**FLYWAY VS LIQUIBASE**

**Similarities Between Liquibase and Flyway**

* Are open-source to an extent and help manage, track and deploy database schema changes.
* Use a versioned migration approach to a database schema change.
* Are based on Java and provides extensive support for Java frameworks like [**Spring Boot**](https://www.baeldung.com/spring-boot) and **[Vert.x](https://www.baeldung.com/vertx)**.
* Support integration with build tools like Maven and Gradle.
* Can run independently from the command line through provided scripts.
* Support a wide variety of databases.

**Differences Between Liquibase and Flyway**

**Defining a Change**

Flyway uses SQL for defining a change. On the other hand, Liquibase provides flexibility to specify a change in different formats including SQL such as XML, YAML, and JSON.

**Storing a Change**

**Both the tools store the deployed change in a table.** Flyway migrations are stored in the database schema with a default table named flyway\_schema\_history. Similarly, Liquibase stores its deployed migrations in a table named databasechangelog. Both the tools support overriding default configuration to change the table name.

**Execution Order of a Change**

Managing the order of a change is comparatively difficult in Flyway. With Flyway, the order depends on the version number and migration type in the filename. Contrarily, Liquibase uses a separate file named master\_changelog in which the changes are deployed in the order they are defined.

**Rollback a Change**

 Liquibase provides a way to roll back everything or undo specific migrations (available only on paid versions).

Flyway also has a undo migration, which can be deployed with a file name that starts with U followed by the version that needs to be undone. Its paid version also offers even more complex undo functionality.

Both the tools offer a decent rollback functionality, but considering only the free version, Flyway offers a good-to-use solution.

**Selective Deployment of a Change**

 Liquibase wins here when we’ve to selectively deploy a change. Flyway is also capable of doing it, but you would have to set up a different configuration file for each environment or database. With Liquibase we can easily add [labels and contexts](https://www.liquibase.com/blog/contexts-vs-labels?_ga=2.161326136.488045199.1647282742-629483959.1647282742) to ensure deployment in certain places.

**Snapshots & Comparing Databases**

**Liquibase allows users to take a snapshot of the current state** of the database. We can use this state to compare it to another database. This would be very helpful in scenarios like failover and database replication. Flyway on the other hand doesn’t support any of the snapshot features.

**Conditional Deployment**

**Liquibase offers an added feature called pre-conditions.** Preconditions allow users to apply changes based on the current state of the database. A changeset will only execute if it passes these preconditions.

**Flyway on the other hand doesn’t support this.** But through procedures, we can apply conditions in most SQL-based databases.