

TUGAS BESAR
MANAJEMEN BASIS DATA
TUNNING DATABASE



DOSEN PENGAMPU :
Ahmad Luky Ramdani, S.Kom., M.Kom.

Disusun Oleh :
Jeans Prima Simaremare (14117018)

PROGRAM STUDI
TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA

2019

DAFTAR ISI

DAFTAR ISI.....	2
BAB I STUDI LITERATUR	3
1.1. TUNNING : INDEXING	3
1.2. TUNNING : SETTING CONFIGURATION DBMS	3
BAB II DESKRIPSI PERCOBAAN	5
2.1. TUNNING INDEXING	5
2.2. TUNNING SETTING CONFIGURATION DBMS	5
BAB III HASIL DAN PEMBAHASAN	7
3.1. HASIL QUERY SEBELUM INDEXING.	7
3.2. HASIL QUERY SETELAH INDEXING	9
3.3. KESIMPULAN	10
DAFTAR PUSTAKA	11

BAB I STUDI LITERATUR

1.1. TUNNING : INDEXING

Index adalah sebuah objek dalam sistem database yang dapat mempercepat proses pencarian (*query*) data. Saat database dibuat tanpa menggunakan index, maka kinerja server database dapat menurun secara drastis. Hal ini dikarenakan resource CPU banyak digunakan untuk pencarian data atau pengaksesan query SQL dengan metode table-scan. Index membuat pencarian data akan lebih cepat dan tidak banyak menghabiskan resource CPU.

Index merupakan objek struktur data tersendiri yang tidak bergantung kepada struktur tabel. Setiap index terdiri dari nilai kolom dan penunjuk (atau ROWID) ke baris yang berisi nilai tersebut. Penunjuk tersebut secara langsung menunjuk ke baris yang tepat pada tabel, sehingga menghindari terjadinya full table-scan. Akan tetapi lebih banyak index pada tabel tidak berarti akan mempercepat query. Semakin banyak index pada suatu tabel menyebabkan kelambatan pemrosesan perintah-perintah DML (*Data Manipulation Language*), karena setiap terjadi perubahan data maka index juga harus disesuaikan.

Index tidak dapat dimodifikasi. Kita harus menghapusnya terlebih dahulu dan menciptakannya kembali. Kita dapat menghapus definisi index dari data dictionary dengan perintah DROP INDEX.

Contoh: DROP INDEX employee_last_name_idx;

1.2. TUNNING : SETTING CONFIGURATION DBMS

Database tuning merupakan sejumlah aktifitas yang dilakukan untuk mengoptimalkan kinerja dari basis data. Database tuning tidak hanya meliputi tuning query, tapi juga pada rancangan basis data, pemilihan DBMS, sistem operasi dan CPU yang menjalankan DBMS. Tujuan dari database tuning yaitu memaksimalkan penggunaan sumber daya sistem untuk melakukan kerja secara efisien dan secepat mungkin. Kebanyakan sistem telah dirancang untuk bekerja secara efisien, tapi bisa dilakukan peningkatan kinerja MySQL secara signifikan

dengan mengkostumisasi pengaturan, konfigurasi database dan meng-tuning DBMS. Hasilnya, kinerja database MySQL lebih cepat dan produktifitas semakin meningkat.

MySQL secara default menggunakan konfigurasi moderate, dalam arti bahwa konfigurasi yang diberikan merupakan konfigurasi yang menyesuaikan dengan sistem server secara umum. Ada penyesuaian antara performa dengan stabilitas. Tidak diset performa tinggi agar tidak mempengaruhi stabilitas sistem.mysql-performance-tuning. Kita bisa dengan mudah melakukan ujicoba tuning sistem dengan cara membackup file konfigurasi standard dan kemudian mengubah beberapa opsi/parameter yang tersedia.

BAB II DESKRIPSI PERCOBAAN

2.1. TUNNING INDEXING

Lakukan Tuning Query database dengan melakukan indexing pada beberapa filed yang anda pilih. Index pada data set 1 sampai data set 3 dilakukan pada table student dan pada table section.

```
MariaDB [tubes1]> create index section_idx on section(sec_id);
```

Pada table student indexing dilakukan pada kolom id dengan nama index adalah student_idx. Kemudian pada table section, indexing dilakukan pada kolom sec_id dengan nama index section_idx.

```
MariaDB [tubes1]> create index student_idx on student(id);
```

Hal berikut dilakukan pada data set satu sampai data set tiga

2.2. TUNNING SETTING CONFIGURATION DBMS

Database yang digunakan pada kasus ini adalah Mysql. Untuk pengaturan konfigurasi database manajemen system. Hal pertama yang harus dilakukan adalah mengatur ukuran file yang bisa diupload pada file C://xampp/mysql/bin/my.ini dengan mengganti ketiga parameter berikut :

- innodb_log_file_size=25M
- innodb_log_buffer_size=32M
- innodb_flush_log_at_trx_commit=1
- innodb_lock_wait_timeout=50

```
# Comment the following if you are using InnoDB tables
#skip-innodb
innodb_data_home_dir="C:/xampp/mysql/data"
innodb_data_file_path=ibdata1:10M:autoextend
innodb_log_group_home_dir="C:/xampp/mysql/data"
#innodb_log_arch_dir="C:/xampp/mysql/data"
## You can set .. buffer_pool_size up to 50 - 80 %
## of RAM but beware of setting memory usage too high
innodb_buffer_pool_size=2500M
## Set .. log_file_size to 25 % of buffer pool size
innodb_log_file_size=25M
innodb_log_buffer_size=32M
innodb_flush_log_at_trx_commit=1
innodb_lock_wait_timeout=50

## UTF 8 Settings
#init-connect='(SET NAMES utf8)'
#collation_server=utf8_unicode_ci
#character_set_server=utf8
#skip-character-set-client-handshake
#character_sets_dir="C:/xampp/mysql/share/charsets"
```



BAB III HASIL DAN PEMBAHASAN

3.1. HASIL QUERY SEBELUM INDEXING.

Pada kasus kali ini, hanya ada tiga data set yang digunakan karena dari segi spesifikasi laptop yang tidak mendukung banyak data, sehingga data set yang digunakan adalah :

- advisor = 100, student = 100, section = 200,takes = 200 sebagai Q1
- advisor = 200, student = 200, section = 400,takes = 400 sebagai Q2
- advisor = 500, student = 500, section = 1000,takes = 1000 sebagai Q3

Sedangkan query yang dijalankan adalah :

- SELECT * FROM student;
- SELECT * FROM student WHERE tot_cred > 30;
- SELECT `name`, department FROM student WHERE tot_cred > 30;
- SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id
- SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id

- **Data Set 1**

Query	Waktu
Q1	0.00057831
Q2	0.05462359
Q3	0.00310190
Q4	0.02704672
Q5	0.36409621

- **Data Set 2**

Query	Waktu
Q1	0.13258467
Q2	0.06420210
Q3	0.53790141
Q4	1.43283213
Q5	0.12674401

- **Data Set 3**

Query	Waktu
Q1	0.04132121
Q2	0.01314218

Q3	0.01356729
Q4	0.02178303
Q5	0.10762034

```

XAMPP for Windows - mysql -u root
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 4 | 0.04373350 | select *from student where tot_cred>30 |
| 5 | 0.00216120 | select name,dept_name from student where tot_cred>30 |
| 6 | 0.12805690 | select *from takes join student on takes.id=student.id join section on takes.course_id=section.course_id |
| 7 | 0.15726640 | SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,cour |
| 8 | 5.20339500 | se.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.cou |
| 9 | 0.00142380 | drop database tubes2 |
| 10 | 15.50643440 | SELECT DATABASE() |
| 11 | 0.09969740 | drop database tubes2 |
| 12 | 0.15464180 | SELECT DATABASE() |
| 13 | 0.00074070 | create database tubes2 |
| 14 | 0.13760350 | SELECT DATABASE() |
| 15 | 0.04900220 | select *from student |
| 16 | 0.00190140 | select *from student where tot_cred>30 |
| 17 | 1.14076090 | select name,dept_name from student where tot_cred>30 |
| 18 | 1.14076090 | select *from takes join student on takes.id=student.id join section on takes.course_id=section.course_id |
| 19 | 0.23884500 | SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,cour |
| 20 | 0.10039780 | se.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.cou |
+-----+-----+-----+
15 rows in set (0.447 sec)

MariaDB [tubes2]>

```

```

XAMPP for Windows - mysql -u root
5975 rows in set (0.158 sec)

MariaDB [tubes3]> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 15 | 0.04900220 | select *from student where tot_cred>30 |
| 16 | 0.00190140 | select name,dept_name from student where tot_cred>30 |
| 17 | 1.14076090 | select *from takes join student on takes.id=student.id join section on takes.course_id=section.course_id |
| 18 | 0.23884500 | SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,cour |
| 19 | 0.10039780 | se.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.cou |
| 20 | 0.00037490 | create database tubes3 |
| 21 | 4.63905580 | SELECT DATABASE() |
| 22 | 0.00112900 | drop database tubes3 |
| 23 | 0.00170230 | SELECT DATABASE() |
| 24 | 0.00031910 | create database tubes3 |
| 25 | 0.04472140 | SELECT DATABASE() |
| 26 | 0.04472140 | select *from student |
| 27 | 0.01244250 | select *from student where tot_cred>30 |
| 28 | 0.00183460 | select name,dept_name from student where tot_cred>30 |
| 29 | 0.07187720 | select *from takes join student on takes.id=student.id join section on takes.course_id=section.course_id |
| 30 | 0.15822370 | SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,cour |
| 31 | 0.10039780 | se.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.cou |
+-----+-----+-----+
15 rows in set (0.000 sec)

MariaDB [tubes3]>

```


3.2. HASIL QUERY SETELAH INDEXING

Pada kasus kali ini, hanya ada tiga data set yang digunakan karena dari segi spesifikasi laptop yang tidak mendukung banyak data, sehingga data set yang digunakan adalah :

- advisor = 100, student = 100, section = 200,takes = 200 sebagai Q1
- advisor = 200, student = 200, section = 400,takes = 400 sebagai Q2
- advisor = 500, student = 500, section = 1000,takes = 1000 sebagai Q3

Sedangkan query yang dijalankan adalah :

- SELECT * FROM student;
- SELECT * FROM student WHERE tot_cred > 30;
- SELECT `name`, department FROM student WHERE tot_cred > 30;
- SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id
- SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id

Data Set 2

Data Set 1

Query	Waktu
Q1	0.00043729
Q2	0.00087963
Q3	0.00022409
Q4	0.00703271
Q5	0.01456437

Query	Waktu
Q1	0.53268921
Q2	0.75342126
Q3	0.00075432
Q4	1.09831563
Q5	0.57328532

Data Set 3

Query	Waktu
Q1	0.532215125

Q2	0.01284294
Q3	0.00320982
Q4	1.03298746
Q5	0.87543214

Dari sejumlah query yang dijalankan dari ketiga data set tersebut, semakin banyak data yang dihasilkan semakin lama response time yang dibutuhkan dalam memproses query sql. Setelah dilakukan indexing pada kolom id table student dan kolom sec_id pada table section, hasilnya query yang dijalankan lebih cepat dari sebelum dilakukan indexing.

3.3. KESIMPULAN

Database tuning merupakan sejumlah aktifitas yang dilakukan untuk mengoptimalkan kinerja dari basis data. Database tuning tidak hanya meliputi tuning query, tapi juga pada rancangan basis data, pemilihan DBMS, sistem operasi dan CPU yang menjalankan DBMS. Tujuan dari database tuning yaitu memaksimalkan penggunaan sumber daya sistem untuk melakukan kerja secara efisien dan secepat mungkin.

Dari percobaan yang telah dilakukan dengan menggunakan tiga buah set data yang disimpan kedalam tiga buah database berbeda dapat diketahui bahwa waktu pengekseskuan sebuah query terbukti lebih cepat di proses ketika dilakukan indexing. Indexing terbukti dapat mempermudah dan mempercepat proses pencarian namun apabila kita melihat pada data set 3 yang tidak mempercepat proses eksekusi query maka indexing tak selalu dapat membantu namun dapat mempercepat pencarian dan eksekusi query. Proses tuning dapat membantu meningkatkan performa dari proses dikarenakan dapat mempercepat proses run. Proses run juga dipengaruhi oleh eksternal seperti penggunaan RAM yang penuh pada saat proses tersebut sehingga membuat peningkatan waktu .

DAFTAR PUSTAKA

1. <https://www.careerride.com/DB-index-tuning.aspx>
2. <https://www.teknojempol.com/2017/12/mysql-database-tuning-bagian-1.html>