

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentiels

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

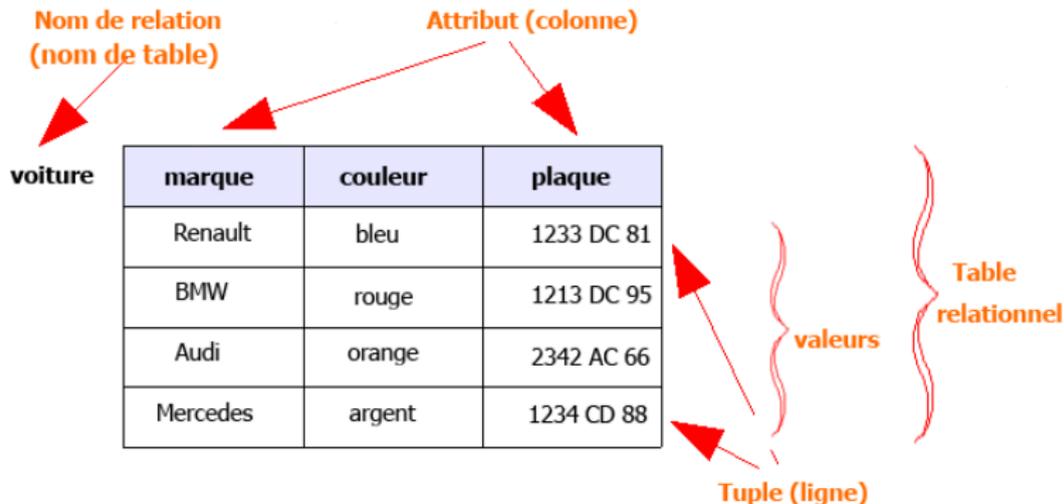
le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- 1 Les concepts généraux
- 2 Notions essentielles pour le modèle relationnel
- 3 Algèbre Relationnelle et SQL
  - Les opérateurs de base unaires (sur une seule opérande)
  - Traduction en SQL d'une requête simple
  - Les opérateurs de jointure
  - Opérateurs ensemblistes :  $\cup$  ,  $\cap$  ,  $\setminus$
  - Opérateurs de SQL qui ne sont pas dans l'algèbre
  - Autres ordres SQL
- 4 Le modèle conceptuel Entités-Associations
- 5 Normalisation d'une relation

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
Jointures  
Opérateurs  
ensemblistes  
Opérateurs hors de  
l'Alg.  
Autres ordres SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

- Le modèle relationnel repose sur la notion de relation assimilable à un tableau.



- les langages classiques (python) manipulent des tableaux. Ils pourraient servir à rechercher les informations dans ce genre de structures de données.
- ⇒ Pourquoi alors avoir introduit des langages de manipulation de données relationnelles ?

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

⇒ Pourquoi alors avoir introduit des langages de manipulation de données relationnelles ?

- 1 Pour qu'un programmeur puisse effectuer une recherche à l'aide d'un langage classique impératif, il est nécessaire qu'il connaisse la structure de la relation.

- La couleur est-elle en 6ème colonne ou en 7ème ?

contraire au principe de *séparation des programmes et des données*

- 2 Les utilisateurs doivent connaître l'organisation des données (tri, existence d'index, ...) pour que leurs requêtes soient efficaces.

- comment aller rapidement au véhicule d'immatriculation FD-446-QM ?

si la structure change, les programmes doivent aussi changer...

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
Jointures  
Opérateurs  
ensemblistes  
Opérateurs hors de  
l'Alg.  
Autres ordres SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

- Les **Langages de Manipulation de données** doivent donc être le plus proche du modèle conceptuel :
  - ne porte que sur les concepts du schéma : relation, attributs, dom.
  - doivent ignorer complètement l'organisation interne des relations.
- Le modèle relationnel a été proposé avec deux LMD de base :
  - l'**algèbre relationnel** et
  - le **calcul relationnel** équivalents en puissanceSQL implémente les relations non pas comme des ensembles mais comme des **multi-ensembles** (il peut y avoir redondance). Ce choix a été fait pour des questions d'efficacité.
- **L'algèbre Relationnelle** repose sur un ens d'opérateurs  $R_1 \square R_2 \rightarrow R_{temp}$
- 5 opérateurs de base : *sélection, projection, union, différence, produit*  
1 opérateur syntaxique : renommer (modifie le schéma/pas les  $n$ -uplets).
- A partir de ces opérateurs, d'autres opérateurs ont été introduits :
  - ils ne sont que la composée de plusieurs opérateurs de base (raccourcis).
  - les plus fréquents : *intersection, jointure naturelle, théta jointure, division*.
- Autre classification des opérateurs :
  - les opérateurs basés sur la théorie des ensembles :  $\cup, \cap, \setminus, \times$
  - les opérateurs construits spécifiquement pour les BD relationnelles :  
sélection, projection, jointure, division et renommage

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

Cet opérateur construit une relation dans laquelle n'apparaissent que certains attributs de la relation opérande.

Intuition : on extrait uniquement certaines colonnes du tableau

Soit  $R(A_1, A_2, \dots, A_n)$  une relation et soient  $\{A_{i_1}, A_{i_2}, \dots, A_{i_j}\}$  un sous ensemble de ses attributs. La projection de  $R$  sur  $\{A_{i_1}, A_{i_2}, \dots, A_{i_j}\}$ , notée  $\pi[A_{i_1}, A_{i_2}, \dots, A_{i_j}]R$  ( $\pi$  comme projection) est une nouvelle relation, temporaire, de schéma  $(A_{i_1}, A_{i_2}, \dots, A_{i_j})$  et de population égale à

$$\{ t \mid \exists r : (r \in R) \wedge (t.A_{i_1} = r.A_{i_1}) \wedge \dots \wedge (t.A_{i_j} = r.A_{i_j}) \}$$

**Remarque.** Il s'agit d'une opération *ensembliste*. Donc si l'opération crée des doublons, ils sont automatiquement éliminés.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentielsAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation**Exemple.** Soit la relation *Personne* suivante :

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

 $\pi[\text{nom}, \text{prénom}] \textit{Personne}$  $\pi[\text{nom}, \text{année}] \textit{Personne}$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentielsAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation**Exemple.** Soit la relation *Personne* suivante :

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

 $\pi[\text{nom}, \text{prénom}] \textit{Personne}$  $\pi[\text{nom}, \text{année}] \textit{Personne}$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQLJointures  
Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.  
Autres ordres SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation**Exemple.** Soit la relation *Personne* suivante :

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

 $\pi[\text{nom}, \text{prénom}] \textit{Personne}$ 

| nom    | prénom    |
|--------|-----------|
| Jean   | Thierry   |
| Talon  | Achille   |
| Rochat | Dominique |
| Martin | Joëlle    |
| Martin | Jules     |

 $\pi[\text{nom}, \text{année}] \textit{Personne}$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation**Exemple.** Soit la relation *Personne* suivante :

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

 $\pi[\text{nom}, \text{prénom}] \textit{Personne}$ 

| nom    | prénom    |
|--------|-----------|
| Jean   | Thierry   |
| Talon  | Achille   |
| Rochat | Dominique |
| Martin | Joëlle    |
| Martin | Jules     |

 $\pi[\text{nom}, \text{année}] \textit{Personne}$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation**Exemple.** Soit la relation *Personne* suivante :

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

 $\pi[\text{nom}, \text{prénom}] \textit{Personne}$ 

| nom    | prénom    |
|--------|-----------|
| Jean   | Thierry   |
| Talon  | Achille   |
| Rochat | Dominique |
| Martin | Joëlle    |
| Martin | Jules     |

 $\pi[\text{nom}, \text{année}] \textit{Personne}$ 

| nom    | année |
|--------|-------|
| Jean   | 1960  |
| Talon  | 1978  |
| Rochat | 1956  |
| Martin | 1968  |
| Martin | 1984  |
| Rochat | 1976  |

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures  
Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Cet opérateur construit une relation dans laquelle seuls certains  $n$ -uplets apparaissent.

Intuition : On sélectionne des lignes ( $n$ -uplets) qui satisfont une certaine propriété, appelé **prédicat de sélection**

Soit  $R(A_1, A_2, \dots, A_n)$  une relation la sélection selon un prédicat  $p$ , notée  $\sigma[p]R$  ( $\sigma$  comme sélection) est une nouvelle relation, temporaire, de schéma identique à celui de  $R$  et de population égale à l'ensemble des  $n$ -uplets de  $R$  pour lesquels le prédicat  $p$  est vrai.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

**Exemple.** A partir de la relation *Personne* on peut créer la relation

**Femme** :=  $\sigma[\text{genre} = \text{'F'}] \text{Personne}$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

**Exemple.** A partir de la relation *Personne* on peut créer la relation

*Femme* :=  $\sigma[\text{genre} = \text{'F'}] \text{Personne}$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
JointuresOpérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

**Exemple.** A partir de la relation *Personne* on peut créer la relation

*Femme* :=  $\sigma[\text{genre} = \text{'F'}] \textit{Personne}$

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

## Syntaxe du prédicat.

|  |     |  |
|--|-----|--|
| $\langle p \rangle$                        | ::= | $\langle \text{condition} \rangle$<br>$\langle p \rangle \langle \text{opérateur logique} \rangle \langle p \rangle$<br>$\neg \langle p \rangle$ <span style="color: green;">négation</span><br>$\text{"("} \langle p \rangle \text{"}"$<br>$\langle p \rangle \vee \langle p \rangle$ <span style="color: green;">disjonction</span><br>$\langle p \rangle \wedge \langle p \rangle$ <span style="color: green;">conjonction</span> |
| $\langle \text{opérateur logique} \rangle$ | ::= | $\langle p \rangle \vee \langle p \rangle$ <span style="color: green;">disjonction</span><br>$\langle p \rangle \wedge \langle p \rangle$ <span style="color: green;">conjonction</span>   |
| $\langle \text{condition} \rangle$         | ::= | Attr $\langle \text{opérateur comp} \rangle$ valeur<br>Attr $\langle \text{opérateur comp} \rangle$ Attr   |
| $\langle \text{opérateur comp} \rangle$    | ::= | $=$   $\neq$   $\leq$   $<$   $>$   $\geq$   |

## Remarques.

- Les opérateurs de comparaisons  $\leq, <, >, \geq$  ne peuvent s'appliquer que sur des attributs dont le domaine est muni d'un ordre :
  - valeurs numériques,
  - dates,
  - chaînes de caractères (ordre alphanumérique  $\sim$  alphabétique généralisé)

Pour les domaines sans ordre, on ne peut utiliser que les opérateurs  $=$  et  $\neq$ .

- Lorsque l'on compare deux attributs, il faut s'assurer que les deux attributs aient même domaine.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

| nom    | prénom    |
|--------|-----------|
| Jean   | Thierry   |
| Rochat | Dominique |

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQLJointures  
Opérateurs  
ensemblistes  
Opérateurs hors de  
l'Alg.  
Autres ordres SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

| nom    | prénom    | jour | mois | année | genre |
|--------|-----------|------|------|-------|-------|
| Jean   | Thierry   | 29   | 02   | 1960  | M     |
| Talon  | Achille   | 20   | 11   | 1978  | M     |
| Rochat | Dominique | 07   | 12   | 1956  | M     |
| Martin | Joëlle    | 01   | 05   | 1968  | F     |
| Martin | Jules     | 03   | 03   | 1984  | M     |
| Rochat | Dominique | 17   | 02   | 1976  | F     |

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

$$H := \pi \left[ \begin{array}{c} \text{nom} \\ \text{prénom} \end{array} \right] \left( \sigma \left[ \begin{array}{c} \text{genre} = 'M' \wedge \\ \text{année} < 1975 \end{array} \right] \text{Personne} \right)$$

| nom    | prénom    |
|--------|-----------|
| Jean   | Thierry   |
| Rochat | Dominique |

L'opérateur de renommage  $\alpha$  permet de changer le nom d'un ou plusieurs attribut(s) d'une relation  $R$ .

$\alpha[\text{Attr1} : \text{newNameForAttr1}, \dots]R$

**Remarque.** Cet opérateur est utile lorsqu'il y a un problème d'homonymie ou de synonymie, ou alors avant des opérations ensemblistes (union, différence, intersection) qui nécessite que les attributs correspondants aient le même nom.

**Exemple.**

$$\alpha \left[ \begin{array}{l} \text{nom} : n \\ \text{prénom} : p \end{array} \right] \pi \left[ \begin{array}{l} \text{nom} \\ \text{prénom} \end{array} \right] \left( \sigma \left[ \begin{array}{l} \text{genre} = 'M' \wedge \\ \text{année} < 1975 \end{array} \right] \text{Personne} \right)$$

| nom    | prénom    |
|--------|-----------|
| Jean   | Thierry   |
| Rochat | Dominique |

$\Rightarrow$

| n      | p         |
|--------|-----------|
| Jean   | Thierry   |
| Rochat | Dominique |

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

## ordre **SELECT**

**SELECT**  $A_1, \dots, A_N$  FROM  $R$  WHERE  $C$ ;

- $A_1, \dots, A_N$  : attributs
- $R$  : nom d'une relation
- $C$  : condition de sélection sur les  $n$ -uplets de  $R$ , par exemple comme la sélection de l'algèbre relationnelle.

Le **SELECT** fait une **projection** (opération algébrique) du résultat de la **sélection** sur les attributs  $A_1, \dots, A_N$ .

**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- 1 Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?
- 2 On peut remplacer la suite exhaustive des attributs d'une table par le symbole \* :

**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- 1 Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?

```
SELECT Titre, long FROM Film
```

```
WHERE nomStudio='Disney' and Date=1999;
```

- 2 On peut remplacer la suite exhaustive des attributs d'une table par le symbole \* :

**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?

```
SELECT Titre, long FROM Film
```

```
WHERE nomStudio='Disney' and Date=1999;
```

$$\pi \left[ \begin{array}{l} \text{Titre} \\ \text{Long} \end{array} \right] \sigma \left[ \begin{array}{l} \text{nomStudio} = ' \text{Disney}' \wedge \\ \text{Date} = 1999; \end{array} \right] \text{Film}$$

- ❷ On peut remplacer la suite exhaustive des attributs d'une table par le symbole \* :

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentiels

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?

```
SELECT Titre, long FROM Film
```

```
WHERE nomStudio='Disney' and Date=1999;
```

$$\pi \left[ \begin{array}{c} \text{Titre} \\ \text{Long} \end{array} \right] \sigma \left[ \begin{array}{c} \text{nomStudio} = ' \text{Disney}' \wedge \\ \text{Date} = 1999; \end{array} \right] \text{Film}$$

- ❷ On peut remplacer la suite exhaustive des attributs d'une table par le symbole \* :

```
SELECT * FROM Film
```

```
WHERE nomStudio='Disney' and Date=1999;
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures  
Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?

```
SELECT Titre, long FROM Film
```

```
WHERE nomStudio='Disney' and Date=1999;
```

$$\pi \left[ \begin{array}{c} \text{Titre} \\ \text{Long} \end{array} \right] \sigma \left[ \begin{array}{c} \text{nomStudio} = ' \text{Disney}' \wedge \\ \text{Date} = 1999; \end{array} \right] \text{Film}$$

- ❷ On peut remplacer la suite exhaustive des attributs d'une table par le symbole \* :

```
SELECT * FROM Film
```

```
WHERE nomStudio='Disney' and Date=1999;
```

$$\sigma \left[ \begin{array}{c} \text{nomStudio} = ' \text{Disney}' \wedge \\ \text{Date} = 1999; \end{array} \right] \text{Film}$$

**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

③ Utilisation des ALIAS :

④ on peut un peu compliquer la sélection :

**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \textit{Film}(\textit{Titre}, \textit{date}, \textit{long}, \textit{couleur}, \textit{nomStudio}, \textit{IdProd}), \\ \textit{Joue}(\textit{TitreFilm}, \textit{FilmDate}, \textit{nomActeur}, \textit{Paye}), \\ \textit{Acteur}(\textit{nom}, \textit{adresse}, \textit{genre}, \textit{dateNaissance}), \\ \textit{Studio}(\textit{nom}, \textit{Adresse}), \\ \textit{Producteur}(\textit{nom}, \textit{adresse}, \textit{Id}) \end{array} \right\}$$

③ Utilisation des ALIAS :

```
SELECT Titre AS nom, long AS Durée FROM Film  
WHERE nomStudio='Disney' and Date=1999;
```

④ on peut un peu compliquer la sélection :

**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

③ Utilisation des ALIAS :

**SELECT** Titre AS nom, long AS Durée FROM Film  
WHERE nomStudio='Disney' and Date=1999;

$$\alpha \left[ \begin{array}{l} \text{Titre : nom} \\ \text{long : Duree} \end{array} \right] \pi \left[ \begin{array}{l} \text{Titre} \\ \text{long} \end{array} \right] \sigma \left[ \begin{array}{l} \text{nomStudio = ' Disney' } \\ \wedge \text{Date = 1999;} \end{array} \right] \text{Film}$$

Attention, les colonnes ont changé de noms.

④ on peut un peu compliquer la sélection :

**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

③ Utilisation des ALIAS :

```
SELECT Titre AS nom, long AS Durée FROM Film
WHERE nomStudio='Disney' and Date=1999;
```

$$\alpha \left[ \begin{array}{l} \text{Titre} : \text{nom} \\ \text{long} : \text{Duree} \end{array} \right] \pi \left[ \begin{array}{l} \text{Titre} \\ \text{long} \end{array} \right] \sigma \left[ \begin{array}{l} \text{nomStudio} = ' \text{Disney}' \\ \wedge \text{Date} = 1999; \end{array} \right] \text{Film}$$

Attention, les colonnes ont changé de noms.

④ on peut un peu compliquer la sélection :

```
SELECT Titre FROM Film
WHERE nomStudio='Disney' and
(Date>1999 OR long <90) AND NOT couleur;
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- 1 la valeur `NULL` code l'absence de valeur
- 2 `NULL op v` est évalué à `NULL` si `op` est un opérateur arithmétique et `v` une valeur.
- 3 `NULL = v` est évalué à `UNKNOWN` (idem pour `NULL > v`).  
En particulier, cette expression n'est pas évaluée à vraie.
- 4 Faire donc attention lorsqu'on manipule l'absence de valeur `NULL`.

`SELECT * FROM Film WHERE Date >= 1970 OR Date < 1970;`  
renvoie la table Film **privée des  $n$ -uplets pour lesquels la valeur Date est null.**

Recherche des  $n$ -uplets pour lesquels une colonne n'a pas de valeur :

- `SELECT * FROM my_table WHERE phone = NULL;`
- `SELECT * FROM my_table WHERE phone IS NULL;`
- `SELECT * FROM my_table WHERE phone = '';`

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

Soient  $\{B_1, \dots, B_k\} \subset \{A_1, \dots, A_n\}$ .

```
SELECT A1, ..., An FROM R WHERE C  
ORDER BY B1, ... Bk;
```

Supposons que  $k = 2$

Si  $n(B_1) < n'(B_1)$  alors  $n$  avant  $n'$ .

Si  $n(B_1) = n'(B_1)$  alors  $n$  avant  $n'$  ssi  $n(B_2) < n'(B_2)$

```
SELECT * FROM Film  
WHERE nomStudio='Disney' and Date='1999'  
ORDER BY Longueur, Titre;
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

**Exemple.** Pour illustrer les jointures, considérons la base de données relationnelle InstitutFormation dont le schéma est le suivant :

$$\mathcal{S} = \left\{ \begin{array}{l} \textit{personne}(\textit{numPers}, \textit{nom}, \textit{adresse}), \\ \textit{personnePrenom}(\textit{numPers}, \textit{numPrenom}, \textit{prenom}), \\ \textit{etudiant}(\textit{numPers}, \textit{numEtud}, \textit{dateN}), \\ \textit{etudiantEtude}(\textit{numEtud}, \textit{annee}, \textit{diplome}), \\ \textit{enseignant}(\textit{numPers}, \textit{tel}, \textit{status}, \textit{numEns}, \textit{banque}, \\ \qquad \qquad \qquad \textit{agence}, \textit{compte}), \\ \textit{cours}(\textit{nomC}, \textit{cycle}, \textit{numEns}), \\ \textit{obtenu}(\textit{numEtud}, \textit{nomC}, \textit{note}), \\ \textit{inscrit}(\textit{numEtud}, \textit{nomC}), \\ \textit{prerequis}(\textit{nomC}, \textit{nomCpre}) \end{array} \right\}$$

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $S(B_1, B_2, \dots, B_p)$  n'ayant aucun attribut en commun. Le produit cartésien, noté  $R \times S$  est la relation temporaire de schéma  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_p)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  et de  $S$  concaténés.

### Exemple.

- Existe-t-il des personnes dont le nom est identique au nom d'un cours ? Donner leurs noms.

On verra plus tard que l'on peut écrire cette requête avec une théta-jointure :

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $S(B_1, B_2, \dots, B_p)$  n'ayant aucun attribut en commun. Le produit cartésien, noté  $R \times S$  est la relation temporaire de schéma  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_p)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  et de  $S$  concaténés.

### Exemple.

- Existe-t-il des personnes dont le nom est identique au nom d'un cours ? Donner leurs noms.

$$\pi[\text{nom}]\sigma[\text{nom} = \text{nomC}](\text{personne} \times \text{Cours})$$

On verra plus tard que l'on peut écrire cette requête avec une théta-jointure :

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $S(B_1, B_2, \dots, B_p)$  n'ayant aucun attribut en commun. Le produit cartésien, noté  $R \times S$  est la relation temporaire de schéma  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_p)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  et de  $S$  concaténés.

### Exemple.

- Existe-t-il des personnes dont le nom est identique au nom d'un cours ? Donner leurs noms.

$$\pi[\text{nom}]\sigma[\text{nom} = \text{nomC}](\text{personne} \times \text{Cours})$$

On verra plus tard que l'on peut écrire cette requête avec une théta-jointure :

$$\pi[\text{nom}](\text{personne} \bowtie [\text{nom} = \text{nomC}]\text{Cours})$$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \\ \text{Producteur}(\text{nom}, \text{Adresse}, \text{ID}) \end{array} \right\}$$

## Différents types de jointures :

- ① **CROSS JOIN** ...  $\equiv$  , : produit cartésien

```
SELECT A1,...,An FROM R1, R2, ..., Rk WHERE C;
```

**Exemple** : Quel est le nom du producteur de "stars war" ?

```
SELECT nom FROM Film, Producteur
```

```
WHERE Titre='star war' AND IdProd=ID
```

Que se passe-t-il si les 2 schémas ont des attributs en commun ?

**Exemple** : les schémas Acteur et Producteur partage

l'attribut Nom. Quels acteurs et quels producteurs ont la même adresse ?

```
SELECT Acteur.nom, Producteur.nom
```

```
FROM Acteur, Producteur
```

```
WHERE Acteur.adresse= Producteur.adresse;
```

Les colonnes s'appellent Acteur.nom et Producteur.nom.

(renommage).

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $S(B_1, B_2, \dots, B_p)$  n'ayant aucun attribut en commun. La théta jointure selon le prédicat  $p$ , noté  $R \bowtie[p] S$  est la relation temporaire de schéma  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_p)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  et de  $S$  concaténés qui satisfont le prédicat  $p$ .

$$R \bowtie[p] S \equiv \sigma[p](R \times S)$$

**Syntaxe du prédicat.** même syntaxe + :

$\langle \text{condition} \rangle ::= \text{nomAttrdeR} \langle \text{opérateur comp} \rangle \text{nomAttrdeS}$

**Exemple.**

- liste des couples de numéros d'étudiants nés le même jours

**Remarque.** Le produit est une théta jointure dont le prédicat serait toujours vrai.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

**Jointures**

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $S(B_1, B_2, \dots, B_p)$  n'ayant aucun attribut en commun. La théta jointure selon le prédicat  $p$ , noté  $R \bowtie[p] S$  est la relation temporaire de schéma  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_p)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  et de  $S$  concaténés qui satisfont le prédicat  $p$ .

$$R \bowtie[p] S \equiv \sigma[p](R \times S)$$

**Syntaxe du prédicat.** même syntaxe + :

$\langle \text{condition} \rangle ::= \text{nomAttrdeR} \langle \text{opérateur comp} \rangle \text{nomAttrdeS}$

**Exemple.**

- liste des couples de numéros d'étudiants nés le même jours

(
Etudiant
)
(
Etudiant
)

**Remarque.** Le produit est une théta jointure dont le prédicat serait toujours vrai.

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $S(B_1, B_2, \dots, B_p)$  n'ayant aucun attribut en commun. La théta jointure selon le prédicat  $p$ , noté  $R \bowtie[p] S$  est la relation temporaire de schéma  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_p)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  et de  $S$  concaténés qui satisfont le prédicat  $p$ .

$$R \bowtie[p] S \equiv \sigma[p](R \times S)$$

**Syntaxe du prédicat.** même syntaxe + :

$\langle \text{condition} \rangle ::= \text{nomAttrdeR} \langle \text{opérateur comp} \rangle \text{nomAttrdeS}$

**Exemple.**

- liste des couples de numéros d'étudiants nés le même jours

$$\left( \text{Etudiant} \quad \pi \begin{bmatrix} \text{numEtud} \\ \text{dateN} \end{bmatrix} \text{Etudiant} \right)$$

**Remarque.** Le produit est une théta jointure dont le prédicat serait toujours vrai.

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $S(B_1, B_2, \dots, B_p)$  n'ayant aucun attribut en commun. La théta jointure selon le prédicat  $p$ , noté  $R \bowtie[p] S$  est la relation temporaire de schéma  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_p)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  et de  $S$  concaténés qui satisfont le prédicat  $p$ .

$$R \bowtie[p] S \equiv \sigma[p](R \times S)$$

**Syntaxe du prédicat.** même syntaxe + :

$\langle \text{condition} \rangle ::= \text{nomAttrdeR} \langle \text{opérateur comp} \rangle \text{nomAttrdeS}$

**Exemple.**

- liste des couples de numéros d'étudiants nés le même jours

$$\left( \text{Etudiant} \quad \alpha \left[ \begin{array}{l} \text{numEtud} : \text{num1} \\ \text{dateN} : \text{dateN1} \end{array} \right] \pi \left[ \begin{array}{l} \text{numEtud} \\ \text{dateN} \end{array} \right] \text{Etudiant} \right)$$

**Remarque.** Le produit est une théta jointure dont le prédicat serait toujours vrai.

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $S(B_1, B_2, \dots, B_p)$  n'ayant aucun attribut en commun. La théta jointure selon le prédicat  $p$ , noté  $R \bowtie [p] S$  est la relation temporaire de schéma  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_p)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  et de  $S$  concaténés qui satisfont le prédicat  $p$ .

$$R \bowtie [p] S \equiv \sigma[p](R \times S)$$

**Syntaxe du prédicat.** même syntaxe + :

$\langle \text{condition} \rangle ::= \text{nomAttrdeR} \langle \text{opérateur comp} \rangle \text{nomAttrdeS}$

**Exemple.**

- liste des couples de numéros d'étudiants nés le même jours

$$\left( \text{Etudiant} \bowtie \left[ \begin{array}{l} (\text{numEtud} < \text{num1}) \\ \wedge (\text{dateN} = \text{dateN1}) \end{array} \right] \alpha \left[ \begin{array}{l} \text{numEtud} : \text{num1} \\ \text{dateN} : \text{dateN1} \end{array} \right] \pi \left[ \begin{array}{l} \text{numEtud} \\ \text{dateN} \end{array} \right] \text{Etudiant} \right)$$

**Remarque.** Le produit est une théta jointure dont le prédicat serait toujours vrai.

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $S(B_1, B_2, \dots, B_p)$  n'ayant aucun attribut en commun. La théta jointure selon le prédicat  $p$ , noté  $R \bowtie [p] S$  est la relation temporaire de schéma  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_p)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  et de  $S$  concaténés qui satisfont le prédicat  $p$ .

$$R \bowtie [p] S \equiv \sigma[p](R \times S)$$

**Syntaxe du prédicat.** même syntaxe + :

$\langle \text{condition} \rangle ::= \text{nomAttrdeR} \langle \text{opérateur comp} \rangle \text{nomAttrdeS}$

**Exemple.**

- liste des couples de numéros d'étudiants nés le même jours

$$\pi \left[ \begin{array}{c} \text{numEtud} \\ \text{num1} \end{array} \right] \left( \text{Etudiant} \bowtie \left[ \begin{array}{c} (\text{numEtud} < \text{num1}) \\ \wedge (\text{dateN} = \text{dateN1}) \end{array} \right] \alpha \left[ \begin{array}{c} \text{numEtud} : \text{num1} \\ \text{dateN} : \text{dateN1} \end{array} \right] \pi \left[ \begin{array}{c} \text{numEtud} \\ \text{dateN} \end{array} \right] \text{Etudiant} \right)$$

**Remarque.** Le produit est une théta jointure dont le prédicat serait toujours vrai.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

## Différents types de jointures :

- ② `[INNER] JOIN ... ON ...` : jointure par défaut de MySQL.  
Ne sont incluses dans le résultat final que les lignes qui se correspondent dans les deux tables.

```
SELECT * FROM t1 JOIN t2 ON condition1  
WHERE condition2
```

- La condition `ON` = une condition `WHERE` .  
Condition de jointure  $\neq$  Condition de filtrage
- La lisibilité des requêtes est plus grande en utilisant
  - la syntaxe à base de `JOIN` ,
  - en isolant ce qui est du filtrage et de la jointure,
  - en isolant avec clarté chaque condition de jointure entre chaque couple de table.

⇒ Optimisation d'exécution de la requête plus pointue avec `JOIN` .

- $T_1 \times T_2 \times \dots \times T_k$  sans `WHERE` (pour tests)  
⇒ mise à genoux du serveur !

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

## Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

La jointure de  $R(X, Y)$  et  $S(Y, Z)$  où  $X, Y, Z$  sont des attributs (ou des ensembles d'attributs) et où  $Y$  n'est pas vide, est la relation notée  $R \bowtie S$ , de schéma  $(X, Y, Z)$  dont la population est l'ensemble des  $n$ -uplets  $\langle x, y, z \rangle$  obtenus par composition d'un  $n$ -uplet  $\langle x, y \rangle$  de  $R$  et d'un  $n$ -uplet  $\langle y, z \rangle$  de  $S$  tels que les valeurs pour  $Y$  sont égales.

- La population de  $R \bowtie S$  contient  $k$   $n$ -uplets avec  $k \in [0 : |R| \times |S|]$ .
- La population est vide ssi il n'y a pas de  $n$ -uplet de  $R$  et  $S$  qui ont même valeur pour  $Y$ .
- Il y a  $|R| \times |S|$   $n$ -uplets dans la jointure lorsque les  $n$ -uplets de  $R$  et de  $S$  ont tous la même valeur pour  $Y$ .

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

## Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

## le modèle

Entités-  
AssociationsNormalisation  
d'une relation

La jointure de  $R(X, Y)$  et  $S(Y, Z)$  où  $X, Y, Z$  sont des attributs (ou des ensembles d'attributs) et où  $Y$  n'est pas vide, est la relation notée  $R \bowtie S$ , de schéma  $(X, Y, Z)$  dont la population est l'ensemble des  $n$ -uplets  $\langle x, y, z \rangle$  obtenus par composition d'un  $n$ -uplet  $\langle x, y \rangle$  de  $R$  et d'un  $n$ -uplet  $\langle y, z \rangle$  de  $S$  tels que les valeurs pour  $Y$  sont égales.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- Quels sont les acteurs qui jouent dans "La syndicaliste" complétés par le numéro du producteur de ce film ?

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

## Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

## le modèle

Entités-  
AssociationsNormalisation  
d'une relation

La jointure de  $R(X, Y)$  et  $S(Y, Z)$  où  $X, Y, Z$  sont des attributs (ou des ensembles d'attributs) et où  $Y$  n'est pas vide, est la relation notée  $R \bowtie S$ , de schéma  $(X, Y, Z)$  dont la population est l'ensemble des  $n$ -uplets  $\langle x, y, z \rangle$  obtenus par composition d'un  $n$ -uplet  $\langle x, y \rangle$  de  $R$  et d'un  $n$ -uplet  $\langle y, z \rangle$  de  $S$  tels que les valeurs pour  $Y$  sont égales.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- Quels sont les acteurs qui jouent dans "La syndicaliste" complétés par le numéro du producteur de ce film ?

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

## Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

## Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

La jointure de  $R(X, Y)$  et  $S(Y, Z)$  où  $X, Y, Z$  sont des attributs (ou des ensembles d'attributs) et où  $Y$  n'est pas vide, est la relation notée  $R \bowtie S$ , de schéma  $(X, Y, Z)$  dont la population est l'ensemble des  $n$ -uplets  $\langle x, y, z \rangle$  obtenus par composition d'un  $n$ -uplet  $\langle x, y \rangle$  de  $R$  et d'un  $n$ -uplet  $\langle y, z \rangle$  de  $S$  tels que les valeurs pour  $Y$  sont égales.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{long}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- Quels sont les acteurs qui jouent dans "La syndicaliste" complétés par le numéro du producteur de ce film ?

$$\pi \left[ \begin{array}{l} \text{nomActeur}, \\ \text{IdProd} \end{array} \right] \sigma \left[ \begin{array}{l} \text{Titre} = \\ \text{"La syndicaliste"} \\ \wedge \text{date} = 2024 \end{array} \right] \left( \text{Film} \bowtie \alpha \left[ \begin{array}{l} \text{TitreFilm} : \text{Titre} \\ \text{FilmDate} : \text{date} \end{array} \right] \text{Joue} \right)$$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \\ \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \\ \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

**Exemples.**

- Donner tous les renseignements sur les étudiants (nom, adresse, date de naissance, numEtud, numPers)
- Donner les noms des étudiants ayant réussi le cours d'algorithmique.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \\ \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \\ \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

## Exemples.

- Donner tous les renseignements sur les étudiants (nom, adresse, date de naissance, numEtud, numPers)

**Etudiant ❌ Personne**

- Donner les noms des étudiants ayant réussi le cours d'algorithmique.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \textit{personne}(\textit{numPers}, \textit{nom}, \textit{adresse}), \\ \textit{personnePrenom}(\textit{numPers}, \textit{numPrenom}, \textit{prenom}), \\ \textit{etudiant}(\textit{numPers}, \textit{numEtud}, \textit{dateN}), \\ \textit{etudiantEtude}(\textit{numEtud}, \textit{annee}, \textit{diplome}), \\ \textit{enseignant}(\textit{numPers}, \textit{tel}, \textit{status}, \textit{numEns}, \textit{banque}, \textit{agence}, \textit{compte}), \\ \textit{cours}(\textit{nomC}, \textit{cycle}, \textit{numEns}), \\ \textit{obtenu}(\textit{numEtud}, \textit{nomC}, \textit{note}), \\ \textit{inscrit}(\textit{numEtud}, \textit{nomC}), \\ \textit{prerequis}(\textit{nomC}, \textit{nomCpre}) \end{array} \right\}$$

## Exemples.

- Donner tous les renseignements sur les étudiants (nom, adresse, date de naissance, numEtud, numPers)

**Etudiant ❌ Personne**

- Donner les noms des étudiants ayant réussi le cours d'algorithmique.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \textit{personne}(\textit{numPers}, \textit{nom}, \textit{adresse}), \\ \textit{personnePrenom}(\textit{numPers}, \textit{numPrenom}, \textit{prenom}), \\ \textit{etudiant}(\textit{numPers}, \textit{numEtud}, \textit{dateN}), \\ \textit{etudiantEtude}(\textit{numEtud}, \textit{annee}, \textit{diplome}), \\ \textit{enseignant}(\textit{numPers}, \textit{tel}, \textit{status}, \textit{numEns}, \textit{banque}, \textit{agence}, \textit{compte}), \\ \textit{cours}(\textit{nomC}, \textit{cycle}, \textit{numEns}), \\ \textit{obtenu}(\textit{numEtud}, \textit{nomC}, \textit{note}), \\ \textit{inscrit}(\textit{numEtud}, \textit{nomC}), \\ \textit{prerequis}(\textit{nomC}, \textit{nomCpre}) \end{array} \right\}$$

## Exemples.

- Donner tous les renseignements sur les étudiants (nom, adresse, date de naissance, numEtud, numPers)

**Etudiant ❌ Personne**

- Donner les noms des étudiants ayant réussi le cours d'algorithmique.

$\pi[\textit{nom}](\sigma[\textit{nomC} = \text{'algo'}](\textit{Etudiant} \bowtie \textit{Personne} \bowtie \textit{obtenu}))$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \\ \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \\ \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

## Exemples.

- Donner tous les renseignements sur les étudiants (nom, adresse, date de naissance, numEtud, numPers)

Etudiant ❌ Personne

- Donner les noms des étudiants ayant réussi le cours d'algorithmique.

$\pi[\text{nom}](\sigma[\text{nomC} = \text{'algo'}](\text{Etudiant} \bowtie \text{Personne} \bowtie \text{obtenu}))$

$\pi[\text{nom}](\text{Etudiant} \bowtie \text{Personne} \bowtie (\sigma[\text{nomC} = \text{'algo'}]\text{obtenu}))$

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

## Différents types de jointures :

③ `[NATURAL] JOIN ... [USING ...]` : ✕

en spécifiant éventuellement les listes de attributs qui doivent exister dans les 2 tables.

```
SELECT * FROM t1 JOIN t2 USING (c1,c2,c3)
```

La clause USING (columnlist) recense la liste des attributs qui doivent exister dans les 2 tables. Les 2 jointures suivantes donnent le même résultat :

```
a JOIN b USING (c1,c2,c3)
```

```
a JOIN b ON a.c1=b.c1 AND a.c2=b.c2 AND a.c3=b.c3
```

Le terme NATURAL permet d'éviter de préciser les colonnes concernées par la jointure.

Lorsqu'on a plusieurs tables à joindre :

```
SELECT * FROM a JOIN b ON a.x=b.y JOIN c ON b.z=c.t
```

...

## Différents types de jointures :

- ④ **LEFT /RIGHT /FULL [OUTER] JOIN ...** : jointures externes.

Exemple :  $\left\{ \begin{array}{l} \text{dept}(\text{numdept}, \text{nomdept}) \\ \text{employeur}(\text{numdept}, \text{nom}, \text{adresse} \dots) \end{array} \right.$

Certains  $n$ -uplets de dept ont un  $n^o$  de dépt qui ne correspond à aucune ligne de employeur.

Ces  $n$ -uplets sont *à l'extérieur de la jointure* entre les 2 tables.

Une **jointure externe** permet de rajouter ces  $n$ -uplets.

**LEFT JOIN**  $\equiv \left\{ \begin{array}{l} \text{les } n\text{-uplets de la } \bowtie \text{ naturelle} \\ + \text{ les } n\text{-uplets de la table de gauche} \end{array} \right.$

- même syntaxe que le **[INNER] JOIN** .

- on peut utiliser une jointure externe pour trouver les enregistrements dans une table qui n'ont pas de correspondances dans une autre.

**RIGHT JOIN**  $\equiv \left\{ \begin{array}{l} \text{les } n\text{-uplets de la } \bowtie \text{ naturelle} + \\ \text{les } n\text{-uplets de la table de droite} \end{array} \right.$

**FULL JOIN**  $\equiv \left\{ \begin{array}{l} \text{les } n\text{-uplets de la } \bowtie \text{ naturelle} + \\ \text{les } n\text{-uplets de la table de gauche} \\ \text{les } n\text{-uplets de la table de droite} \end{array} \right.$

(le **FULL** n'est pas implémenté dans MySQL, mais est calculable)

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

## Différents types de jointures :

- ④ **LEFT /RIGHT /FULL [OUTER] JOIN ...** : jointures externes.

Les jointures externes sont extrêmement pratiques pour rapatrier le maximum d'information disponible, même si des lignes de table ne sont pas renseignées entre les différentes tables jointes.

### Exemple d'utilisation.

```
{ Disque(Id, Nom_Disque, Commentaires, IdPays)
  Pays(IdPays, NPays)
```

Pour certains Disque, le code du pays n'a pas été précisé.

⇒ Lister tous les disques avec le nom du pays s'il existe.

Jointure naturelle :

```
SELECT * FROM Disque JOIN Pays USING (IdPays);
```

Produit cartésien :

```
SELECT * FROM Disque, Pays
WHERE Disque.IdPays=Pays.IdPays;
```

Jointures externes :

```
SELECT * FROM Disque LEFT JOIN Pays USING (IdPays);
```

Ainsi, en utilisant cette jointure, on précise que la table Disque et la table Pays sont liées par le champs IdPays.

Et voilà !

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

## Différents types de jointures :

- ④ **LEFT /RIGHT /FULL [OUTER] JOIN ...** : jointures externes.

**Autre exemple d'utilisation de la jointure externe :**

```
SELECT table1.* FROM table1  
LEFT JOIN table2 ON table1.id=table2.id  
WHERE table2.id IS NULL;
```

Cet exemple retourne

- toutes les lignes trouvées dans **table1** avec une valeur de **id** qui n'est pas présente dans **table2**
- (autrement dit) toutes les lignes de **table1** sans correspondance dans la table **table2** .

⇒ Cela demande que **table2.id** soit déclaré **NOT NULL**

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

**Jointures**

Opérateurs  
ensemblistes

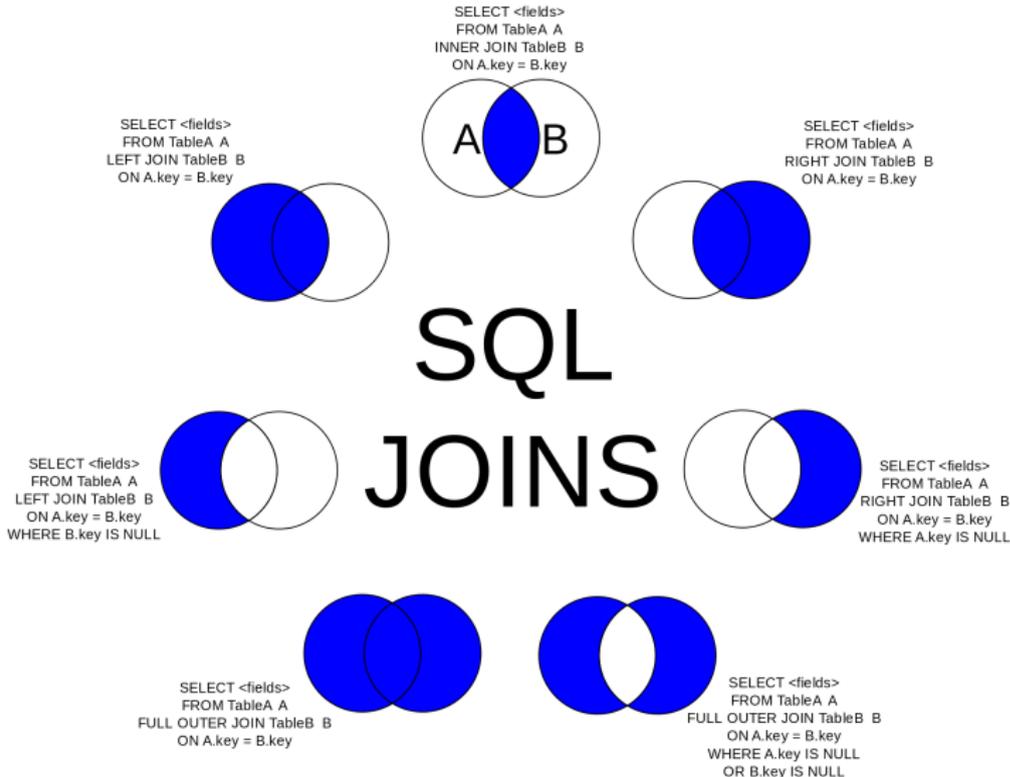
Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation



This work is licensed under a Creative Commons Attribution 3.0 Unported License.  
Author: <http://commons.wikimedia.org/wiki/User:Arbeck>

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
JointuresOpérateurs  
ensemblistesOpérateurs hors de  
l'Alg.  
Autres ordres SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

Soient deux relations de même schéma  $R(A_1, A_2, \dots, A_n)$  et  $S(A_1, A_2, \dots, A_n)$ .

**Union** :  $R \cup S$  est la relation temporaire de même schéma et de population égale à l'ensemble des  $n$ -uplets de  $R$  et de ceux de  $S$  (avec élimination des doublons éventuels).

**Différence** :  $R \setminus S$  est la relation temporaire de même schéma et de population égale à l'ensemble des  $n$ -uplets de  $R$  moins ceux de  $S$ , c'est à dire à l'ensemble des  $n$ -uplets qui se trouvent dans  $R$  mais pas dans  $S$ .

**Intersection** :  $R \cap S$  est la relation temporaire de même schéma et de population égale à l'ensemble des  $n$ -uplets qui sont présents à la fois dans  $R$  et dans  $S$ .

**Exemple.** Pour illustrer les opérateurs ensemblistes, nous reprenons la base de données relationnelle InstitutFormation dont le schéma est le suivant :

$$S = \left\{ \begin{array}{l} \textit{personne}(\textit{numPers}, \textit{nom}, \textit{adresse}), \\ \textit{personnePrenom}(\textit{numPers}, \textit{numPrenom}, \textit{prenom}), \\ \textit{etudiant}(\textit{numPers}, \textit{numEtud}, \textit{dateN}), \\ \textit{etudiantEtude}(\textit{numEtud}, \textit{annee}, \textit{diplome}), \\ \textit{enseignant}(\textit{numPers}, \textit{tel}, \textit{status}, \textit{numEns}, \textit{banque}, \textit{agence}, \textit{compte}), \\ \textit{cours}(\textit{nomC}, \textit{cycle}, \textit{numEns}), \\ \textit{obtenu}(\textit{numEtud}, \textit{nomC}, \textit{note}), \\ \textit{inscrit}(\textit{numEtud}, \textit{nomC}), \\ \textit{prerequis}(\textit{nomC}, \textit{nomCpre}) \end{array} \right\}$$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation**Exemple.**

- Donnez la liste composée d'une part de la liste des numéros des enseignants en BD et d'autre part de la liste des numéros des étudiants inscrits en cours de BD.
- Donnez la liste des numéros d'étudiants qui ne sont pas inscrits au cours de BD.
- Donnez la liste des numéros des personnes qui sont à la fois enseignants et étudiants.

## Exemple.

- Donnez la liste composée d'une part de la liste des numéros des enseignants en BD et d'autre part de la liste des numéros des étudiants inscrits en cours de BD.

$\pi[\text{numEns}]\sigma[\text{nomC} = \text{BD}]\text{Cours}$

- Donnez la liste des numéros d'étudiants qui ne sont pas inscrits au cours de BD.
- Donnez la liste des numéros des personnes qui sont à la fois enseignants et étudiants.

## Exemple.

- Donnez la liste composée d'une part de la liste des numéros des enseignants en BD et d'autre part de la liste des numéros des étudiants inscrits en cours de BD.

$$\pi[\text{numEns}]\sigma[\text{nomC} = \text{BD}]\text{Cours}$$
$$\pi[\text{numEtud}]\sigma[\text{nomC} = \text{BD}]\text{inscrit}$$

- Donnez la liste des numéros d'étudiants qui ne sont pas inscrits au cours de BD.
- Donnez la liste des numéros des personnes qui sont à la fois enseignants et étudiants.

## Exemple.

- Donnez la liste composée d'une part de la liste des numéros des enseignants en BD et d'autre part de la liste des numéros des étudiants inscrits en cours de BD.

$$\pi[\text{numEns}]\sigma[\text{nomC} = \text{BD}]\text{Cours}$$
$$\cup$$
$$\pi[\text{numEtud}]\sigma[\text{nomC} = \text{BD}]\text{inscrit}$$

- Donnez la liste des numéros d'étudiants qui ne sont pas inscrits au cours de BD.
- Donnez la liste des numéros des personnes qui sont à la fois enseignants et étudiants.

**Exemple.**

- Donnez la liste composée d'une part de la liste des numéros des enseignants en BD et d'autre part de la liste des numéros des étudiants inscrits en cours de BD.

$$H = \begin{cases} \alpha[\text{numEns} : \text{num}](\pi[\text{numEns}]\sigma[\text{nomC} = \text{BD}]\text{Cours}) \\ \cup \\ \alpha[\text{numEtud} : \text{num}](\pi[\text{numEtud}]\sigma[\text{nomC} = \text{BD}]\text{inscrit}) \end{cases}$$

- Donnez la liste des numéros d'étudiants qui ne sont pas inscrits au cours de BD.
- Donnez la liste des numéros des personnes qui sont à la fois enseignants et étudiants.

**Exemple.**

- Donnez la liste composée d'une part de la liste des numéros des enseignants en BD et d'autre part de la liste des numéros des étudiants inscrits en cours de BD.

$$H = \begin{cases} \alpha[\text{numEns} : \text{num}](\pi[\text{numEns}]\sigma[\text{nomC} = \text{BD}]\text{Cours}) \\ \cup \\ \alpha[\text{numEtud} : \text{num}](\pi[\text{numEtud}]\sigma[\text{nomC} = \text{BD}]\text{inscrit}) \end{cases}$$

- Donnez la liste des numéros d'étudiants qui ne sont pas inscrits au cours de BD.

$$\pi[\text{numEtud}]\text{Etudiant} \setminus \pi[\text{numEtud}]\sigma[\text{nomC} = \text{BD}]\text{inscrit}$$

- Donnez la liste des numéros des personnes qui sont à la fois enseignants et étudiants.

**Exemple.**

- Donnez la liste composée d'une part de la liste des numéros des enseignants en BD et d'autre part de la liste des numéros des étudiants inscrits en cours de BD.

$$H = \begin{cases} \alpha[\text{numEns} : \text{num}](\pi[\text{numEns}]\sigma[\text{nomC} = \text{BD}]\text{Cours}) \\ \cup \\ \alpha[\text{numEtud} : \text{num}](\pi[\text{numEtud}]\sigma[\text{nomC} = \text{BD}]\text{inscrit}) \end{cases}$$

- Donnez la liste des numéros d'étudiants qui ne sont pas inscrits au cours de BD.

$$\pi[\text{numEtud}]\text{Etudiant} \setminus \pi[\text{numEtud}]\sigma[\text{nomC} = \text{BD}]\text{inscrit}$$

- Donnez la liste des numéros des personnes qui sont à la fois enseignants et étudiants.

$$\pi[\text{numPers}]\text{Enseignant} \cap \pi[\text{numPers}]\text{Etudiant}$$

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

**Opérateurs  
ensemblistes**

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- syntaxe SQL : **UNION** , **INTERSECTION** , **EXCEPT** ou **MINUS** .
- Quels sont les noms et adresses des acteurs qui sont des producteurs ?  
Solution en SQL :

Solution en algèbre relationnelle :

- Seul l'opérateur **UNION** est implémenté en MySQL  
**Opérateur IN** . L'opérateur **IN** (resp. **NOT IN** ) permet de simuler les autres opérateurs ensemblistes de l'algèbre relationnelle.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

**Opérateurs  
ensemblistes**

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- syntaxe SQL : **UNION** , **INTERSECTION** , **EXCEPT** ou **MINUS** .
- Quels sont les noms et adresses des acteurs qui sont des producteurs ?

Solution en SQL :

```
(SELECT nom,adresse FROM Acteur)
```

```
(SELECT nom,adresse FROM Producteur)
```

Solution en algèbre relationnelle :

- Seul l'opérateur **UNION** est implémenté en MySQL
- Opérateur IN** . L'opérateur **IN** (resp. **NOT IN** ) permet de simuler les autres opérateurs ensemblistes de l'algèbre relationnelle.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

**Opérateurs  
ensemblistes**

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- syntaxe SQL : **UNION** , **INTERSECTION** , **EXCEPT** ou **MINUS** .
- Quels sont les noms et adresses des acteurs qui sont des producteurs ?

Solution en SQL :

```
(SELECT nom,adresse FROM Acteur)
```

```
INTERSECTION
```

```
(SELECT nom,adresse FROM Producteur)
```

Solution en algèbre relationnelle :

- Seul l'opérateur **UNION** est implémenté en MySQL
- Opérateur IN** . L'opérateur **IN** (resp. **NOT IN** ) permet de simuler les autres opérateurs ensemblistes de l'algèbre relationnelle.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- syntaxe SQL : **UNION** , **INTERSECTION** , **EXCEPT** ou **MINUS** .
- Quels sont les noms et adresses des acteurs qui sont des producteurs ?

Solution en SQL :

(SELECT nom,adresse FROM Acteur)

INTERSECTION

(SELECT nom,adresse FROM Producteur)

Solution en algèbre relationnelle :

$$\pi \left[ \begin{array}{l} \textit{nom}, \\ \textit{adresse} \end{array} \right] (\textit{Acteur}) \cap \pi \left[ \begin{array}{l} \textit{nom}, \\ \textit{adresse} \end{array} \right] (\textit{Producteur})$$

- Seul l'opérateur **UNION** est implémenté en MySQL

**Opérateur IN** . L'opérateur **IN** (resp. **NOT IN** ) permet de simuler les autres opérateurs ensemblistes de l'algèbre relationnelle.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- syntaxe SQL : **UNION** , **INTERSECTION** , **EXCEPT** ou **MINUS** .
- Quels sont les noms et adresses des acteurs qui **ne** sont **pas** des producteurs ?

Solution en SQL :

(SELECT nom,adresse FROM Acteur)

(SELECT nom,adresse FROM Producteur)

Solution en algèbre relationnelle :

$$\pi \left[ \begin{array}{c} \text{nom,} \\ \text{adresse} \end{array} \right] (\text{Acteur}) \quad \pi \left[ \begin{array}{c} \text{nom,} \\ \text{adresse} \end{array} \right] (\text{Producteur})$$

- Seul l'opérateur **UNION** est implémenté en MySQL

**Opérateur IN** . L'opérateur **IN** (resp. **NOT IN** ) permet de simuler les autres opérateurs ensemblistes de l'algèbre relationnelle.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentiels

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- syntaxe SQL : **UNION** , **INTERSECTION** , **EXCEPT** ou **MINUS** .
- Quels sont les noms et adresses des acteurs qui **ne** sont **pas** des producteurs ?

Solution en SQL :

```
(SELECT nom,adresse FROM Acteur)
```

```
EXCEPT
```

```
(SELECT nom,adresse FROM Producteur)
```

Solution en algèbre relationnelle :

$$\pi \left[ \begin{array}{c} \textit{nom}, \\ \textit{adresse} \end{array} \right] (\textit{Acteur}) \quad \setminus \quad \pi \left[ \begin{array}{c} \textit{nom}, \\ \textit{adresse} \end{array} \right] (\textit{Producteur})$$

- Seul l'opérateur **UNION** est implémenté en MySQL

**Opérateur IN** . L'opérateur **IN** (resp. **NOT IN** ) permet de simuler les autres opérateurs ensemblistes de l'algèbre relationnelle.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

**Opérateurs  
ensemblistes**

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

$$\mathcal{S} = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- 1 Quels sont les noms des acteurs qui ne sont pas producteurs ?
- 2 Quels sont les noms des producteurs de films dans lesquels joue Matt Damon qui sont sortis avant 2006 ?

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- 1 Quels sont les noms des acteurs qui ne sont pas producteurs ?

```
SELECT nom FROM Acteur
```

```
WHERE nom NOT IN (SELECT Nom FROM Producteur)
```

- 2 Quels sont les noms des producteurs de films dans lesquels joue Matt Damon qui sont sortis avant 2006 ?

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

## Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

## Autres ordres SQL

## le modèle

Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- 1 Quels sont les noms des acteurs qui ne sont pas producteurs ?

```
SELECT nom FROM Acteur
```

```
WHERE nom NOT IN (SELECT Nom FROM Producteur)
```

$$\pi[\text{nom}] \text{Acteur} \setminus \alpha[\text{Nom} : \text{nom}] \pi[\text{Nom}] \text{Producteur}$$

- 2 Quels sont les noms des producteurs de films dans lesquels joue Matt Damon qui sont sortis avant 2006 ?

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les noms des acteurs qui ne sont pas producteurs ?

```
SELECT nom FROM Acteur
```

```
WHERE nom NOT IN (SELECT Nom FROM Producteur)
```

$$\pi[\text{nom}] \text{Acteur} \setminus \alpha[\text{Nom} : \text{nom}] \pi[\text{Nom}] \text{Producteur}$$

- ❷ Quels sont les noms des producteurs de films dans lesquels joue Matt Damon qui sont sortis avant 2006 ?

```
(SELECT TitreFilm, FilmDate FROM Joue
```

```
WHERE nomActeur = 'Matt Damon' and FilmDate <= 2006)
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les noms des acteurs qui ne sont pas producteurs ?

```
SELECT nom FROM Acteur
```

```
WHERE nom NOT IN (SELECT Nom FROM Producteur)
```

$$\pi[\text{nom}] \text{Acteur} \setminus \alpha[\text{Nom} : \text{nom}] \pi[\text{Nom}] \text{Producteur}$$

- ❷ Quels sont les noms des producteurs de films dans lesquels joue Matt Damon qui sont sortis avant 2006 ?

```
(SELECT IdProd FROM Film WHERE
```

```
(Titre,date) IN (SELECT TitreFilm,FilmDate FROM Joue  
WHERE nomActeur = 'Matt Damon' and FilmDate<=2006) )
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

## Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

## Autres ordres SQL

## le modèle

Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les noms des acteurs qui ne sont pas producteurs ?

```
SELECT nom FROM Acteur
```

```
WHERE nom NOT IN (SELECT Nom FROM Producteur)
```

$\pi[\text{nom}] \text{Acteur} \setminus \alpha[\text{Nom} : \text{nom}] \pi[\text{Nom}] \text{Producteur}$

- ❷ Quels sont les noms des producteurs de films dans lesquels joue Matt Damon qui sont sortis avant 2006 ?

```
SELECT Nom FROM Producteur WHERE
```

```
Id IN (SELECT IdProd FROM Film WHERE
```

```
(Titre,date) IN (SELECT TitreFilm,FilmDate FROM Joue  
WHERE nomActeur = 'Matt Damon' and FilmDate<=2006) )
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les noms des acteurs qui ne sont pas producteurs ?

```
SELECT nom FROM Acteur
```

```
WHERE nom NOT IN (SELECT Nom FROM Producteur)
```

$$\pi[\text{nom}] \text{Acteur} \setminus \alpha[\text{Nom} : \text{nom}] \pi[\text{Nom}] \text{Producteur}$$

- ❷ Quels sont les noms des producteurs de films dans lesquels joue Matt Damon qui sont sortis avant 2006 ?

```
SELECT Nom FROM Producteur WHERE
```

```
Id IN (SELECT IdProd FROM Film WHERE
```

```
(Titre,date) IN (SELECT TitreFilm,FilmDate FROM Joue
```

```
WHERE nomActeur = 'Matt Damon' and FilmDate<=2006) )
```

$$\pi[\text{Nom}] \left( \text{Producteur} \bowtie_{\left[ \begin{array}{l} \text{IdProd} \\ = \text{Id} \end{array} \right]} \text{Film} \bowtie_{\left[ \begin{array}{l} \text{Titre} = \text{TitreFilm} \wedge \\ \text{Date} = \text{FilmDate} \end{array} \right]} \sigma_{\left[ \begin{array}{l} \text{DateFilm} \leq 2006 \\ \wedge (\text{nomActeur} = \\ \text{"MattDamon"}) \end{array} \right]} \text{Joue} \right)$$

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

**Opérateurs  
ensemblistes**

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \textit{personne}(\textit{numPers}, \textit{nom}, \textit{adresse}), \\ \textit{personnePrenom}(\textit{numPers}, \textit{numPrenom}, \textit{prenom}), \\ \textit{etudiant}(\textit{numPers}, \textit{numEtud}, \textit{dateN}), \\ \textit{etudiantEtude}(\textit{numEtud}, \textit{annee}, \textit{diplome}), \\ \textit{enseignant}(\textit{numPers}, \textit{tel}, \textit{status}, \textit{numEns}, \textit{banque}, \textit{agence}, \textit{compte}), \\ \textit{cours}(\textit{nomC}, \textit{cycle}, \textit{numEns}), \\ \textit{obtenu}(\textit{numEtud}, \textit{nomC}, \textit{note}), \\ \textit{inscrit}(\textit{numEtud}, \textit{nomC}), \\ \textit{prerequis}(\textit{nomC}, \textit{nomCpre}) \end{array} \right\}$$

- 1 Quels sont les  $n^o$  des personnes qui sont enseignants mais pas étudiants ?

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \textit{personne}(\textit{numPers}, \textit{nom}, \textit{adresse}), \\ \textit{personnePrenom}(\textit{numPers}, \textit{numPrenom}, \textit{prenom}), \\ \textit{etudiant}(\textit{numPers}, \textit{numEtud}, \textit{dateN}), \\ \textit{etudiantEtude}(\textit{numEtud}, \textit{annee}, \textit{diplome}), \\ \textit{enseignant}(\textit{numPers}, \textit{tel}, \textit{status}, \textit{numEns}, \textit{banque}, \textit{agence}, \textit{compte}), \\ \textit{cours}(\textit{nomC}, \textit{cycle}, \textit{numEns}), \\ \textit{obtenu}(\textit{numEtud}, \textit{nomC}, \textit{note}), \\ \textit{inscrit}(\textit{numEtud}, \textit{nomC}), \\ \textit{prerequis}(\textit{nomC}, \textit{nomCpre}) \end{array} \right\}$$

- ❶ Quels sont les  $n^o$  des personnes qui sont enseignants mais pas étudiants ?

```
SELECT numPers FROM enseignant  
WHERE numPers NOT IN  
(SELECT numPers FROM etudiant)
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
JointuresOpérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \\ \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \\ \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

- ① Quels sont les  $n^{\circ}$  des personnes qui sont enseignants mais pas étudiants ?

```
SELECT numPers FROM enseignant
WHERE numPers NOT IN
  (SELECT numPers FROM etudiant)
π[numPers]enseignant \ π[numPers]etudiant
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

**Opérateurs  
ensemblistes**

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \\ \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \\ \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

- 2 Quels sont les  $n^{\circ}$  de téléphone des enseignants qui enseignent dans un cours validé par un étudiant dont le nom est Gautier ?

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

**Opérateurs  
ensemblistes**

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \textit{personne}(\textit{numPers}, \textit{nom}, \textit{adresse}), \\ \textit{personnePrenom}(\textit{numPers}, \textit{numPrenom}, \textit{prenom}), \\ \textit{etudiant}(\textit{numPers}, \textit{numEtud}, \textit{dateN}), \\ \textit{etudiantEtude}(\textit{numEtud}, \textit{annee}, \textit{diplome}), \\ \textit{enseignant}(\textit{numPers}, \textit{tel}, \textit{status}, \textit{numEns}, \textit{banque}, \textit{agence}, \textit{compte}), \\ \textit{cours}(\textit{nomC}, \textit{cycle}, \textit{numEns}), \\ \textit{obtenu}(\textit{numEtud}, \textit{nomC}, \textit{note}), \\ \textit{inscrit}(\textit{numEtud}, \textit{nomC}), \\ \textit{prerequis}(\textit{nomC}, \textit{nomCpre}) \end{array} \right\}$$

- 2 Quels sont les  $n^{\circ}$  de téléphone des enseignants qui enseignent dans un cours validé par un étudiant dont le nom est Gautier ?

(SELECT numPers FROM personne WHERE nom = 'Gautier')

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
JointuresOpérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \\ \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \\ \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

- 2 Quels sont les  $n^{\circ}$  de téléphone des enseignants qui enseignent dans un cours validé par un étudiant dont le nom est Gautier ?

```
(SELECT numEtud FROM Etudiants WHERE numPers IN
  (SELECT numPers FROM personne WHERE nom = 'Gautier')
)
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
JointuresOpérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \\ \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \\ \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

- 2 Quels sont les  $n^{\circ}$  de téléphone des enseignants qui enseignent dans un cours validé par un étudiant dont le nom est Gautier ?

```
(SELECT nomC FROM obtenu WHERE numEtud IN
  (SELECT numEtud FROM Etudiants WHERE numPers IN
    (SELECT numPers FROM personne WHERE nom = 'Gautier')
  )
)
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
JointuresOpérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \\ \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \\ \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

- 2 Quels sont les  $n^{\circ}$  de téléphone des enseignants qui enseignent dans un cours validé par un étudiant dont le nom est Gautier ?

```
(SELECT numEns FROM cours WHERE nomC IN
  (SELECT nomC FROM obtenu WHERE numEtud IN
    (SELECT numEtud FROM Etudiants WHERE numPers IN
      (SELECT numPers FROM personne WHERE nom = 'Gautier')
    )
  )
)
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
JointuresOpérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$S = \left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \\ \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \\ \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

- 2 Quels sont les  $n^{\circ}$  de téléphone des enseignants qui enseignent dans un cours validé par un étudiant dont le nom est Gautier ?

```
SELECT tel FROM enseignant WHERE numEns IN
(SELECT numEns FROM cours WHERE nomC IN
(SELECT nomC FROM obtenu WHERE numEtud IN
(SELECT numEtud FROM Etudiants WHERE numPers IN
(SELECT numPers FROM personne WHERE nom = 'Gautier')
)
)
)
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

## Opérateurs unaires

## Traduction en SQL

## Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

## Autres ordres SQL

## le modèle

Entités-  
AssociationsNormalisation  
d'une relation

- Les variables de  $n$ -uplets vont être utiles lorsque l'on cherche à raisonner sur plusieurs  $n$ -uplets d'une même relation et à les comparer.
- L'idée est de prendre deux  $n$ -uplets dans une relation (autrement dit un  $n$ -uplet de  $R \times R$ ) et de sélectionner les couples de  $n$ -uplets satisfaisant certaines contraintes.

**Exemple : Quels acteurs ont la même adresse ?**

```
SELECT Star1.Nom, Star2.Nom FROM Acteur Star1, Acteur Star2
```

Ici Star1 et Star2 sont des variables pour les  $n$ -uplets. On aurait pu nommé ces variables différemment par exemple t1 et t2.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

- Les variables de  $n$ -uplets vont être utiles lorsque l'on cherche à raisonner sur plusieurs  $n$ -uplets d'une même relation et à les comparer.
- L'idée est de prendre deux  $n$ -uplets dans une relation (autrement dit un  $n$ -uplet de  $R \times R$ ) et de sélectionner les couples de  $n$ -uplets satisfaisant certaines contraintes.

### Exemple : Quels acteurs ont la même adresse ?

```
SELECT Star1.Nom, Star2.Nom FROM Acteur Star1, Acteur Star2  
WHERE Star1.adresse = Star2.adresse
```

Ici Star1 et Star2 sont des variables pour les  $n$ -uplets. On aurait pu nommé ces variables différemment par exemple t1 et t2.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

## Opérateurs unaires

## Traduction en SQL

## Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

## Autres ordres SQL

## le modèle

Entités-  
AssociationsNormalisation  
d'une relation

- Les variables de  $n$ -uplets vont être utiles lorsque l'on cherche à raisonner sur plusieurs  $n$ -uplets d'une même relation et à les comparer.
- L'idée est de prendre deux  $n$ -uplets dans une relation (autrement dit un  $n$ -uplet de  $R \times R$ ) et de sélectionner les couples de  $n$ -uplets satisfaisant certaines contraintes.

**Exemple : Quels acteurs ont la même adresse ?**

```
SELECT Star1.Nom, Star2.Nom FROM Acteur Star1, Acteur Star2  
WHERE Star1.adresse = Star2.adresse
```

Ici Star1 et Star2 sont des variables pour les  $n$ -uplets. On aurait pu nommé ces variables différemment par exemple t1 et t2.

- ❶ Problème 1 : on obtient des couples dont les deux valeurs sont identiques.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- Les variables de  $n$ -uplets vont être utiles lorsque l'on cherche à raisonner sur plusieurs  $n$ -uplets d'une même relation et à les comparer.
- L'idée est de prendre deux  $n$ -uplets dans une relation (autrement dit un  $n$ -uplet de  $R \times R$ ) et de sélectionner les couples de  $n$ -uplets satisfaisant certaines contraintes.

### Exemple : Quels acteurs ont la même adresse ?

```
SELECT Star1.Nom, Star2.Nom FROM Acteur Star1, Acteur Star2
```

```
WHERE Star1.adresse = Star2.adresse AND Star1.nom <> Star2.nom
```

Ici Star1 et Star2 sont des variables pour les  $n$ -uplets. On aurait pu nommé ces variables différemment par exemple t1 et t2.

- ❶ Problème 1 : on obtient des couples dont les deux valeurs sont identiques.  $\Rightarrow$  Star1.nom <> Star2.nom

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

## le modèle

Entités-  
AssociationsNormalisation  
d'une relation

- Les variables de  $n$ -uplets vont être utiles lorsque l'on cherche à raisonner sur plusieurs  $n$ -uplets d'une même relation et à les comparer.
- L'idée est de prendre deux  $n$ -uplets dans une relation (autrement dit un  $n$ -uplet de  $R \times R$ ) et de sélectionner les couples de  $n$ -uplets satisfaisant certaines contraintes.

**Exemple : Quels acteurs ont la même adresse ?**

```
SELECT Star1.Nom, Star2.Nom FROM Acteur Star1, Acteur Star2
```

```
WHERE Star1.adresse = Star2.adresse AND Star1.nom <> Star2.nom
```

Ici Star1 et Star2 sont des variables pour les  $n$ -uplets. On aurait pu nommé ces variables différemment par exemple t1 et t2.

- ❶ Problème 1 : on obtient des couples dont les deux valeurs sont identiques.  $\Rightarrow$  Star1.nom <> Star2.nom
- ❷ Problème 2 : on a les couples en double !

- Les variables de  $n$ -uplets vont être utiles lorsque l'on cherche à raisonner sur plusieurs  $n$ -uplets d'une même relation et à les comparer.
- L'idée est de prendre deux  $n$ -uplets dans une relation (autrement dit un  $n$ -uplet de  $R \times R$ ) et de sélectionner les couples de  $n$ -uplets satisfaisant certaines contraintes.

### Exemple : Quels acteurs ont la même adresse ?

```
SELECT Star1.Nom, Star2.Nom FROM Acteur Star1, Acteur Star2
```

```
WHERE Star1.adresse = Star2.adresse AND Star1.nom < Star2.nom
```

Ici Star1 et Star2 sont des variables pour les  $n$ -uplets. On aurait pu nommé ces variables différemment par exemple t1 et t2.

- 1 Problème 1 : on obtient des couples dont les deux valeurs sont identiques.  $\Rightarrow$  Star1.nom  $\langle \rangle$  Star2.nom
- 2 Problème 2 : on a les couples en double!  
 $\Rightarrow$  Star1.nom < Star2.nom

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

## Opérateur ANY $\equiv$ SOME

Quels sont les titres et dates des films dont le titre a été réutilisés par la suite pour au moins un autre film ?

```
SELECT Titre, date FROM Film t1
```

```
WHERE date < ANY
```

```
(SELECT date FROM Film WHERE Titre=t1.Titre)
```

- On sélectionne les titre et date des  $n$ -uplets
- chaque  $n$ -uplet sera appelé provisoirement  $t1$
- pour lesquels la date est strictement antérieure à une date quelconque (**ANY**) de la table obtenue en sélectionnant toutes les dates associées à un film dont le titre est égal au titre du  $n$ -uplet  $t1$ .

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

## Opérateur ALL

Quels acteurs n'ont joué dans aucun film paru avant 2000 ?

```
SELECT nomActeur FROM Joue t1
```

```
WHERE t1.nomActeur <> ALL
```

```
(SELECT nomActeur FROM Joue WHERE FilmDate<=2000)
```

On sélectionne les noms d'acteurs qui sont différents de tous les noms d'acteurs (**ALL**) obtenus en sélectionnant les noms d'acteurs ayant joué dans un film dont la date est antérieure à 2000.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

## Rappel :

- l'implémentation des relations dans SQL est basée sur des multi-ensembles (dans un multi-ensemble, chaque élément peut apparaître plusieurs fois contrairement au cas d'un ensemble).

L'utilisation de **DISTINCT** permet alors de supprimer les doublons.

**Exemple.** Quel est l'ensemble des producteurs de films dans lesquels Matt Damon joue ?

```
SELECT DISTINCT Nom FROM Producteur, Film, Joue
WHERE IdProd = Id AND
      Titre = TitreFilm AND
      date = FilmDate AND
      nomActeur = "Matt Damon";
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

**Exemple.** A combien s'élèvent les gains de Matt Damon ?

```
SELECT SUM(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

**Exemple.** A combien s'élèvent les gains de Matt Damon ?

```
SELECT SUM(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

**Exemple.** Quel est la moyenne des cachets de Matt Damon ?

```
SELECT AVG(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

**Exemple.** A combien s'élèvent les gains de Matt Damon ?

```
SELECT SUM(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

**Exemple.** Quel est la moyenne des cachets de Matt Damon ?

```
SELECT AVG(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

**Exemple.** Quel est le cachet minimal que Matt Damon a accepté ?

```
SELECT MIN(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

**Exemple.** A combien s'élèvent les gains de Matt Damon ?

```
SELECT SUM(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

**Exemple.** Quel est la moyenne des cachets de Matt Damon ?

```
SELECT AVG(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

**Exemple.** Quel est le cachet minimal que Matt Damon a accepté ?

```
SELECT MIN(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

**Exemple.** Quel est le cachet maximal de Matt Damon ?

```
SELECT MAX(Paye) FROM Joue  
WHERE nomActeur='Matt Damon';
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

## Exemple.

- 1 nombre de valeurs non nulles dans une colonne :  
`SELECT COUNT(nomActeur) FROM joue`
- 2 nombre de valeurs distinctes non nulles :  
`SELECT COUNT(DISTINCT nomActeur) FROM joue`
- 3 nombre total de lignes (même s'il y a des valeurs nulles) :  
`SELECT COUNT(*) FROM joue`

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures  
Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Après la clause **WHERE** , l'opérateur **GROUP BY** appliqué à un attribut A permet de partitionner les n-uplets selon la valeur de l'attribut A.

**Exemple.** On veut, film par film, la liste des acteurs.

```
SELECT TitreFilm, nomActeur FROM Joueur  
GROUP BY TitreFilm;
```

**Exercice 1.** On veut, par acteur, la somme total des cachets obtenus.

**Exercice 2.** Quels sont les acteurs qui jouent dans le plus de films ?  
Le résultat sera trié par nombre de films décroissant.

**Exercice 3.** Quels sont les acteurs qui ont gagné le plus ? Le résultat  
sera trié par total des cachets décroissant.

**Exercice 4.** Trier les films selon la somme des cachets par ordre  
décroissant.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures  
Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Après la clause **WHERE** , l'opérateur **GROUP BY** appliqué à un attribut A permet de partitionner les n-uplets selon la valeur de l'attribut A.

**Exemple.** On veut, film par film, la liste des acteurs.

```
SELECT TitreFilm, nomActeur FROM Joue  
GROUP BY TitreFilm;
```

**Exercice 1.** On veut, par acteur, la somme total des cachets obtenus.

```
SELECT SUM(Paye) FROM Joue GROUP BY nomActeur;
```

**Exercice 2.** Quels sont les acteurs qui jouent dans le plus de films ?  
Le résultat sera trié par nombre de films décroissant.

**Exercice 3.** Quels sont les acteurs qui ont gagné le plus ? Le résultat  
sera trié par total des cachets décroissant.

**Exercice 4.** Trier les films selon la somme des cachets par ordre  
décroissant.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures  
Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Après la clause **WHERE** , l'opérateur **GROUP BY** appliqué à un attribut A permet de partitionner les n-uplets selon la valeur de l'attribut A.

**Exemple.** On veut, film par film, la liste des acteurs.

```
SELECT TitreFilm, nomActeur FROM Joue  
GROUP BY TitreFilm;
```

**Exercice 1.** On veut, par acteur, la somme total des cachets obtenus.

```
SELECT SUM(Paye) FROM Joue GROUP BY nomActeur;
```

**Exercice 2.** Quels sont les acteurs qui jouent dans le plus de films ?  
Le résultat sera trié par nombre de films décroissant.

```
SELECT nomActeur, COUNT(*) AS nbFilms FROM Joue  
GROUP BY nomActeur ORDER BY nbFilms DESC;
```

**Exercice 3.** Quels sont les acteurs qui ont gagné le plus ? Le résultat sera trié par total des cachets décroissant.

**Exercice 4.** Trier les films selon la somme des cachets par ordre décroissant.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes  
Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Après la clause **WHERE** , l'opérateur **GROUP BY** appliqué à un attribut A permet de partitionner les n-uplets selon la valeur de l'attribut A.

**Exemple.** On veut, film par film, la liste des acteurs.

```
SELECT TitreFilm, nomActeur FROM Joue  
GROUP BY TitreFilm;
```

**Exercice 1.** On veut, par acteur, la somme total des cachets obtenus.

```
SELECT SUM(Paye) FROM Joue GROUP BY nomActeur;
```

**Exercice 2.** Quels sont les acteurs qui jouent dans le plus de films ?  
Le résultat sera trié par nombre de films décroissant.

```
SELECT nomActeur, COUNT(*) AS nbFilms FROM Joue  
GROUP BY nomActeur ORDER BY nbFilms DESC;
```

**Exercice 3.** Quels sont les acteurs qui ont gagné le plus ? Le résultat sera trié par total des cachets décroissant.

```
SELECT nomActeur, SUM(Paye) AS total FROM Joue  
GROUP BY nomActeur ORDER BY total DESC;
```

**Exercice 4.** Trier les films selon la somme des cachets par ordre décroissant.

Après la clause **WHERE** , l'opérateur **GROUP BY** appliqué à un attribut A permet de partitionner les n-uplets selon la valeur de l'attribut A.

**Exemple.** On veut, film par film, la liste des acteurs.

```
SELECT TitreFilm, nomActeur FROM Joue  
GROUP BY TitreFilm;
```

**Exercice 1.** On veut, par acteur, la somme total des cachets obtenus.

```
SELECT SUM(Paye) FROM Joue GROUP BY nomActeur;
```

**Exercice 2.** Quels sont les acteurs qui jouent dans le plus de films ?  
Le résultat sera trié par nombre de films décroissant.

```
SELECT nomActeur, COUNT(*) AS nbFilms FROM Joue  
GROUP BY nomActeur ORDER BY nbFilms DESC;
```

**Exercice 3.** Quels sont les acteurs qui ont gagné le plus ? Le résultat sera trié par total des cachets décroissant.

```
SELECT nomActeur, SUM(Paye) AS total FROM Joue  
GROUP BY nomActeur ORDER BY total DESC;
```

**Exercice 4.** Trier les films selon la somme des cachets par ordre décroissant.

```
SELECT TitreFilm, SUM( Paye ) AS cachets FROM Joue  
GROUP BY TitreFilm  
ORDER BY cachets DESC;
```

- la clause **HAVING** sélectionne les **groupes** qui satisfont une condition.
- Cette condition porte non pas sur un  $n$ -uplet mais sur l'ensemble des  $n$ -uplets d'un groupe.
- La condition du **HAVING** compare le résultat d'une fonction d'agrégation portant sur un attribut qui ne fait pas partie de la clause **GROUP BY** :

**Exemple.** Quels sont les films pour lesquels la somme des cachets dépassent 3000000 ?

```
SELECT TitreFilm FROM Joue GROUP BY TitreFilm
HAVING SUM( Paye ) >=3000000
ORDER BY SUM( Paye ) DESC
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

- la clause **HAVING** sélectionne les groupes qui satisfont une condition.
- Cette condition porte non pas sur un  $n$ -uplet mais sur l'ensemble des  $n$ -uplets d'un groupe.
- La condition du **HAVING** compare le résultat d'une fonction d'agrégation portant sur un attribut qui ne fait pas partie de la clause **GROUP BY** :

**Exercice.** Quels sont les films dans lesquels Matt Damon joue et pour lesquels la somme des cachets dépassent 3000000 ?

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

- la clause **HAVING** sélectionne les **groupes** qui satisfont une condition.
- Cette condition porte non pas sur un  $n$ -uplet mais sur l'ensemble des  $n$ -uplets d'un groupe.
- La condition du **HAVING** compare le résultat d'une fonction d'agrégation portant sur un attribut qui ne fait pas partie de la clause **GROUP BY** :

**Exercice.** Quels sont les films dans lesquels Matt Damon joue et pour lesquels la somme des cachets dépassent 3000000 ?

```
(SELECT DISTINCT TitreFilm FROM Joue  
WHERE nomActeur='Matt Damon')
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

- la clause **HAVING** sélectionne les **groupes** qui satisfont une condition.
- Cette condition porte non pas sur un  $n$ -uplet mais sur l'ensemble des  $n$ -uplets d'un groupe.
- La condition du **HAVING** compare le résultat d'une fonction d'agrégation portant sur un attribut qui ne fait pas partie de la clause **GROUP BY** :

**Exercice.** Quels sont les films dans lesquels Matt Damon joue et pour lesquels la somme des cachets dépassent 3000000 ?

```
SELECT TitreFilm FROM Joue WHERE TitreFilm IN
    (SELECT DISTINCT TitreFilm FROM Joue
     WHERE nomActeur='Matt Damon')
GROUP BY TitreFilm
HAVING SUM( Paye ) >=3000000
ORDER BY SUM( Paye ) DESC
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

## le modèle

Entités-  
AssociationsNormalisation  
d'une relation

Soient 2 relations  $R(A_1, A_2, \dots, A_n)$  et  $V(A_1, A_2, \dots, A_f)$  avec  $f < n$  telles que tous les attributs de  $V$  soient attributs de  $R$ . La division, noté  $R/V$  est la relation temporaire de schéma  $(A_{f+1}, A_{f+2}, \dots, A_n)$  dont la population est l'ensemble des  $n$ -uplets de  $R$  tronqués à  $(A_{f+1}, A_{f+2}, \dots, A_n)$  tels que, si on les complète par **n'importe quel**  $n$ -uplet de  $V$ , le  $n$ -uplet obtenu est dans  $R$ .

$$\left\{ \langle a_{f+1}, a_{f+2}, \dots, a_n \rangle \mid \forall \langle a_1, \dots, a_f \rangle \in V, \langle a_1, \dots, a_f, a_{f+1}, \dots, a_n \rangle \in R \right\}$$

**Illustration.** On considère la relation  $R$  ci-contre qui associe à chaque nom de client d'une boutique de vins, les crus et années des vins achetés. On suppose ici que le nom est suffisant pour caractériser un client (autrement dit que `nom` est un identifiant de la table *client*).

| nom    | cru             | année |
|--------|-----------------|-------|
| Dupont | St Emilion      | 2001  |
| Dupont | Nuit St Georges | 2000  |
| Dupont | Nuit St Georges | 2001  |
| Dupont | Aligoté         | 2000  |
| Durant | St Emilion      | 2001  |
| Durant | Aligoté         | 2003  |
| Martin | St Emilion      | 2001  |
| Martin | Nuit St Georges | 2001  |
| Martin | Nuit St Georges | 2000  |

- ① Quels sont les clients qui ont déjà acheté à la fois du St Emilion 2001 et du Nuit St georges 2000 ?

Soit  $V =$ 

| cru             | année |
|-----------------|-------|
| St Emilion      | 2001  |
| Nuit St Georges | 2000  |

,  $R/V =$

| nom    | cru             | année |
|--------|-----------------|-------|
| Dupont | St Emilion      | 2001  |
| Dupont | Nuit St Georges | 2000  |
| Dupont | Nuit St Georges | 2001  |
| Dupont | Aligoté         | 2000  |

|        |                 |      |
|--------|-----------------|------|
| Durant | St Emilion      | 2001 |
| Durant | Aligoté         | 2003 |
| Martin | St Emilion      | 2001 |
| Martin | Nuit St Georges | 2001 |
| Martin | Nuit St Georges | 2000 |

- ① Quels sont les clients qui ont déjà acheté à la fois du St Emilion 2001 et du Nuit St georges 2000 ?

$$\text{Soit } V = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{St Emilion} & 2001 \\ \text{Nuit St Georges} & 2000 \\ \hline \end{array}, R/V = \begin{array}{|c|} \hline \text{nom} \\ \hline \text{Dupont} \\ \text{Martin} \\ \hline \end{array}$$

| nom    | cru             | année |
|--------|-----------------|-------|
| Dupont | St Emilion      | 2001  |
| Dupont | Nuit St Georges | 2000  |
| Dupont | Nuit St Georges | 2001  |
| Dupont | Aligoté         | 2000  |

|        |                 |      |
|--------|-----------------|------|
| Durant | St Emilion      | 2001 |
| Durant | Aligoté         | 2003 |
| Martin | St Emilion      | 2001 |
| Martin | Nuit St Georges | 2001 |
| Martin | Nuit St Georges | 2000 |

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

- ❶ Quels sont les clients qui ont déjà acheté à la fois du St Emilion 2001 et du Nuit St georges 2000 ?

$$\text{Soit } v = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{St Emilion} & 2001 \\ \text{Nuit St Georges} & 2000 \\ \hline \end{array}, \quad R/V = \begin{array}{|c|} \hline \text{nom} \\ \hline \text{Dupont} \\ \text{Martin} \\ \hline \end{array}$$

- ❷ Quels sont les clients qui ont déjà acheté du St Emilion 2001 ?

$$\text{Soit } v' = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{St Emilion} & 2001 \\ \hline \end{array}, \quad R/V' =$$

| nom    | cru             | année |
|--------|-----------------|-------|
| Dupont | St Emilion      | 2001  |
| Dupont | Nuit St Georges | 2000  |
| Dupont | Nuit St Georges | 2001  |
| Dupont | Aligoté         | 2000  |

|        |                 |      |
|--------|-----------------|------|
| Durant | St Emilion      | 2001 |
| Durant | Aligoté         | 2003 |
| Martin | St Emilion      | 2001 |
| Martin | Nuit St Georges | 2001 |
| Martin | Nuit St Georges | 2000 |

- ① Quels sont les clients qui ont déjà acheté à la fois du St Emilion 2001 et du Nuit St georges 2000 ?

$$\text{Soit } v = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{St Emilion} & 2001 \\ \text{Nuit St Georges} & 2000 \\ \hline \end{array}, \quad R/V = \begin{array}{|c|} \hline \text{nom} \\ \hline \text{Dupont} \\ \text{Martin} \\ \hline \end{array}$$

- ② Quels sont les clients qui ont déjà acheté du St Emilion 2001 ?

$$\text{Soit } v' = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{St Emilion} & 2001 \\ \hline \end{array}, \quad R/V' = \begin{array}{|c|} \hline \text{nom} \\ \hline \text{Dupont} \\ \text{Durant} \\ \text{Martin} \\ \hline \end{array}$$

| nom    | cru             | année |
|--------|-----------------|-------|
| Dupont | St Emilion      | 2001  |
| Dupont | Nuit St Georges | 2000  |
| Dupont | Nuit St Georges | 2001  |
| Dupont | Aligoté         | 2000  |

|        |                 |      |
|--------|-----------------|------|
| Durant | St Emilion      | 2001 |
| Durant | Aligoté         | 2003 |
| Martin | St Emilion      | 2001 |
| Martin | Nuit St Georges | 2001 |
| Martin | Nuit St Georges | 2000 |

- ❶ Quels sont les clients qui ont déjà acheté à la fois du St Emilion 2001 et du Nuit St georges 2000 ?

$$\text{Soit } v = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{St Emilion} & 2001 \\ \text{Nuit St Georges} & 2000 \\ \hline \end{array}, \quad R/V = \begin{array}{|c|} \hline \text{nom} \\ \hline \text{Dupont} \\ \text{Martin} \\ \hline \end{array}$$

- ❷ Quels sont les clients qui ont déjà acheté du St Emilion 2001 ?

$$\text{Soit } v' = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{St Emilion} & 2001 \\ \hline \end{array}, \quad R/V' = \begin{array}{|c|} \hline \text{nom} \\ \hline \text{Dupont} \\ \text{Durant} \\ \text{Martin} \\ \hline \end{array}$$

- ❸ Quels sont les clients qui ont déjà acheté de l'Aligoté de 2005 ?

$$\text{Soit } v'' = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{Aligoté} & 2005 \\ \hline \end{array}, \quad R/V'' =$$

| nom    | cru             | année |
|--------|-----------------|-------|
| Dupont | St Emilion      | 2001  |
| Dupont | Nuit St Georges | 2000  |
| Dupont | Nuit St Georges | 2001  |
| Dupont | Aligoté         | 2000  |

|        |                 |      |
|--------|-----------------|------|
| Durant | St Emilion      | 2001 |
| Durant | Aligoté         | 2003 |
| Martin | St Emilion      | 2001 |
| Martin | Nuit St Georges | 2001 |
| Martin | Nuit St Georges | 2000 |

- ① Quels sont les clients qui ont déjà acheté à la fois du St Emilion 2001 et du Nuit St georges 2000 ?

$$\text{Soit } v = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{St Emilion} & 2001 \\ \text{Nuit St Georges} & 2000 \\ \hline \end{array}, \quad R/V = \begin{array}{|c|} \hline \text{nom} \\ \hline \text{Dupont} \\ \text{Martin} \\ \hline \end{array}$$

- ② Quels sont les clients qui ont déjà acheté du St Emilion 2001 ?

$$\text{Soit } v' = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{St Emilion} & 2001 \\ \hline \end{array}, \quad R/V' = \begin{array}{|c|} \hline \text{nom} \\ \hline \text{Dupont} \\ \text{Durant} \\ \text{Martin} \\ \hline \end{array}$$

- ③ Quels sont les clients qui ont déjà acheté de l'Aligoté de 2005 ?

$$\text{Soit } v'' = \begin{array}{|c|c|} \hline \text{cru} & \text{année} \\ \hline \text{Aligoté} & 2005 \\ \hline \end{array}, \quad R/V'' = \begin{array}{|c|} \hline \text{nom} \\ \hline \end{array}$$

| nom    | cru             | année |
|--------|-----------------|-------|
| Dupont | St Emilion      | 2001  |
| Dupont | Nuit St Georges | 2000  |
| Dupont | Nuit St Georges | 2001  |
| Dupont | Aligoté         | 2000  |

|        |                 |      |
|--------|-----------------|------|
| Durant | St Emilion      | 2001 |
| Durant | Aligoté         | 2003 |
| Martin | St Emilion      | 2001 |
| Martin | Nuit St Georges | 2001 |
| Martin | Nuit St Georges | 2000 |

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

**Exemple (institut de formation).** Donnez la liste des étudiants qui peuvent s'inscrire au cours de drug design (un étudiant peut s'inscrire à un cours s'il a validé les cours prérequis).

Rappel du schéma :

$$\left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \quad \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \quad \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

- 1 on recherche tous les prérequis pour le cours de drug design :
- 2 on calcule la liste des étudiants qui peuvent s'inscrire au cours :

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

**Exemple (institut de formation).** Donnez la liste des étudiants qui peuvent s'inscrire au cours de drug design (un étudiant peut s'inscrire à un cours s'il a validé les cours prérequis).

Rappel du schéma :

$$\left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \quad \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \quad \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

- 1 on recherche tous les prérequis pour le cours de drug design :

$$H_1 = \alpha[\text{nomCpre:nomC}] \pi[\text{nomCpre}] \left( \sigma[\text{nomC}=\text{"drug design"}] \text{prerequis} \right)$$

- 2 on calcule la liste des étudiants qui peuvent s'inscrire au cours :

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
AssociationsNormalisation  
d'une relation

**Exemple (institut de formation).** Donnez la liste des étudiants qui peuvent s'inscrire au cours de drug design (un étudiant peut s'inscrire à un cours s'il a validé les cours prérequis).

Rappel du schéma :

$$\left\{ \begin{array}{l} \text{personne}(\text{numPers}, \text{nom}, \text{adresse}), \\ \text{personnePrenom}(\text{numPers}, \text{numPrenom}, \text{prenom}), \\ \text{etudiant}(\text{numPers}, \text{numEtud}, \text{dateN}), \\ \text{etudiantEtude}(\text{numEtud}, \text{annee}, \text{diplome}), \\ \text{enseignant}(\text{numPers}, \text{tel}, \text{status}, \text{numEns}, \text{banque}, \text{agence}, \text{compte}), \\ \text{cours}(\text{nomC}, \text{cycle}, \text{numEns}), \quad \text{obtenu}(\text{numEtud}, \text{nomC}, \text{note}), \\ \text{inscrit}(\text{numEtud}, \text{nomC}), \quad \text{prerequis}(\text{nomC}, \text{nomCpre}) \end{array} \right\}$$

- 1 on recherche tous les prérequis pour le cours de drug design :

$$H_1 = \alpha[\text{nomCpre:nomC}] \pi[\text{nomCpre}] \left( \sigma[\text{nomC}=\text{"drug design"}] \text{prerequis} \right)$$

- 2 on calcule la liste des étudiants qui peuvent s'inscrire au cours :

$$\left( \pi[\text{numEtud}, \text{nomC}] \text{obtenus} / H_1 \right)$$

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation



Il n'existe pas en SQL d'équivalent direct à la division.

Cependant il est toujours possible de trouver une autre solution, notamment par l'intermédiaire

- des opérations de calcul et
- de regroupement.

Donnez la liste des étudiants qui peuvent s'inscrire au cours de drug design (un étudiant peut s'inscrire à un cours s'il a validé les cours prérequis).

- 1 on recherche les cours prérequis pour le cours de drug design :
- 2 on recherche LE NOMBRE de cours prérequis pour le cours de drug design :
- 3 on recherche les étudiants qui ont un nombre de cours prérequis pour "drug design" égal à celui calculé.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation



Il n'existe pas en SQL d'équivalent direct à la division.

Cependant il est toujours possible de trouver une autre solution, notamment par l'intermédiaire

- des opérations de calcul et
- de regroupement.

Donnez la liste des étudiants qui peuvent s'inscrire au cours de drug design (un étudiant peut s'inscrire à un cours s'il a validé les cours prérequis).

- 1 on recherche les cours prérequis pour le cours de drug design :  

```
SELECT nomCpre FROM prerequis  
WHERE nomC = "drug design"
```
- 2 on recherche LE NOMBRE de cours prérequis pour le cours de drug design :
- 3 on recherche les étudiants qui ont un nombre de cours prérequis pour "drug design" égal à celui calculé.



Il n'existe pas en SQL d'équivalent direct à la division.

Cependant il est toujours possible de trouver une autre solution, notamment par l'intermédiaire

- des opérations de calcul et
- de regroupement.

Donnez la liste des étudiants qui peuvent s'inscrire au cours de drug design (un étudiant peut s'inscrire à un cours s'il a validé les cours prérequis).

- 1 on recherche les cours prérequis pour le cours de drug design :

```
SELECT nomCpre FROM prerequis  
WHERE nomC = "drug design"
```

- 2 on recherche LE NOMBRE de cours prérequis pour le cours de drug design :

```
SELECT COUNT(DISTINCT nomCpre) FROM prerequis  
WHERE nomC = "drug design"
```

- 3 on recherche les étudiants qui ont un nombre de cours prérequis pour "drug design" égal à celui calculé.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation



Il n'existe pas en SQL d'équivalent direct à la division.

Cependant il est toujours possible de trouver une autre solution, notamment par l'intermédiaire

- des opérations de calcul et
- de regroupement.

Donnez la liste des étudiants qui peuvent s'inscrire au cours de drug design (un étudiant peut s'inscrire à un cours s'il a validé les cours prérequis).

- 1 on recherche les cours prérequis pour le cours de drug design :

```
SELECT nomCpre FROM prerequis  
WHERE nomC = "drug design"
```

- 2 on recherche LE NOMBRE de cours prérequis pour le cours de drug design :

```
SELECT COUNT(DISTINCT nomCpre) FROM prerequis  
WHERE nomC = "drug design"
```

- 3 on recherche les étudiants qui ont un nombre de cours prérequis pour "drug design" égal à celui calculé.

```
SELECT numEtud FROM obtenu  
WHERE nomC IN (SELECT nomCpre FROM prerequis  
WHERE nomC = "drug design")  
  
GROUP BY numEtud  
HAVING COUNT(DISTINCT nomC) =  
(SELECT COUNT(DISTINCT nomCpre) FROM prerequis  
WHERE nomC = "drug design")
```

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

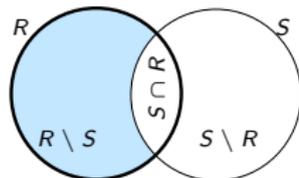
Entités-  
AssociationsNormalisation  
d'une relation

- Intersection :

$$R \cap S = R \setminus (R \setminus S) = S \setminus (S \setminus R)$$

ou alors

$$R \cap S = (R \cup S) \setminus ((R \setminus S) \cup (S \setminus R))$$



- Jointure naturelle : Soient  $R(X, Y)$  et  $S(Y, Z)$ .

$$R \bowtie S = \pi[X, Y, Z] \sigma[Y = Y'] (R \times \alpha[Y : Y'] S)$$

- Théta Jointure : Soient  $R(X, Y)$  et  $S(U, V)$ .

$$R \bowtie_p S = \sigma[p](R \times S)$$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

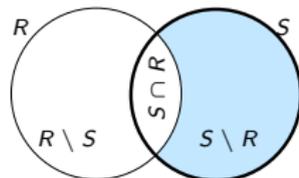
Entités-  
AssociationsNormalisation  
d'une relation

- Intersection :

$$R \cap S = R \setminus (R \setminus S) = S \setminus (S \setminus R)$$

ou alors

$$R \cap S = (R \cup S) \setminus ((R \setminus S) \cup (S \setminus R))$$



- Jointure naturelle : Soient  $R(X, Y)$  et  $S(Y, Z)$ .

$$R \bowtie S = \pi[X, Y, Z] \sigma[Y = Y'] (R \times \alpha[Y : Y'] S)$$

- Théta Jointure : Soient  $R(X, Y)$  et  $S(U, V)$ .

$$R \bowtie_p S = \sigma[p](R \times S)$$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

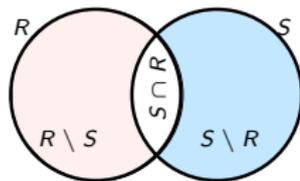
Entités-  
AssociationsNormalisation  
d'une relation

- Intersection :

$$R \cap S = R \setminus (R \setminus S) = S \setminus (S \setminus R)$$

ou alors

$$R \cap S = (R \cup S) \setminus ((R \setminus S) \cup (S \setminus R))$$



- Jointure naturelle : Soient  $R(X, Y)$  et  $S(Y, Z)$ .

$$R \bowtie S = \pi[X, Y, Z] \sigma[Y = Y'] (R \times \alpha[Y : Y'] S)$$

- Théta Jointure : Soient  $R(X, Y)$  et  $S(U, V)$ .

$$R \bowtie_p S = \sigma[p](R \times S)$$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQL

## Opérateurs unaires

## Traduction en SQL

## Jointures

Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

## Autres ordres SQL

## le modèle

Entités-  
AssociationsNormalisation  
d'une relation

- Division : Soient  $R(X, Y)$  et  $S(Y)$ .

$$R / S = \pi[X]R \setminus \underbrace{\pi[X]((\pi[X]R \times S) \setminus R)}_1$$

Explications :

- $\pi[X]R$  : toutes les valeurs de  $X$  dans  $R$
- $(\pi[X]R \times S)$  : une valeur de  $X$  (présente dans  $R$ ) suivie d'un  $n$ -uplet de  $S$
- $((\pi[X]R \times S) \setminus R)$  : les  $n$ -uplets  $(X, Y)$  n'appartenant pas à  $R$  avec une valeur de  $X$  présente dans  $R$
- $\pi[X]((\pi[X]R \times S) \setminus R)$  : les valeurs de  $X$  (présentes dans  $R$ ) pour lesquelles il existe un  $n$ -uplet de  $S$  tel que  $(X, Y)$  n'appartient pas à  $R$ .

Le temps de réponse d'une requête dépend essentiellement du nombre d'instructions élémentaires nécessaires au calcul de la réponse. L'ordre de grandeur de ce nombre d'instructions élémentaires est appelé *complexité*. Cette complexité s'exprime en fonction de la taille des données en entrée.

- Sélection  $\sigma[p]R$  : on doit balayer tous les  $n$ -uplets de la relation  $R$ .  
 $\Rightarrow$  la complexité est en  $O(|R|)$ .  
La taille de la relation résultat est donc comprise entre 0 et  $|R|$ .
- Projection  $\pi[A_i, A_k \dots]R$  : il faut parcourir toute la table.  $\Rightarrow$  la complexité est en  $O(|R|)$ .  
S'il y a une autre opération à faire simultanément, comme par exemple  $\pi[nom]\sigma[p]Personne$ , alors la projection est faite en même temps que la sélection.
- Jointure naturelle, théta jointure, produit cartésien : il faut parcourir tous les couples de  $n$ -uplets.  $\Rightarrow$  complexité en  $O(|R| \times |S|)$ .  
Dans le cas du produit cartésien, la taille du résultat est exactement  $|R| \times |S|$ .  
 $\Rightarrow$  on a intérêt à faire des jointures sur des petites relations. Pour optimiser le temps de réponse à une requête, il vaut mieux faire d'abord des sélections (diminuer la taille des relations) avant de faire des jointures.  
**Exemple.** Quels sont les cours assurés par Muller ?

Le temps de réponse d'une requête dépend essentiellement du nombre d'instructions élémentaires nécessaires au calcul de la réponse. L'ordre de grandeur de ce nombre d'instructions élémentaires est appelé *complexité*. Cette complexité s'exprime en fonction de la taille des données en entrée.

- Sélection  $\sigma[p]R$  : on doit balayer tous les  $n$ -uplets de la relation  $R$ .  
 $\Rightarrow$  la complexité est en  $O(|R|)$ .  
La taille de la relation résultat est donc comprise entre 0 et  $|R|$ .
- Projection  $\pi[A_i, A_k \dots]R$  : il faut parcourir toute la table.  $\Rightarrow$  la complexité est en  $O(|R|)$ .  
S'il y a une autre opération à faire simultanément, comme par exemple  $\pi[nom]\sigma[p]Personne$ , alors la projection est faite en même temps que la sélection.
- Jointure naturelle, théta jointure, produit cartésien : il faut parcourir tous les couples de  $n$ -uplets.  $\Rightarrow$  complexité en  $O(|R| \times |S|)$ .  
Dans le cas du produit cartésien, la taille du résultat est exactement  $|R| \times |S|$ .  
 $\Rightarrow$  on a intérêt à faire des jointures sur des petites relations. Pour optimiser le temps de réponse à une requête, il vaut mieux faire d'abord des sélections (diminuer la taille des relations) avant de faire des jointures.  
**Exemple.** Quels sont les cours assurés par Muller ?

*Personne*  $\bowtie$  *Enseignant*  $\bowtie$  *Cours*

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

Le temps de réponse d'une requête dépend essentiellement du nombre d'instructions élémentaires nécessaires au calcul de la réponse. L'ordre de grandeur de ce nombre d'instructions élémentaires est appelé *complexité*. Cette complexité s'exprime en fonction de la taille des données en entrée.

- Sélection  $\sigma[p]R$  : on doit balayer tous les  $n$ -uplets de la relation  $R$ .  
 $\Rightarrow$  la complexité est en  $O(|R|)$ .  
La taille de la relation résultat est donc comprise entre 0 et  $|R|$ .
- Projection  $\pi[A_i, A_k \dots]R$  : il faut parcourir toute la table.  $\Rightarrow$  la complexité est en  $O(|R|)$ .  
S'il y a une autre opération à faire simultanément, comme par exemple  $\pi[nom]\sigma[p]Personne$ , alors la projection est faite en même temps que la sélection.
- Jointure naturelle, théta jointure, produit cartésien : il faut parcourir tous les couples de  $n$ -uplets.  $\Rightarrow$  complexité en  $O(|R| \times |S|)$ .  
Dans le cas du produit cartésien, la taille du résultat est exactement  $|R| \times |S|$ .  
 $\Rightarrow$  on a intérêt à faire des jointures sur des petites relations. Pour optimiser le temps de réponse à une requête, il vaut mieux faire d'abord des sélections (diminuer la taille des relations) avant de faire des jointures.  
**Exemple.** Quels sont les cours assurés par Muller ?

$\sigma[nom = Muller](Personne \bowtie Enseignant \bowtie Cours)$

Le temps de réponse d'une requête dépend essentiellement du nombre d'instructions élémentaires nécessaires au calcul de la réponse. L'ordre de grandeur de ce nombre d'instructions élémentaires est appelé *complexité*. Cette complexité s'exprime en fonction de la taille des données en entrée.

- Sélection  $\sigma[p]R$  : on doit balayer tous les  $n$ -uplets de la relation  $R$ .  
 $\Rightarrow$  la complexité est en  $O(|R|)$ .  
La taille de la relation résultat est donc comprise entre 0 et  $|R|$ .
- Projection  $\pi[A_i, A_k \dots]R$  : il faut parcourir toute la table.  $\Rightarrow$  la complexité est en  $O(|R|)$ .  
S'il y a une autre opération à faire simultanément, comme par exemple  $\pi[nom]\sigma[p]Personne$ , alors la projection est faite en même temps que la sélection.
- Jointure naturelle, théta jointure, produit cartésien : il faut parcourir tous les couples de  $n$ -uplets.  $\Rightarrow$  complexité en  $O(|R| \times |S|)$ .  
Dans le cas du produit cartésien, la taille du résultat est exactement  $|R| \times |S|$ .  
 $\Rightarrow$  on a intérêt à faire des jointures sur des petites relations. Pour optimiser le temps de réponse à une requête, il vaut mieux faire d'abord des sélections (diminuer la taille des relations) avant de faire des jointures.  
**Exemple.** Quels sont les cours assurés par Muller ?

$\sigma[nom = Muller](Personne \bowtie Enseignant \bowtie Cours)$

$\sigma[nom = Muller]Personne$

Le temps de réponse d'une requête dépend essentiellement du nombre d'instructions élémentaires nécessaires au calcul de la réponse. L'ordre de grandeur de ce nombre d'instructions élémentaires est appelé *complexité*. Cette complexité s'exprime en fonction de la taille des données en entrée.

- Sélection  $\sigma[p]R$  : on doit balayer tous les  $n$ -uplets de la relation  $R$ .  
 $\Rightarrow$  la complexité est en  $O(|R|)$ .  
La taille de la relation résultat est donc comprise entre 0 et  $|R|$ .
- Projection  $\pi[A_i, A_k \dots]R$  : il faut parcourir toute la table.  $\Rightarrow$  la complexité est en  $O(|R|)$ .  
S'il y a une autre opération à faire simultanément, comme par exemple  $\pi[nom]\sigma[p]Personne$ , alors la projection est faite en même temps que la sélection.
- Jointure naturelle, théta jointure, produit cartésien : il faut parcourir tous les couples de  $n$ -uplets.  $\Rightarrow$  complexité en  $O(|R| \times |S|)$ .  
Dans le cas du produit cartésien, la taille du résultat est exactement  $|R| \times |S|$ .  
 $\Rightarrow$  on a intérêt à faire des jointures sur des petites relations. Pour optimiser le temps de réponse à une requête, il vaut mieux faire d'abord des sélections (diminuer la taille des relations) avant de faire des jointures.  
**Exemple.** Quels sont les cours assurés par Muller ?

$\sigma[nom = Muller](Personne \bowtie Enseignant \bowtie Cours)$

$(\sigma[nom = Muller]Personne) \bowtie Enseignant \bowtie Cours$

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- ① Cascades de projections :

$$\begin{cases} \text{Si } \{A_1, \dots, A_j\} \subseteq \{B_1, \dots, B_l\} \text{ alors} \\ \pi[A_1, \dots, A_j] \left( \pi[B_1, \dots, B_l] R \right) = \pi[A_1, \dots, A_j] R \end{cases}$$

- ② Cascades de sélections :

$$\sigma[p_1] \left( \sigma[p_2] R \right) = \sigma[p_2] \left( \sigma[p_1] R \right) = \sigma[p_1 \wedge p_2] R$$

- ③ Propriétés des jointures et produits.

• Commutativité :  $R \bowtie_p S = S \bowtie_p R$      $R \bowtie_p S = S \bowtie_p R$      $R \times S = S \times R$

- Associativité :

- Jointure naturelle. Si  $R$  et  $S$  ont au moins un attribut en commun et si  $S$  et  $T$  ont au moins un attribut en commun :

$$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$$

- Théta Jointure. si les relations  $R$ ,  $S$  et  $T$  n'ont aucun attribut en commun, et si le prédicat  $p_1$  (resp.  $p_2$ ) ne porte que sur des attributs de  $R$  et  $S$  (resp.  $S$  et  $T$ ) :

$$(R \bowtie_{p_1} S) \bowtie_{p_2} T = R \bowtie_{p_1} (S \bowtie_{p_2} T)$$

- Produit cartésien. si  $R$ ,  $S$  et  $T$  n'ont aucun attribut en commun :

$$(R \times S) \times T = R \times (S \times T)$$

- ④ Les opérateurs union et intersection sont commutatifs et associatifs. La différence n'est ni commutative ni associative.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentielsAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
JointuresOpérateurs  
ensemblistesOpérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
AssociationsNormalisation  
d'une relation

Illustration d'une démarche à utiliser pour écrire des requêtes.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \quad \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

*Quelle est la liste des noms des producteurs de films dans lesquels joue au moins un acteur qui habite Beverly Hill ?*

- 1 Identifier les relations utiles pour exprimer la requête.  
Producteur (pour le nom des producteurs), Acteurs (pour les acteurs qui habitent Beverly Hill), Joue et Film (pour faire le lien)
- 2 Recopier les schémas de ces relations et indiquer dans ces schémas
  - les attributs qui font partie du résultat de la requête (nom des producteurs),
  - les conditions portant sur les attributs (adresse="Beverly Hill")
  - les liens entre les relations (jointures) :
 

```
Film.idProd = Producteur.id
Joue.nomActeur = Acteur.nomActeur
Joue.TitreFilm = Film.Titre   Joue.date = Film.date
```

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Illustration d'une démarche à utiliser pour écrire des requêtes.

$$\mathcal{S} = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \quad \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

*Quelle est la liste des noms des producteurs de films dans lesquels joue au moins un acteur qui habite Beverly Hill ?*

Producteur (id, nom, adresse)

Film (idProd, titre, date, longueur, couleur, nomstudio)

Joue( titreFilm, date, nomActeur, paye)

Acteur( nom, adresse, genre, ...)

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentiell

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

Illustration d'une démarche à utiliser pour écrire des requêtes.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{Nom}, \text{Adresse}), \quad \text{Producteur}(\text{Nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

*Quelle est la liste des noms des producteurs de films dans lesquels joue au moins un acteur qui habite Beverly Hill ?*

Producteur (id, nom, adresse)  
?

Film (idProd, titre, date, longueur, couleur, nomstudio)

Joue( titreFilm, date, nomActeur, paye)

Acteur( nom, adresse, genre, ...)

|  
="Beverly Hill"



Introduction aux BDR

Jean-Paul Comet,  
Nadia Abchiche-Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs ensemblistes

Opérateurs hors de l'Alg.

Autres ordres SQL

le modèle

Entités-Associations

Normalisation d'une relation

Producteur (id, nom, adresse)  
?

Film (idProd, titre, date, longueur, couleur, nomstudio)

Joue( titreFilm, date, nomActeur, paye)

Acteur( nom, adresse, genre, ...)

= "Beverly Hill"

3 Traduire cette figure en expression algébrique :

- faire les sélections avec les conditions sur les attributs
- faire les jointures (naturelle ou theta) selon les liens entre relations (il y aura une jointure par lien)
- projeter sur les attributs recherchés.

$$\pi[nom] \left( \text{Producteur} \bowtie \text{Film} \bowtie \text{Joue} \bowtie \left( \sigma \left[ \begin{array}{l} \text{adresse} \\ = \\ \text{Beverly} \end{array} \right] \text{Acteur} \right) \right)$$

Conditions for joins:

- Producteur  $\bowtie$  Film:  $\left[ \begin{array}{l} id \\ = \\ idProd \end{array} \right]$
- Film  $\bowtie$  Joue:  $\left[ \begin{array}{l} \text{film.date} = \text{joue.date} \\ \wedge \\ \text{titre} = \text{TitreFilm} \end{array} \right]$
- Joue  $\bowtie$  Acteur:  $\left[ \begin{array}{l} \text{nomActeur} \\ = \\ \text{acteur.nom} \end{array} \right]$

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLOpérateurs unaires  
Traduction en SQL  
Jointures  
Opérateurs  
ensemblistesOpérateurs hors de  
l'Alg.  
Autres ordres SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

$$\pi[\text{nom}] \left( \text{Producteur} \bowtie_{\begin{bmatrix} id \\ = \\ idProd \end{bmatrix}} \text{Film} \bowtie_{\begin{bmatrix} film.date = joue.date \\ \wedge \\ titre = TitreFilm \end{bmatrix}} \text{Joue} \bowtie_{\begin{bmatrix} nomActeur \\ = \\ acteur.nom \end{bmatrix}} \left( \sigma_{\begin{bmatrix} adresse \\ = \\ Beverly \end{bmatrix}} \text{Acteur} \right) \right)$$

#### ④ Traduire expression algébrique en SQL :

```
SELECT nom FROM
```

```
Producteur JOIN
```

```
Film ON Id=IdProd JOIN
```

```
Joue ON film.date=joue.date AND titre=titreFilm JOIN
```

```
Acteur ON nomActeur=acteur.nom
```

```
WHERE adresse="Beverly"
```

Quelle est la liste des acteurs

- *qui habitent au même endroit qu'un acteur ayant joué dans Green Book et*
- *qui ont touché au moins un cachet de plus de 500000 euros ?*

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL  
Jointures  
Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.  
Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

## Quelle est la liste des acteurs

- *qui habitent au même endroit qu'un acteur ayant joué dans Green Book et*
- *qui ont touché au moins un cachet de plus de 500000 euros ?*

Les adresses  
des acteurs qui  
ont joué dans  
"Green Book"

Joue (nomFilm, nomActeur, paye, FilmDate)

Acteur (nom, adresse, genre, dateN)

Les adresses  
des acteurs  
qui ont touché  
un cachet > 500000

Joue (nomFilm, nomActeur, paye, FilmDate)

Acteur (nom, adresse, genre, dateN)

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

## Quelle est la liste des acteurs

- *qui habitent au même endroit qu'un acteur ayant joué dans Green Book et*
- *qui ont touché au moins un cachet de plus de 500000 euros ?*

Les adresses  
des acteurs qui  
ont joué dans  
"Green Book"

Les adresses  
des acteurs  
qui ont touché  
un cachet > 500000

= "Green Book"

Joue (nomFilm, nomActeur, paye, FilmDate)

Acteur (nom, adresse, genre, dateN)

Acteur (nom, adresse, genre, dateN)

Joue (nomFilm, nomActeur, paye, FilmDate)

>500000

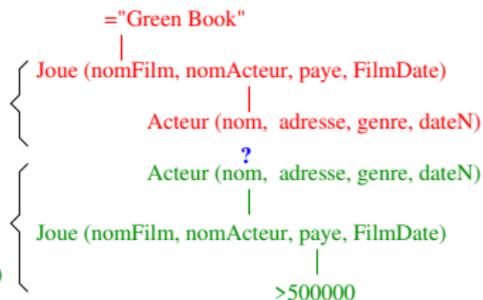
?

## Quelle est la liste des acteurs

- *qui habitent au même endroit qu'un acteur ayant joué dans Green Book et*
- *qui ont touché au moins un cachet de plus de 500000 euros ?*

Les adresses des acteurs qui ont joué dans "Green Book"

Les adresses des acteurs qui ont touché un cachet > 500000



Introduction aux BDR

Jean-Paul Comet,  
Nadia Abchiche-Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs ensemblistes

Opérateurs hors de l'Alg.

Autres ordres SQL

le modèle Entités-Associations

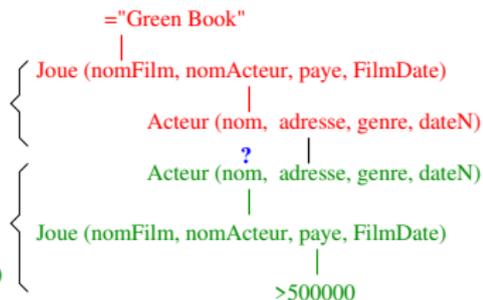
Normalisation d'une relation

## Quelle est la liste des acteurs

- *qui habitent au même endroit qu'un acteur ayant joué dans Green Book et*
- *qui ont touché au moins un cachet de plus de 500000 euros ?*

Les adresses des acteurs qui ont joué dans "Green Book"

Les adresses des acteurs qui ont touché un cachet > 500000



Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

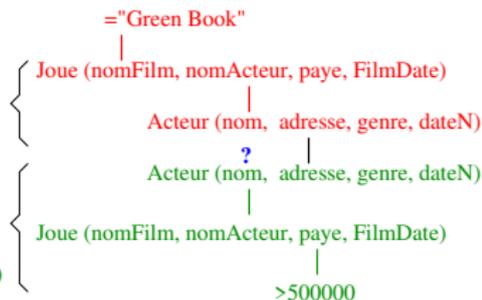
Normalisation  
d'une relation

## Quelle est la liste des acteurs

- *qui habitent au même endroit qu'un acteur ayant joué dans Green Book et*
- *qui ont touché au moins un cachet de plus de 500000 euros ?*

Les adresses des acteurs qui ont joué dans "Green Book"

Les adresses des acteurs qui ont touché un cachet > 500000



$$\pi[nom] \left( \left( \left( \text{Acteur} \bowtie \left[ \begin{matrix} nom = \\ nomActeur \end{matrix} \right] \sigma[paye > 500\text{keuros}] \text{Joue} \right) \bowtie [adr=adresse] \right) \left( \alpha[adresse : adr] \pi[adresse] \left( \text{Acteur} \bowtie \left[ \begin{matrix} nom = \\ nomActeur \end{matrix} \right] \sigma \left[ \begin{matrix} TitreFilm = \\ GreenBook \end{matrix} \right] \text{Joue} \right) \right) \right)$$

**Rq.** les requêtes avec un "pour tout" ou "aucun" ne se représentent que mal graphiquement.

## Introduction aux BDR

Jean-Paul Comet,  
Nadia Abchiche-Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

Opérateurs unaires  
Traduction en SQL  
Jointures

Opérateurs ensemblistes  
Opérateurs hors de l'Alg.

Autres ordres SQL

le modèle  
Entités-Associations

Normalisation d'une relation

$$\pi[\text{nom}] \left( \left( \left( \text{Acteur} \bowtie \left[ \begin{array}{l} \text{nom} = \\ \text{nomActeur} \end{array} \right] \sigma[\text{paye} > 500\text{keuros}] \text{Joue} \right) \bowtie [\text{adr} = \text{adresse}] \right) \bowtie \left( \alpha[\text{adresse} : \text{adr}] \pi[\text{adresse}] \left( \text{Acteur} \bowtie \left[ \begin{array}{l} \text{nom} = \\ \text{nomActeur} \end{array} \right] \sigma \left[ \begin{array}{l} \text{TitreFilm} = \\ \text{GreenBook} \end{array} \right] \text{Joue} \right) \right) \right)$$

**SELECT nom FROM**

```
(
  SELECT * FROM Acteur JOIN Joue ON nom=nomActeur WHERE paye>500000
)
```

**JOIN**

```
(
  SELECT adresse AS adr FROM Acteur JOIN Joue ON nom=nomActeur
  WHERE TitreFilm = "Green Book"
)
```

**ON adr = adresse**

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

```
ALTER TABLE nom_table alter_specification [, alter_specification] ...
```

```
alter_specification:
```

```
    ADD [COLUMN] column_definition [FIRST | AFTER col_name ]  
    | ADD [COLUMN] (column_definition,...)  
    | ADD [CONSTRAINT [symbol]]  
        PRIMARY KEY (index_col_name,...)  
    | ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}  
    | CHANGE [COLUMN] old_col_name column_definition  
    | MODIFY [COLUMN] column_definition  
    | DROP col_name  
    | DROP PRIMARY KEY  
    | RENAME [TO] new_tbl_name
```

---

```
DROP TABLE tbl_name [, tbl_name] ...
```

---

```
INSERT INTO tbl_name [(nomCol1 [, nomCol2]..)]  
VALUES (valeur1[,valeur2]...);
```

- Si on ne précise pas les noms des colonnes, les valeurs seront insérées dans les colonnes correspondant à l'ordre de leur création dans la table.
- A l'inverse, si la liste des colonnes est incomplète, les autres colonnes seront remplies avec la valeur `NULL`.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

Opérateurs unaires  
Traduction en SQL

Jointures

Opérateurs  
ensemblistes

Opérateurs hors de  
l'Alg.

Autres ordres SQL

le modèle

Entités-  
Associations

Normalisation  
d'une relation

```
UPDATE nomTable  
SET nomCol1=expression1 [,nomCol2=expression2]...  
WHERE condition;
```

On ne peut modifier qu'une seule table à la fois. Cependant, la condition **WHERE** peut faire référence à d'autres tables.

```
UPDATE joue SET paye=paye*1.1  
WHERE nomActeur IN (SELECT nom FROM acteur  
                     WHERE genre='F')
```

---

```
DELETE FROM nomTable WHERE condition;
```

L'instruction **DELETE** ne touche pas à la structure de la table. Si la condition est omise, tous les  $n$ -uplets sont effacés. La condition **WHERE** peut faire référence à plusieurs tables.