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

```
-- [INFO]:
---- You must add the file of load_salaries.dump to the same directory of this current file "creations_queries.sql" and run:
------mysql -uroot < creations_queries.sql
---- PS: I ignore the file for github because its large file.

DROP DATABASE IF EXISTS employees;
CREATE DATABASE IF NOT EXISTS employees;
USE employees;

SELECT 'CREATING DATABASE STRUCTURE' as 'INFO';

DROP TABLE IF EXISTS dept_emp,
                     dept_manager,
                     titles,
                     salaries,
                     employees,
                     departments;

   set storage_engine = InnoDB;

select CONCAT('storage engine: ', @@storage_engine) as INFO;

CREATE TABLE salaries (
    emp_no      INT             NOT NULL,
    salary      INT             NOT NULL,
    from_date   DATE            NOT NULL,
    to_date     DATE            NOT NULL
);

SELECT 'LOADING salaries' as 'INFO';
source load_salaries.dump;
```

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:

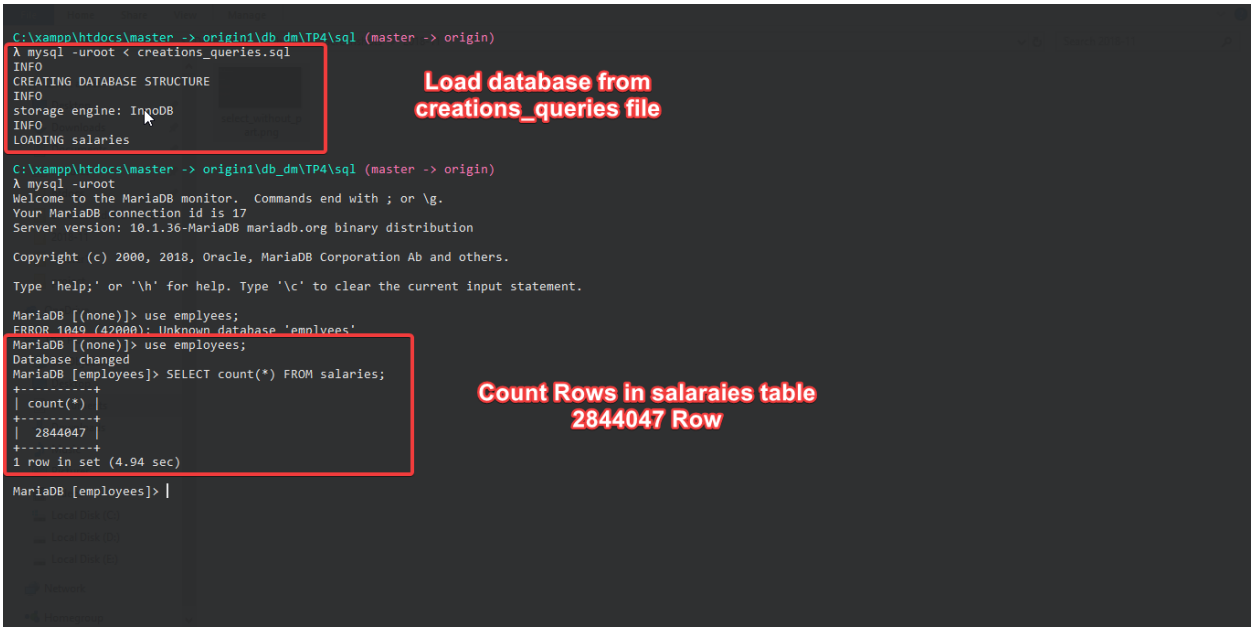

Figure 2 SQL Loader Script

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:

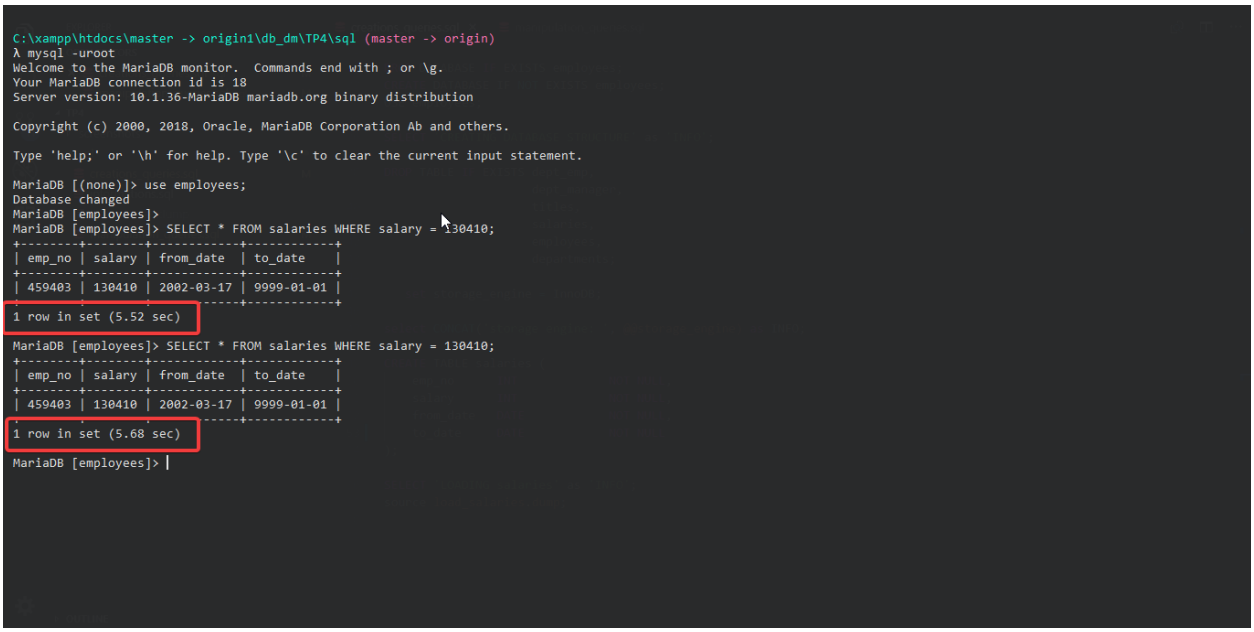

Figure 3 - Execution time without partitions.

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:

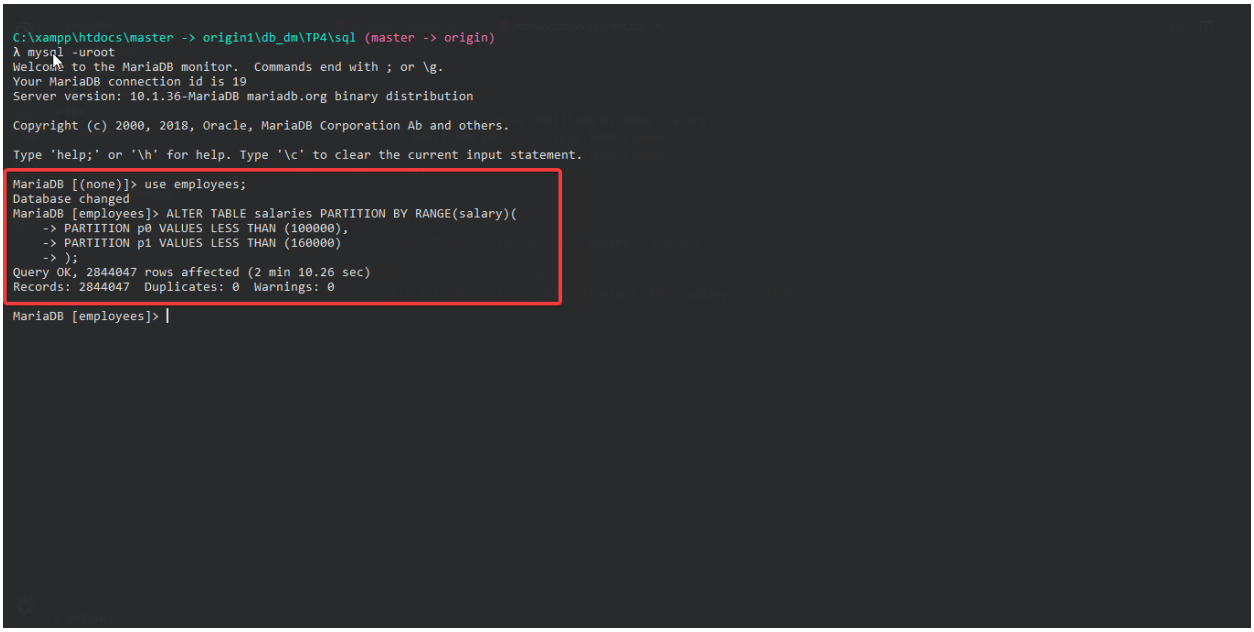

Figure 5 - PARTITION on exesting table

Step Six:

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

The following script show the query and the result:

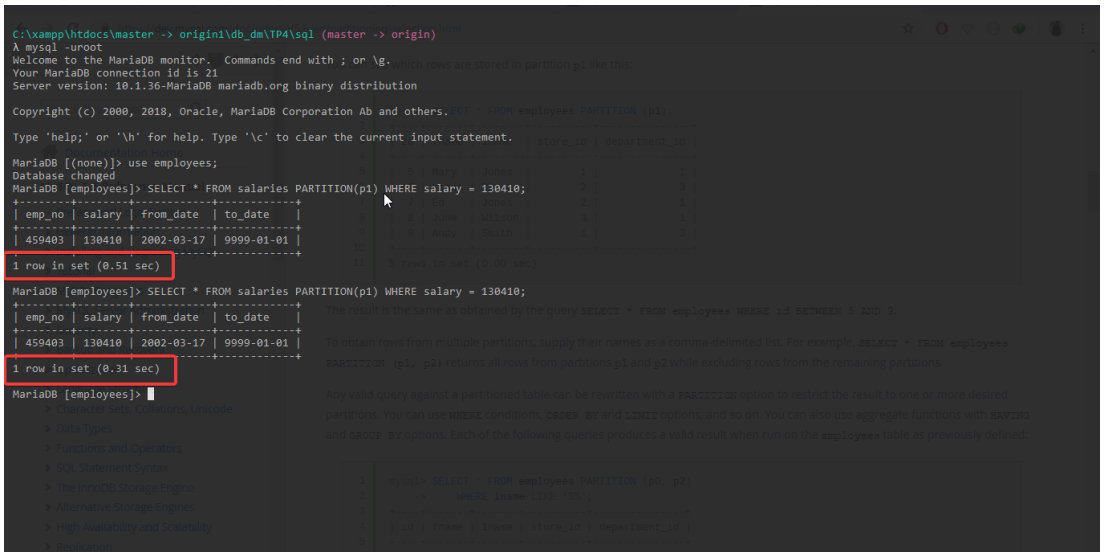

Figure 6 Select using partition

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.