Established – 1961 Subject: OSDBMS **SEVA SADAN'S**

R. K. TALREJA COLLEGE

OF

ARTS, SCIENCE & COMMERCE

ULHASNAGAR – 421003

CERTIFICATE

This is to certify that Mr./Ms. AYUSH RAJARAM VIKAL of S.Y. Information Technology (SYIT) Roll No. 2542051 has Satisfactorily completed the Open-Source Data Base Management System Mini Project entitled Municipal Tax Collection Database Management System during the academic year 2025 – 2026, as a part of the practical requirement. The project work is found to be satisfactory and is approved for Submission.

PROF. INCHARGE **HEAD OF DEPT**

INDEX

1. INTRODUCTION

A Secure Multi-User Database System is designed to allow multiple authorized users to access a database while maintaining data security, integrity, and confidentiality. In a multi-user environment, different users perform different operations such as data entry, updating records, report generation, and auditing.
This project implements a secure database using MySQL and Structured Query Language (SQL). Security is achieved using **GRANT** and **REVOKE** commands, which control user access to database objects.
The system demonstrates how role-based access control can be applied to protect sensitive information and ensure that users can only perform operations according to their responsibilities.

---

2. PROBLEM DEFINITION

In many organizations, multiple users access the same database simultaneously. Without proper security mechanisms:

- Unauthorized users may modify or delete records.
- Sensitive data may be exposed.
- Data integrity may be compromised.
- There may be misuse of administrative privileges.

Traditional systems without proper access control create security risks. Therefore, a secure multi-user database using privilege management is required to ensure safe and controlled database access.

3. OBJECTIVES OF THE PROJECT

The objectives of this project are:

- To design a secure multi-user database system.
- To implement Role-Based Access Control (RBAC).
- To use GRANT and REVOKE commands effectively.
- To restrict unauthorized access to sensitive data.
- To maintain data integrity and confidentiality.
- To provide practical understanding of database security concepts.

4. SCOPE OF THE PROJECT

The scope of this project includes:

- Creating database users.
- Assigning privileges based on roles.
- Restricting access using GRANT and REVOKE.
- Testing user permissions.

Limitations:

- The project is limited to database-level security.
- It does not include network-level or application-level security.
- It is implemented for academic purposes only.

5. REQUIREMENT SPECIFICATION

5.1 Hardware Requirements

- Computer/Laptop
- Minimum 4GB RAM
- 10GB Free Storage

5.2 Software Requirements

Software	Purpose
MySQL Server	Database Management
MySQL Workbench	Query Execution
Windows/Linux OS	Platform
SQL	Query Language

---

6. SYSTEM DESIGN

The system follows a simple client-server architecture:

1. Users connect to the MySQL Server.
2. Authentication is performed.
3. Based on assigned privileges, users can perform allowed operations.
4. Unauthorized actions are denied automatically by the database system.

User Roles in the System

1. **Administrator** – Full database control
2. **Data Entry User** – Insert and update data
3. **Auditor** – Read-only access

This design ensures separation of duties and secure data management.

---

7. DATABASE DESIGN

7.1 Sample Database

```
CREATE DATABASE SecureDB;
USE SecureDB;
```

7.2 Sample Table

```
CREATE TABLE Employees (
    Emp_ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Department VARCHAR(50),
    Salary DECIMAL(10,2)
);
```

The database contains sensitive information such as employee salary, which requires restricted access.

---

8. UML DIAGRAMS

8.1 Use Case Diagram (Conceptual)

Actors:

- Administrator
- Data Entry User
- Auditor

Use Cases:

- Create User
- Grant Privileges
- Insert Data
- View Records
- Revoke Access

8.2 Sequence Flow

1. User Login
2. Authentication
3. Privilege Verification
4. Query Execution

5. Access Granted or Denied

---

9. SQL IMPLEMENTATION

9.1 Creating Users

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY
'admin123';
CREATE USER 'data_user'@'localhost' IDENTIFIED
BY 'data123';
CREATE USER 'auditor'@'localhost' IDENTIFIED BY
'audit123';
```

9.2 Granting Privileges

Grant All Privileges to Admin

```
GRANT ALL PRIVILEGES ON Secure DB. *
TO 'admin'@'localhost';
```

Grant Limited Privileges to Data User

```
GRANT SELECT, INSERT, UPDATE
ON SecureDB.Employees
TO 'data_user'@'localhost';
```

Grant Read-Only Access to Auditor

```
GRANT SELECT
ON SecureDB.Employees
TO 'auditor ‘@’localhost';
```

Apply changes:

```
FLUSH PRIVILEGES;
```

9.3 Revoking Privileges

Revoke update permission:

REVOKE UPDATE
ON SecureDB.Employees
FROM 'data_user'@'localhost';
Revoke all privileges:
REVOKE ALL PRIVILEGES
ON Secure DB. *
FROM 'auditor'@'localhost';

---

10. SYSTEM TESTING AND RESULT

Testing was conducted to verify:

- Admin can perform all operations.
- Data user can insert and update but cannot delete.
- Auditor can only view data.
- Unauthorized actions generate error messages.

Result:
The system successfully enforced role-based restrictions using GRANT and REVOKE commands.

---

11. SECURITY, BACKUP AND RECOVERY

11.1 Security

- Role-Based Access Control implemented.
- Unauthorized access restricted.
- Sensitive data protected.

11.2 Backup

Backup using:
mysqldump -u root -p SecureDB > backup.sql

11.3 Recovery

Restore database:
mysql -u root -p SecureDB < backup.sql

These mechanisms ensure data safety and reliability.

---

12. FUTURE SCOPE AND CONCLUSION

Future Scope

- Integration with web-based authentication systems.
- Implementation of encrypted connections (SSL).
- Implementation of password policies.
- Multi-level security roles.

Conclusion

The Secure Multi-User Database using GRANT and REVOKE successfully demonstrates how database-level security can be implemented in MySQL.

The system ensures:

- Controlled access
- Data confidentiality
- Data integrity
- Operational security

Thus, the project provides practical knowledge of database security mechanisms and demonstrates effective privilege management in a multi-user environment.

---

13. REFERENCES

- MySQL Official Documentation – Oracle Corporation
- Database Management System Textbooks
- SQL Tutorials (W3Schools, TutorialsPoint)
- Academic DBMS Reference Materials

---

14. GLOSSARY

DBMS: Software used to manage databases.
SQL: Structured Query Language used to manage data.
GRANT: SQL command used to provide privileges to users.
REVOKE: SQL command used to remove privileges from users.
Role-Based Access Control (RBAC): Security model that assigns permissions based on user roles.
MySQL: Open-source relational database management system used in this project.