

ADR-3

Migration from MySQL to PostgreSQL with Lombok

Date: 13-04-2025

Status

Accepted

Author

System Architect Team

Context

We initially planned to use MySQL as its a relational database and has manually written boilerplate Java code for entity classes. However, the growing need for advanced SQL features (such as partial indexes, GIN-based full-text search, and advanced concurrency control) and developer productivity improvements necessitated reconsideration. We evaluated the opportunity to adopt PostgreSQL for its richer feature set and integrate Lombok to reduce boilerplate code in our Java stack.

Decision

We decided to:

- Migrate from MySQL to PostgreSQL as the primary relational database.
- Use Lombok in our Java-based backend services to simplify model/entity-related code.

Alternatives

Option	Pros	Cons
Stay with MySQL	Stable and familiar; wide support	Lacks full-text search, partial indexes, UPSERTs, native UUID

Use PostgreSQL + Lombok (Chosen)	Advanced SQL, type safety, code reduction, cleaner models	Learning curve for Lombok annotations, build dependency
----------------------------------	---	---

Rationale

PostgreSQL provides several essential capabilities not available or inefficient in MySQL, including:

- Full-text search (GIN indexes)
- More efficient and concurrent data handling (MVCC implementation)
- Better JSONB support for hybrid relational/non-relational needs

Lombok drastically reduces boilerplates in Java classes, making the codebase leaner and easier to maintain. This improves productivity and consistency, especially when working with DTOs, JPA entities, and immutable value objects.

Consequences

- Better query performance and search capabilities using PostgreSQL.
- Cleaner and more maintainable codebase with Lombok.
- Developers can focus more on business logic than boilerplate code.

References

- <https://www.geeksforgeeks.org/difference-between-mysql-and-postgresql/>