Q1 Tools Analysis

Selection Tools : Flyway and Liquibase

Answers of each questions:

1. Overview and Key Features:

Flyway:

Overview: - Flyway is an open-source database migration tool which is more concerned with simplicity and version control of database changes. It can be combined with CI/CD pipelines and can be able to migrate using SQL based migrations using versioned scripts (V1___Create_Table.sql). It is lightweight and Java based and is compatible with environments (local, Docker, or cloud).

Key Features: - SQL migration scripts Versioned such as: V1, V2 - Allows migrating and un-migrating. - API support, Command-line and Docker support. - Several databases (MySQL, PostgreSQL, Oracle, SQL Server, etc.) are supported. - This feature is able to run migrations automatically on the startup of Java projects. - Close connectivity implies a strong, seamless integration or compatibility with the listed CI/CD tools.

Liquibase:

Overview: - Liquibase is a comprehensive, enterprise-focused database change management tool designed to facilitate the tracking, management, and deployment of database schema changes in a controlled and auditable manner. It supports multiple changelog formats including XML, YAML, JSON, and SQL, making it flexible for various development and operational preferences.

Key Features: - Liquibase allows users to define database changes using XML, YAML, JSON, or SQL. This flexibility helps teams adopt the tool - regardless of their preferred syntax or toolchain. - The tool maintains a detailed history of schema changes, enabling teams to track modifications over time with precision. - Liquibase supports rollback operations, allowing teams to reverse database changes safely and reliably, which is critical for rapid recovery from deployment issues or errors. - Diff capabilities: It offers diff tools that compare database schemas, assisting in identifying changes between different database states or environments. - Liquibase provides an audit trail of database changes, which is particularly important for regulated industries or environments demanding high levels of compliance and traceability.

2. Comparison table: Ease of Use | CI/CD Integration | Supported DBs

Ease of Use
Flyway: Simple setup with SQL scripts; lightweight CLI
Liquibase: Steeper learning curve due to changelog formats

CI/CD Integration: Flyway: Excellent with Jenkins, GitHub Actions, GitLab CI
Liquibase:Excellent with advanced rollback & validation options

Supported Databases
Flyway: MySQL, PostgreSQL, Oracle, SQL Server, MariaDB, H2
Liquibase: MySQL, PostgreSQL, Oracle, SQL Server, DB2, SQLite, Sybase

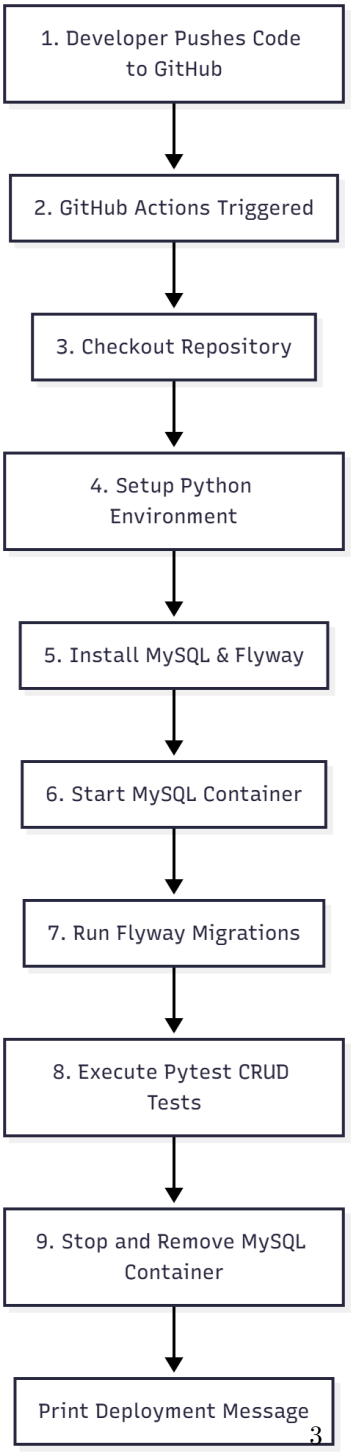
Migration Tracking
Flyway: Tracks versions via flyway_schema_history table
Liquibase: Tracks using databasechangelog and databasechangeloglock tables

Use Case Fit
Flyway: Ideal for small to medium projects with simple automation needs
Liquibase: Best suited for enterprise environments requiring auditing and rollback

Rollback Support
Flyway: Basic rollback via undo scripts
Liquibase: Full rollback support built-in

3.Integration strategy in a CI/CD pipeline (diagram + steps):

Diagram:



Steps:
1. Checkout Code
Pulls the latest version of the repository from GitHub.

2. Set Up Python Installs Python 3.11 and testing dependencies (`pytest`, etc.).

3. Install MySQL Client & Flyway
Downloads Flyway CLI and MySQL tools for migration and connectivity.

4. Start MySQL Container Launches a temporary MySQL instance for testing.

5. Wait for MySQL Readiness Ensures MySQL is fully initialized before migrations start. |

6. Apply Flyway Migrations Runs both initial (`migrations_initial`) and incremental (`migrations_incremental`) SQL migrations. |

7. Run CRUD Tests Executes automated Pytest scripts to verify CREATE, READ, UPDATE, and DELETE operations.

8. Clean Up Environment Stops and removes the MySQL container to keep the environment clean.

9. Deployment Message Prints: `CI/CD pipeline complete for commit <repo link>`confirming successful completion.