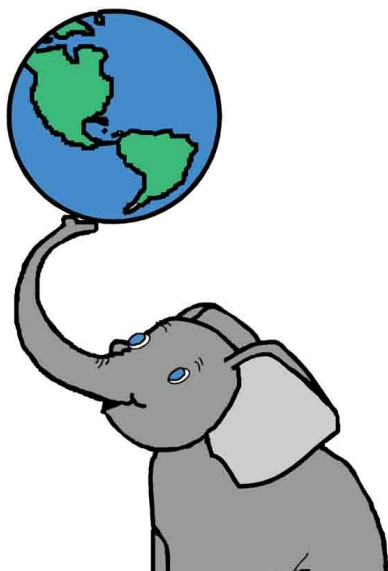


Exercice 10 et sa correction



version 1.3

Septembre 2019



Table des matières

I - Exercice : 10 - sous-requêtes

5

Solution des exercices

7

Exercice : 10 - sous-requêtes

Exercice permettant de mettre en pratique les sous-requêtes

Question

[Solution n°1 p 7]

A partir des tables travail.structure, travail.Delegation et travail.communes64 trouver la liste des communes qui appartiennent à la "COMMUNAUTE DE COMMUNES NIVE-ADOUR"

Pour réaliser cet exercice

- 1) utiliser des sous-requêtes sans WITH
- 2) Utiliser WITH
- 3) utiliser une jointure.

Comparer les temps de traitement entre 1,2 et 3. Commenter.

Le résultat à obtenir :

	Nom_Commune character varying(50)
1	URCUIT
2	URT
3	MOUGUERRE
4	LAHONCE
5	SAINT-PIERRE-D'IRUBE
6	VILLEFRANQUE

Envoyer vos requêtes et commentaires sur les temps de traitement au tuteur.



Solution des exercices

> Solution n°1 (exercice p. 5)

```
SELECT n_siren FROM travail.structure as struc WHERE struc.nom_struct = 'COMMUNAUTE DE COMMUNES NIVE-ADOUR'
```

renvoi le n_siren de cette structure.

Il faut ensuite rechercher dans *travail.Delegation* Toutes les codes insee des communes qui correspondent :

```
select code_insee from travail.delegation where code_siren=(SELECT n_siren FROM travail.structure as struc WHERE struc.nom_struct = 'COMMUNAUTE DE COMMUNES NIVE-ADOUR')
```

renvoi

	code_insee character varying(10)
1	64540
2	64546
3	64407
4	64304
5	64496
6	64558

reste plus qu'à afficher les noms en clair des communes à partir des code_insee

La requête complète peut s'écrire :

```
select "Nom_Commune" from travail.commune64 where "INSEE_Commune" IN (select code_insee from travail.delegation where code_siren=(SELECT n_siren FROM travail.structure as struc WHERE struc.nom_struct = 'COMMUNAUTE DE COMMUNES NIVE-ADOUR'))
```

cette requête est exécutée en 11 ms.

Avec **WITH** (+ jointure) on peut écrire par exemple :

WITH

```
lagor AS (SELECT n_siren FROM travail.structure as struc WHERE struc.nom_struct = 'COMMUNAUTE DE COMMUNES NIVE-ADOUR'),
```

```
delegation_lagor AS (SELECT code_insee FROM travail.delegation JOIN lagor ON code_siren = n_siren)
```

```
SELECT "Nom_Commune" from travail.commune64 AS com64
```

```
JOIN delegation_lagor AS dl
```

ON com64."INSEE_Commune" = dl.code_insee;

résultat en 12 ms.

Solution uniquement avec jointure :

```
SELECT
  *
FROM
  travail.commune64,
  travail.delegation,
  travail.structure
WHERE
  commune64."INSEE_Commune" = delegation.code_insee AND
  delegation.code_siren = structure.n_siren AND
  structure.nom_struct = 'COMMUNAUTE DE COMMUNES NIVE-ADOUR';
```

On peut, si on le souhaite, en dériver une syntaxe avec **JOIN** qui est l'écriture que nous recommandons :

```
SELECT
  *
FROM travail.communes64 as com
JOIN travail.delegation as del ON del.code_insee = com."INSEE_Commune"
JOIN travail.structure as str ON del.code_siren = str.code_siren
WHERE str.nom_struct = 'COMMUNAUTE DE COMMUNES NIVE-ADOUR'
```

Le résultat est obtenu en 12 ms.

Commentaire : Dans notre cas il n'y a pas de gain dans une solution par rapport à une autre.

L'optimisation de requête n'est pas toujours évidente. Il convient de regarder de près pour des requêtes qui auraient des temps d'exécution très longs en faisant des tests sur des jeux de tests réduits.

Comme il a été indiqué en général l'utilisation de IN est plus coûteuse que l'utilisation de jointures.

On retiendra que généralement il est possible de transformer une sous-requête en jointure, mais ce n'est pas toujours vrai (exemple pour une sous-requête dans la clause FROM contenant un *opérateur d'agrégat statistique*¹)

nb : Un petit inconvénient du constructeur graphique est qu'il ne génère pas une syntaxe utilisant JOIN...ON, mais une jointure avec WHERE.



Complément

On utilise ici pour comparer les requêtes les temps affichés par PdAdmin. Sur un aussi petit jeu de données, ce n'est pas forcément très significatif. Nous verrons plus loin dans le cours sur la partie optimisation comment utiliser EXPLAIN ANALYZE qui donne un indicateur beaucoup plus fiable.



Remarque

Les solutions proposées ne tiennent pas compte que dans la réalité une même commune peut déléguer plusieurs compétences à une même structure (cas qui ne se présente pas dans le jeu de données proposé). Si on veut en tenir compte il faut ajouter aux requêtes un GROUP BY sur le nom ou le N° INSEE de la commune.

1 - <http://docs.postgresql.fr/9.4/functions-aggregate.html#functions-aggregate-statistics-table>