r_1 \bowtie_P r_2 = \sigma_P(r_1 \times r_2)$.

- Si les relations r_1 et r_2 sont telles que tous les attributs de r_2 sont des attributs de r_1 et non réciproquement : $E_2 \subsetneq E_1$, leur **Division** :

$$r_1 \div r_2 = \{e; \forall e_2 \in r_2, (e, e_2) \in r_1\}$$

(Remarque : lorsque $r_1 = r \times r_2$ alors $r_1 \div r_2 = r$. En général $(r_1 \div r_2) \times r_2 \subset r_1$).

- Pour effectuer un produit cartésien : `SELECT * FROM table1, table2`

```
SELECT * FROM eleves, notes
```

	nom	prenom	adresse	telephone	naissance	langue	nom	prenom	math	physique	français	anglais
1	Dupond	Tom	3 place de la gare	609080706	1995-03-15	esp	Dupond	Tom	10	12	14	15
2	Dupond	Tom	3 place de la gare	609080706	1995-03-15	esp	Dumas	Anne	10	12	14	15
3	Dupond	Tom	3 place de la gare	609080706	1995-03-15	esp	Laurin	Theo	11	8	13	12
4	Dupond	Tom	3 place de la gare	609080706	1995-03-15	esp	Charron	Ella	9	10	11	7
5	Dupond	Tom	3 place de la gare	609080706	1995-03-15	esp	Henri	Thomas	8	11	7	6
6	Dupond	Tom	3 place de la gare	609080706	1995-03-15	esp	Cuvelier	Paul	5	10	8	12
7	Charron	Ella	10 rue de la gare	607080908	1994-12-10	ita	Dupond	Tom	10	12	14	15
8	Charron	Ella	10 rue de la gare	607080908	1994-12-10	ita	Dumas	Anne	10	12	14	15
9	Charron	Ella	10 rue de la gare	607080908	1994-12-10	ita	Laurin	Theo	11	8	13	12
10	Charron	Ella	10 rue de la gare	607080908	1994-12-10	ita	Charron	Ella	9	10	11	7

36 enregistrements ramenés depuis `SELECT * FROM eleves, notes;` (en 4ms)

- Un produit cartésien est rarement utile ; une jointure l'est beaucoup plus. Même si une jointure est théoriquement la composée d'un produit cartésien suivi d'une sélection, l'implémentation d'une jointure est beaucoup plus rapide qu'un produit cartésien suivi d'une sélection.
- Il n'existe pas d'opérateurs en SQL pour la division relationnelle, mais on peut en réaliser une par une requête complexe.

- Pour effectuer une jointure symétrique simple :
(Condition : égalité de certains attributs) :

```
SELECT * FROM table1 JOIN table2  
ON table1.attribut = table2.attribut
```

Par exemple, seulement sur quelques attributs (projection) :

```
SELECT eleves.nom, eleves.prenom, naissance, langue,  
(math+physique+français+anglais)/4.0 FROM eleves JOIN notes  
ON eleves.nom = notes.nom
```

	nom	prenom	naissance	langue	(math+physique+français+anglais)/4.0
1	Dupond	Tom	1995-03-15	esp	12.75
2	Charron	Ella	1994-12-10	ita	9.25
3	Cuvelier	Paul	1995-09-12	all	8.75
4	Dumas	Anne	1994-10-01	rus	12.75
5	Henri	Thomas	1995-01-24	all	8.0
6	Laurin	Theo	1995-06-07	esp	11.0

Pour obtenir le classement : avec ORDER BY et AS (renommage) :

```
SELECT eleves.nom, eleves.prenom, naissance, langue,  
       (math+physique+français+anglais)/4.0 AS moyenne  
FROM eleves JOIN notes ON eleves.nom = notes.nom  
ORDER BY moyenne DESC
```

	nom	prenom	naissance	langue	moyenne
1	Dupond	Tom	1995-03-15	esp	12.75
2	Dumas	Anne	1994-10-01	rus	12.75
3	Laurin	Theo	1995-06-07	esp	11.0
4	Charron	Ella	1994-12-10	ita	9.25
5	Cuvelier	Paul	1995-09-12	all	8.75
6	Henri	Thomas	1995-01-24	all	8.0

- Récupérer les noms et n° de téléphone des élèves n'ayant pas la moyenne :

```
SELECT eleves.nom, telephone  
FROM eleves JOIN notes ON eleves.nom = notes.nom  
WHERE (math+physique+français+anglais)/4.0 < 10
```

	nom	telephone
1	Charron	607080908
2	Cuvelier	665544332
3	Henri	607677878

- Récupérer les noms et moyenne des élèves nés après 1995 :

```
SELECT eleves.nom, (math+physique+français+anglais)/4.0  
AS moyenne FROM eleves JOIN notes  
ON eleves.nom = notes.nom  
WHERE naissance >= '1995-01-01'
```

	nom	moyenne
1	Dupond	12.75
2	Cuvelier	8.75
3	Henri	8.0
4	Laurin	11.0