



INSTITUTE OF INFORMATICS AND COMMUNICATION

M.Sc. Informatics

(2022-2024)

Semester - 3

COMPUTER COMMUNICATION AND NETWORKS

(ITEC-304)

SUBMITTED BY: -

NAME - BHARAT AGGARWAL

ROLL NO - 22/1408

SUBMITTED TO: -

Mr. Unmesh Shukla

Question 1: Can you store these data objects in a MySQL table along with a primary key (INT type) ID column and a DESCRIPTION column (TEXT type)?

Answer:

Storing large data objects (e.g., files of 10 MB or more) directly in a MySQL table, especially within a TEXT column, is not typically recommended. While it is technically possible to store such data in a database, doing so can have several drawbacks and is not considered a best practice for large files or multimedia content. Here are some issues you might encounter:

- **Performance:** Storing large data objects in a MySQL table can negatively impact database performance. Retrieving or updating these objects can be slow and resource-intensive, especially if many users are accessing the database concurrently.
- **Database Bloat:** Large data objects can quickly lead to database bloat, which can result in slower backups, longer query times, and increased storage costs.
- **Lack of Indexing:** TEXT columns are typically not indexed for searching or sorting, making it inefficient to search for specific content within these large objects.
- **Limited Scalability:** As your dataset grows, it becomes increasingly challenging to scale your database, and it may require more resources to handle the large data objects effectively.
- **Complexity:** Managing large data objects in a database introduces complexity in your application code and database maintenance.

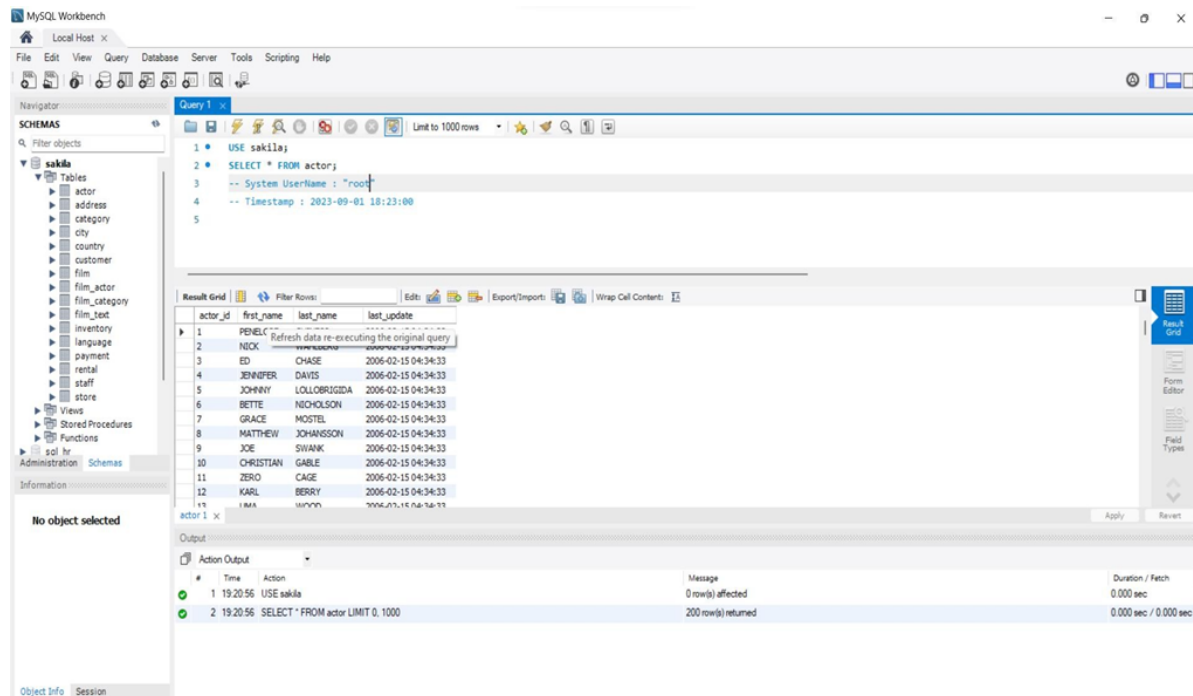
Question 2: According to you, what can be an ideal solution for storing the information of such objects in a table?

Answer: Instead of storing large data objects directly in the database, here's a more common and recommended approach:

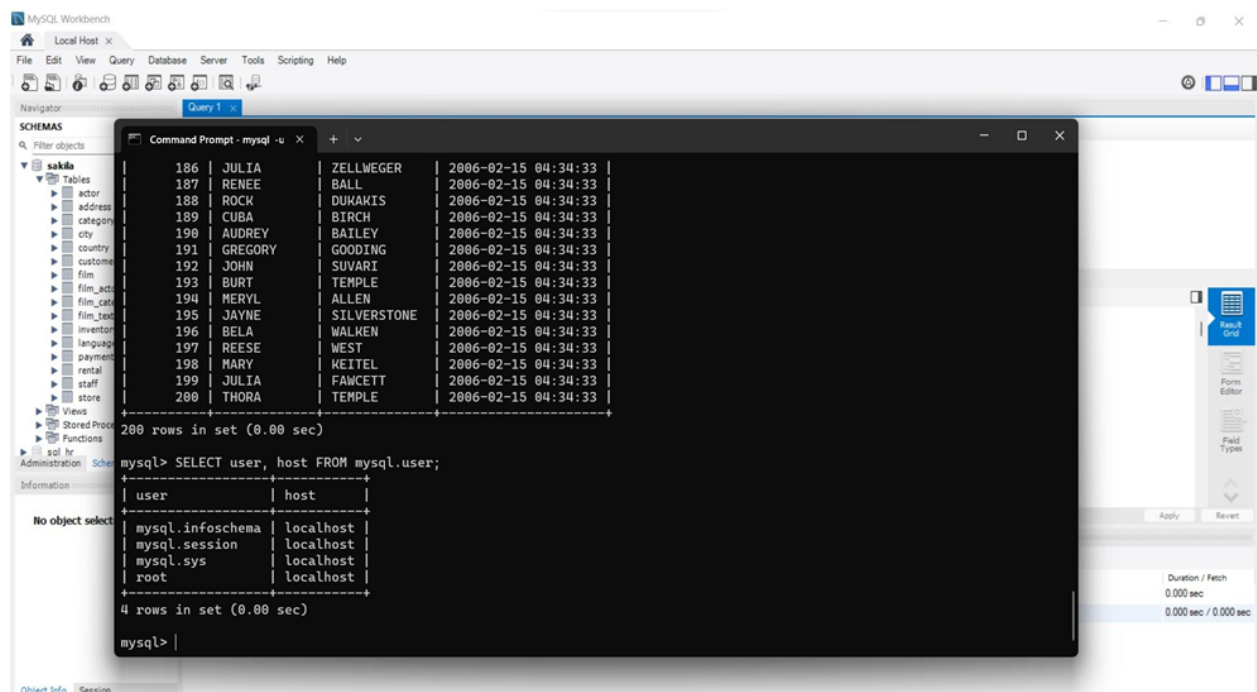
- **File System Storage:** Store the large data objects (multimedia files, large text documents, etc.) in the file system of your server or a dedicated file storage service like Amazon S3, Google Cloud Storage, or a content delivery network (CDN). Each object should be saved as a file, and you can store the file paths or references in the database.
- **Database Metadata:** In your database, store metadata about these objects in a structured manner. This metadata might include information like the file name, description, file path or URL, file size, date uploaded, and any other relevant attributes. You can use a table structure similar to what you suggested, with an ID column and a DESCRIPTION column, to manage this metadata.

Deliverables:

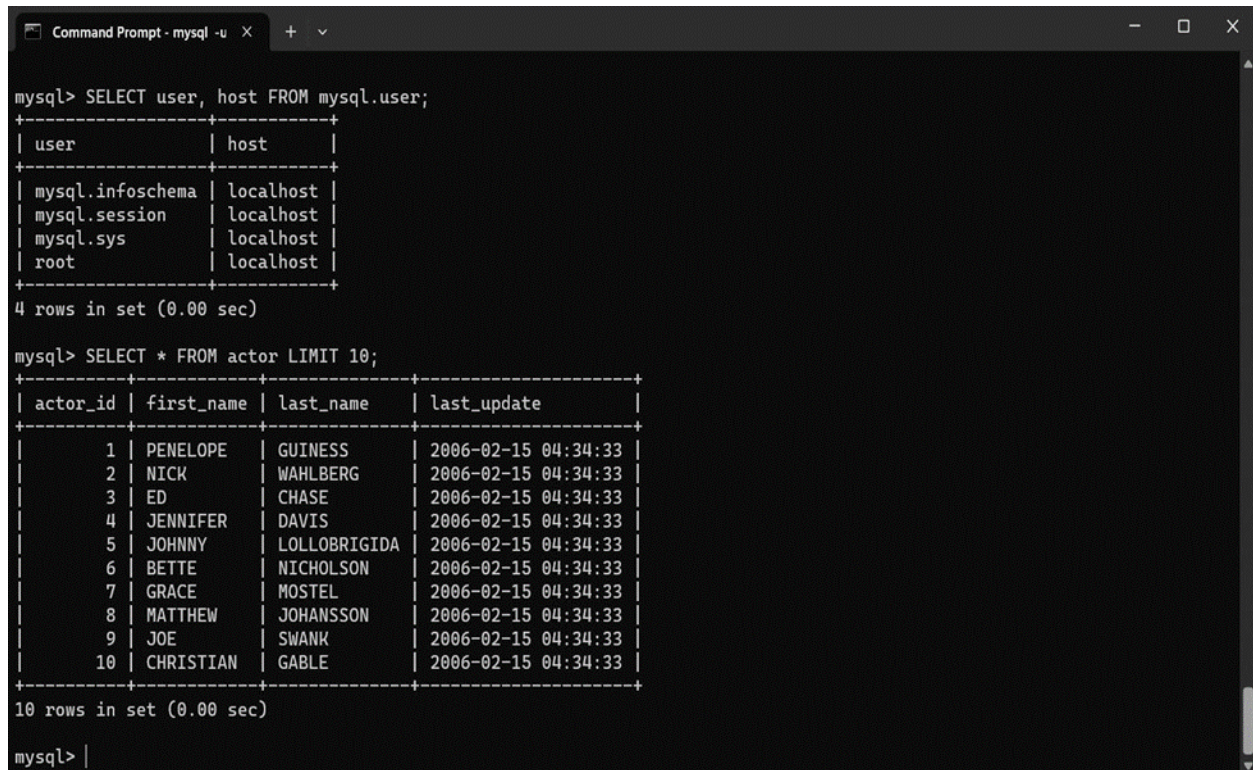
1. Screenshot of timestamp and system user name with MySQL workbench open.



2. Screenshot of MySQL running on terminal or command prompt.



3. Screenshot of the top 10 objects retrieved from the table using SELECT query.



```
mysql> SELECT user, host FROM mysql.user;
+-----+-----+
| user          | host      |
+-----+-----+
| mysql.infoschema | localhost |
| mysql.session   | localhost |
| mysql.sys       | localhost |
| root            | localhost |
+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM actor LIMIT 10;
+-----+-----+-----+-----+
| actor_id | first_name | last_name | last_update |
+-----+-----+-----+-----+
| 1 | PENELOPE | GUINNESS | 2006-02-15 04:34:33 |
| 2 | NICK | WAHLBERG | 2006-02-15 04:34:33 |
| 3 | ED | CHASE | 2006-02-15 04:34:33 |
| 4 | JENNIFER | DAVIS | 2006-02-15 04:34:33 |
| 5 | JOHNNY | LOLLOBRIGIDA | 2006-02-15 04:34:33 |
| 6 | BETTE | NICHOLSON | 2006-02-15 04:34:33 |
| 7 | GRACE | MOSTEL | 2006-02-15 04:34:33 |
| 8 | MATTHEW | JOHANSSON | 2006-02-15 04:34:33 |
| 9 | JOE | SWANK | 2006-02-15 04:34:33 |
| 10 | CHRISTIAN | GABLE | 2006-02-15 04:34:33 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> |
```