

Jointures et vues

L3Pro SCT – Bases de données et programmation

Mathieu Sassolas

IUT de Sénart Fontainebleau
Département Informatique

Année 2015-2016
Cours 3



Jointures et
vues

M. Sassolas
L3Pro SCT – M7

Cours 3

Jointures

Vues

TD/TP

1 Jointures

2 Vues

3 TD/TP

2 / 18

Jointures et
vues

M. Sassolas
L3Pro SCT – M7

Cours 3

Jointures

Vues

TD/TP

1 Jointures

2 Vues

3 TD/TP

3 / 18

Jointures et
vues

M. Sassolas
L3Pro SCT – M7

Cours 3

Jointures

Vues

TD/TP

Énoncé

« Trouver le nom et prénom de tous les employés
avec le nom et prénom de leur supérieur. »

On avait :

- ▶ utilisé deux « copies » de la table employees
- ▶ comparé la colonne reportsto de la première copie avec la colonne employeenumber de la seconde.
- ▶ ce qui donnait une requête du genre :

```
SELECT sub.firstname,sub.lastname,  
       sup.firstname AS boss_firstname,  
       sup.lastname AS boss_lastname  
FROM employees AS sub, employees AS sup  
WHERE sub.reportsto = sup.employeenumber;
```

4 / 18

Énoncé

Trouver les informations sur toutes les hospitalisations du patient 195..., y compris le nom et prénom du médecin responsable.

- ▶ On va sélectionner sur les tables hospitalisations et medecin en même temps.
- ▶ On va comparer la colonne responsable avec l'identifiant du médecin.

```
SELECT hos.*,m.prenom AS prenom_med,m.nom AS nom_med
FROM hospitalisations AS hos, medecin AS m
WHERE hos.responsable = m.id_medecin
AND patient='195...';
```

- ▶ On a **joint** les deux tables (ou copies) en utilisant l'égalité de valeur dans deux colonnes.
- ▶ On a sélectionné depuis cette nouvelle table (avec plus de colonnes).
- ▶ On appelle cette table la **jointure** entre les tables.
- ▶ Il y a une syntaxe prévue pour cela.

```
SELECT * FROM table1 JOIN table2
ON table1.colA = table2.colB;
```

- ▶ On a le droit d'utiliser des conditions complexes (voir autre chose que des égalités, mais c'est très rare) :

```
SELECT * FROM table1 JOIN table2
ON table1.colA = table2.colB
AND table1.colC >= table2.colD;
```
- ▶ Si la colonne sur laquelle se fait la jointure porte le même nom (il y aura fusion des colonnes) :

```
SELECT * FROM table1 JOIN table2
ON table1.col = table2.col;
```

```
SELECT * FROM table1 JOIN table2 USING (col);
```
- ▶ Jointure sur toutes les colonnes portant le même nom, c'est une **jointure naturelle** :

```
SELECT * FROM table1 NATURAL JOIN table2;
```

Il est donc important que deux colonnes qui ont le même nom représentent la même chose.

medicaments

nom_m	fournisseur_m
Xanax	1651
Depakine	241
Valium	42
Plavix	241
Viagra	1651
Voltaren	315

fournisseurs

id_f	nom_f	ville_f
1651	Pfizer	New York
241	Sanofi	Paris
42	Roche	Bâle
315	Novartis	Bâle
471	GSK	Londres

```
SELECT * FROM medicaments JOIN fournisseurs
ON id\_f = fournisseur\_m;
```

nom_m	fournisseur_m	id_f	nom_f	ville_f
Depakine	241	241	Sanofi	Paris
Valium	42	42	Roche	Bâle
Plavix	241	241	Sanofi	Paris
Voltaren	315	315	Novartis	Bâle
Viagra	1651	1651	Pfizer	New York
Xanax	1651	1651	Pfizer	New York

Exemple du début, avec des jointures

Jointures et vues

M. Sassolas
L3Pro SCT - M7
Cours 3

Jointures

Vues
TD/TP

Les employés et leur supérieur

```
SELECT sub.firstname,sub.lastname,
       sup.firstname AS boss_firstname,
       sup.lastname AS boss_lastname
FROM employees AS sub JOIN employees AS sup
ON sub.reportsto=sup.employee_number;
```

L'hôpital

```
SELECT hos.*,m.prenom AS prenom_med,
       m.nom AS nom_med
FROM hospitalisations AS hos JOIN medecin AS m
ON hos.responsable = m.id_medecin
WHERE AND patient='195...';
```

9 / 18

Jointures externes

Jointures et vues

M. Sassolas
L3Pro SCT - M7
Cours 3

Jointures

Vues
TD/TP

Cas des lignes qui ne correspondent à rien

- ▶ La jointure ne garde que les lignes sur lesquelles la jointure peut se faire.
- ▶ Dans nos exemples : le « President » n'apparaît plus car il n'a pas de supérieur, les employés n'étant supérieurs de personne n'apparaissent pas à gauche ; le laboratoire GSK n'apparaît plus car il ne fournit pas de médicaments.

Syntaxe des jointures externes

```
SELECT * FROM table1 LEFT JOIN table 2 ON <...>;
SELECT * FROM table1 RIGHT JOIN table 2 ON <...>;
SELECT * FROM table1 FULL JOIN table 2 ON <...>;
```

Les lignes de la table de gauche (resp. de droite, les deux) apparaîtront toute, avec une valeur null dans les colonnes de l'autre table.

10 / 18

Exemple simple

Jointures et vues

M. Sassolas
L3Pro SCT - M7
Cours 3

Jointures

Vues
TD/TP

medicaments

nom_m	fournisseur_m
Xanax	1651
Depakine	241
Valium	42
Plavix	241
Viagra	1651
Voltaren	315

fournisseurs

id_f	nom_f	ville_f
1651	Pfizer	New York
241	Sanofi	Paris
42	Roche	Bâle
315	Novartis	Bâle
471	GSK	Londres

```
SELECT * FROM medicaments RIGHT JOIN fournisseurs
ON id\_f = fournisseur\_m;
```

nom_m	fournisseur_m	id_f	nom_f	ville_f
Depakine	241	241	Sanofi	Paris
Valium	42	42	Roche	Bâle
Plavix	241	241	Sanofi	Paris
Voltaren	315	315	Novartis	Bâle
Viagra	1651	1651	Pfizer	New York
Xanax	1651	1651	Pfizer	New York
		471	GSK	Londres

11 / 18

Plan de la séance

Jointures et vues

M. Sassolas
L3Pro SCT - M7
Cours 3

Jointures

Vues

TD/TP

1 Jointures

2 Vues

3 TD/TP

12 / 18

- ▶ Pour utiliser une jointure comme une seule table (si on y accède souvent).
- ▶ Pour donner une vue partielle des données : masquer des colonnes ou des lignes.
- ▶ Pour ajouter des colonnes avec des valeurs calculées (pas de détails dans ce cours).

Création

```
CREATE VIEW <nom_vue> AS SELECT ...;
CREATE VIEW <nom_vue> WITH (security_barrier)
AS SELECT ...;
```

↔ S'utilise ensuite comme une table :

```
SELECT * FROM <nom_vue>;
SELECT * FROM <nom_vue> JOIN <table> ON ...;
DROP VIEW <nom_vue>;
```

L'option security_barrier

- ▶ Applique le SELECT complètement quand la vue est utilisée.
- ▶ Empêche des attaques mais est plus lent.
- ▶ Utile quand la vue est créée pour cacher des lignes (pour des raisons de droit d'accès).

```
CREATE VIEW medic_labos AS
SELECT nom_m AS nom,
       nom_f AS fournisseur,
       ville_f as ville_fournisseur
FROM medicaments RIGHT JOIN fournisseurs
ON id_f = fournisseur_m;
```

nom	fournisseur	ville_fournisseur
Depakine	Sanofi	Paris
Valium	Roche	Bâle
Plavix	Sanofi	Paris
Voltaren	Novartis	Bâle
Viagra	Pfizer	New York
Xanax	Pfizer	New York
	GSK	Londres

```
CREATE VIEW hospi2014anon AS
SELECT entree,sortie,lit,medecin_responsable
FROM hospitalisations
WHERE where extract(year from entree)=2014;
```

```
CREATE VIEW hospiDrDupont WITH (security_barrier) AS
SELECT entree,sortie,lit,patient
FROM hospitalisations
WHERE where medecin_responsable = 1421588;
```

```
CREATE VIEW hospi_en_cours WITH (security_barrier) AS
SELECT h.*, patients.*
FROM hospitalisations h JOIN patients
ON h.patient = patient.num_secu
WHERE where sortie IS NULL;
```

1 Jointures

2 Vues

3 TD/TP

↳ C'est l'heure du TP ◀