ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25jul1399

Design and Performance Analysis of a Secure Multi-User Database with Role- Based Access Control and Audit Trail

Pratiksha Patil[1]

[1]Department of Computer Science

Publication Date: 2025/07/28

Abstract: In multi-user database systems, guaranteeing both secure access and traceability is essential. This research introduces a relational database architecture leveraging Role-Based Access Control (RBAC) and comprehensive Audit Logging. Users are classified into Admin, Editor, and Viewer roles, each assigned distinct permissions. Every database operation is recorded to ensure accountability. Through performance tests measuring query latency and storage impact, we find that adding role enforcement and logging results in a minor slowdown—well within acceptable limits for most institutional use cases. This balance makes the system practical for environments demanding both security and transparency.

Keywords: *Database Security, Role-Based Access Control, Audit Trail, Multi-User System, SQL, Access Management.*

How to Cite: Pratiksha Patil (2025), Design and Performance Analysis of a Secure Multi-User Database with Role- Based Access Control and Audit Trail. *International Journal of Innovative Science and Research Technology*, 10(7), 2184. https://doi.org/10.38124/ijisrt/25jul1399

I. INTRODUCTION

Modern database systems often require strict control over who can access data and a clear record of who performed which actions. Traditional permission models lack granularity and accountability. This paper presents an integrated system that enforces role-specific access and tracks all operations, ensuring both security and traceability.

II. RELATED WORK

RBAC simplifies the assignment of permissions through roles, reducing administrative overhead and improving security. Audit trails, meanwhile, enable forensic analysis and policy compliance. However, very few implementations combine both techniques in a cohesive academic solution—that's the gap this work addresses.

III. METHODOLOGY / SYSTEM DESIGN

The proposed system defines three main roles: Admin, Editor, and Viewer. Each role has associated privileges that dictate how users interact with the database. Audit logs capture every action, including timestamp, user ID, and operation type. MySQL is used as the backend, with Python (Tkinter or Flask) handling the interface and access logic.

IV. RESULTS AND DISCUSSION

Performance was measured in terms of SELECT and INSERT query times, both with and without logging. The addition of audit trail logging introduced an average latency increase of 18–33%, which is acceptable given the gain in accountability. Log storage per 1000 operations was approximately 210 KB.

V. CONCLUSION AND FUTURE SCOPE

This paper demonstrates that integrating RBAC with audit trails offers a secure, scalable, and traceable database solution with manageable performance overhead. Future work includes implementing data encryption, multi-factor authentication, and extending the system to distributed or cloud databases.

ACKNOWLEDGEMENTS

The authors would like to thank the faculty and peers for their valuable input during the development of this project.

REFERENCES

[1]. Sandhu, R.S., et al. 'Role-Based Access Control Models.' IEEE Computer, 1996.
[2]. Vacca, J.R. Computer and Information Security Handbook. Elsevier, 2012.