**iDigitalCloudTech Pvt. Ltd.**

**POC on Automated DB deployment tool ( Flyway )**

**Flyway** is based around seven basic commands: Migrate, Clean, Info, Validate, Undo, Baseline, and Repair.

Migrations can be written in [SQL](https://en.wikipedia.org/wiki/SQL) (database-specific syntax such as [PL/SQL](https://en.wikipedia.org/wiki/PL/SQL), [T-SQL](https://en.wikipedia.org/wiki/T-SQL), etc is supported) or [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) (for advanced data transformations or dealing with LOBs).

It has a command-line client, a Java API (also works on [Android](https://en.wikipedia.org/wiki/Android_(operating_system))) for migrating the database on application startup, a [Maven](https://en.wikipedia.org/wiki/Apache_Maven) plugin, and a [Gradle](https://en.wikipedia.org/wiki/Gradle) plugin.

Supported databases include [Oracle](https://en.wikipedia.org/wiki/Oracle_Database), [SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server), [DB2](https://en.wikipedia.org/wiki/IBM_Db2), [MySQL](https://en.wikipedia.org/wiki/MySQL) (including [Amazon RDS](https://en.wikipedia.org/wiki/Amazon_RDS), Aurora MySQL, [MariaDB](https://en.wikipedia.org/wiki/MariaDB)), Percona XtraDB, [PostgreSQL](https://en.wikipedia.org/wiki/PostgreSQL) (including [Amazon RDS](https://en.wikipedia.org/wiki/Amazon_RDS) and [Heroku](https://en.wikipedia.org/wiki/Heroku)), Aurora PostgreSQL, [YugabyteDB](https://en.wikipedia.org/wiki/YugabyteDB), [CockroachDB](https://en.wikipedia.org/wiki/CockroachDB), [Redshift](https://en.wikipedia.org/wiki/Amazon_Redshift), [Informix](https://en.wikipedia.org/wiki/Informix), [H2](https://en.wikipedia.org/wiki/H2_(DBMS)), [Hsql](https://en.wikipedia.org/wiki/Hsql), [Derby](https://en.wikipedia.org/wiki/Apache_Derby), [SQLite](https://en.wikipedia.org/wiki/SQLite), [SAP HANA](https://en.wikipedia.org/wiki/SAP_HANA), [Sybase ASE](https://en.wikipedia.org/wiki/Sybase_ASE), [Phoenix](https://en.wikipedia.org/wiki/Apache_Phoenix), and [Firebird](https://en.wikipedia.org/wiki/Firebird_(database_server)).

**Flyway Commands:**

**Migrate**: Migrates the schema to the latest version. Flyway will create the schema history table automatically if it doesn’t exist. Migrate is the centerpiece of the Flyway workflow. It will scan the filesystem or your classpath for available migrations. It will compare them to the migrations that have been applied to the database. If any difference is found, it will migrate the database to close the gap. Migrate should preferably be executed on application startup to avoid any incompatibilities between the database and the expectations of the code.

**Info**: Prints the details and status information about all the migrations.

**Validate**: Validate applied migrations against resolved ones (on the filesystem or classpath) to detect accidental changes that may prevent the schema(s) from being recreated exactly.

Validation fails if

* differences in migration names, types, or checksums are found
* versions have been applied that aren’t resolved locally anymore
* versions have been resolved that haven’t been applied yet

**Baseline**: Baselines an existing database, excluding all migrations up to and including baselineVersion.

**Repair**: Repairs the Flyway schema history table. This will perform the following actions:

* Remove any failed migrations on databases without DDL transactions  
  (User objects left behind must still be cleaned up manually)
* Realign the checksums, descriptions, and types of the applied migrations with the ones of the available migrations

**Undo**: [Undoes](https://flywaydb.org/documentation/command/undo) the most recently applied versioned migration.

**Clean**: Drops all objects (tables, views, procedures, triggers, …) in the configured schemas.  
The schemas are cleaned in the order specified by the schemas property.

**Naming convention:**

The default naming patterns within Flyway are [documented clearly](https://flywaydb.org/documentation/migrations#naming). You have a prefix that determines the type of file, whether a versioned migration (V), an undo migration (U), or a repeatable migration (R). That goes at the front of the file name. Next, you have a version number. This can be almost any format that you like, but it has to be unique for any given migration (although a versioned migration and an undo migration must share a common version number) and it has to be logically ordered. Then, you add two underscores, in order to separate the functional naming aspects of the file from the purely descriptive. After that, it’s just text. You can use an underscore between words and it will be translated as spaces.

V1\_\_create\_idigicloudemp.sql

V3\_\_insert\_idcemp.sql

R\_\_insert\_IDCEMPTEST.sql

**PoC Scenarios:**

**Scenario1: Single script for migration using flyway migrate**

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Table

Description automatically generated with medium confidence

**Scenarios 2: Executing Flyway migrate again without any change in version.**

Text

Description automatically generated

**Scenario 3: Executing multiple versions in sequence**

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Table

Description automatically generated with medium confidence

A picture containing graphical user interface

Description automatically generated

**Scenario 4: Executing Flyway migrate again without any change in a version for multiple scripts**

Text

Description automatically generated

**Scenario 5: To test Incorrect script placed for migration**

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

**After correction,**

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

**Scenario 6: Flyway Clean**

Text

Description automatically generated

**Scenario 7: Repeatable migration**

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

**Scenario 8: Change in the repeatable migration and executing flyway migrate**

**Text

Description automatically generated**

**Scenario 9: Baseline the version**

Graphical user interface, application

Description automatically generated

PoC Acceptance Criteria:

1 ) Execute DDL to Environment  
2 ) Execute DML to Environment  
3 ) Maintain Scripts execution history  
4 ) Incremental DB release to environment  
5 ) Any standard practices like naming convention, handling commits document those.

Text

Description automatically generated

Text

Description automatically generated