

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

NoSQL

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.

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

NoSQL

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

NoSQL

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

NoSQL

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.

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

NoSQL

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.

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

NoSQL

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** :

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

NoSQL

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

NoSQL

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

NoSQL

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
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
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

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

NoSQL

Normalisation d'une relation

- 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

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

NoSQL

Normalisation d'une relation

- 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
Dupont
Martin

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

NoSQL

Normalisation d'une relation

- 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
Dupont
Martin

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

Soit $V' =$

cru	année
St Emilion	2001

, $R/V' =$

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

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

NoSQL

Normalisation d'une relation

- 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
Dupont
Martin

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

Soit $V' =$

cru	année
St Emilion	2001

, $R/V' =$

nom
Dupont
Durant
Martin

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

NoSQL

Normalisation d'une relation

- 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
Dupont
Martin

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

Soit $V' =$

cru	année
St Emilion	2001

, $R/V' =$

nom
Dupont
Durant
Martin

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

Soit $V'' =$

cru	année
Aligoté	2005

, $R/V'' =$

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

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

NoSQL

Normalisation d'une relation

- 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
Dupont
Martin

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

Soit $V' =$

cru	année
St Emilion	2001

, $R/V' =$

nom
Dupont
Durant
Martin

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

Soit $V'' =$

cru	année
Aligoté	2005

, $R/V'' =$

nom

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

NoSQL

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 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

NoSQL

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 :

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

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

Introduction
aux 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

NoSQL

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 :

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

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 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-
Associations

NoSQL

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

NoSQL

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.

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

NoSQL

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.

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-
Associations

NoSQL

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
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

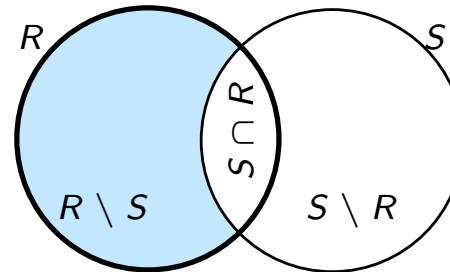
Normalisation
d'une relation

● 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-
Associations

NoSQL

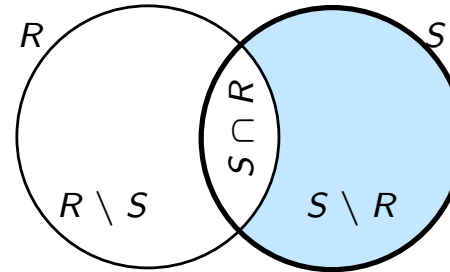
Normalisation
d'une relation

● 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-
Associations

NoSQL

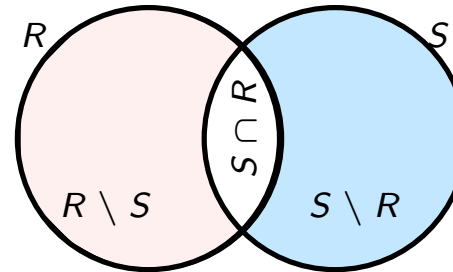
Normalisation
d'une relation

● 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-
Associations

NoSQL

Normalisation
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 .

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

NoSQL

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 ?

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

NoSQL

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 $\pi[A_i, A_k...]R$: 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 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-
Associations

NoSQL

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 $\pi[A_i, A_k...]R$: 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 ?

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

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

NoSQL

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 ?

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

$\sigma[nom = Muller]Personne$

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-
Associations

NoSQL

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 $\pi[A_i, A_k...]R$: 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

NoSQL

Normalisation d'une relation

1 Cascades de projections :

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

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

- 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 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

NoSQL

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 ?

- 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) :
 $\text{Film.idProd} = \text{Producteur.id}$
 $\text{Joue.nomActeur} = \text{Acteur.nomActeur}$
 $\text{Joue.TitreFilm} = \text{Film.Titre} \quad \text{Joue.date} = \text{Film.date}$

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

NoSQL

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 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-
Associations

NoSQL

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, ...)
|
="Beverly Hill"

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-
Associations

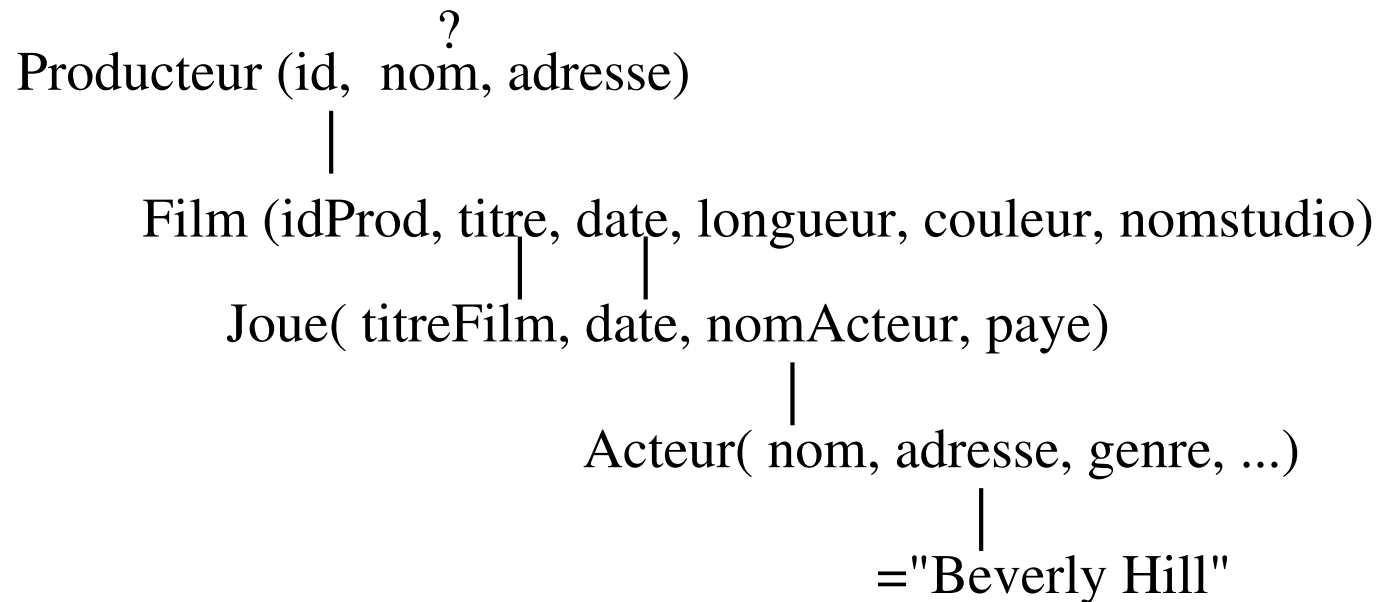
NoSQL

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 ?



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-
Associations

NoSQL

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.

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

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-
Associations

NoSQL

Normalisation
d'une relation

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

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

NoSQL

Normalisation
d'une relation

Quelle est la liste des acteurs

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

NoSQL

Normalisation
d'une relation

- *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

Joue (nomFilm, nomActeur, paye, FilmDate)

Acteur (nom, adresse, genre, dateN)

Acteur (nom, adresse, genre, dateN)

Joue (nomFilm, nomActeur, paye, FilmDate)

Exemple plus compliqué.

Quelle est la liste des acteurs

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

NoSQL

Normalisation
d'une relation

- *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

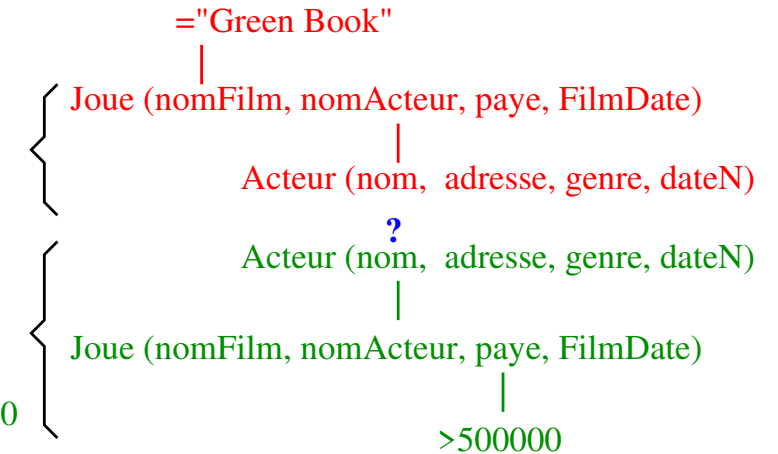
Exemple plus compliqué.

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

NoSQL

Normalisation
d'une relation

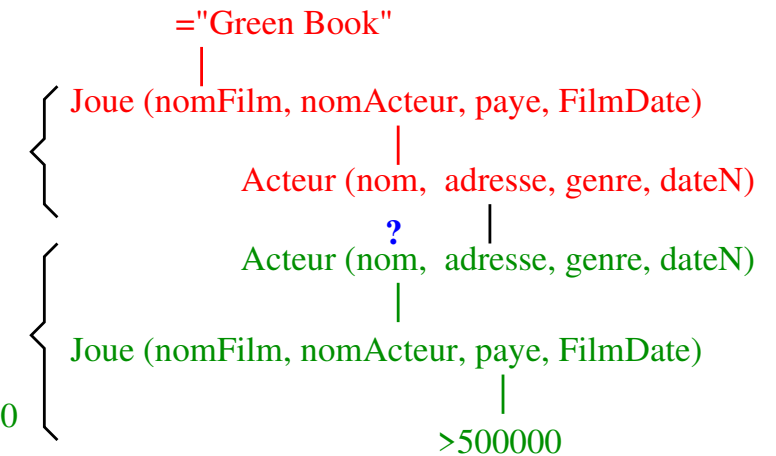
Exemple plus compliqué.

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

NoSQL

Normalisation
d'une relation

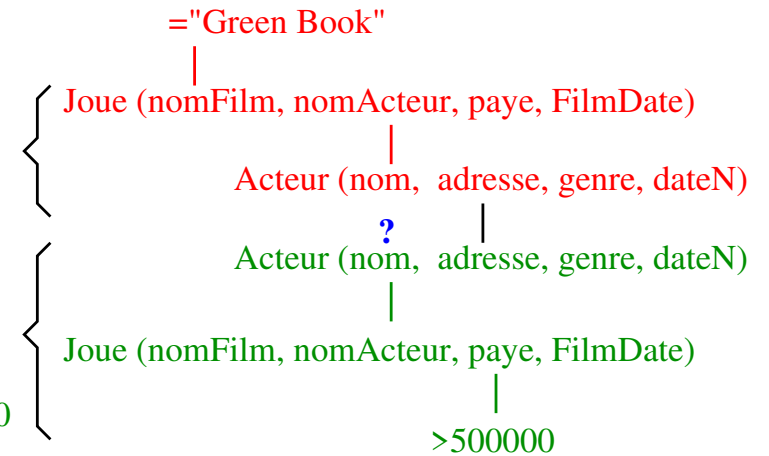
Exemple plus compliqué.

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



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

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

NoSQL

Normalisation d'une relation

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

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

NoSQL

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

NoSQL

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.