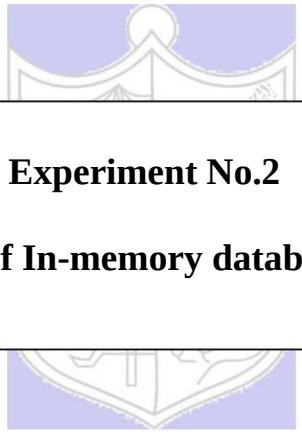


Experiment No.2

Title: Execution of In-memory database queries

Batch: B1 **Roll No.:** 16010420133 **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)

CREATING AN EMP TABLE IN MAIN MEMORY:

Query:

```
CREATE TABLE EMP (emp_id INT, emp_name CHAR (30), emp_gender CHAR(10),
emp_age INT, PRIMARY KEY(emp_id)) ENGINE = MEMORY;
```

OUTPUT:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0441 seconds.)

```
CREATE TABLE EMP (emp_id INT, emp_name CHAR (30), emp_gender CHAR(10), emp_age INT, PRIMARY
KEY(emp_id)) ENGINE = MEMORY
```

INSERTING VALUES IN TABLE:

Query:

```
insert into EMP values(101, "Pradnya", “Female”, 19);
insert into EMP values(102,"Pranav", “Male”, 24);
insert into EMP values(103,“Deepak”, “Male”, 51);
insert into EMP values(104,"Jyoti", “Female”, 46);
```

OUTPUT:

1 row inserted. (Query took 0.0060 seconds.)

```
insert into EMP values(101, "Pradnya", "Female", 19)
```

1 row inserted. (Query took 0.0008 seconds.)

```
insert into EMP values(102,"Pranav", "Male", 24)
```

1 row inserted. (Query took 0.0007 seconds.)

```
insert into EMP values(103,"Deepak", "Male", 51)
```

1 row inserted. (Query took 0.0008 seconds.)

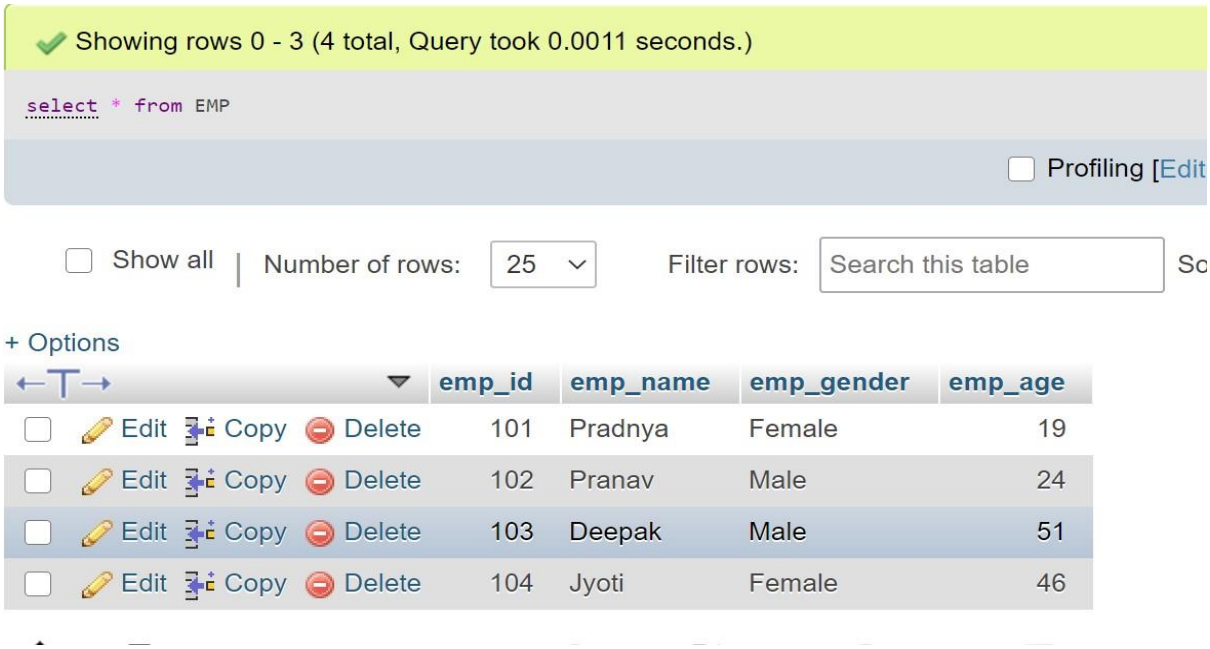
```
insert into EMP values(104,"Jyoti", "Female", 46)
```

DISPLAY EMP TABLE:

Query:

```
Select * from EMP;
```

OUTPUT:

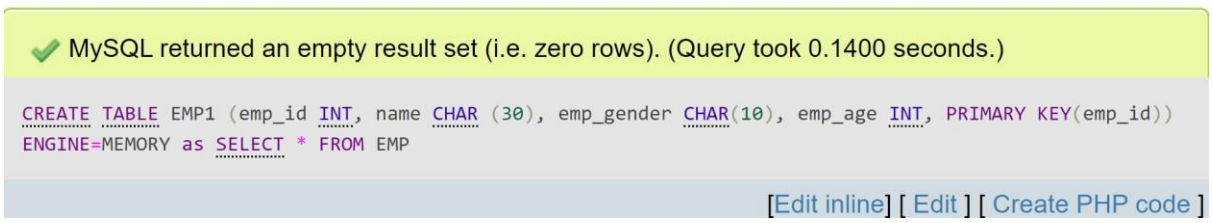


LOADING THE DATA IN MEMORY FROM OTHER EXISTING TABLE:

Query:

```
CREATE TABLE EMP1 (emp_id INT, name CHAR (30), emp_gender CHAR(10), emp_age INT, PRIMARY KEY(emp_id)) ENGINE=MEMORY as SELECT * FROM EMP;
```

OUTPUT:

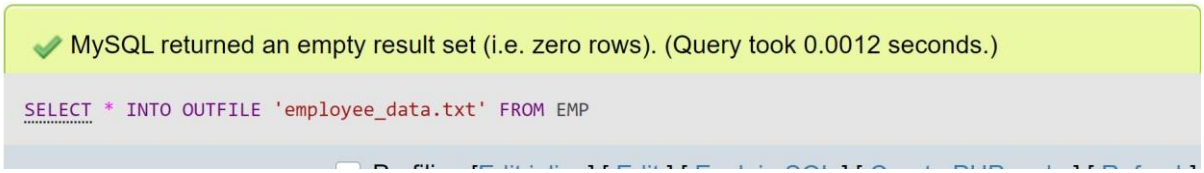


MOVING THE DATA FROM IN-MEMORY TABLE TO HARD DRIVE (USING ANY TEXT FILE)

Query:

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

OUTPUT:

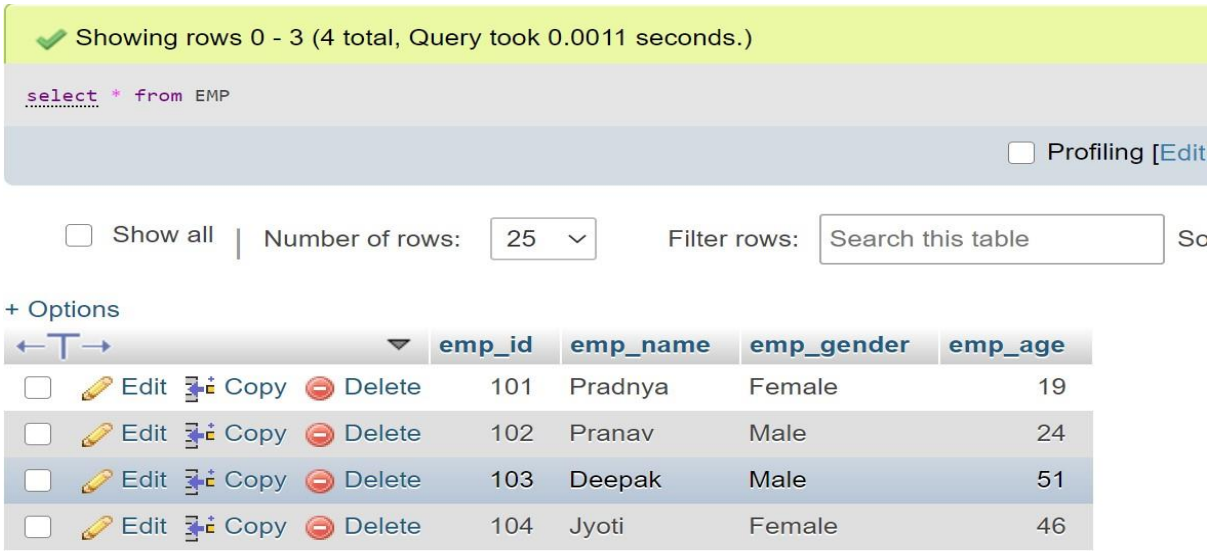


DISPLAYING MEMORY TABLE AND THE NEW TABLE WITH THE DUPLICATE VALUES:

Query:

```
SELECT * FROM EMP;
```

OUTPUT:



	emp_id	emp_name	emp_gender	emp_age
Edit Copy Delete	101	Pradnya	Female	19
Edit Copy Delete	102	Pranav	Male	24
Edit Copy Delete	103	Deepak	Male	51
Edit Copy Delete	104	Jyoti	Female	46

Query:

```
SELECT * FROM EMP1;
```

OUTPUT:

Showing rows 0 - 3 (4 total, Query took 0.0010 seconds.)

```
SELECT * FROM EMP1
```

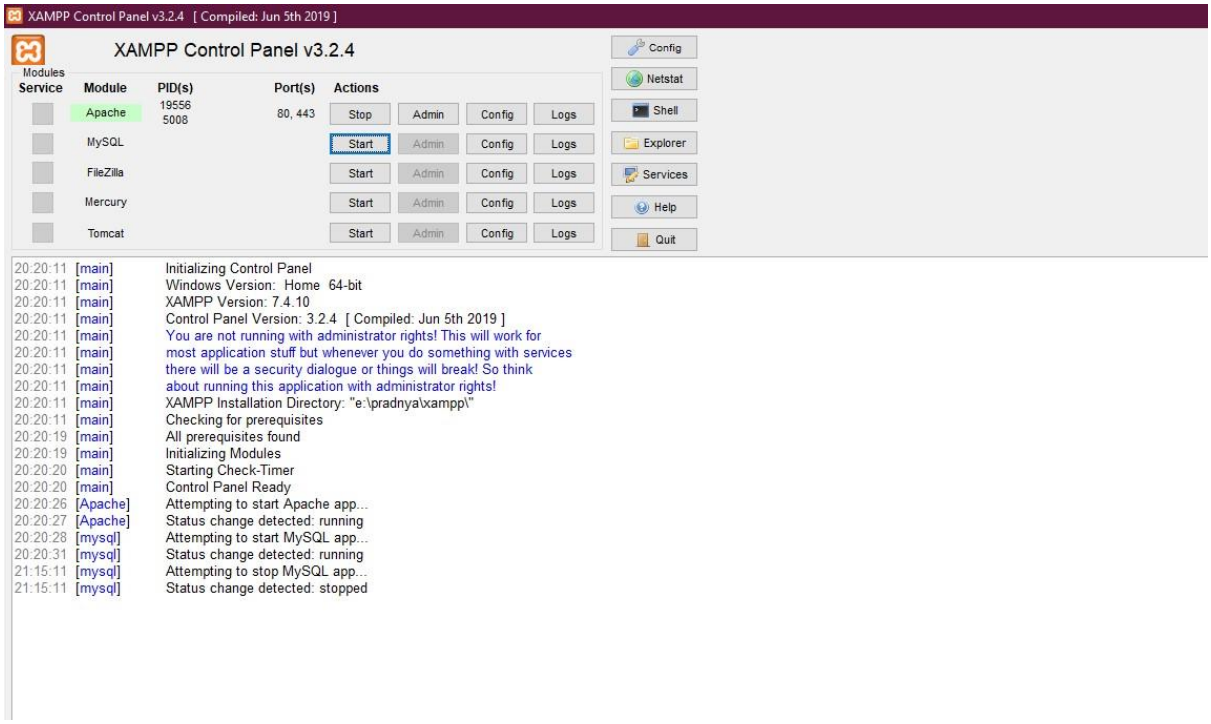
Profiling [Edit inline] [

Show all | Number of rows: 25 Filter rows: Search this table Sort by ke

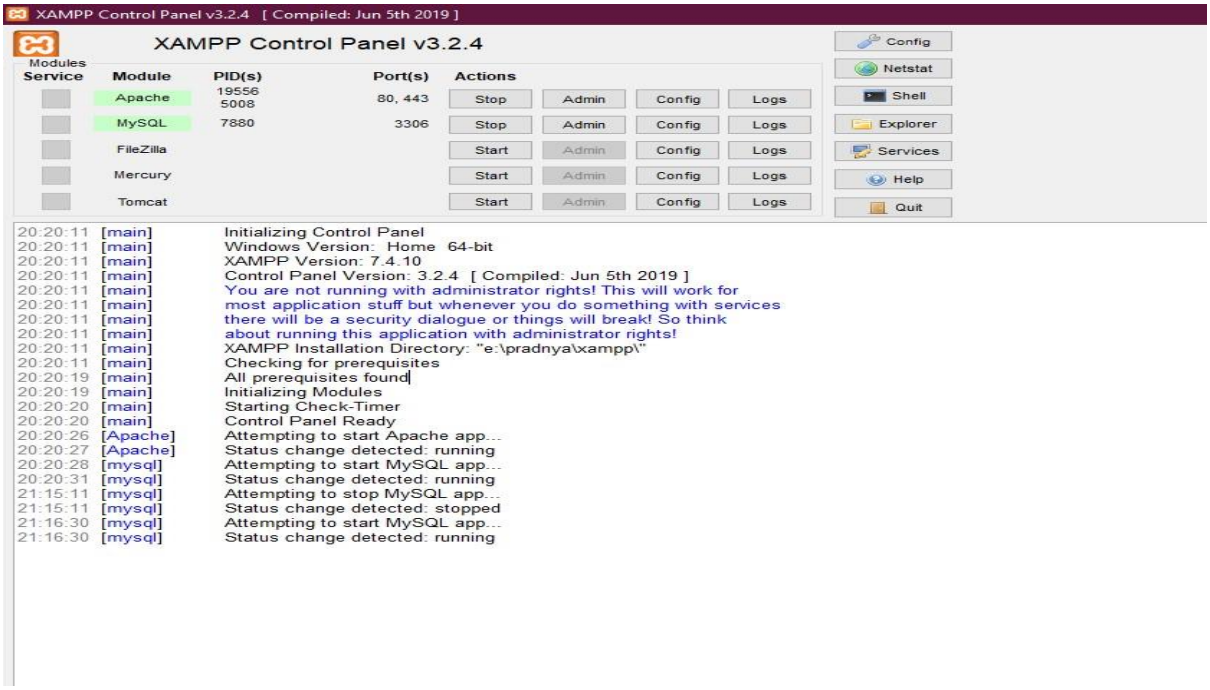
+ Options

	name	emp_id	emp_name	emp_gender	emp_age
Edit Copy Delete	NULL	101	Pradnya	Female	19
Edit Copy Delete	NULL	102	Pranav	Male	24
Edit Copy Delete	NULL	103	Deepak	Male	51
Edit Copy Delete	NULL	104	Jyoti	Female	46

SERVER STOP (stop the mysql from the xampp control panel):



SERVER RUNNING (Start the mysql from the xampp control panel again):

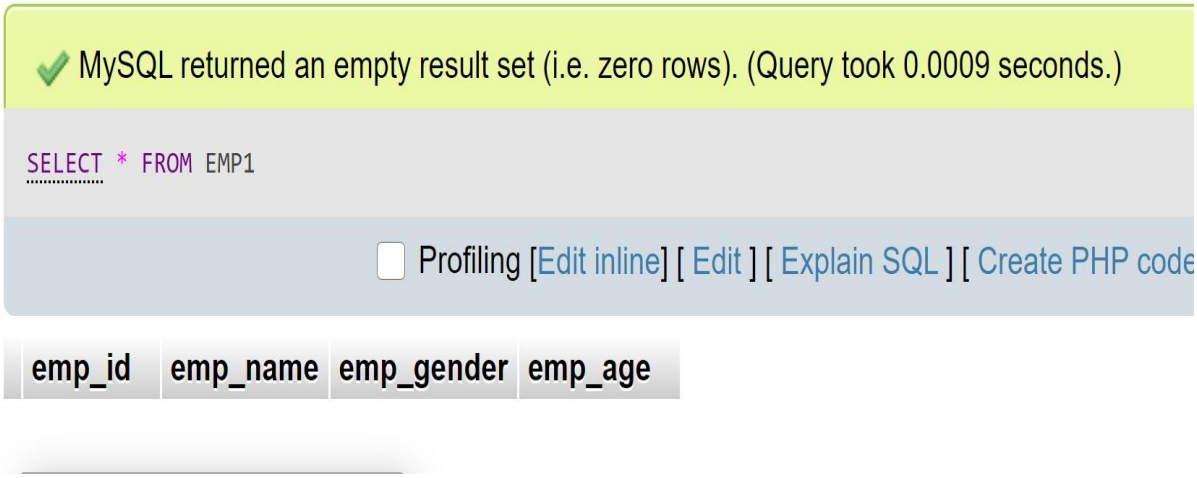


AFTER STARTING THE SERVER THE VALUES IN BOTH TABLE WILL BE LOST:

Query:

```
SELECT * FROM EMP1;
```

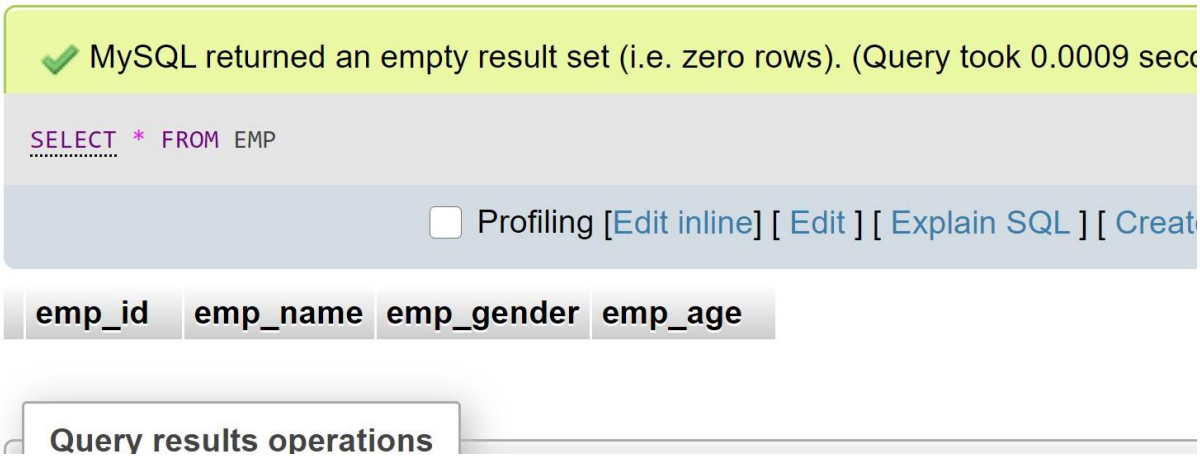
OUTPUT:



Query:

```
SELECT * FROM EMP;
```

OUTPUT:

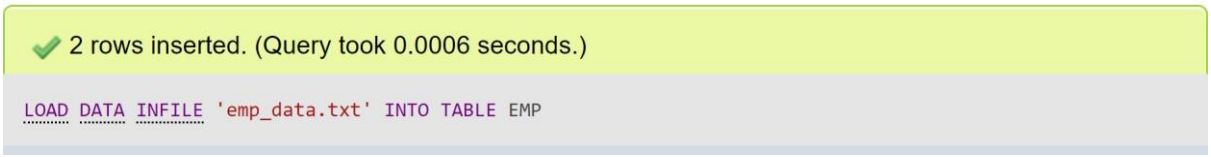

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 secc
SELECT * FROM EMP
Profiling [Edit inline] [Edit] [Explain SQL] [Creat
emp_id
emp_name
emp_gender
emp_age
Query results operations

TO POPULATE A MEMORY TABLE WHEN THE MYSQL SERVER (using xampp) STARTS, USE THE INFILE OPTION:

Query:

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

OUTPUT:



Query:

```
SELECT * FROM EMP;
```

OUTPUT:

Showing rows 0 - 3 (4 total, Query took 0.0009 seconds.)

```
SELECT * FROM emp
```

Show all | Number of rows: 25 Filter rows: Search this

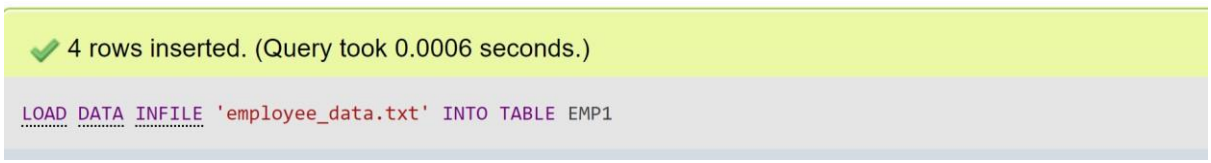
+ Options

emp_id	emp_name	emp_gender	emp_age
101	Pradnya	Female	19
102	Pranav	Male	24
103	Deepak	Male	51
104	Jyoti	Female	46

Query:

```
LOAD DATA INFILE 'emp_data.txt' INTO TABLE EMP1;
```

OUTPUT:

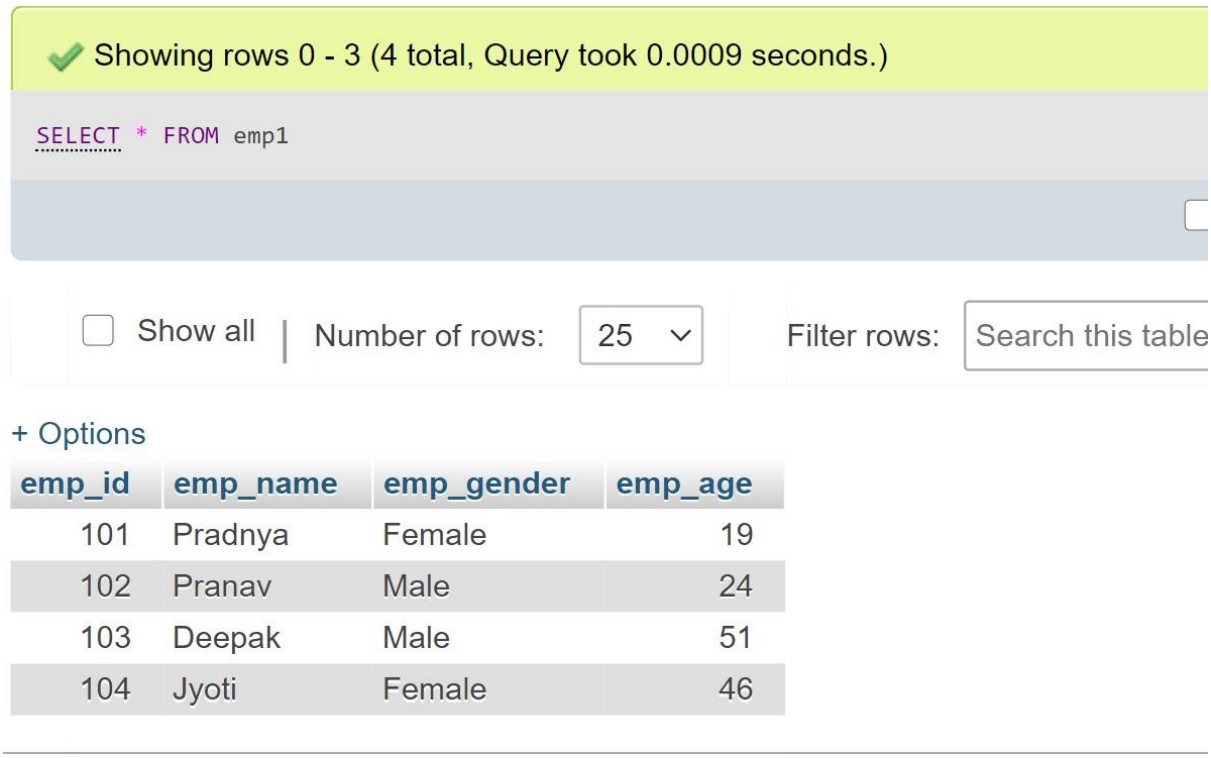


AFTER LOADING THE CONTENTS FROM THE TEXT FILE <THE TABLES WILL HAVE ITS CONTENTS:

Query:

Page No:SELECT * FROM EMP1;

OUTPUT:



emp_id	emp_name	emp_gender	emp_age
101	Pradnya	Female	19
102	Pranav	Male	24
103	Deepak	Male	51
104	Jyoti	Female	46

Questions:

1. What is the difference between traditional and In-memory databases? ANS: An in-memory database (IMDB; also main memory database system or MMDB or memory resident database) is a database management system that primarily relies on main memory for computer data storage. It is contrasted with database management systems that employ a disk storage mechanism. Main memory databases are faster than disk-optimized databases since the internal optimization algorithms are simpler and execute fewer CPU instructions. Accessing data in memory eliminates seek time when querying the data, which provides faster and more predictable performance than disk.

Applications where response time is critical, such as those running telecommunications network equipment and mobile advertising networks, often use main-memory databases.

In reply to your query, yes it loads the data in RAM of your computer.

On-Disk Databases

- All data stored on disk, disk I/O needed to move data into main memory when needed.
- Data is always persisted to disk.
- Traditional data structures like B-Trees designed to store tables and indices efficiently on disk.
- Virtually unlimited database size.
- Support very broad set of workloads, i.e. OLTP, data warehousing, mixed workloads, etc.

Page No:

In-Memory Databases

- All data stored in main memory, no need to perform disk I/O to query or update data.
- Data is persistent or volatile depending on the in-memory database product.
- Specialized data structures and index structures assume data is always in main memory.
- Optimized for specialized workloads; i.e. communications industry-specific HLR/HSS workloads.
- Database size limited by the amount of main memory.

2. List applications using in-memory database. Explain any one of it stressing upon advantage of using in-memory database.

ANS: In-memory databases are commonly used for:

 Real-time banking, retail, advertising, medical device analytics, machine learning and billing/subscriber applications
 Online interactive gaming
 Geospatial/GIS processing
 Processing of streaming sensor data
 Developing embedded software systems
 Applications in transport systems, network switches and routers
 Fulfilling the requirements of e-commerce applications Online

Interactive Gaming:

A relative gaming leaderboard shows a gamer's position relative to other players of a similar rank. A relative gaming leaderboard can help to build engagement among players and meanwhile keep gamers from becoming demotivated when compared only to top players. For a game with millions of players, in-memory databases can deliver sorting results quickly and keep the leaderboard updated in real time.

Outcomes:

CO1: Design advanced database systems using Parallel, Distributed and In-memory Databases and its implementation.

Conclusion: (Conclusion to be based on outcomes achieved)

The implementation of in-memory database is seen in this experiment where we could observe how data is stored in RAM and how it can be retrieved even after server restarts. Also understood the reasons behind practical application of in-memory database in fields like real -time analytics, gaming etc.

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