

420-C42

Langages d'exploitation des bases de données

Partie 10

DQL IV

Requêtes corrélées et autres opérateurs ensemblistes

DQL IV

requête non corrélée

- Jusqu'à maintenant nous avons vu des requêtes imbriquées non corrélée. C'est-à-dire que la requête interne ne possède aucun lien spécifique avec la requête externe.
- Généralement, ces requêtes servent à trouver de l'information sans considération pour la requête externe. Plus spécifiquement, la requête interne est exécutée une seule fois avant la requête externe. Finalement, la requête externe utilise le résultat obtenu par la requête interne.
 - Par exemple : quel est le id du département des ventes?
`SELECT * FROM employe WHERE departement =
 (SELECT id FROM departement WHERE nom = 'Ventes');`
 - Si la requête interne retourne 5, on obtient :
`SELECT * FROM employe WHERE departement = (5);`
- La requête interne est traitée une seule fois et retourne l'identifiant du département des ventes. Ce résultat est ensuite injecté dans la requête externe.

DQL IV

requête corrélée

- Une requête corrélée est une requête imbriquée qui utilise les informations de la requête externe.
- En fait, la requête interne **est réalisée pour chaque ligne** de la requête externe.
- Évidemment, ce type de requête peut être beaucoup plus longue à exécuter. Néanmoins, elle permet des requêtes impossibles autrement.
- Elle se reconnaissent facilement car la requête interne fait référence explicitement à au moins un élément de la requête externe. On retrouve souvent aussi l'utilisation des alias de table.

DQL IV

requête corrélée

- Un premier exemple :

-- Trouver les employés dont leur superviseur sont
-- les superviseurs de leur département.

```
SELECT nom, prenom  
FROM employe  
WHERE superviseur =  
      (SELECT departement.superviseur  
        FROM departement  
        WHERE departement.id = employe.departement);
```

critère de corrélation
avec la requête externe

requête interne réalisée pour chaque ligne de la requête externe

DQL IV

requête corrélée

- Un deuxième exemple :

-- Trouver les employés qui gagnent le plus petit salaire
-- de leur département.

```
SELECT nom, prenom  
FROM employe AS emp  
WHERE salaire = ( SELECT MIN(es.salaire)  
                  FROM employe AS es  
                  WHERE es.departement = emp.departement);
```

critère de corrélation
avec la requête externe

requête interne réalisée pour chaque ligne de la requête externe

DQL IV

requête corrélée

- Un troisième exemple :

-- Trouver les employés qui gagnent au moins la moyenne salariale
-- de leur département et qui sont en minorité de genre dans leur département.

```
SELECT nom, prenom
FROM employe AS emp
WHERE salaire >= ( SELECT AVG(es.salaire)
                   FROM employe AS es
                   WHERE es.departement = emp.departement)
AND genre = ( SELECT eg.genre
              FROM employe AS eg
              WHERE eg.departement = emp.departement
              GROUP BY eg.genre
              ORDER BY COUNT(*) ASC
              LIMIT 1);
```

DQL IV

opérateurs ensemblistes

- L'opérateur IN est un opérateur ensembliste, c'est-à-dire qu'il effectue une certaine logique sur un ensemble de valeurs (une liste).
- Les opérateurs ANY, ALL et EXISTS permettent eux aussi d'effectuer des opérations sur des ensembles:
 - [*opérateur comparaison* ANY(...)]
si au moins un élément retourne vrai à l'opérateur de comparaison
 - [*opérateur de comparaison* ALL(...)]
si tous les éléments retournent vrai à l'opérateur de comparaison
 - [EXISTS(...)]
s'il existe au moins une valeur

DQL IV

opérateurs ensemblistes

- Tous ces opérateurs s'appliquent sur une liste.
- Voici l'équivalent logique entre les opérateurs ensemblistes et les opérateurs de base :
 - $a \text{ IN } (x1, x2, x3)$ $a = x1 \text{ OR } a = x2 \text{ OR } a = x3$
 - $a > \text{ANY}(x1, x2, x3)$ $a > x1 \text{ OR } a > x2 \text{ OR } a > x3 \text{ (=ANY } \Rightarrow \text{ IN)}$
 - $a > \text{ALL}(x1, x2, x3)$ $a > x1 \text{ AND } a > x2 \text{ AND } a > x3$
 - $\text{EXISTS}(\{ensemble\})$ $\text{COUNT}(*\{ensemble\}) > 0$

DQL IV

ANY

- les employés ayant un salaire plus grand
- qu'au moins un employé ayant un nom débutant par la lettre B

```
SELECT nom, prenom
FROM employe
WHERE salaire > ANY( SELECT salaire
                      FROM employe
                      WHERE nom LIKE('B%'));
```

DQL IV

ALL

-- les employés ayant un salaire plus grand ou égal
-- à tous les employés du département des ventes

```
SELECT nom, prenom  
FROM employe  
WHERE salaire >= ALL( SELECT salaire  
                        FROM employe  
                        WHERE departement =  
                        (SELECT id FROM departement WHERE nom = 'Ventes'));
```

DQL IV

EXISTS

-- les départements ayant au moins
-- un employé embauché après le mois de juin 2002

```
SELECT nom
  FROM departement AS dep
 WHERE EXISTS(  SELECT nas
                FROM employe AS emp
                WHERE date_embauche > '2002-06-01' AND
                      emp.departement = dep.id); -- corrélation
```