SQL Alchemy

Postgresql

In postgresql we can create the table by simple right click the table in it and also we can add the columns in it

In postgresql there different types of datatypes in it like int ,float varchar ,serial(it used to generate serial numbers)…..  
  
After create the data we can right click and view the data by simple right click the view/edit in that we can view the number table we need to view in it

SQLALCHEMY

In sql alchemy we need to install the sql alchemy in the venv using the command of pip install sqlalchemy

Then we need to use ,create\_engine ,sessionmaker,

from sqlalchemy import create\_engine

from sqlalchemy.ext.declarative import declarative\_base

from sqlalchemy.orm import sessionmaker

to connect sqlalchemy to python we need to :

SQLALCHEMY\_DATABASE\_URL=”postgres://<username>:<password>@<ip-address>/<hostname>”

After creating the sqlalchemy database url we need to create the engine to work in it:create\_engine(sqlalchmemy\_db\_url)

Then we need to create the session maker to bind the engine session\_maker(autocommit=false,autoflash=false,bind=engine)

Session\_maker is response for taking with the database

In sqlalchemy we need to create the new class for each table in the database  
  
\_\_tablename\_\_=”tablename in database” 🡪 it used to tells sqlalchemy this the table name

To insert the value we need to import the Column(start letter should capital),Integer, String, etc we need import sqlalchemy

To get the post by using sql orm we need to used session maker need to be import from the database

To get the details we need to type the

db.query(table\_filename.tablename).all()

**TO enter the data**

we need to need to insert the session=depends(get\_db) in the function parameters

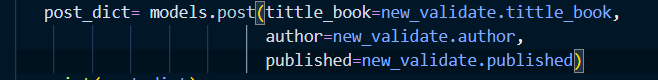
to insert the value we need to take the table create file name like

A computer screen with text and images

Description automatically generated

After add the data with the help of the class validate we need to add the post to the db by using the db.add(variable\_name) and we need to commit it by db.commit() that need to refresh by db.refresh(variable\_name)

Instead of typing this type

 of data we can type with the help \*\*post.dict()

To get the specific data using get(“/{id}”}

Db.query(file name.tablename).filter(filename.column==name of column).first()

To put the delete same as a upper query but use variable name.delete or update

We skip to 5:30:31

Realationship

Relationship is used to connect two table to gether with the help of realtionalship()  
  
 In that relationship we have different type in it they are one to one, one to many, many to one, many to many  
  
 in that relationship we used to pass the some parameters in like back\_populates, uselist, foreignkeys.

Back\_populates 🡪 is used to connect the table in bi-directional ways so we must used it

Foreignkey 🡪 is used to connect the table together

Variablename=Column(“datatype”,Foreignkey(“tablename need”))

Uselist 🡪

Relationship

Relationship is used to connect the table which show the relationship among them

To use 🡪 relationship(“classname need “ back\_populates=”variable that store”)

Marshmellow