420-C42

Langages d'exploitation des bases de données

Partie 8

DQL II

Fonctions d’agrégation et regroupement

DQL II fonctions d'agrégation

- Le langage SQL offre quelques fonctions d'agrégation (fonctions synthétisant un ensemble de données; aussi appelées fonctions de groupe ou fonctions statistiques).
  - COUNT(*) le nombre de ligne (incluant les valeurs nulles)
  - COUNT(*colonne*) le nombre de ligne non nulle sur la colonne indiquée
  - COUNT(DISTINCT *col*) le nombre de ligne distincte non nulle sur la col. indiquée
  - MIN(*colonne*) la valeur minimum
  - MAX(*colonne*) la valeur maximum
  - SUM(*colonne*) la somme des valeurs
  - AVG(*colonne*) la moyenne des valeurs
- Ces fonctions retournent 1 seul tuple (par regroupement).
- À l'exception de COUNT(*), toutes les fonctions d'agrégation ignorent les valeurs nulles.

DQL II fonctions d'agrégation

```
-- retourne le nombre d'employé, le plus petit et le plus grand salaire,
-- la somme et la moyenne des salaires du département des ventes
SELECT COUNT(*) AS "Nombre",
                        MIN(salaire) AS "Salaire minimum",
                        MAX(salaire) AS "Salaire maximum",
                        SUM(salaire) AS "Masse salariale",
                        AVG(salaire) AS "Moyenne des salaires"
        FROM employe
        WHERE departement = (SELECT id
                                FROM departement WHERE nom = 'Ventes');


-- retourne le(s) nom(s) de l'employé ou des employés le(s) mieux payé.
SELECT nom, prenom
    FROM employe
    WHERE salaire = (SELECT MAX(salaire) FROM employe);
```

DQL II fonctions d'agrégation

- PostgreSQL propose plusieurs autres fonctions d'agrégation dont celles-ci :
  - BOOL_AND(*colonne*) ET logique sur toutes les valeurs
  - BOOL_OR(*colonne*) OU logique sur toutes les valeurs
  - STRING_AGG(*colonne*) concaténation de toutes les chaînes de caractères
  - VAR[_POP][_SAMP](*colonne*) variance de la population ou d'un échantillon
  - STDDEV[_POP][_SAMP](*colonne*) écart type de la population ou d'un échantillon

- Attention, dans la clause SELECT, il est impossible de mélanger les colonnes à des fonctions d'agrégation. Par exemple, cette requête est impossible et ne fais pas de sens de toute façon :

```
SELECT nom, SUM(salaire)
        FROM employe
        WHERE ville IN ('Montréal', 'Québec');
```

DQL II

GROUP BY

- La clause SELECT permet, par la clause GROUP BY, d'effectuer des regroupements et d'utiliser les fonctions d'agrégation sur ces derniers.
- Les critères de regroupement peuvent être sur une colonne (un critère) ou plusieurs colonnes (multi critères).
- Dans une requête de regroupement seul ces informations peuvent être directement retournées :
  - le(s) critère(s) de regroupement
  - le résultat des fonctions d'agrégation appliqué aux regroupements.
- La sélection de ligne (clause WHERE) est appliquée **avant** le regroupement.

DQL II

GROUP BY

- SELECT departement,
  COUNT(salaire),
  MIN(salaire),
  MAX(salaire),
  SUM(salaire),
  AVG(salaire)
  FROM employe
  WHERE departement IS NOT NULL
  GROUP BY departement;

DQL II regroupement

- Exemple de requête erronée avec un regroupement sur un critère.

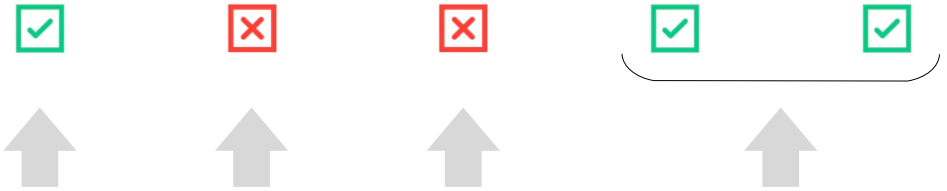
employe

nas	nom	genre	salaire	departement
111	Dupuis	h	20.00	3
222	Lebel	f	25.00	5
333	Lapierre	f	22.00	3
444	Bordeleau	h	18.00	3
555	Pignon	h	20.00	5
666	Sasseur	f	15.00	2
777	Leblanc	h	30.00	5
888	Latendresse	f	25.00	2

```
SELECT departement, nom, salaire, SUM(salaire), MIN(nom)
FROM employe
GROUP BY departement;
```

regroupement par département

departement	nom	salaire	SUM(salaire)	MIN(nom)
3	?	?	60.00	Bordeleau
5	?	?	75.00	Lebel
2	?	?	40.00	Latendresse



critère de regroupement

quel nom choisir?

quel salaire choisir?

fonctions d'agrégation

DQL II regroupement

- Exemple de requête erronée avec un regroupement sur deux critères.

employe

nas	nom	genre	salaire	departement
111	Dupuis	h	20.00	3
222	Lebel	f	25.00	5
333	Lapierre	f	22.00	3
444	Bordeleau	h	18.00	3
555	Pignon	h	20.00	5
666	Sasseur	f	15.00	2
777	Leblanc	h	30.00	5
888	Latendresse	f	25.00	2

```
SELECT departement, genre, nom, COUNT(*), MIN(salaire)
FROM employe
GROUP BY departement, genre;
```

regroupement par combinaison : département et genre

departement	genre	nom	COUNT(*)	MIN(salaire)
3	h	?	2	18.00
5	f	?	1	25.00
3	f	?	1	22.00
5	h	?	2	20.00
2	f	?	2	15.00

critères de regroupement

quel nom choisir?

fonctions d'agrégation

DQL II

HAVING

- Il est possible d'exclure certains regroupements avec la clause HAVING.
- La clause WHERE limite les lignes alors que la clause HAVING limite les regroupements.
- La clause HAVING ne peut être utilisée sans la clause GROUP BY.
- Il est important de se rappeler que la clause WHERE est exécutée **avant** la clause HAVING (puisque la clause WHERE est exécutée avant GROUP BY et que HAVING est exécutée après GROUP BY).

DQL II

HAVING

```
-- retourne le nombre d'employés par département
-- seulement pour les départements ayant au moins 5 employés
SELECT departement, COUNT(*)
    FROM employe
    GROUP BY departement
    HAVING COUNT(*) >= 5;

-- retourne la moyenne salariale des hommes pour les départements dont la
-- moyenne salariale est supérieure à 35$  -  retourne seulement les trois
-- départements ayant la plus grande moyenne salariale
SELECT departement, AVG(salaire) AS "Moyenne des salaires"
    FROM employe
    WHERE genre = 'h'
    GROUP BY departement
    HAVING AVG(salaire) > 35.0
    ORDER BY "Moyenne des salaires" DESC
    LIMIT 3;
```

DQL II

HAVING

- L’usage simultanée des clauses WHERE et HAVING requiert une certaine attention car il est facile d’écrire une requête erronée.
- Mise en situation :
  - On désire compter le nombre d’employés par département ayant un salaire égal ou supérieur à 30$.
  - Cependant, nous ne sommes intéressés que par les départements de plus de 5 employés.
- On comprend que :
  - Il y aura un regroupement sur la colonne departement
  - Il y aura une restriction sur les employés selon le salaire
  - Il y aura une restriction sur les départements selon le nombre d’employés

DQL II

HAVING

- Une solution facile mais erronée :

```
SELECT departement, COUNT(*)
    FROM employe
    WHERE salaire >=  30
    GROUP BY department
    HAVING COUNT(*) > 5;
```

- Cette requête ne retourne que les départements qui ont plus de 5 employés gagnant plus de 30$.

DQL II

HAVING

- Voici une solution correcte même si elle est moins intuitive :

```
SELECT departement, COUNT(*)
    FROM employe
    WHERE    salaire >=  30 AND
                departement IN (SELECT departement
                                    FROM employe
                                    GROUP BY departement
                                    HAVING COUNT(*) > 5)
    GROUP BY department;
```

DQL II ordre d'évaluation des clauses

- Une requête SELECT complète est évaluée dans un ordre différent de la position des clauses dans la requête :

```
SELECT DISTINCT …
    FROM …
    WHERE …
    GROUP BY …
    HAVING …
    ORDER BY …
    LIMIT … OFFSET …;
```