

Report

Subject: Horizontal Fragmentation.

Students: CHARFAOUI Younes, BOURBAI Ismail.

Github Repository: https://github.com/IsmailBourbie/master-one-practical-work/tree/master/db_dm/TP4

Step One:

We've installed *MySQL* in our computers, and we've used Command line and Text editor for manipulating and queries the DBMS

Step Two:

In this step we create a sample table without partitions using the following SQL Script

```
--
27 -- [INFO]:
28 ---- You must add the file of load_salaries.dump to the same directory of this current file "creations_queries.sql" and run:
29 -----mysql -uroot < creations_queries.sql
30 ---- PS: I ignore the file for github because its large file.
31
32 DROP DATABASE IF EXISTS employees;
33 CREATE DATABASE IF NOT EXISTS employees;
34 USE employees;
35
36 SELECT 'CREATING DATABASE STRUCTURE' as 'INFO';
37
38 DROP TABLE IF EXISTS dept_emp,
39 dept_manager,
40 titles,
41 salaries,
42 employees,
43 departments;
44
45 set storage_engine = InnoDB;
46
47 select CONCAT('storage engine: ', @@storage_engine) as INFO;
48
49 CREATE TABLE salaries (
50     emp_no    INT          NOT NULL,
51     salary    INT          NOT NULL,
52     from_date DATE         NOT NULL,
53     to_date   DATE         NOT NULL
54 );
55
56 SELECT 'LOADING salaries' as 'INFO';
57 source load_salaries.dump;
58
```

Figure 1 Example of Creation table

Step Three:

In This Step we have used script to load our prepared data into the table and count all rows inserted, the following screenshot explain that:

```
C:\xampp\htdocs\master -> origin1\db_dm\TP4\sql (master -> origin)
λ mysql -uroot < creations_queries.sql
INFO
CREATING DATABASE STRUCTURE
INFO
storage engine: InnoDB
INFO
LOADING salaries

C:\xampp\htdocs\master -> origin1\db_dm\TP4\sql (master -> origin)
λ mysql -uroot
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 17
Server version: 10.1.36-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use employees;
ERROR 1040 (42000): Unknown database 'employees'
MariaDB [(none)]> use employees;
Database changed
MariaDB [employees]> SELECT count(*) FROM salaries;
+-----+
| count(*) |
+-----+
| 2844047 |
+-----+
1 row in set (4.94 sec)

MariaDB [employees]> |
```

Load database from creations_queries file

**Count Rows in salaries table
2844047 Row**

Figure 2 SQL Loader Script

PS: we didn't add The Loader file in the repo of GitHub because it is a large file.

Step Four:

The following screenshots show the execution time of select operation on our table without using partitions:

```
C:\xampp\htdocs\master -> origin1\db_dm\TP4\sql (master -> origin)
λ mysql -uroot
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 18
Server version: 10.1.36-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use employees;
Database changed
MariaDB [employees]>
MariaDB [employees]> SELECT * FROM salaries WHERE salary = 130410;
+-----+
| emp_no | salary | from_date | to_date |
+-----+
| 459403 | 130410 | 2002-03-17 | 9999-01-01 |
+-----+
1 row in set (5.52 sec)

MariaDB [employees]> SELECT * FROM salaries WHERE salary = 130410;
+-----+
| emp_no | salary | from_date | to_date |
+-----+
| 459403 | 130410 | 2002-03-17 | 9999-01-01 |
+-----+
1 row in set (5.68 sec)

MariaDB [employees]> |
```

Figure 3 - Execution time without partitions.

Comment:

Its take 5.5sec;

Step Five:

The execution of the last script (Figure 4) take a long time, So we can add Fragmentation to our table (Partitions), and use partitioning BY RANGE.

We can do that by creating partitioned table or just ADD partition on our existing table using the following script:

```
C:\xampp\htdocs\master -> origin\db_dm\TP4\sql (master -> origin)
A mysql -uroot
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 19
Server version: 10.1.36-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use employees;
Database changed
MariaDB [employees]> ALTER TABLE salaries PARTITION BY RANGE(salary)(
-> PARTITION p0 VALUES LESS THAN (100000),
-> PARTITION p1 VALUES LESS THAN (160000)
-> );
Query OK, 2844047 rows affected (2 min 10.26 sec)
Records: 2844047  Duplicates: 0  Warnings: 0

MariaDB [employees]> |
```

Figure 5 - PARTITION on exesting table

Step Six:

We execute the same query of selection operation but this time using partition.

The following script show the query and the result:

```
C:\xampp\htdocs\master -> origin\db_dm\TP4\sql (master -> origin)
A mysql -uroot
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 21
Server version: 10.1.36-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use employees;
Database changed
MariaDB [employees]> SELECT * FROM salaries PARTITION(p1) WHERE salary = 130410;
+-----+-----+-----+-----+
| emp_no | salary | from_date | to_date |
+-----+-----+-----+-----+
| 459403 | 130410 | 2002-03-17 | 9999-01-01 |
+-----+-----+-----+-----+
1 row in set (0.51 sec)

MariaDB [employees]> SELECT * FROM salaries PARTITION(p1) WHERE salary = 130410;
+-----+-----+-----+-----+
| emp_no | salary | from_date | to_date |
+-----+-----+-----+-----+
| 459403 | 130410 | 2002-03-17 | 9999-01-01 |
+-----+-----+-----+-----+
1 row in set (0.31 sec)

MariaDB [employees]> |
```

The result is the same as obtained by the query `SELECT * FROM employees WHERE id BETWEEN 3 AND 5`.

To obtain rows from multiple partitions, supply their names as a comma-delimited list. For example, `SELECT * FROM employees PARTITION (p1, p2)` returns all rows from partitions p1 and p2 while excluding rows from the remaining partitions.

Any valid query against a partitioned table can be rewritten with a `PARTITION` option to restrict the result to one or more desired partitions. You can use `WHERE` conditions, `ORDER BY` and `LIMIT` options, and so on. You can also use aggregate functions with `HAVING` and `GROUP BY` options. Each of the following queries produces a valid result when run on the `employees` table as previously defined.

```
mysql> SELECT * FROM employees PARTITION (p0, p1)
      WHERE id < 100000;
+----+-----+-----+-----+
| id | first_name | last_name | salary | department_id |
+----+-----+-----+-----+
1 10 | Thomas | King | 11000 | 10
```

Figure 6 Select using partition

Comment:

Its take just 0.3-0.5sec!

Conclusion:

In This Practical word we saw that Fragmentation is very useful, it optimize the execution time of queries.