

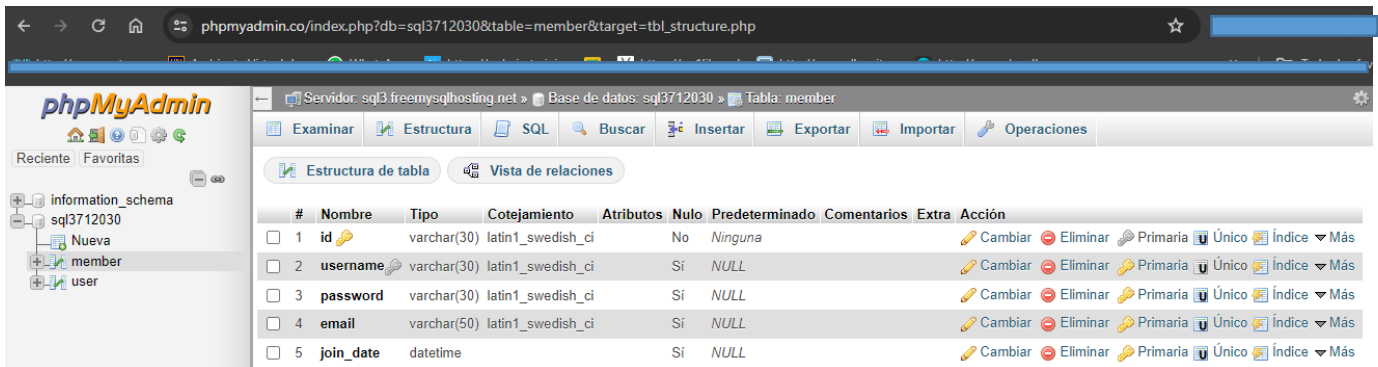
Flask-SQLAlchemy ORM

1) Creation of a table

```
#Create table
class Member(db.Model): #Table name
    id = db.Column(db.String(30), primary_key = True)
    username = db.Column(db.String(30), unique=True)
    password = db.Column(db.String(30))
    email = db.Column(db.String(50))
    join_date = db.Column(db.DateTime)

    def __repr__(self): # When is called, get the string representation of the instance
        return '<Member %r>' % self.username
```

```
root@DESKTOP-829P21V:/mnt/c/Users/Paul Manriquez/Desktop/FlaskProyects/4_Flask_SQL_Alchemy# python3 app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 125-845-134
```



The screenshot shows the phpMyAdmin interface for a database named 'sql3712030'. The 'member' table structure is displayed with the following columns:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	varchar(30)	latin1_swedish_ci		No	Ninguna			Cambiar, Eliminar, Primaria, Único, Índice, Más
2	username	varchar(30)	latin1_swedish_ci		Si	NULL			Cambiar, Eliminar, Primaria, Único, Índice, Más
3	password	varchar(30)	latin1_swedish_ci		Si	NULL			Cambiar, Eliminar, Primaria, Único, Índice, Más
4	email	varchar(50)	latin1_swedish_ci		Si	NULL			Cambiar, Eliminar, Primaria, Único, Índice, Más
5	join_date	datetime			Si	NULL			Cambiar, Eliminar, Primaria, Único, Índice, Más






2) Add a member from the shell

```
root@DESKTOP-829P21V:/mnt/c/Users/Paul Manriquez/Desktop/FlaskProyects/4_Flask_SQL_Alchemy
root@DESKTOP-829P21V:/mnt/c/Users/Paul Manriquez/Desktop/FlaskProyects/4_Flask_SQL_Alchemy# python3
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from app import db, Member
>>> from datetime import date
>>> date.today()
datetime.date(2024, 6, 5)
>>> paul = Member(username='paul', password='secret', email='paul@mail.com', join_date=date.today())
>>> db.session.add(paul)
>>> db.session.commit()
>>>
```

```
# Create the SQLAlchemy db instance
db = SQLAlchemy(app)
app.app_context().push()
```

its important to add app.app_... or you'll get an error

phpMyAdmin



Reciente Favoritas

+

information_schema

+

sql3712030

Nueva

member

user

←

Servidor: sql3.freemysqlhosting.net » Base de datos: sql3712030 » Tabla: member

Examinar

Estructura

SQL

Buscar

Insertar

Exportar

Importar

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0.0007 segundos.)

SELECT * FROM `member`

Perfilar

☐ Mostrar todo

Número de filas: 25

Filtrar filas:

+ Opciones

←

→

id


username


password


email

join_date

☐

 Editar

 Copiar

 Borrar

paul

secret

paul@mail.com

2024-06-05 00:00:00

Flask-SQLAlchemy ORM

3) Modify the data user like an object method.


Important the next steps can be performed if you are in the instance shell when you created and commit the users, otherwise you will need to retrieve the row as an object and then modify it.






```
>>> from datetime import date
>>> paul = Member(username='paul',password='secret',email='paul@mail.com',join_date=date.today())
>>> db.session.add(paul)
>>> db.commit()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/lib/python3.10/dist-packages/flask_sqlalchemy/extension.py", line 1008, in __getattr__
    raise AttributeError(name)
AttributeError: commit
>>> db.session.commit()
>>> user = Member(username='user',password='password',email='user@mail.com',join_date=date.today())
>>> db.session.add(user)
>>> db.session.commit()
>>> paul.password='newpassword'
>>> db.session.commit()
>>> user.email = 'newuseremail@email.com'
>>> db.session.commit()
```

4) To delete, in the same instance, put the object-user and commit the changes

```
>>> db.session.delete(user)
>>> db.session.commit()
>>>
```

+ Opciones

	id	username	password	email	join_date
<input type="checkbox"/>  Editar  Copiar  Borrar	1	paul	newpassword	paul@mail.com	2024-06-05 00:00:00

 ☐ Seleccionar todo Para los elementos que están marcados:  Editar  Copiar  Borrar  Exportar

5) Performing queries

Like query

```
Type "help", "copyright", "credits" or "license" for more information.
>>> from app import db,Member
>>> Member.query.all()
[<Member 'paul'>, <Member 'user'>, <Member 'user2'>]
>>> likeQuery = Member.query.filter(Member.username.like('%us%')).all()
>>> likeQuery
[<Member 'user'>, <Member 'user2'>]
```

And query

```
>>> q = Member.query.filter(Member.username == 'paul').filter(Member.email == 'paul@mail.com').all()
>>> q
[<Member 'paul'>]
```

Order by

```
>>> Member.query.order_by(Member.username).all()
[<Member 'paul'>, <Member 'user'>, <Member 'user2'>]
>>> Member.query.order_by(Member.email).all()
[<Member 'paul'>, <Member 'user2'>, <Member 'user'>]
>>>
```