

Bases de Données (II)

Requêtes

D.Malka

MPSI 2019-2020

Sommaire

1 Requête

2 Opérations sur le modèle relationnel

3 Jointure

- Produit cartésien
- Jointure (symétrique)

4 Fonctions agrégat

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

Sommaire

1 Requête

2 Opérations sur le modèle relationnel

3 Jointure

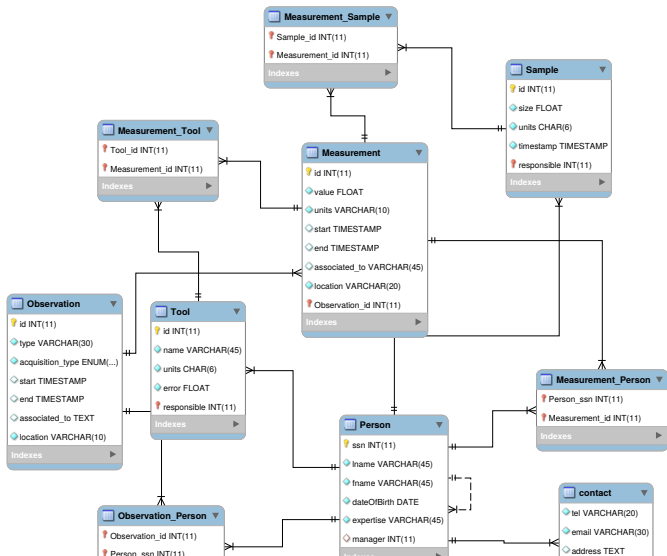
- Produit cartésien
- Jointure (symétrique)

4 Fonctions agrégat

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

Schéma de base de données

Une base de données est constituée d'un ensemble de table liées les unes aux autres.



Requête

Une requête extrait ou construit de l'information à partir d'une base de donnée.

Exemple :

Selecting all the measurements performed by 'Stephane LEBLOND'.

```
1 SELECT p.lname,m.id,m.start,m.end
2 FROM Person p JOIN Measurement_Person mp ON mp.Person_ssn=p.ssn
3     JOIN Measurement m ON mp.Measurement_id=m.id
4 WHERE lname='LEBLOND'
```

Requête

```

1 SELECT p.lname,m.id,m.start,m.end
2 FROM Person p JOIN Measurement_Person mp ON mp.Person_ssn=p.ssn
3      JOIN Measurement m ON mp.Measurement_id=m.id
4 WHERE lname='LEBLOND'
```

Résultat :

lname	id	start	end
LEBLOND	1	"2019-02-23 08 :00 :00"	"2019-02-23 08 :01 :00"
LEBLOND	2	"2019-02-23 12 :00 :00"	"2019-02-23 12 :01 :00"
LEBLOND	5	"2019-02-24 08 :00 :00"	"2019-02-24 08 :01 :00"
LEBLOND	6	"2019-02-24 12 :00 :00"	"2019-02-24 12 :01 :00"
LEBLOND	9	"2019-02-25 08 :00 :00"	"2019-02-25 08 :01 :00"
LEBLOND	10	"2019-02-25 12 :00 :00"	"2019-02-25 12 :01 :00"

Le résultat d'une requête est une table.

Sommaire

1 Requête

2 Opérations sur le modèle relationnel

3 Jointure

- Produit cartésien
- Jointure (symétrique)

4 Fonctions agrégat

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

Opérateurs ensemblistes

Opérateurs ensemblistes

- ▶ Union : $R_1(S) \cup R_2(S)$
- ▶ Intersection : $R_1(S) \cap R_2(S)$
- ▶ Différence : $R_1(S) - R_2(S)$

En SQL

- ▶ Union : `(SELECT * FROM R1) UNION (SELECT * FROM R2)`
- ▶ Intersection : `(SELECT * FROM R1) INTERSECT (SELECT * FROM R2)`
- ▶ Différence : `(SELECT * FROM R1) EXCEPT (SELECT * FROM R2)`

Opérateurs ensemblistes

Exemple de résultat

Les relations `bibliotheque 1` et `bibliotheque 2` obéissent au même schéma relationnel :

`livre={numéro, auteur, titre}`

Requête : `(SELECT * FROM bibliotheque 1) EXCEPT (SELECT * FROM bibliotheque 2)`

numéro	auteur	titre
"78 BA"	"Barbusse"	"Le Feu"
"156 ES"	"Eschyle"	"Les Perses"
"007 CL"	"Clausewitz"	"De la Guerre"
...
"126 OR"	"Orwell"	"1984"

La table renvoyée contient les livres disponibles uniquement dans la bibliothèque 1.

Projection

Projection

Soit $R(S)$ une relation de S et $X \subset S$. On appelle *projection* de R selon X la relation :

$$\pi_X(R) = \{e(X) \mid e \in R\}$$

Le schéma relationnel de $\pi_X(R)$ est donc X .

En SQL

```
SELECT A1, A4, ... FROM R
```

Projection

Exemple de résultat

Requête : `SELECT nom, salaire FROM employe`

nom	salaire
"Dinateur"	2565
"Cutier"	1500
...	...
"Bouneur"	5260

Restriction

Restriction

Soit $R(S)$ une relation de S , $A \in S$ et $a \in \text{dom}(A)$. On appelle *restriction* de $R(S)$ selon $A = a$ la relation :

$$\sigma_{A=a}(R) = \{e \in R \mid e.A = a\}$$

Le schéma relationnel de $\sigma_{A=a}(R)$ est toujours S .

En SQL

```
SELECT * FROM R WHERE A=a
```

Restriction

Exemple de résultat

Requête : `SELECT * FROM employe WHERE salaire>2000`

nss	nom	prénom	salaire	département
1 75 ...	"Dinateur"	"Laure"	2565	"Informatique"
1 83 ...	"Citron"	"Théo"	2050	"Marketing"
...
2 94 ...	"Bombeur"	"Jean"	5260	"Achat"

Une requête simple

Une requête simple

Renvoyer le nom et le département des employés touchant plus de 2000 euros.

$$\pi_{\text{nom,departement}}(\sigma_{\text{salaire}>2000}(\text{employe}))$$

En SQL

```
SELECT nom, departement FROM employe WHERE salaire>2000
```

Exemple de résultat

nom	département
"Dinateur"	"Informatique"
"Citron"	"Marketing"
...	...
"Bombeur"	"Achat"

Sommaire

1 Requête

2 Opérations sur le modèle relationnel

3 Jointure

- Produit cartésien
- Jointure (symétrique)

4 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 contiennent 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 ce 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	Nono	"78 BA"
59	Roux	"156 ES"

- ▶ 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 Requête

2 Opérations sur le modèle relationnel

3 Jointure

- Produit cartésien
- Jointure (symétrique)

4 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.