

Relación entre tablas con Flask (OneToMany)

1. Creamos a estrutura do proxecto

- `python -m venv venv`
- `./venv/Scripts/activate`
- `pip install flask`
- `pip install flask-cors`
- `pip install flask-marshmallow`
- `pip install flask-sqlalchemy`

2. Executamos o comando: *pip list*

Con este comando podemos visualizar as recentes instalacións.

3. Creamos o arquivo `main.py`

4. Realizamos as importacións necesarias

```
from flask import Flask, request, jsonify
from flask_sqlalchemy import SQLAlchemy
import flask_marshmallow import Schema, fields

app = Flask(__name__)

if __name__ == '__main__':
    app.run(debug=True)
```

5. Configuramos a base de datos

...

```
app = Flask(__name__)
```

```
#database config
```

```
app.config['SQLALCHEMY_DATABASE_URI'] = "mysql://root@localhost/escuela"
```

```
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
```

```
db = SQLAlchemy(app)
```

```
ma = Marshmallow(app)
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

6. Creamos as clases

...

```
class Alumno(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    nombre = db.Column(db.String(250), nullable=False)

    def __init__(self, nombre):
        self.nombre = nombre

class Nota(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    nota_alumno = db.Column(db.Float, nullable=False)
    alumno_id = db.Column(db.Integer, db.ForeignKey('alumno.id'))
    alumno = db.relationship('Alumno', backref='notas')

    def __init__(self, nota_alumno, alumno_id):
        self.nota_alumno = nota_alumno,
        self.alumno_id = alumno_id

class NotaSchema(Schema):
    id = fields.Integer()
    nota_alumno = fields.Float()

class AlumnoSchema(Schema):
    id = fields.Integer()
    nombre = fields.String()
    notas = fields.Nested(NotaSchema, many=True)

alumno_schema = AlumnoSchema()
alumnos_schema = AlumnoSchema(many=True)

nota_schema = NotaSchema()
notas_schema = NotaSchema(many=True)

if __name__ == '__main__':
    app.run(debug=True)
```

7. Realizamos as peticións GET

...

```
@app.get("/")
def get_alumno():
    alumnos = Alumno.query.all()
    return alumnos_schema.dump(alumnos)

@app.get("/notas")
def get_notas():
    notas = Nota.query.all()
    return notas_schema.dump(notas)

if __name__ == '__main__':
    app.run(debug=True)
```

Executamos o código e dirixímonos a pypmyadmin. Creamos a base de datos escuela cunha táboa alummno con 2 campos (id, nombre) e outra táboa nota con 4 campos (id, nota_alumno, alumno_id). Despois executaremos en SQL a seguinte consulta:

- ALTER TABLE nota ADD CONSTRAINT fk_nota_alumno FOREIGN KEY(alumno_id) REFERENCES alumno(id);

Añadimos valores aos campos das táboas. Para ver os datos dirixímonos a localhost:5000 e comprobamos que se mostra un json con todo.

8. Realizamos as peticións POST

```
@app.post("/save-alumno")
def save_alumno():
    nombre = request.json['nombre']
    alumno = Alumno(nombre)
    db.session.add(alumno)
    db.session.commit()
    return alumno_schema.dump(alumno)

@app.post("/save-nota")
def save_notas():
    nota_alumno = request.json['nota_alumno']
    alumno_id = request.json['alumno_id']
    nota = Nota(nota_alumno, alumno_id)
    db.session.add(nota)
    db.session.commit()
    return nota_schema.dump(nota)
```

Podemos probar que funcione dende Postman.