

## TP3 - BASES DE DONNEES

Le but de ce TP est d'utiliser un générateur de bases de données afin de mesurer des temps d'exécution de requêtes SQL sur de grosses bases de données.

### I. INSTALLATION

---

Nous avons installé un générateur de base de données pour générer automatiquement une base de données de grande taille.

### II. EXERCICE 1 : INDEX

---

1. On commence par activer le timer. On mesure le temps d'exécution cinq fois d'une requête avec un code existant dans la base de données.

Real	User	Sys
0.088	0.079445	0.007964
0.090	0.080602	0.008124
0.085	0.076590	0.007767
0.088	0.078418	0.008453
0.091	0.077850	0.008894

2. On réitère l'opération avec un code inexistant dans la base de données.

Real	User	Sys
0.080	0.069247	0.008672
0.086	0.077139	0.007004
0.088	0.080683	0.006728
0.084	0.076905	0.006492
0.086	0.079380	0.006823

On remarque que les temps d'exécutions sont très similaires alors que, dans le second cas, on doit parcourir l'entièreté de la table qui est de grande taille.

3. On crée un index sur l'attribut `code`. On recommence alors les mesures pour les deux requêtes.

Pour la requête avec un code qui existe dans la base de données, on obtient :

Real	User	Sys
0.001	0.000177	0.000303
0.000	0.000162	0.000112
0.001	0.000169	0.000126
0.001	0.000160	0.000109
0.001	0.000175	0.000127

Pour la requête avec un code non existant dans la base de données, on obtient :

Real	User	Sys
0.001	0.000132	0.000092
0.001	0.000123	0.000092
0.001	0.000210	0.000151
0.000	0.000156	0.000111
0.000	0.000144	0.000103

On constate que les temps d'exécutions, dans un cas comme dans l'autre, ont considérablement diminué. Un index sur un attribut pertinent permet donc de diminuer le temps d'exécutions de certaines requêtes.

### III. EXERCICE 2 : TEMPS D'EXECUTION D'UNE REQUETE

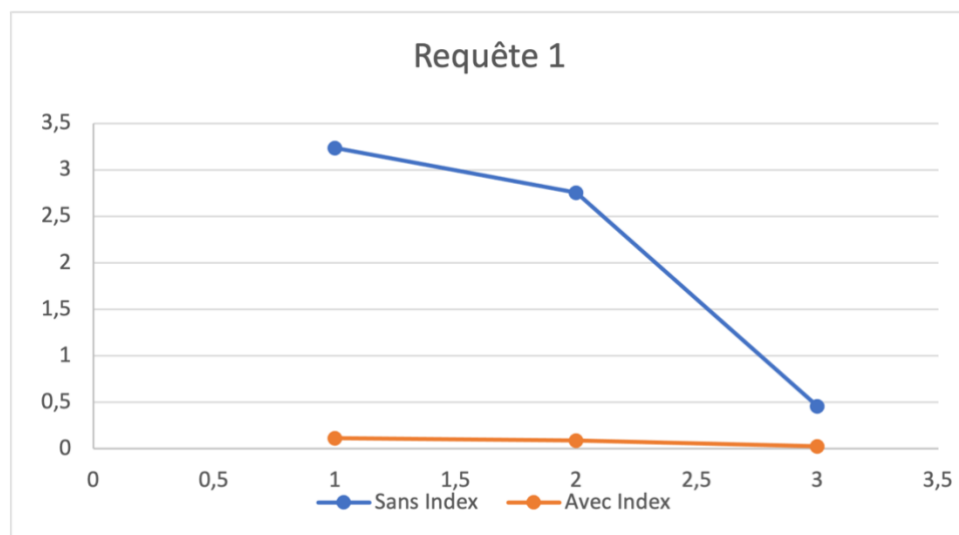
1. Nous pouvons nous attendre à avoir des temps d'exécutions différents. Les requêtes 1 et 4 pourraient notamment prendre plus de temps à cause des liens entre les tables.
- 2.

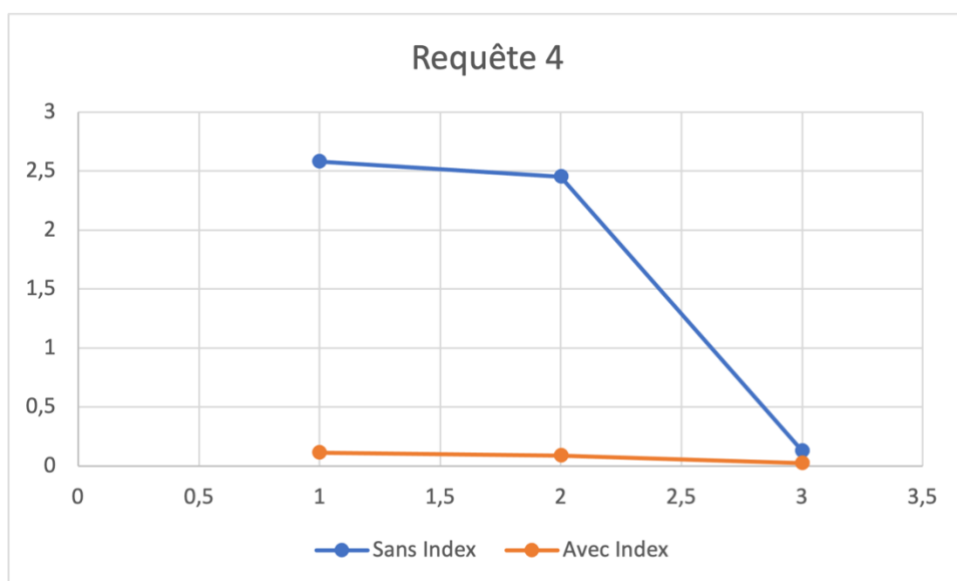
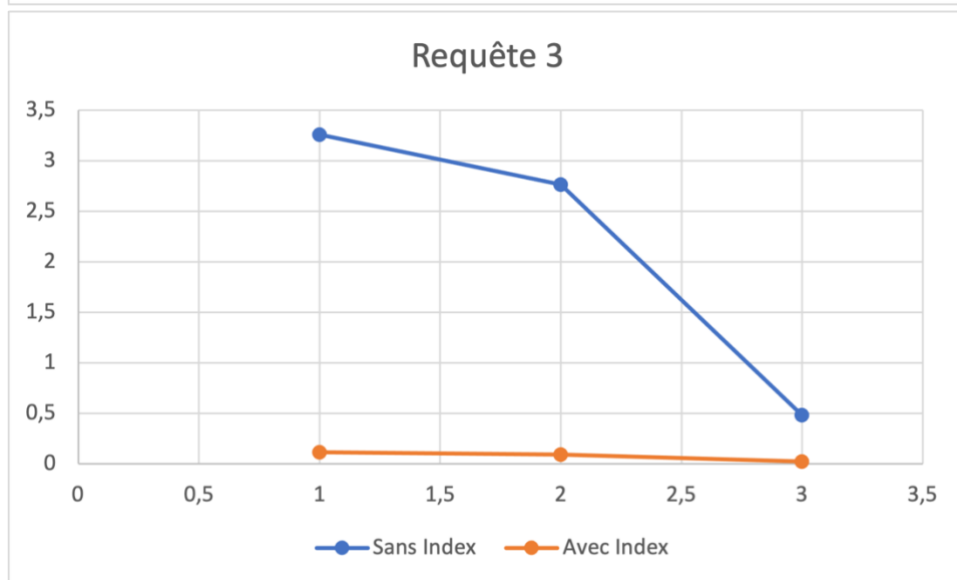
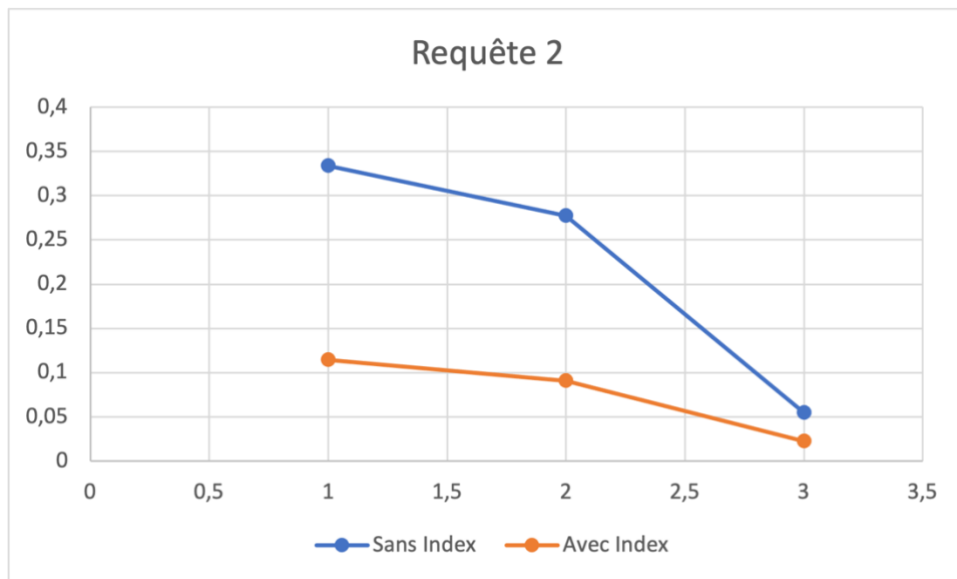
Résultats par requêtes	Real	User	Sys
Requête 1	3.232	2.756857	0.460279
Requête 2	0.334	0.277084	0.054734
Requête 3	3.256	2.764119	0.480746
Requête 4	2.582	2.452172	0.126528

3. Voir annexe pour les schémas d'exécution des requêtes.

4. Pour améliorer le temps d'exécution nous avons créé un index sur la colonne customerID de la table Customer car il semble apparaître régulièrement dans les schémas d'exécution.

Résultats par requêtes	Real	User	Sys
Requête 1	0.118	0.090930	0.026306
Requête 2	0.114	0.090554	0.022382
Requête 3	0.116	0.090793	0.023268
Requête 4	0.113	0.090261	0.022385





## IV. EXERCICE 3 : EXPERIMENTATION

---

Voici la requête que nous avons imaginé :

```
SELECT c.customerId FROM customer c JOIN facture f ON f.customerId =  
c.customerId ORDER BY name LIMIT 10;
```

Pour diminuer le temps d'exécution de cette requête nous avons fait deux indices, un sur la colonne name de la table Customer et un autre sur la colonne customerId de la table Customer.

Résultats	Real	User	Sys
Sans index	8.520	3.000933	2.118755
Avec index	0.04	0.000618	0.000826

Les résultats sont assez impressionnants : on a divisé le temps d'exécution par plus de 200 !

Schéma d'exécution de la requête :

```
QUERY PLAN  
|--SCAN c USING INDEX customernameIDX  
`--SEARCH f USING INDEX i (customerId=?)  
Memory Used:          9076768 (max 57231872) bytes  
Number of Outstanding Allocations: 4358 (max 23252)  
Number of Pcache Overflow Bytes: 137344 (max 726336) bytes  
Largest Allocation:    8192000 bytes  
Largest Pcache Allocation: 4104 bytes  
Lookaside Slots Used:  49 (max 110)  
Successful lookaside attempts: 1802799  
Lookaside failures due to size: 54  
Lookaside failures due to OOM: 0  
Pager Heap Usage:      8737072 bytes  
Page cache hits:        7865418  
Page cache misses:      3551281  
Page cache writes:      22264  
Page cache spills:      0  
Schema Heap Usage:      1760 bytes  
Statement Heap/Lookaside Usage: 7440 bytes  
Fullscan Steps:         0  
Sort Operations:         0  
Autoindex Inserts:      0  
Virtual Machine Steps:   0  
Reprepare operations:    0  
Number of times run:     0  
Memory used by prepared stmt: 7440
```

On peut voir qu'un index est utilisé à chaque étape.

Les index sont des outils très efficaces pour diminuer le temps d'exécution d'une requête à condition que ceux-ci soient bien utilisés.

## Annexes

Schéma d'exécution de la requête 1. de la partie 3. :

```
QUERY PLAN
|--SCAN c
`--SEARCH f USING INDEX i (customerId=?)
Memory Used:          148912 (max 149440) bytes
Number of Outstanding Allocations: 201 (max 205)
Number of Pcache Overflow Bytes: 4672 (max 4672) bytes
Largest Allocation:    122400 bytes
Largest Pcache Allocation: 4104 bytes
Lookaside Slots Used:  43 (max 109)
Successful lookaside attempts: 157
Lookaside failures due to size: 3
Lookaside failures due to OOM: 0
Pager Heap Usage:      9976 bytes
Page cache hits:        1
Page cache misses:      1
Page cache writes:      0
Page cache spills:      0
Schema Heap Usage:      1344 bytes
Statement Heap/Lookaside Usage: 6800 bytes
Fullscan Steps:         0
Sort Operations:        0
Autoindex Inserts:      0
Virtual Machine Steps:  0
Reprepare operations:    0
Number of times run:    0
Memory used by prepared stmt: 6800
```

Schéma d'exécution de la requête 2. de la partie 3. :

```
QUERY PLAN
|--SCAN customer
`--LIST SUBQUERY 1
  |--SCAN f
Memory Used:          148912 (max 149440) bytes
Number of Outstanding Allocations: 201 (max 206)
Number of Pcache Overflow Bytes: 4672 (max 4672) bytes
Largest Allocation:    122400 bytes
Largest Pcache Allocation: 4104 bytes
Lookaside Slots Used:  49 (max 109)
Successful lookaside attempts: 231
Lookaside failures due to size: 4
Lookaside failures due to OOM: 0
Pager Heap Usage:      9976 bytes
Page cache hits:        0
Page cache misses:      0
Page cache writes:      0
Page cache spills:      0
Schema Heap Usage:      1344 bytes
Statement Heap/Lookaside Usage: 7568 bytes
Fullscan Steps:         0
Sort Operations:        0
```

Autoindex Inserts:	0	
Virtual Machine Steps:	0	
Reprepare operations:	0	
Number of times run:	0	
Memory used by prepared stmt:	7568	



### Schéma d'exécution de la requête 3. de la partie 3. :

#### QUERY PLAN

|--SCAN customer

`--SEARCH facture USING INDEX i (customerId=?)

Memory Used: 148912 (max 149440) bytes

Number of Outstanding Allocations: 201 (max 206)

Number of Pcache Overflow Bytes: 4672 (max 4672) bytes

Largest Allocation: 122400 bytes

Largest Pcache Allocation: 4104 bytes

Lookaside Slots Used: 43 (max 109)

Successful lookaside attempts: 283

Lookaside failures due to size: 6

Lookaside failures due to OOM: 0

Pager Heap Usage: 9976 bytes

Page cache hits: 0

Page cache misses: 0

Page cache writes: 0

Page cache spills: 0

Schema Heap Usage: 1344 bytes

Statement Heap/Lookaside Usage: 6800 bytes

Fullscan Steps: 0

Sort Operations: 0

Autoindex Inserts: 0

Virtual Machine Steps: 0

Reprepare operations: 0

Number of times run: 0

Memory used by prepared stmt: 6800

### Schéma d'exécution de la requête 4. de la partie 3. :

#### QUERY PLAN

|--SCAN customer

`--LIST SUBQUERY 1

|--SCAN f

`--SEARCH c USING AUTOMATIC COVERING INDEX (customerId=?)

Memory Used: 151472 (max 151856) bytes

Number of Outstanding Allocations: 202 (max 208)

Number of Pcache Overflow Bytes: 4672 (max 4672) bytes

Largest Allocation: 122400 bytes

Largest Pcache Allocation: 4104 bytes

Lookaside Slots Used: 56 (max 109)

Successful lookaside attempts: 394

Lookaside failures due to size: 9

Lookaside failures due to OOM: 0

Pager Heap Usage: 9976 bytes

Page cache hits: 0

Page cache misses: 0

Page cache writes: 0

Page cache spills: 0

Schema Heap Usage: 1344 bytes

Statement Heap/Lookaside Usage: 10896 bytes

Fullscan Steps: 0

Sort Operations: 0

Autoindex Inserts: 0

Virtual Machine Steps: 0

Reprepare operations: 0

Number of times run: 0  
Memory used by prepared stmt: 10896