

# Base de données

## L'algèbre relationnelle

## historique

- Années 70 Edgar Franck Codd formalise les opérations réalisables sur les relations

## historique

- Années 70 Edgar Franck Codd formalise les opérations réalisables sur les relations
- Description requêtes

# historique

- Années 70 Edgar Franck Codd formalise les opérations réalisables sur les relations
- Description requêtes
- Garantie optimisation des requêtes

## historique

- Années 70 Edgar Franck Codd formalise les opérations réalisables sur les relations
- Description requêtes
- Garantie optimisation des requêtes
- Passons en revue les opérations de l'algèbre relationnelle et leurs traductions en SQL (version Mysql)

## historique

- Années 70 Edgar Franck Codd formalise les opérations réalisables sur les relations
- Description requêtes
- Garantie optimisation des requêtes
- Passons en revue les opérations de l'algèbre relationnelle et leurs traductions en SQL (version Mysql)
- Opérateurs unaires : fournissent une nouvelle table à partir d'une seule table

## historique

- Années 70 Edgar Franck Codd formalise les opérations réalisables sur les relations
- Description requêtes
- Garantie optimisation des requêtes
- Passons en revue les opérations de l'algèbre relationnelle et leurs traductions en SQL (version Mysql)
- Opérateurs unaires : fournissent une nouvelle table à partir d'une seule table
- Opérateurs binaires : fournissent une nouvelle table à partir de deux tables

## Opérateurs ensemblistes : produit cartésien

- Le **produit cartésien** est un opérateur binaire qui produit une relation  $T$



## Opérateurs ensemblistes : produit cartésien

- Le **produit cartésien** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,

## Opérateurs ensemblistes : produit cartésien

- Le **produit cartésien** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est la concaténation de ceux de  $R$  et  $S$ ,

## Opérateurs ensemblistes : produit cartésien

- Le **produit cartésien** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est la concaténation de ceux de  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont obtenus en concaténant tout tuple/enregistrement de la première relation  $R$  avec chacun des tuples/enregistrements  $S$  de la deuxième relation.

## Opérateurs ensemblistes : produit cartésien

- Le **produit cartésien** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est la concaténation de ceux de  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont obtenus en concaténant tout tuple/enregistrement de la première relation  $R$  avec chacun des tuples/enregistrements  $S$  de la deuxième relation.
- Notation

$$T = R \times S$$

# Opérateurs ensemblistes : produit cartésien

- Réalisation SQL

# Opérateurs ensemblistes : produit cartésien

- Réalisation SQL
- `SELECT * FROM R,S;`

# Opérateurs ensemblistes : produit cartésien

- Réalisation SQL
- `SELECT * FROM R,S;`
- Par exemple (produit cartésien de trois tables Entrée, Plat, Dessert), donne :

## Opérateurs ensemblistes : produit cartésien

- Réalisation SQL
- `SELECT * FROM R,S;`
- Par exemple (produit cartésien de trois tables Entrée, Plat, Dessert), donne :
- Tous les menus possibles

```
SELECT * FROM Entrée;
+-----+
| Nom           |
+-----+
| Terrine de lapin |
| Oeufs mayonnaise |
+-----+
2 rows in set (0.02 sec)
```



## Opérateurs ensemblistes : produit cartésien

- ```

SELECT * FROM Plat;
+-----+
| Nom          |
+-----+
| Blanquette de veau |
| Paella       |
+-----+
2 rows in set (0.01 sec)

SELECT * FROM Dessert;
+-----+
| Nom      |
+-----+
| Banane   |
| Yaourt   |
| Glace    |
+-----+
3 rows in set (0.00 sec)

```

## Opérateurs ensemblistes : produit cartésien

- SELECT \* FROM Entrée, Plat, Dessert;

| Nom              | Nom                | Nom    |
|------------------|--------------------|--------|
| Terrine de lapin | Blanquette de veau | Banane |
| Oeufs mayonnaise | Blanquette de veau | Banane |
| Terrine de lapin | Paella             | Banane |
| Oeufs mayonnaise | Paella             | Banane |
| Terrine de lapin | Blanquette de veau | Yaourt |
| Oeufs mayonnaise | Blanquette de veau | Yaourt |
| Terrine de lapin | Paella             | Yaourt |
| Oeufs mayonnaise | Paella             | Yaourt |
| Terrine de lapin | Blanquette de veau | Glace  |
| Oeufs mayonnaise | Blanquette de veau | Glace  |
| Terrine de lapin | Paella             | Glace  |
| Oeufs mayonnaise | Paella             | Glace  |

12 rows in set (0.00 sec)

# Opérateurs ensemblistes : relations union compatibles

- Deux relations sont **union-compatibles** si elles ont le même nombre d'attributs et que deux à deux, ceux-ci ont le même domaine.

# Opérateurs ensemblistes : relations union compatibles

- Deux relations sont **union-compatibles** si elles ont le même nombre d'attributs et que deux à deux, ceux-ci ont le même domaine.
- Les trois prochains opérateurs : union, intersection et différence, portent sur deux tables  $R$  et  $S$  union-compatibles.

# Opérateurs ensemblistes : union

- L'**union** est un opérateur binaire qui produit une relation  $T$

## Opérateurs ensemblistes : union

- L'**union** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,

## Opérateurs ensemblistes : union

- L'**union** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,

## Opérateurs ensemblistes : union

- L'**union** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  ou à  $S$ .



## Opérateurs ensemblistes : union

- L'**union** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  ou à  $S$ .
- Elimination des doublons.

## Opérateurs ensemblistes : union

- L'**union** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  ou à  $S$ .
- Elimination des doublons.
- Notation

$$T = R \cup S$$

## Opérateurs ensemblistes : union

- L'**union** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  ou à  $S$ .
- Elimination des doublons.
- Notation

$$T = R \cup S$$

- C'est l'opérateur ensembliste habituel.



# Opérateurs ensemblistes : union

- Réalisation SQL

```
SELECT ...
```

```
UNION
```

```
SELECT ...
```

- Exemple. Pour récupérer toutes les préparations :

```
SELECT * FROM Entrée  
UNION  
SELECT * FROM Plat  
UNION  
SELECT * FROM Dessert;
```

```

SELECT * FROM Entrée
UNION
SELECT * FROM Plat
UNION
SELECT * FROM Dessert;
+-----+
| Nom                |
+-----+
| Terrine de lapin   |
| Oeufs mayonnaise  |
| Blanquette de veau|
| Paella             |
| Banane             |
| Yaourt            |
| Glace             |
+-----+

```

## Opérateurs ensemblistes : intersection

- L'**intersection** est un opérateur binaire qui produit une relation  $T$

## Opérateurs ensemblistes : intersection

- L'**intersection** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,



## Opérateurs ensemblistes : intersection

- L'**intersection** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,

## Opérateurs ensemblistes : intersection

- L'**intersection** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  et à  $S$ .

## Opérateurs ensemblistes : intersection

- L'**intersection** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  et à  $S$ .
- Elimination des doublons.

## Opérateurs ensemblistes : intersection

- L'**intersection** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  et à  $S$ .
- Elimination des doublons.
- Notation

$$T = R \cap S$$

## Opérateurs ensemblistes : intersection

- L'**intersection** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  et à  $S$ .
- Elimination des doublons.
- Notation

$$T = R \cap S$$

- C'est l'opérateur ensembliste habituel.

## Opérateurs ensemblistes : intersection

- Réalisation SQL

```
SELECT ...
```

```
INTERSECT
```

```
SELECT ...
```

## Opérateurs ensemblistes : intersection

- Réalisation SQL

```
SELECT ...
```

```
INTERSECT
```

```
SELECT ...
```

- Exemple.

```
SELECT * FROM Entrée
INTERSECT
SELECT * FROM Plat;
ERROR 1064 (42000): You have an error in
your SQL syntax;
check the manual that corresponds to your
MySQL server version
for the right syntax to use near 'SELECT *
FROM Plat' at line 1
```

## Opérateurs ensemblistes : intersection

- Réalisation SQL

```
SELECT ...
```

```
INTERSECT
```

```
SELECT ...
```

- Exemple.

```
SELECT * FROM Entrée
INTERSECT
SELECT * FROM Plat;
ERROR 1064 (42000): You have an error in
your SQL syntax;
check the manual that corresponds to your
MySQL server version
for the right syntax to use near 'SELECT *
FROM Plat' at line 1
```

- Zut ! INTERSECT n'est pas supporté par Mysql ! Il l'est par d'autres SGBD !



## Opérateurs ensemblistes : intersection

- On peut s'en sortir en utilisant un opérateur booléen EXISTS et des requêtes imbriquées que nous étudierons à la toute fin du cours :

## Opérateurs ensemblistes : intersection

- On peut s'en sortir en utilisant un opérateur booléen EXISTS et des requêtes imbriquées que nous étudierons à la toute fin du cours :
- Dans le cas où les deux relations ont deux attributs en commun :

```
SELECT a,b FROM R
WHERE EXISTS (SELECT c,d FROM S
              WHERE a=c AND b=d);
```

ou bien sans requête imbriquée :

```
SELECT a, b FROM R, S
WHERE R.a = S.a AND R.b = S.b;
```

## Opérateurs ensemblistes : intersection

- On peut s'en sortir en utilisant un opérateur booléen EXISTS et des requêtes imbriquées que nous étudierons à la toute fin du cours :
- Dans le cas où les deux relations ont deux attributs en commun :

```
SELECT a,b FROM R
WHERE EXISTS (SELECT c,d FROM S
              WHERE a=c AND b=d);
```

ou bien sans requête imbriquée :

```
SELECT a, b FROM R, S
WHERE R.a = S.a AND R.b = S.b;
```

- Il faut faire autant de tests d'égalité qu'il y a d'attributs en commun.

## Opérateurs ensemblistes : différence

- La **différence** est un opérateur binaire qui produit une relation  $T$

## Opérateurs ensemblistes : différence

- La **différence** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,

## Opérateurs ensemblistes : différence

- La **différence** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,

## Opérateurs ensemblistes : différence

- La **différence** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  mais pas à  $S$ .

## Opérateurs ensemblistes : différence

- La **différence** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  mais pas à  $S$ .
- Notation

$$T = R - S$$



## Opérateurs ensemblistes : différence

- La **différence** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- dont le schéma est le schéma commun à  $R$  et  $S$ ,
- et dont les tuples/enregistrements sont ceux qui appartiennent à  $R$  mais pas à  $S$ .
- Notation

$$T = R - S$$

- C'est l'opérateur ensembliste habituel.

## Opérateurs ensemblistes : différence

- Réalisation SQL

```
SELECT ...
```

```
EXCEPT
```

```
SELECT ...
```

## Opérateurs ensemblistes : différence

- Réalisation SQL

```
SELECT ...
```

```
EXCEPT
```

```
SELECT ...
```

- Exemple. Pour récupérer toutes les préparations :

```
SELECT * FROM Entrée
EXCEPT
SELECT * FROM Plat;
ERROR 1064 (42000): You have an error in
    your SQL syntax;
check the manual that corresponds to your
    MySQL server version
for the right syntax to use near 'SELECT *
    FROM Plat' at line 1
```

## Opérateurs ensemblistes : différence

- Réalisation SQL

```
SELECT ...
```

```
EXCEPT
```

```
SELECT ...
```

- Exemple. Pour récupérer toutes les préparations :

```
SELECT * FROM Entrée
EXCEPT
SELECT * FROM Plat;
ERROR 1064 (42000): You have an error in
your SQL syntax;
check the manual that corresponds to your
MySQL server version
for the right syntax to use near 'SELECT *
FROM Plat' at line 1
```

- Re-Zut ! EXCEPT n'est pas supporté par Mysql ! Il l'est par d'autres SGBD (parfois sous le nom MINUS) !

## Opérateurs ensemblistes : différence

- On peut s'en sortir en utilisant les opérateurs booléens EXISTS et NOT.

## Opérateurs ensemblistes : différence

- On peut s'en sortir en utilisant les opérateurs booléens EXISTS et NOT.
- Dans le cas où les deux relations ont deux attributs en commun :

```
SELECT a,b FROM R
WHERE NOT EXISTS (SELECT c,d FROM S
                  WHERE a=c AND b=d);
```

## Opérateurs ensemblistes : différence

- On peut s'en sortir en utilisant les opérateurs booléens EXISTS et NOT.
- Dans le cas où les deux relations ont deux attributs en commun :

```
SELECT a,b FROM R
WHERE NOT EXISTS (SELECT c,d FROM S
                  WHERE a=c AND b=d);
```

- Il faut faire autant de tests d'égalité qu'il y a d'attributs en commun.

# Besoin

- Exemple de la bibliothèque : Emprunteur, Auteur, Document, Emprunt



# Besoin

- Exemple de la bibliothèque : Emprunteur, Auteur, Document, Emprunt
- Richesse du modèle relationnel : relation Emprunt

# Besoin

- Exemple de la bibliothèque : Emprunteur, Auteur, Document, Emprunt
- Richesse du modèle relationnel : relation Emprunt
- pour relier un Emprunteur et un Document

# Besoin

- Exemple de la bibliothèque : Emprunteur, Auteur, Document, Emprunt
- Richesse du modèle relationnel : relation Emprunt
- pour relier un Emprunteur et un Document
- Mais l'information complète est répartie sur plusieurs tables, ici 4

# Besoin

- Exemple de la bibliothèque : Emprunteur, Auteur, Document, Emprunt
- Richesse du modèle relationnel : relation Emprunt
- pour relier un Emprunteur et un Document
- Mais l'information complète est répartie sur plusieurs tables, ici 4
- D'où besoin de regrouper des informations différentes issues de différentes relations

## Solution : jointure

- La **jointure** est un opérateur binaire qui produit une relation  $T$

## Solution : jointure

- La **jointure** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,

## Solution : jointure

- La **jointure** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- ayant pour schéma la concaténation des schémas de deux relations  $R$  et  $S$

## Solution : jointure

- La **jointure** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- ayant pour schéma la concaténation des schémas de deux relations  $R$  et  $S$
- et dont les tuples/enregistrements sont obtenus en concaténant les tuples/enregistrements des deux relations



## Solution : jointure

- La **jointure** est un opérateur binaire qui produit une relation  $T$
- à partir de deux relations  $R$  et  $S$ ,
- ayant pour schéma la concaténation des schémas de deux relations  $R$  et  $S$
- et dont les tuples/enregistrements sont obtenus en concaténant les tuples/enregistrements des deux relations
- qui vérifient un critère  $P$ .

# Solution : jointure

- Réalisation SQL

## Solution : jointure

- Réalisation SQL
- `SELECT * FROM R JOIN S ON P;`

## Solution : jointure

- Par exemple pour récupérer (entre autres) toutes les dates de retour des documents empruntés par tous les emprunteurs ainsi que leurs noms et prénoms

| id | nom      | prénom      |
|----|----------|-------------|
| 1  | Martin   | Jacques     |
| 2  | Lagaffe  | Gaston      |
| 3  | Liègeois | Jean-Pierre |
| 4  | Colluci  | Michel      |
| 5  | Dupont   | Jean        |

| idDocument | Titre                 | Auteur             | Genre  |
|------------|-----------------------|--------------------|--------|
| 1          | Ruy Blas              | Victor Hugo        | Roman  |
| 2          | Les fleurs du mal     | Charles Baudelaire | Poésie |
| 3          | Don Quichotte         | Cervantès          | Roman  |
| 4          | Les lauriers de César | Gosciny            | BD     |

| idEmprunteur | idDocument | DateEmprunt | DateRetour |
|--------------|------------|-------------|------------|
| 1            | 3          | 2013-05-02  | 2013-05-22 |
| 1            | 4          | 2013-05-04  | 2013-05-25 |
| 2            | 1          | 2013-06-08  | 2013-06-29 |

## Solution : jointure

- Par exemple pour récupérer (entre autres) toutes les dates de retour des documents empruntés par tous les emprunteurs ainsi que leurs noms et prénoms

## Solution : jointure

- Par exemple pour récupérer (entre autres) toutes les dates de retour des documents empruntés par tous les emprunteurs ainsi que leurs noms et prénoms
- `SELECT * FROM Emprunteur JOIN Emprunt  
ON Emprunt.idEmprunteur=Emprunteur.id;`

## Solution : jointure

- Par exemple pour récupérer (entre autres) toutes les dates de retour des documents empruntés par tous les emprunteurs ainsi que leurs noms et prénoms
- `SELECT * FROM Emprunteur JOIN Emprunt  
ON Emprunt.idEmprunteur=Emprunteur.id;`
- Trop d'informations... On projette pour ne garder que nom, prénom, date de retour

```
SELECT nom, prénom, DateRetour
FROM Emprunteur JOIN Emprunt
ON Emprunt.idEmprunteur=Emprunteur.id ;
```

| nom     | prénom  | DateRetour |
|---------|---------|------------|
| Martin  | Jacques | 2013-05-22 |
| Martin  | Jacques | 2013-05-25 |
| Lagaffe | Gaston  | 2013-06-29 |

3 rows in set (0.00 sec)



## Solution : jointure

- Si on veut récupérer en plus les titres des livres empruntés, il faut faire une jointure supplémentaire



## Solution : jointure

- Si on veut récupérer en plus les titres des livres empruntés, il faut faire une jointure supplémentaire
- sur la table Document, en utilisant la correspondance par idDocument.

## Solution : jointure

- Si on veut récupérer en plus les titres des livres empruntés, il faut faire une jointure supplémentaire
- sur la table Document, en utilisant la correspondance par idDocument.

```
SELECT nom, prénom, Titre, DateEmprunt from Emprunteur
      JOIN Emprunt ON Emprunt.idEmprunteur=Emprunteur.id
      JOIN Document ON Emprunt.idDocument=Document.idDocument;
```

| nom     | prénom  | Titre                 | DateRetour |
|---------|---------|-----------------------|------------|
| Lagaffe | Gaston  | Ruy Blas              | 2013-06-08 |
| Martin  | Jacques | Don Quichotte         | 2013-05-02 |
| Martin  | Jacques | Les lauriers de César | 2013-05-04 |

3 rows in set (0.00 sec)

## Solution : jointure

- Retrouver le numéro de téléphone de l'emprunteur de Ruy Blas pour lui signaler qu'il a oublié de rendre le livre !

## Solution : jointure

- Retrouver le numéro de téléphone de l'emprunteur de Ruy Blas pour lui signaler qu'il a oublié de rendre le livre !

```
SELECT telephone from Emprunteur
      JOIN Emprunt ON Emprunt.idEmprunteur=
                Emprunteur.id
      JOIN Document ON Emprunt.idDocument=
                Document.idDocument
      WHERE Titre='Ruy Blas';
```

| telephone |
|-----------|
| 987654321 |

```
1 row in set (0.00 sec)
```

# Statut de ce qui est renvoyé par une requête SELECT

- Une requête SELECT renvoie

# Statut de ce qui est renvoyé par une requête SELECT

- Une requête SELECT renvoie
- une table (relation), avec des attributs et des enregistrements

# Statut de ce qui est renvoyé par une requête SELECT

- Une requête SELECT renvoie
- une table (relation), avec des attributs et des enregistrements
- MAIS

## Statut de ce qui est renvoyé par une requête SELECT

- Une requête SELECT renvoie
- une table (relation), avec des attributs et des enregistrements
- MAIS
- cette table **n'est pas stockée** dans la base de données.



# Statut de ce qui est renvoyé par une requête SELECT

- Une requête SELECT renvoie
- une table (relation), avec des attributs et des enregistrements
- MAIS
- cette table **n'est pas stockée** dans la base de données.
- Elle est **perdue** après l'exécution de la requête.

## Statut de ce qui est renvoyé par une requête SELECT

- Une requête SELECT renvoie
- une table (relation), avec des attributs et des enregistrements
- MAIS
- cette table **n'est pas stockée** dans la base de données.
- Elle est **perdue** après l'exécution de la requête.
- Il y a des mécanismes de requêtes imbriquées permettant de réutiliser le résultat d'une requête.

# Intérêt

- Écriture de toutes les requêtes à l'aide des opérateurs présentés (ou presque...)

# Intérêt

- Écriture de toutes les requêtes à l'aide des opérateurs présentés (ou presque...)
- L'algèbre relationnelle justifie la commutation ou l'associativité de certains opérateurs

# Intérêt

- Écriture de toutes les requêtes à l'aide des opérateurs présentés (ou presque...)
- L'algèbre relationnelle justifie la commutation ou l'associativité de certains opérateurs
- ce qui permet de réécrire les requêtes formulées

# Intérêt

- Écriture de toutes les requêtes à l'aide des opérateurs présentés (ou presque...)
- L'algèbre relationnelle justifie la commutation ou l'associativité de certains opérateurs
- ce qui permet de réécrire les requêtes formulées
- dans un but d'optimisation...

## Intérêt

- Écriture de toutes les requêtes à l'aide des opérateurs présentés (ou presque...)
- L'algèbre relationnelle justifie la commutation ou l'associativité de certains opérateurs
- ce qui permet de réécrire les requêtes formulées
- dans un but d'optimisation...
- avec la garantie que le résultat sera le bon !