

A "migration" is the set of steps needed whenever you change the structure of your SQLAlchemy models, add a new attribute, etc. to replicate those changes in the database, add a new column, a new table, etc.

First we need to install Alembic. "pip install alembic"

Usage of Alembic starts with creation of the *Migration Environment*. This directory contains scripts that are specific to a particular application. The migration environment is created once and then maintained alongside the application's source code. It is created using the "init" command of Alembic, and can be customized to fit the specific requirements of the application.

The directory includes the following:

1. "yourproject": This is the root directory of your application's source code or a subdirectory within it.
2. "alembic": This directory is where the migration environment resides. It can have any name and can be located within your application's source tree. If your project uses multiple databases, there may be multiple "alembic" directories.
3. "env.py": This Python script is executed when the alembic migration tool is used. It configures and generates a SQLAlchemy engine, establishes a connection to the database, and invokes the migration engine using the connection. This script can be customized to handle multiple engines, accept custom arguments, and load application-specific libraries and models.
4. "README": This file, included with the environment templates, provides informative instructions or explanations.
5. "script.py.mako": This is a Mako template file used to generate new migration scripts. It defines the structure and content of each migration file, including imports and the structure of the "upgrade()" and "downgrade()" functions. In some cases, such as the multodb environment, it allows for the generation of multiple functions with specific naming schemes.
6. "versions/": This directory contains individual version scripts for migrations. Unlike other migration tools that use ascending integers, Alembic uses a partial GUID approach. The ordering of version scripts is relative to directives within the scripts themselves. It is possible to merge migration sequences from different branches by manually "splicing" version files together, but it should be done carefully.

Procedure :

1. Firstly we need to create an environment for migration with command:
alembic init alembic #here "alembic" is name of command "init" is a command to generate

the migrations directory and last parameter is the name of the directory.

2. After this we need to edit .ini file (its a file placed in a current directory)

"This is a file that the Alembic script searches for when it is executed. The file can be located in a different directory, and its location can be specified using either the "--config" option or the "ALEMBIC_CONFIG" environment variable when running the Alembic runner. If both options are provided, the "--config" option takes precedence."

Changes to be made in this file

sqlalchemy.url = url of your database(SQLALCHEMY_DATABASE_URL)

3. Go to env.py file which will be inside alembic/env.py. Then update "target_metadata" =Base.metadata -----here first import Base and all models .

4. Now we need to make the migration file.

It can be done by : alembic revision -m "descriptive message you want to give"
or you can also auto generate migration using command:
alembic revision --autogenerate

5. After creating the migration we can finally run our migration .

The `alembic upgrade` command will run upgrade operations, proceeding from the current database revision, in this example `None` , to the given target revision. We can specify `1975ea83b712` as the revision we'd like to upgrade to, but it's easier in most cases just to tell it "the most recent", in this case `head`