**Database Migration**

Git වැනි version control system වලදී code changes history එකක maintain වන අතර එම නිසා කවුද ඒ ඒ changes කලේ කොයිකාලේද කලේ වගේ full history එකක් බලාගන්න පුලුවන්.ඒ වගේම database එකටත් කරන වෙනස්කම් track කරලා තියාගන්න ඕනේ වෙනවනම් එකට use වන version control system තිබෙන අතර flyway, liquibase යනු එවැනි version control system දෙකක් වේ. Database schema changes track කරගන තියන open source version control application එකකි.Any changes which we do with data base scheme will be track and save in **flywayschamahistory** table in flyway and **databasechangelog** table in liquibase.

[Liquibase or flyway](https://www.liquibase.org/)is an open-source database-independent library for tracking, managing and applying database schema changes.

Advantage

Database independent

Automatic update (When application up, Changes will be automatically applied.)

Automatic creation and execution of rollback operation (මොනා හරි ප්‍රශ්නයක් වුනොත් execute වෙන time එකේදී එක automatically roll back වෙලා previous version එකට එනවා)

Changesset - describe a set of chagnges that liquibase execute within one transaction. මෙහිදී roll back එක වෙනවා වගේම changesset එක එය database eke databasechangelog table eke තියන් ඉන්නවා.තවද it is identified by author name and id(Unique id)

**Differences between Liquibase and Flyway.**

**Flyway** - Use SQL for defining changes.

**Liquibase** - Use SQL, XML, JSON, YMAL for defining changes (With Liquibase we can work with database-agnostic languages and easily apply schema changes to different database types).

**Flyway** - Follow file name conventions (a migration should have a convention of prefixes as V (for versioned), U (for undo), and R (for repeatable). It will be followed by a version number and a separator \_\_(two underscores) followed by a description and a suffix .sql such as V01\_\_Add\_New\_Column.sql).

**Liquibase** - changes are managed by one ledger known as a master changelog  (db.changelog-master) which will be defined as including all the migrations.

**Flyway** - Store changes in flyway\_schema\_history table.

**Liquibase** - Store changes in databasechangelog table.

**Flyway** - Execution order depends on the version number and migration type in the filename.

**Liquibase** - Execution order depends on separate file named master\_changelog in which the changes are deployed in the order they are defined.

**Flyway** - 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.

**Liquibase** -Liquibase provides a way to roll back everything or undo specific migrations.

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**

Database changes environment wise deploy කරන්න වන time තියනවා.Example එකක් ලෙස local එකේ table එක update වෙන්න ඕනේ නමුත් dev එකේ ඕනේ වෙන්නේ නෑ වගේ එකක්.මෙය flyway and liquibase දෙකම use කරලාකරගන්න පුලුවන් වුනත් liquibase වලදී එය වඩා පහසුවෙන් කරගන්න පුලුවන් (Flyway is also capable of doing it, but you would have to set up a different configuration file for each environment or database)

In the liquibase

<changeSet id="id" author="lahiru" context="dev"></changeSet>

In the application.property file

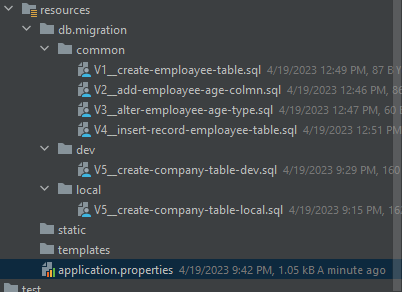
spring.profiles.active=dev

In flyway we can achieved it as bellow way.

In the application.property file

spring.flyway.locations=classpath:db/migration/common,classpath:db/migration/${spring.profiles.active}

File structure



**Conditional Deployment**

**Flyway** - Flyway on the other hand doesn't support this.

**Liquibase** -Preconditions allow users to apply changes based on the current state of the database. A changeset will only execute if it passes these preconditions.

Guidance documentation have been given in this repository and it can be found from bellow blog as well.

**Snapshots and comparing Database**

**Flyway** – Doses not allow.

**Liquibase** - Libuibase allows users to take a snapshot of the current state of the Database.

Snapshot is read only static view of source database as I exist at snapshot creation (Snapshot යනු existing database එකේ read only ලෙස තියන copy එකක් වේ).

