Cambodia Academy of Digital Technology

MySQL vs PostgreSQL Report

Group : SE G3

Lecturer : Mr. Thea Sophol

Team: Group 6

Members: Prak Pychey & Lim Lyheang

Key Similarities

Authentication & Privileges: Both systems provide comprehensive authentication mechanisms and support fine-grained privilege control at database, table, and column levels.

Role-Based Access Control: Both support RBAC, allowing administrators to group permissions into roles and assign them to users efficiently.

SQL-Based Management: Both use standard SQL commands (CREATE USER, GRANT, REVOKE) for managing users, roles, and privileges.

Privilege Inheritance: Both systems allow roles to be granted to users, enabling efficient permission management through role membership.

Key Differences

Conceptual Architecture

MySQL: Maintains clear separation between users and roles. Users are created with `CREATE USER`, roles with `CREATE ROLE`, then roles are assigned to users using `GRANT role TO user`.

PostgreSQL: Users and roles are the same entity. `CREATE USER` is just an alias for `CREATE ROLE` with login privileges. This unified approach treats everything as roles.

Host-Based Access Control

MySQL: User accounts are tied to specific hosts (e.g., 'username'@'localhost'). Each user-host combination is a separate account, providing strict network-level security but requiring multiple entries for different connection sources.

PostgreSQL: Uses pg_hba.conf for host-based authentication, separating connection control from user identity. Roles can connect from multiple hosts without duplicate accounts.

Role Management

MySQL: Roles must be explicitly activated in sessions. Clear hierarchy where users receive roles, and role privileges are applied when activated.

PostgreSQL: Automatic privilege inheritance through role membership. More sophisticated nested role relationships with flexible inheritance models.

Practical Impact

MySQL provides clarity through distinct user-role separation and explicit host-based security, making it straightforward for traditional access control scenarios.

PostgreSQL offers greater flexibility with its unified role system, better suited for complex enterprise environments requiring dynamic privilege management.

Conclusion

Both systems effectively manage database security, with MySQL emphasizing simplicity and explicit control, while PostgreSQL focuses on flexibility and unified role management.