**Database schema migrations**

- postgreSQL, python, SQLAlchemy, Alembic

**Steps**

1. Set up a Postgres server locally. Using the user interface called pgAdmin, make a server (username = postgress, password = password). Create a database.

In order to connect to this database in Python you will need to remember:

a. The name of the database - books in our case

b. The port - defaults to 5432

c. The username - defaults to postgress

d. The password you used when creating the server

2. *python models.py* // to create the table in database

3. *pip install psycopg2* // Connecting to Postgres with Python

4. *python config.py*  // change db name and password

5. *python crud.py* --> add values to database, querying from database

( *python crud2.py* --> same as crud.py. using sessions instead of engines)

6. Alembic

initialize Alembic in the root of our project

*pip install alembic*

*alembic init alembic*

7. Setting up Alembic

edit env.py file

replace *fileConfig(config.config\_file\_name)* by:

*fileConfig(config.config\_file\_name)*

*import sys, os*

*sys.path.insert(0, os.path.dirname(os.path.dirname(\_\_file\_\_)))*

*import config as my\_config*

*config.set\_main\_option('sqlalchemy.url', my\_config.DATABASE\_URI)*

replace *target\_metadata = None* by *:*

*from models import Base*

*target\_metadata = Base.metadata*

8. *alembic stamp head* // creates alembic\_version table in our books database to track the current version of our models.

9. Changes in DB(migrations)

Uncomment price = Column(MONEY) in models.py //adds new column

*alembic revision --autogenerate -m "Added price column"*  // to autogenerate a migration script

10. *alembic upgrade head*

*11. python cli.py*  //updates value to “prices” column