LiquidBase:

**What is Liquibase?**

Liquibase is an open source database change management tool built on Java.  Rather than writing SQL directly against the database to create, update or drop database objects, developers define their desired database changes in XML files.  The XML file, called a [changelog](http://www.liquibase.org/manual/databasechangelog), contains a list of [changesets](http://www.liquibase.org/manual/changeset" \o "Liquibase Changesets" \t "_blank) that define a desired database change in an database agnostic abstraction.  The changelog is intended to contain an evolving list of database changes the team would like to apply to a target database.

**How does it work?**

When Liquibase is executed, you must specify the database against which to apply your changesets.  Liquibase uses two tables to manage changes to the database: databasechangelog and databasechangelock.  If the tables don’t exist on the target database, they are created.  An entry is added to the databasechangelock table which ensures only one instance of Liquibase is running at a time.  The databasechangelog table contains a listing of every changeset that has been applied to this database.  Liquibase conducts a diff of the table contents with the XML files and determines which changes still need to be applied.  Once this is determined, Liquibase will then apply the changes to the database.  If you are new to Liquibase, then I recommend checking out their documentation on how to setup and get [started](http://www.liquibase.org/quickstart).

* For every user story if there is need to change in database schema then we can have dbchangelog.yml file.
* A dbchangelog.yml file has changeset which represents each operation.
* Think of each change set as an atomic change that you want to apply to your database.
* In paysafe the flow is like below
  + Mock data is created in xls file
  + These xls file are provided as a input to dbchangelog files to mock the data to respective db