

TP 5 - Indexation

Le but de ce TP est d'observer et de pratiquer la construction, le stockage et l'utilisation des différents types d'index sous Postgres.

Tous les exercices sont à faire sur la base de données `clients.sql`.

► Exercice 1 : Stockage et mise-à-jour des index

1. Utilisez le script `clients.sql` pour créer la table `client`. Lorsqu'on examine le schéma de la table avec `\d`, on observe qu'un index est déjà associé à la table. Pourquoi ?

La commande `\timing` permet de demander à Postgres de mesurer le temps d'exécution de chaque instruction.

2. Le fichier `clients_fill.sql` contient un script pour remplir la table avec les informations relatives aux clients. Mesurez le temps de remplissage dans les cas suivants :
 - (a) Sans rien changer à la table.
 - (b) Après avoir construit un index b-tree sur la colonne `nom`.
 - (c) Après avoir construit un index hash sur la colonne `nom`.
 - (d) Après avoir construit des index des **deux** types sur la colonne `nom` et des index des **deux** types sur la colonne `tel`.
 - (e) ♣ Sans aucun index, pas même celui de la question 1.
3. Utilisez la vue `pg_class` pour comparer les tailles (en nombre d'enregistrements et de blocs) des deux types d'index et de la table elle-même.
Que pouvez-vous en déduire sur le type d'index utilisés par Postgres ?

► Exercice 2 : Utilisation des index

Le but de cet exercice est d'observer dans quels cas les index sont utilisés par le *query planner* (optimiseur de requêtes) et de quelle manière.

1. Utilisez le script `clients.sql` pour créer la table. Créez quatre index : un index b-tree et un index hash sur la colonne `nom` et la même chose sur la colonne `tel`, puis chargez les données avec le script `clients_fill.sql`.

Il est en général assez maladroit de maintenir plusieurs index sur les mêmes colonnes. Le but ici est de savoir quel est l'index préféré par le query planner de Postgres lorsqu'il a le choix, ce qui nous donnera une bonne estimation de l'index le plus efficace selon le cas de figure. Pour ce faire, nous allons utiliser la clause **EXPLAIN** qui demande à Postgres d'annoncer le plan d'exécution choisi pour la requête donnée en argument.

2. Exécutez l'instruction ci-dessous et expliquez ce que vous observez.

```
EXPLAIN SELECT * FROM client;
```

3. Regardez les plans d'exécutions pour les requêtes ci-dessous. Dites dans chaque cas quel index est utilisé. Vos observations correspondent-elles à vos connaissances sur le fonctionnement des index ?

- (a) `SELECT nom FROM client;`
- (b) `SELECT nom, prenom FROM client WHERE numcli = 96224;`
- (c) `SELECT nom, prenom FROM client WHERE tel = '0341460953';`
- (d) `SELECT nom, prenom FROM client WHERE tel < '07' AND tel > '06';`
- (e) `SELECT prenom FROM client WHERE nom = 'Delauney';`
- (f) `SELECT prenom FROM client WHERE nom < 'Delauney';`

4. Expliquez la différence entre les plans d'exécution des deux requêtes ci-dessous :

- (a) `SELECT prenom, tel FROM client WHERE nom < 'Delauney';`
- (b) `SELECT prenom, tel FROM client WHERE nom > 'Delauney';`

5. Expliquez la différence entre les plans d'exécution des deux requêtes ci-dessous :

- (a) `SELECT nom, prenom, prenom2, prenom3 FROM client WHERE tel IS NULL;`
- (b) `SELECT nom, prenom, prenom2, prenom3 FROM client WHERE tel IS NOT NULL;`

Essayez de confirmer votre hypothèse avec des index et des requêtes du même type sur prenom2 et prenom3. Le résultat change-t-il après un **ANALYZE** de la table ?

6. ♣ Expliquez la différence entre les plans d'exécution des deux requêtes ci-dessous :

- (a) `SELECT nom, prenom FROM client WHERE numcli < 250;`
- (b) `SELECT nom, prenom FROM client WHERE nom < 'Ali';`

► Exercice 3 : ♣ Index sur plusieurs colonnes

Postgres permet de créer des index sur une clef de recherche composée de plusieurs colonnes. Cet exercice donne un aperçu de leur fonctionnement et de leur utilisation.

1. Essayez de créer un index b-tree et un index hash sur la clef (nom,prenom).
2. Comparez les tailles des index sur une colonne et sur plusieurs colonnes. Expliquez vos observations.

À partir de cette question, effacez **tous** vos index **sauf** l'index b-tree sur (nom,prenom).

3. L'index est-il utilisé pour les requêtes suivantes ? Expliquez vos observations.

- (a) `SELECT tel FROM client WHERE prenom = 'Fleur' AND nom = 'Delacour';`
- (b) `SELECT prenom, tel FROM client WHERE nom = 'Delacour';`
- (c) `SELECT nom, tel FROM client WHERE prenom = 'Fleur';`

4. Regardez le plan d'exécution de la requête ci-dessous. Faites un **VACUUM** de la table client et réessayez. Expliquez vos observations.

`SELECT prenom FROM client WHERE nom = 'Delacour';`