

Bases de Données (II)

Requêtes

D.Malka

MPSI 2018-2019

Sommaire

- 1 Clé
 - Clé primaire
 - Clé étrangère
- 2 Jointure
 - Produit cartésien
 - Jointure (symétrique)
- 3 Fonctions agrégat
 - Agrégats
 - Conditions sur les agrégats
 - Fonctions sur les agrégats

Sommaire

1 Clé

- Clé primaire
- Clé étrangère

2 Jointure

- Produit cartésien
- Jointure (symétrique)

3 Fonctions agrégat

- Agrégats
- Conditions sur les agrégats
- Fonctions sur les agrégats

Clé

Clé d'une relation

Soit $R(S)$ est une relation de S . Une clé est un ensemble d'attributs $K = \{A_i, \dots, A_j\}$ du schéma relationnel S tel que pour tous tuples $t_1, t_2 \in R(S)$, $t_1(K) = t_2(K) \Rightarrow t_1 = t_2$.

Autrement dit, une clé identifie de façon unique chaque tuple t de la relation $R(S)$.

- ▶ **L'ensemble des attributs constituant un schéma relationnel forme une clé pour ce schéma ! Mais ce n'est peut-être pas la meilleur clé !**
- ▶ **Contrainte d'intégrité du SGBD : tout schéma relationnel doit posséder au moins une clé.**

Clé

Exemple de clé – Relation livre

numero	auteur	titre
"78 BA"	"Barbusse"	"Le Feu"
"156 ES"	"Eschyle"	"Les Perses"
"18 CL"	"Clausewitz"	"De la Guerre"
"155 ES"	"Eschyle"	"L'Orestie"

auteur n'est pas une clé.

auteur, titre est une clé. numero aussi.

Clé primaire

Clé primaire d'une relation

Une clé primaire d'une relation $R(S)$ est une clé contenant un nombre minimal d'attributs.

Exemple de clé primaire – Relation livre :

numero	auteur	titre
"78 BA"	"Barbusse"	"Le Feu"
"156 ES"	"Eschyle"	"Les Perses"
"18 CL"	"Clausewitz"	"De la Guerre"
"156 ES"	"Eschyle"	"L'Orestie"

auteur, titre est une clé mais pas primaire.

numero est une clé primaire. titre est une clé primaire.

Il est préférable que la clé primaire n'est aucune valeur sémantique.

Clé étrangère

Clé étrangère d'une relation

L'expression $S_1[A_1, \dots, A_n] \subset S_2[B_1, \dots, B_n]$ est une clé étrangère de S_1 référençant S_2 si B_1, \dots, B_n est une clé de S_2 et si pour tout $t_1 \in R_1(S)$ il existe $t_2 \in R_2(S)$, tel que $t_1[A_1, \dots, A_n] = t_2[B_1, \dots, B_n]$

- ▶ **Une clé étrangère indique un lien entre deux relations**
- ▶ **Une clé étrangère définit une contrainte d'intégrité référentielle :**
 - lors d'une insertion dans la table référençant, la valeur des attributs doit exister dans la table référencée
 - lors d'une suppression dans la table référencée, les tuples référençant doivent être supprimés

Exemple de clé étrangère

- ▶ relation `livre(numero, auteur, titre)`
- ▶ relation `emprunt(id_emprunt, emprunteur, num)`

`num` est une clé étrangère de `emprunt` référençant l'attribut `numero` de la relation `livre`.

Sommaire

- 1 Clé
 - Clé primaire
 - Clé étrangère
- 2 Jointure
 - Produit cartésien
 - Jointure (symétrique)
- 3 Fonctions agrégat
 - Agrégats
 - Conditions sur les agrégats
 - Fonctions sur les agrégats

Produit cartésien

Opération d'origine ensembliste.

Produit cartésien

Soit $R(S)$ et $R'(S')$ deux relations de schémas disjoints, leur produit cartésien est :

$$R \times R' = \{(a_1, \dots, a_n, a'_1, \dots, a'_m) \mid (a_1, \dots, a_n) \in R \text{ et } (a'_1, \dots, a'_m) \in R'\}$$

Son schéma est :

$$S \cup S' = (A_1, \dots, A_n, A'_1, \dots, A'_m)$$

où $S = (A_1, \dots, A_n)$ et $S' = (A'_1, \dots, A'_m)$.

SQL multi-enssembliste : le résultat d'un produit cartésien peut contenir des doublons à moins d'utiliser le mot clé **DISTINCT** dans la requête.

Produit cartésien

Exemple de produit cartésien

- ▶ Relation livre :

numero	auteur	titre
"78 BA"	"Barbusse"	"Le Feu"
"156 ES"	"Eschyle"	"Les Perses"

- ▶ Relation emprunt :

id_emprunt	emprunteur	num
12	Nono	"78 BA"
59	Roux	"156 ES"

- ▶ Résultat du produit cartésien :

numero	auteur	titre	num	emprunteur
"78 BA"	"Barbusse"	"Le Feu"	"78 BA"	Nono
"156 ES"	"Eschyle"	"Les Perses"	"156 ES"	Roux
"78 BA"	"Barbusse"	"Le Feu"	"156 ES"	Roux
"156 ES"	"Eschyle"	"Les Perses"	"78 BA"	Nono

Produit cartésien

En SQL

```
SELECT numero, auteur, titre, num, emprunteur # Projection  
FROM livre, emprunt # Produit cartésien
```

Jointure (symétrique)

Jointure (symétrique)

Soient $R(S)$ et $R'(S')$ deux relations de schémas disjoints, et $A \in S$, $A' \in S'$ tels que $dom(A) = dom(A')$, on note :

$$R[A = A']R' = \{e \in R \times R' \mid e.A = e.A'\} = \sigma_{A=A'}(R \times R')$$

est appelé la jointure symétrique de S et S' selon (A, A') .

- ▶ En théorie, les tuples de la table résultant de la jointure contient les attributs A et A'
- ▶ En pratique, on réalise le plus souvent une jointure entre une clé primaire et une clé étrangère : les attributs A et A' sont alors redondants.
- ▶ Dans cas, on peut réaliser une projection pour éliminer cette redondance.

Jointure

Exemple de jointure symétrique

- ▶ Relation livre :

numero	auteur	titre
"78 BA"	"Barbusse"	"Le Feu"
"156 ES"	"Eschyle"	"Les Perses"

- ▶ Relation emprunt :

id_emprunt	emprunteur	num
12	"78 BA"	Nono
59	"156 ES"	Roux

- ▶ Résultat de la jointure : $livre[numero = num]emprunt$

numero	auteur	titre	num	emprunteur
"78 BA"	"Barbusse"	"Le Feu"	"78 BA"	Nono
"156 ES"	"Eschyle"	"Les Perses"	"156 ES"	Roux

Les tuples du produit cartésien qui n'avaient pas de sens ont disparu !

Jointure

Littéralement, en SQL

```
SELECT numero, auteur, titre, num, emprunteur # Projection  
FROM livre, emprunt # Produit cartésien  
WHERE num=numero AND ... # Restriction
```

Mauvaise idée car alors la complexité mémoire et temporelle est $O(n_1 \times n_2)$ avec n_1 et n_2 les tailles des deux tables.

Jointure

En SQL avec JOIN...ON

```
SELECT numero, auteur, titre, num, emprunteur # Projection  
FROM livre  
JOIN emprunt ON num=numero # Jointure symétrique  
WHERE ... # Restriction
```

La complexité de la requête est alors $O(n\log(n))$.

Sommaire

- 1 Clé
 - Clé primaire
 - Clé étrangère
- 2 Jointure
 - Produit cartésien
 - Jointure (symétrique)
- 3 Fonctions agrégat
 - Agrégats
 - Conditions sur les agrégats
 - Fonctions sur les agrégats

Fonctions sur les agrégats

Un exemple

Comment exprimer la requête suivante :

Renvoyer le salaire moyen des employés de chaque département de l'entreprise comptant plus de 10 salariés.

La condition *plus de 10 salariés* ne porte pas sur un tuple mais sur des groupes de tuples (des partitions). La restriction via le mot clé `WHERE` n'est donc pas adaptée à cette situation.

De même, *le salaire moyen* est calculé sur des groupes de tuples.

Agrégats

Agrégats

Partitions de la table renvoyée par une requête : **GROUP BY**.

N'a d'intérêt que si combiné avec une fonction d'agrégation.

EN SQL

```
SELECT A1,...  
FROM R1,R2...  
WHERE ...  
GROUP BY A1
```

Conditions sur les agrégats

Condition sur les Agrégats

Condition appliquée à chaque partition (et non pas l'ensemble des tuples) d'une table :

HAVING

EN SQL

```
SELECT A1,...  
FROM R1,R2...  
WHERE ...  
GROUP BY A1  
HAVING <condition sur les partitions>
```

Fonction sur les agrégats

Fonction sur les agrégats

Fonctions appliquées à chaque partition (et non pas à l'ensemble des tuples) :

- ▶ MAX
- ▶ MIN
- ▶ COUNT (nombre de tuples dans la partition)
- ▶ AVG (moyenne des valeurs de l'attributs de la partition)
- ▶ SUM (somme des valeurs de l'attributs de la partition)

Fonctions sur les agrégats

Utilisation des fonctions d'agrégation

- ▶ `employe(nss, nom, prénom, salaire, departement)`
- ▶ `departement(nom, localisation, responsable)`

```
SELECT e.departement, AVG(salaire)
FROM employe e
GROUP BY departement
HAVING COUNT(departement)>10
```

Cette requête renvoie une table du type :

d.nom	AVG(salaire)
"Achat"	3020
"Marketing"	3250
"Production"	1865

Les départements de moins de 10 employés ne figurent pas dans le résultat.