

BASES DE DONNÉES

ESIR 1, TP3

Installation du TP

Ces travaux pratiques utilisent une base de données de grande taille générée automatiquement. Les consignes qui suivent permettent d'installer le générateur de la base de données, puis de l'exécuter.

1. Créer un répertoire sous `/tmp` avec un nom unique, par exemple en utilisant votre numéro d'étudiant `/tmp/etudiant123456/`

ATTENTION : Vous allez travailler dans ce TP avec des bases de données qui sont plus grandes que votre quota d'espace disque. Vous pouvez dépasser ce quota en mettant les fichiers temporaires dans le répertoire `/tmp`. Par contre, ces fichiers ne sont pas sauvegardés.

2. Télécharger depuis la page Moodle du cours BDD le fichier `generate.zip` dans ce répertoire, et extraire les fichiers `generateDB.sh`, `DBgenerate.java`, et `DBgenerate1.java`.
3. Vérifier les droits d'exécution (`ls -l`), et si cela est nécessaire, rendre le fichier exécutable, par exemple en utilisant la commande `chmod u+x generateDB.sh`.
4. Utiliser la commande `./generateDB.sh` pour engendrer les fichiers `database.sql` et `database1.sql`. Si cela est nécessaire (par exemple, pas assez de quota), diminuer la taille de la base de données générée (deuxième paramètre de `generateDB.sh`).

Exercice 1 : Index

Créer une base de données à l'aide de la commande `sqlite3 test < database.sql`. La base contient une table de codes. Vérifier la structure de la base à l'aide de la commande `cat database.sql`.

Utiliser la commande `.timer ON` pour déterminer le temps d'exécution d'une requête.

1. Mesurer le temps d'exécution de la requête `SELECT * FROM demo WHERE code=53772617;`. Répéter la mesure 5 fois.
2. Mesurer le temps d'exécution de la requête `SELECT * FROM demo WHERE code=53772618;` (ce code n'existe pas dans la base de données). Répéter la mesure 5 fois.
3. Créer un index sur l'attribut `code` à l'aide de la commande `CREATE INDEX demoIDX ON demo(code);`.
4. Recommencer les mesures.

Qu'avez-vous observé ?

Exercice 2 : Comprendre le temps d'exécution d'une requête, optimisation

Construire une nouvelle base de données à l'aide de la commande `sqlite3 test1 < database1.sql`. Cette base de données comporte les tables suivantes :

```
CREATE TABLE facture (id INTEGER, customerId TEXT, amount REAL);
CREATE TABLE customer (customerId TEXT, name TEXT);
```

La base contient des informations correspondant aux factures et aux clients d'une société. Le responsable des ventes recherche les noms des clients qui ont une facture d'un montant total de plus de 1000 euros.

Supposons que seulement une petite fraction des factures ont un montant total supérieur à 1000 euros.

- Comparer sans les exécuter les requêtes SQL suivantes :

1. `SELECT c.name FROM customer c, facture f WHERE f.customerId=c.customerId AND f.amount>1000;`
2. `SELECT name FROM customer WHERE customerId IN (SELECT f.customerId FROM facture f WHERE amount>1000);`
3. `SELECT name FROM (customer NATURAL JOIN facture) WHERE amount>1000;`
4. `SELECT name FROM customer WHERE customerId IN (SELECT c.customerId FROM customer c, facture f WHERE c.customerId=f.customerId AND f.amount>1000);`

Essayer de prévoir lesquelles seront les longues à s'exécuter.

- Mesurer le temps d'exécution des requêtes. Utiliser la commande `PRAGMA auto_index = 0` pour analyser le temps d'évaluation sans indices automatiques. Préparer un fichier tableur pour y accumuler les résultats de ces expériences (voir `measurements.xlsx`).
- Utiliser la commande `EXPLAIN QUERY PLAN` et l'option `.stats ON` pour comprendre le schéma d'exécution de chaque requête.
- Définir des index qui pourraient réduire le temps d'exécution de ces requêtes. Mesurer les nouveaux temps d'exécution et ajouter les résultats au fichier tableur. Faire une présentation graphique de l'ensemble de ces expériences et commenter.

Fin de TP : N'oubliez pas de sauvegarder dans votre répertoire personnel les fichiers que vous avez utilisés (par exemple, les résultats de vos expériences, les fichiers de compte-rendu, etc.). Tout ce qui reste dans le répertoire `/tmp` risque d'être perdu.