AT05.04- MongoDB desde Python

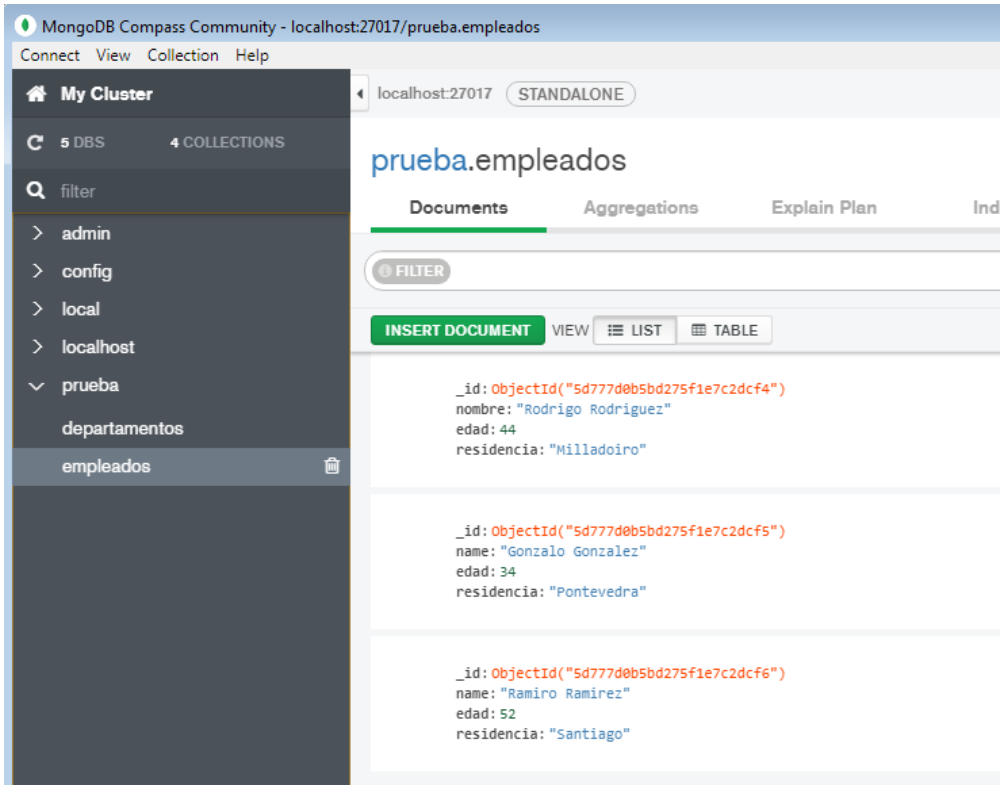
ACCESO A MONGO DESDE PYTHON

Vamos a ver ahora como podemos desde python acceder a una base de datos mongodb y crear colecciones, insertar objetos, etc.
Vamos a utilizar para ello el siguiente programa:

```
from pymongo import MongoClient

try:
    conn = MongoClient('localhost',27017)
    print("Conexion establecida!!!")
except:
    print("No se puede establecer la conexion")

# seleccionamos la base de datos
db = conn.AT05_04

# Creamos o seleccionamos la coleccion "departamentos" para insertar los
departamentos
deptos = db.departamentos
depto1={
   "id":"1",
   "departamento":"ventas",
   "sede":"Santiago"
   }
depto2={
   "id":"2",
   "departamento":"distribucion",
   "sede":"Pontevedra"
   }
# Insertamos los datos
reg1=deptos.insert_one(depto1)
reg2=deptos.insert_one(depto2)

#Hacemos lo mismo con los empleados
emps = db.empleados

emple1 = {
        "nombre":"Rodrigo Rodriguez",
        "edad":44,
        "residencia":"Milladoiro",
         "departamento":"1"
        }
emple2 = {
        "nombre":"Gonzalo Gonzalez",
        "edad":34,
        "residencia":"Pontevedra",
         "departamento":"2"
        }
emple3= {
        "nombre":"Ramiro Ramirez",
        "edad":52,
        "residencia":"Santiago",
         "departamento":"1"
        }
# Insertamos los datos, ahora todos juntos
reg1 = emps.insert_many([emple1,emple2,emple3])

# Mostramos los datos insertados
cursor = emps.find()
```

```
for registro in cursor:
    print(registro)
```

Comprobamos que efectivamente se ha insertado la información en la base de datos:

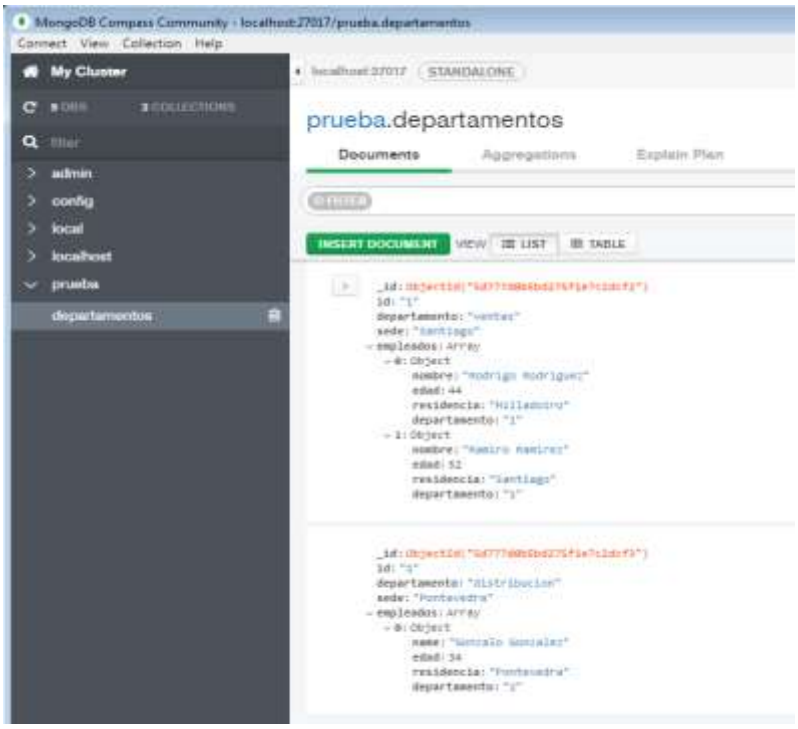


Realmente esta estructura de empleados relacionados con departamentos (típica del modelo relacional) no es la más adecuada para trabajar en MongoDB. En este caso lo que tendríamos que hacer sería incluir a los empleados como elementos dentro de cada departamento.
Vamos por tanto a hacer modificaciones en la base de datos para eliminar los empleados creados, e insertarlos en un array dentro del departamento al que pertenezcan:

```
from pymongo import MongoClient

try:
    conn = MongoClient('localhost',27017)
    print("Conexion establecida!!!")
except:
    print("No se puede establecer la conexion")

# seleccionamos la base de datos
db = conn.AT05_04

# Creamos o seleccionamos la coleccion "departamentos" para insertar los
departamentos
deptos = db.departamentos
depto1={
   "id":"1",
   "departamento":"ventas",
   "sede":"Santiago"
   }
depto2={
   "id":"2",
   "departamento":"distribucion",
```

```
   "sede":"Pontevedra",

   }
#Seleccionamos la coleccion que vamos a modificar
deptos = db.departamentos

#añadimos los empleados de cada departamento. Al departamento 1 le vamos a
cambiar tambien el nombre
resultado1=deptos.update_one(
   {"id":"2"},
   {"$set":{
          "empleados":[{
         "name":"Gonzalo Gonzalez",
         "edad":34,
         "residencia":"Pontevedra",
          "departamento":"2"
         }]}
   })
resultado2 = deptos.update_many(
   {"id":"1"},
    {
          "$set":{
                 "departamento":"comercial"
          },
         "$set":{
                 "empleados":[{
                 "nombre":"Rodrigo Rodriguez",
                 "edad":44,
                 "residencia":"Milladoiro",
                 "departamento":"1"
                 },
                 {
                 "nombre":"Ramiro Ramirez",
                 "edad":52,
                 "residencia":"Santiago",
                 "departamento":"1"
                 }]
          }
    })

# Mostramos los datos modificados
cursor = deptos.find()
for registro in cursor:
    print(registro)

# borramos los empleados, que ya no los necesitamos
emps=db.empleados
emps.delete_one({"nombre":"Gonzalo Gonzalez"})
emps.delete_many({"nombre":{"$regex": "ez$"}})
# y borramos la conexion (podiamos haberlo hecho directamente sin haber borrado
antes los empleados)
emps.drop()
```

Comprobamos que ya no está la colección de empleados y que éstos están dentro del departamento:



Vamos a hacer ahora algunas búsquedas para recuperar información de la base de datos. Para realizar búsquedas usamos el método find(), que funcionará exactamente igual que en MongoDB.

```
from pymongo import MongoClient

try:
    conn = MongoClient('localhost',27017)
    print("Conexion establecida!!!")
except:
    print("No se puede establecer la conexion")

# seleccionamos la base de datos
db = conn.AT05_04

# seleccionamos la coleccion "departamentos"
deptos = db.departamentos

print('\nRecuperamos unicamente el primer departamento:')
print (deptos.find_one())

print ('\nRecuperamos todos los departamentos:')
for depto in deptos.find():
   print(depto)

print('\nRecuperamos el departamento con id=1:')
print (deptos.find_one({"id":"1"}))

print('\nRecuperamos los departamentos con id mayor que 1:')
for depto in deptos.find({ "id": { "$gt": "1" } }):
   print(depto)

print('\nRecuperamos todos los datos de los departamentos 1 y 2:')
for depto in deptos.find({"id":{"$in":["1","2"]}}):
   print(depto)

print('\nRecuperamos todos los datos del departamento menos los empleados:')
for depto in deptos.find({},{"empleados": 0}):
   print(depto)
```

```
print('\nRecuperamos unicamente el id y el nombre del departamento:')
# el Object_id hay que indicar explicitamente que no lo muestre
for depto in deptos.find({},{ "_id":0,"id": 1,"departamento":1}):
   print(depto)


print('\nRecuperamos el nombre y edad de los empleados del departamento cuya edad
sea mayor que el id del departamento')
for depto in
deptos.find({"id":{"$lt":"empleados.edad"}},{"_id":0,"empleados.edad":1,"empleado
s.nombre":1}):
   print(depto)
```

Indica las sentencias necesarias para buscar la siguiente información:

1. Empleados del departamento 1
2. Empleados con edad mayor a 40 años
3. Empleados con edad menor que 50 años
4. Empleados cuyo apellido acaba en “ez”
5. Empleados que residen en la misma ciudad en que está su departamento.
6. Edad de los empleados cuyo nombre empieza por “Ro”
7. Nombre del departamento y nombre de los empleados cuyo nombre sea mayor al nombre de su departamento