# Database Schema Migration Strategy for Devcloud Gen-2

## Need for migration strategy

Just like we have version control in code side and a well-defined build and deployment process (CICD) to deal with multiple environments (dev, QA, prod), we need a similar process for the database so we don’t have to deal with the following problems:

* What is the current version/state of the database?
* Has the migration/script already been applied?
* Is the database compatible with the current deployment. Are all the Object Relational Mappings valid?
* Did the hotfix we did for the dev has it been applied to the qa?
* Can we recreate the database from scratch and have consistent schema easily?

## Challenges

This is especially more challenging when we are dealing with a single database for all microservices as a result, we are not only coupled in terms of API Contracts but also in the Database layer. Any change/update in database by team responsible for microservice A must be communicated to all the teams using the database (even if they are independent logic wise) and they need to update their ORM models appropriately if required. Also deployments on every environment has to have synchronization between Object Relational Mapping of all the microservices with the database schema.

## Solution

Database migration tools like flyway and Liquibase can be used to recreate a database from scratch and make it clear all times what database state we are in and allows us to correctly migrate from the current version to next and rollback if required deterministically.

The above solution would have been enough if there was database per microservice or schema per microservice. But due to monolithic database we need proper synchronization procedure to communicate all teams to do required changes in their respective microservices before deploying the changes and running migration on database. Also it has to be made sure that all the versions are compatible with each other during deployments.

Which microservice should contain the logic for migration?

* Have the ownership of migration be with a specific microservice among existing services. Everytime the microservice starts the migration can run.
* Figure out a way if this flyway script can be run during deployments always using some sort of hooks.

## How does flyway works

A separate table in the database to store the current version. Flyway then scans the code for migrations (sql or java files) and then orders the execution of these migrations according to the version number and adds it to the table accordingly. If support for undo is required, the undo sql file can also be written and the flyway will run the undo in order.