Assignment 4

Rishikumar Patel (8972657)
Database Automation
PROG8850 - Spring 2025 - Section 1
Prof. Richard Hildred
July 19, 2025

Question 1: Analysis and Integration of Database Migration Tools

1.1 Tool Overview and Comparison

Tool 1: Flyway

Overview:
Flyway is an open-source tool for managing database migrations and it considers databases to be a set of SQL scripts of progressively changing versions. It focuses on ease of use, dependability, and works well with most CI/CD pipelines.

Key Features:

- Migration through SQL is supported (both versioned and repeatable)
- MySQL, PostgreSQL, Oracle and SQL Server are part of the 20+ supported relational databases.
- It provides integration through command line and plugins supporting Java, Maven, Gradle, Docker and CLI.
- Integration through command line and plugins is straightforward.
- It provides tracking of complete migrations through schema history tables

Tool 2: Liquibase

Overview:
Liquibase is a tool for managing changes made to databases, supporting alteration definitions in XML, YAML, JSON and SQL. It is also an open-source and cross-platform tool. Moreover, Liquibase integrates with CI/CD workflows and keeps track of the database's version history.

Key Features:

- Supports declarative change logs (XML, JSON, YAML)
- Rollback support for each migration
- Most relational databases supported
- Customizable visual change history tracking
- Wide range of plugins for CI/CD tools such as Jenkins and GitHub Actions.

Comparison Table: Flyway vs Liquibase

Feature	Flyway	Liquibase
Ease of Use	Simple setup with SQL scripts	More complex (declarative syntax)
CI/CD Integration	Strong with CLI, Maven, Gradle	Strong with rollback & changelogs
Supported Databases	20+ (MySQL, PostgreSQL, Oracle)	30+ (more enterprise DBs supported)
Migration Format	Versioned SQL scripts	XML, JSON, YAML, or SQL
Rollback Support	Limited (manual SQL only)	Built-in automatic rollback support

1.2 Integration Strategy for CI/CD Pipeline

To integrate **Flyway** and **Liquibase** into a CI/CD pipeline:

Flyway Integration:

1. Add `flyway.conf` to configure DB credentials and schema.
2. Use GitHub Actions to run `flyway migrate` for each commit or PR.
3. Create separate folders for initial vs incremental migrations.
4. Track schema state using Flyway's built-in schema table.

Liquibase Integration:

1. Maintain changelog files in XML/YAML.
2. Configure Liquibase in CI tools like Jenkins or GitHub Actions.
3. Add validation steps before deployments (`liquibase validate`).
4. Automate rollback on failure using `liquibase rollbackCount`.

Unified Strategy:

- Use **Flyway** for straightforward dev/test environments (SQL-driven).
- Use **Liquibase** for production environments where rollback safety is crucial.
- CI/CD pipeline should detect the environment (e.g., via GitHub Actions `env`) and run the respective tool.
- Store all migration files in version control for traceability.