Experiment No. 3

Title: Execution of In-memory database queries

Batch: **Roll No.:** **Experiment No.:2**

Aim: To execute In-memory database queries

Resources needed: MySQL

Theory

In-Memory database is a database that uses a system's main memory for data storage rather than the disk-based storage typically utilized by traditional databases. In-memory databases, or IMDBs, are frequently employed in high-volume environments where response time is critical, as access times and database requests are typically considerably faster when system memory is used as opposed to hard disk storage.

The traditional databases and in-memory databases can be used together and referred as hybrid databases, which support both in-memory and disk-based storage in order to maximize performance as well as reliability of the system. All most all RDBMS systems available in market supports In-Memory databases.

MySQL In-Memory database:

In MySQL DB, the MEMORY storage engine creates special-purpose tables with contents that are stored in memory. Because the data is vulnerable to crashes, hardware issues, or power outages, use of these tables are limited to temporary work areas or read-only caches for data pulled from other tables.

A typical use case for the MEMORY engine involves these characteristics:

- Operations involving transient, non-critical data such as session management or caching. When the MySQL server halts or restarts, the data in MEMORY tables is lost.
- In-memory storage for fast access and low latency. Data volume can fit entirely in memory without causing the operating system to swap out virtual memory pages.
- A read-only or read-mostly data access pattern (limited updates).
- MEMORY tables cannot contain BLOB or TEXT columns.

To create a MEMORY table, specify the clause ENGINE=MEMORY on the CREATE TABLE statement

```
CREATE TABLE EMP (emp_Id INT, name CHAR (30)) ENGINE = MEMORY;
```

As indicated by the engine name, MEMORY tables are stored in memory. They use hash indexes by default, which makes them very fast for single-value lookups, and very useful for creating temporary tables. However, when the server shuts down, all rows stored in MEMORY tables are lost. The tables themselves continue to exist because their definitions are stored in .frm files on disk, but they are empty when the server restarts.

To load the data in memory from other existing table use,

```
CREATE TABLE EMP (emp_Id INT, name CHAR (30))) ENGINE=MEMORY
as SELECT * FROM EMP;
```

To move the data from In-Memory table to hard drive (using any text file) use the following syntax,

```
SELECT * INTO OUTFILE ''emp_data.txt' FROM EMP;
```

To populate a MEMORY table when the MySQL server starts, use the INFILE option. For example,

```
LOAD DATA INFILE 'emp_data.txt' INTO TABLE EMP;
```

Where, emp_data.txt is a data file.

Procedure:

Perform following tasks:

1. Create In-memory table using Engine as Memory.
2. Insert values in that table.
3. Attempt to retrieve values from the table after restarting the database server.
4. Load the data into table using file load.

Results: (Program printout with output)

Questions:

1. **What is the difference between traditional and In-memory databases?**
2. **List applications using in-memory database. Explain any one of it stressing upon advantage of using in-memory database.**

Outcomes:

Conclusion: (Conclusion to be based on outcomes achieved)

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

1. https://dev.mysql.com/doc/refman/5.5/en/memory-storage-engine.html
2. http://opensourceforu.efytimes.com/2012/01/importance-of-in-memory-databases/
3. http://pages.cs.wisc.edu/~jhuang/qual/main-memory-db-overview.pdf
4. http://docs.memsql.com