

# Les groupements, requetes imbriquées et opérateurs ensemblistes

Toutes ces choses ont l'avantage de permettre de faire des requêtes plus compliquées en assemblant des requêtes plus simple ou en utilisant des fonctions pour nous simplifier la vie.

## ? Les groupements

Les groupements permettent de grouper plusieurs résultats en même temps. Par exemple pour faire la moyenne d'une/plusieurs colonne(s), les additionner, etc.

```
-- Ici on a une fonction sum() qui permet d'additionner tous les résultats ensemble
SELECT c.description, sum(b.surface_habitable + b.surface_jardin) || 'm2' AS surface
FROM Categorie_bien c
JOIN Bien b ON b.id_catbien = c.id_catbien

-- On peut grouper les choses par description
-- (on y met juste toutes les colonnes qui sont dans le select mais pas dans la fonction de
groupeement)
GROUP BY c.description

-- HAVING fait la même chose qu'un WHERE mais sur le résultat d'un groupement
HAVING sum(b.surface_habitable + b.surface_jardin) BETWEEN 2000 AND 3000
```

Les fonctions de groupements que l'on peut utiliser sont :

- `AVG()` pour faire une moyenne de plusieurs résultats
- `SUM()` pour additionner plusieurs résultats
- `MAX()` pour avoir le plus grand résultat
- `MIN()` pour avoir le plus petit résultat
- `COUNT()` pour compter le nombre de résultats
  - `COUNT(*)` pour compter le nombre de tuples
  - `COUNT(attr)` pour compter le nombre de valeurs dans une colonne
  - `COUNT(DISTINCT attr)` pour compter le nombre de valeurs distinctes d'une colonne

# ? Les requêtes imbriquées

Les requêtes imbriquées permettent de diviser des problèmes complexes en requêtes plus simples que l'on assemble entre elles.

```
-- trouvez le nom du ou des candidats qui ont fait l'offre la plus haute
SELECT c.nom
FROM candidat c
JOIN offre o ON o.id_candidat = c.id_candidat
WHERE o.montant = (SELECT max(montant) FROM offre)
```

Dans cette requête on divise le problème en deux parties :

1. Trouver le montant le plus haut (c'est la requête imbriquée)
2. Trouver le(s) candidat·e·s qui ont une offre qui correspond à ce montant

# ? Les opérateurs ensemblistes

```
-- écrivez une requête qui affichera en 3 colonnes les noms, prénoms et rôle (Candidat,
Client, Vendeur) des candidats, des clients et des vendeurs
select nom, prenom, 'Vendeur' as role
from vendeur
union
select nom, prenom, 'Candidat' as role
from candidat
union
select nom, prenom, 'Client' as role
from client
```

Ici étant donné que l'on a les mêmes noms de colonnes pour chaque requête on peut les combiner ensemble (comme vu avec les ensembles en math ou avec les jointures)

Il existe 3 types d'opérateurs ensemblistes en Oracle SQL :

- Le **UNION** qui permet de combiner les deux (additionner en somme)
- Le **MINUS** qui va prendre tout ce qui est dans le premier mais pas après (différence en math)
- Le **INTERSECT** qui va prendre seulement ce qui est commun aux deux requêtes

Voici d'autres exemples avec d'autres opérateurs :

```
-- Sélectionner tous les biens qui n'ont pas fait l'objet d'une offre
SELECT id_bien
FROM bien

MINUS

SELECT id_bien
FROM offre
```

Ou encore

```
-- affichez les identifiants des biens qui ont une offre d'un montant supérieur à 200.000€
dans la province de Liège

-- On prends tous les biens qui sont dans la province de liège
SELECT id_bien
FROM bien
WHERE code_postal BETWEEN 4000 AND 4999

INTERSECT

-- Et on prends l'intersection avec toutes les offres qui ont un montant supérieur à 200000
SELECT id_bien
FROM offre
WHERE montant > 200000
```

---

Revision #1

Created 26 April 2023 19:01:35 by SnowCode

Updated 26 April 2023 19:01:35 by SnowCode