

Experiment No. 3

Title: Execution of In-memory database queries

Batch:SY-IT(B3)

Roll No.:16010423076

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:

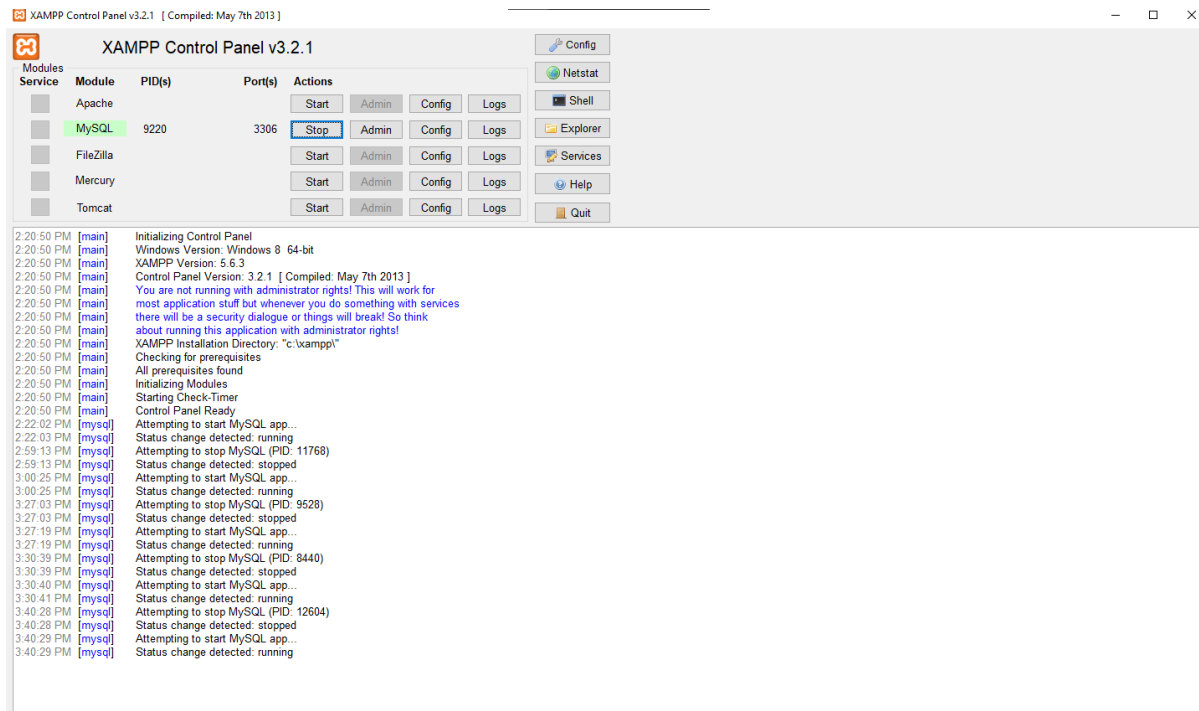
Perform following tasks:

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)

Step 1: Start XAMPP and Open MySQL Shell

1. Open XAMPP Control Panel.
2. Start Apache and MySQL.
3. Click on "Shell" to open the MySQL command-line interface.



Step 2: Select or Create a Database

Run the following command to select or create a database:

```
CREATE DATABASE IF NOT EXISTS InMemoryDB;
USE InMemoryDB;
```

Step 3: Create an In-Memory Table

Run the following SQL command to create a table using the MEMORY engine:

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

Step 4: Insert Values into the In-Memory Table

Insert sample employee records into the EMP table:

```
INSERT INTO EMP (emp_Id, name) VALUES
(1, 'Alice'),
(2, 'Bob'),
```

(3, 'Charlie');

Step 5: Retrieve Data from the In-Memory Table

Verify that the data has been inserted:

SELECT * FROM EMP;

```

XAMPP for Windows - mysql -u root -p

Setting environment for using XAMPP for Windows.
Exam@1601TB314-08 c:\xampp
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE DATABASE IF NOT EXISTS InMemoryDB;
Query OK, 1 row affected (0.00 sec)

mysql> USE InMemoryDB;
Database changed
mysql> CREATE TABLE EMP (
  ->   emp_id INT PRIMARY KEY,
  ->   name CHAR(30)
  -> ) ENGINE=MEMORY;
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO EMP (emp_id, name) VALUES
  -> (1, 'Alice'),
  -> (2, 'Bob'),
  -> (3, 'Charlie');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM EMP;
+----+-----+
| emp_id | name |
+----+-----+
| 1      | Alice|
| 2      | Bob  |
| 3      | Charlie|
+----+-----+
3 rows in set (0.00 sec)

mysql>
  
```

Step 6: Restart the MySQL Server

1. Stop MySQL from XAMPP Control Panel.
2. Start MySQL again.

Open MySQL Shell and select the database:

USE InMemoryDB;

3. Check if data persists in the table:
SELECT * FROM EMP;

4. Since it is an In-Memory table, the data will be lost after restarting.
-

Step 7: Save Data to a File Before Restart

To persist data before restarting MySQL, save it to a file:

SELECT * INTO OUTFILE 'C:/xampp/mysql/data/emp_data.txt'
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
FROM EMP;

```

XAMPP for Windows - mysql -u root -p
+-----+
| 2 | Bob |
| 3 | Charlie |
+-----+
3 rows in set (0.00 sec)

mysql> USE InMemoryDB;
No connection. Trying to reconnect...
Connection id: 1
Current database: *** NONE ***

Database changed
mysql> SELECT * FROM EMP;
Empty set (0.00 sec)

mysql> SELECT * INTO OUTFILE 'C:/xampp/mysql/data/emp_data.txt'
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> FROM EMP;
ERROR 1086 (HY000): File 'C:/xampp/mysql/data/emp_data.txt' already exists
mysql> SELECT * INTO OUTFILE 'C:/xampp/mysql/data/emp_data.txt'
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> FROM EMP;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE EMP (
-> emp_id INT PRIMARY KEY,
-> name CHAR(30)
-> ) ENGINE=MEMORY;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 1
Current database: InMemoryDB

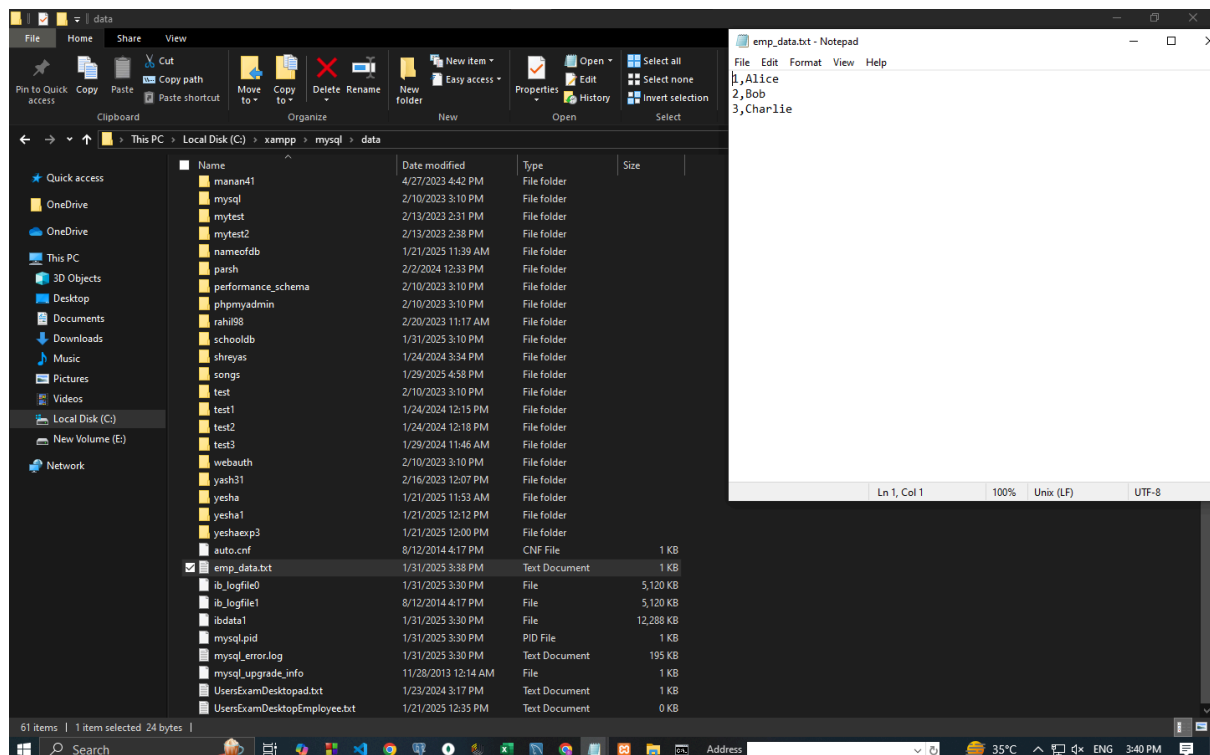
ERROR 1050 (42501): Table 'emp' already exists
mysql> DROP TABLE IF EXISTS EMP;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE EMP (
-> emp_id INT PRIMARY KEY,
-> name CHAR(30)
-> ) ENGINE=MEMORY;
Query OK, 0 rows affected (0.03 sec)

mysql> LOAD DATA INFILE 'C:/xampp/mysql/data/emp_data.txt'
-> INTO TABLE EMP
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n';
Query OK, 0 rows affected (0.01 sec)
Records: 0 Deleted: 0 Skipped: 0 Warnings: 0

mysql>

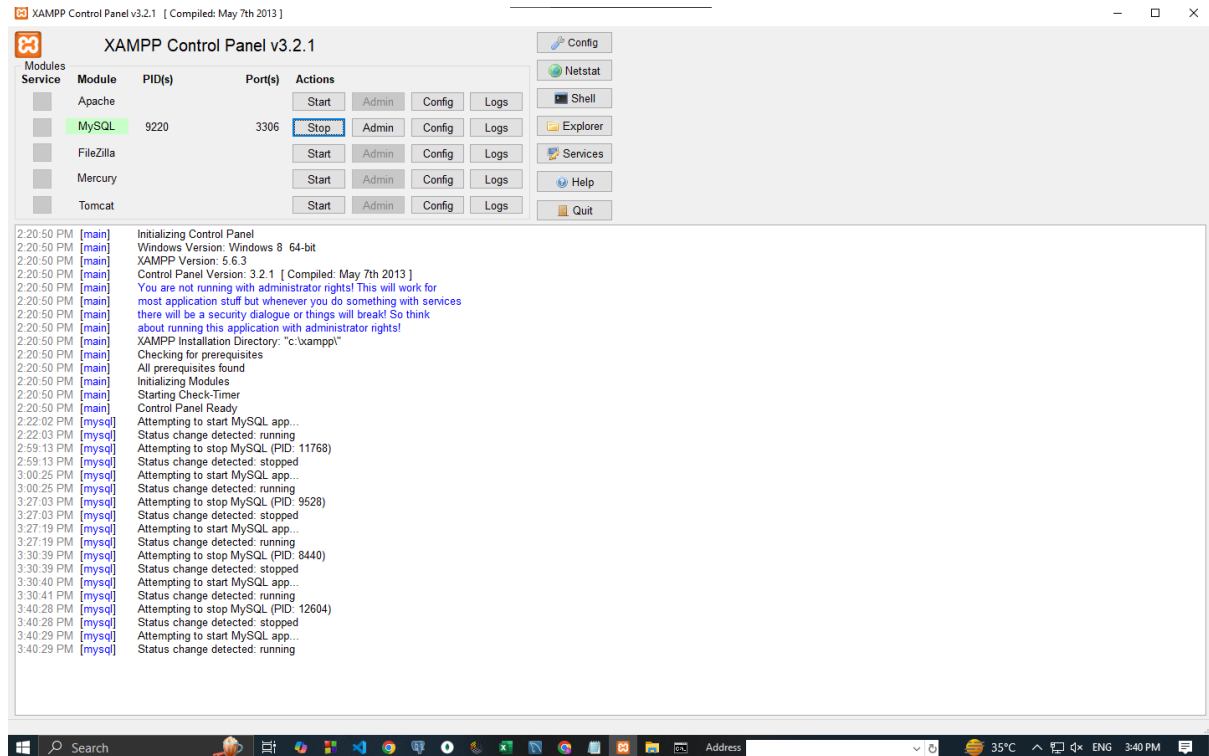
```



Step 8: Restart MySQL and Reload Data

1. Stop and Start MySQL from XAMPP.

Recreate the MEMORY table since the table definition remains but the data is lost:



```

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

2.

Load data from the file back into the table:

```

LOAD DATA INFILE 'C:/xampp/mysql/data/emp_data.txt'
INTO TABLE EMP
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';
  
```

3.

Verify that data is restored:

```

SELECT * FROM EMP;
  
```



```

XAMPP for Windows - mysql -u root -p
-> name CHAR(30)
-> ) ENGINE=MEMORY;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 1
Current database: *** NONE ***

ERROR 1046 (3D000): No database selected
mysql> CREATE TABLE EMP (
-> emp_id INT PRIMARY KEY,
-> name CHAR(30)
-> ) ENGINE=MEMORY;
ERROR 1046 (3D000): No database selected
mysql> USE InMemoryDB;
Database changed
mysql> CREATE TABLE EMP (
-> emp_id INT PRIMARY KEY,
-> name CHAR(30)
-> ) ENGINE=MEMORY;
ERROR 1050 (42S01): Table 'emp' already exists
mysql> DROP TABLE IF EXISTS EMP;
Query OK, 0 rows affected (0.00 sec)

mysql> DROP TABLE IF EXISTS EMP;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql>
mysql> CREATE TABLE EMP (
-> emp_id INT PRIMARY KEY,
-> name CHAR(30)
-> ) ENGINE=MEMORY;
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> LOAD DATA INFILE 'C:/xampp/mysql/data/emp_data.txt'
-> INTO TABLE EMP
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n';
Query OK, 3 rows affected (0.01 sec)
Records: 3 Deleted: 0 Skipped: 0 Warnings: 0

mysql> SELECT * FROM EMP;
+-----+-----+
| emp_id | name |
+-----+-----+
| 1 | Alice |
| 2 | Bob |
| 3 | Charlie |
+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

Questions:

- What is the difference between traditional and In-memory databases?**
Traditional databases use disk storage to save data, which can be slower when accessing large volumes of data due to the time it takes to read from and write to the disk. On the other hand, in-memory databases store data directly in the system's RAM. This makes them much faster because accessing data in memory is quicker than accessing data on disk. Traditional databases are suitable for long-term storage and can handle large datasets with reliability, while in-memory databases are designed for speed, often used for real-time applications or temporary data that does not need to persist long-term.
- List applications using in-memory database. Explain any one of it stressing upon advantage of using in-memory database.**

In-memory databases are used in many applications where speed is crucial. One example is caching in web applications. Caching involves temporarily storing frequently accessed data in memory to speed up response times for users. For example, social media platforms and e-commerce websites use in-memory databases to cache user profiles or product information.

Example: Web Caching in E-Commerce

In an e-commerce website, customer browsing data and product information are often cached in-memory. This allows the website to quickly load product pages, reducing the time spent waiting for data from a traditional disk-based database. The advantage of using an in-memory database here is that it greatly improves user experience by providing faster page loads and quicker search results. This is especially important

when traffic spikes during sales or promotional events, where real-time speed is critical for maintaining performance.

Outcomes:

CO2: Design advanced database systems using In-memory, Spatial and NOSQL databases and its implementation.

Conclusion: (Conclusion to be based on outcomes achieved)

From this experiment, I learned how in-memory databases operate differently from traditional disk-based databases and the advantages they offer in terms of speed and efficiency. By creating and manipulating in-memory tables, I gained hands-on experience with the MEMORY storage engine in MySQL and explored how quickly data can be processed and retrieved when stored in system memory. I also understood the practical applications of in-memory databases, especially in scenarios requiring real-time access to data, such as web caching in e-commerce. This experiment highlighted the trade-offs between data persistence and speed, emphasizing when and why in-memory databases are used for performance optimization.

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>