

Prácticas MongoDB

1. Modelo de datos

Uno a Uno de tipo Embebido

- Crear un modelo Uno a Uno de tipo embebido con los siguientes datos.

COCHES

Matricula
Marca
Modelo
Caballos

CONDUCTORES

DNI
Nombre
Apellidos
Edad

- Probarlo creando la colección insertando algún dato y comprobar que funciona

```
db.coches.insertOne(  
{
```

```

    matricula: "XX|XXXX1",
    i,,,,,,,,, marca: "Ford",
    modelo: "Mustang",
    caballos: 300,
    conductor:
    {
        dni: "eeeeeee",
        nombre: "Pedro",
        apellidos: "PErez",
        edad: 40
    }
})

```

- Visualizar la marca, modelo y nombre del conductor

```

curso> db.coches.find({}, {_id:0, marca: 1, modelo:1,
'conductor.nombre': 1})
[
  { marca: 'Ford', modelo: 'Mustang', conductor: { nombre:
'Pedro' } }
]

```

Uno a Uno de tipo Referencia

- Ahora, borramos la colección y creamos un modelo Uno a Uno de tipo referencia . En la colección “coches” usamos la matricula para hacer la referencia. En la colección “conductores” usamos el atributo “cod_matricula” para hacer la referencia.
- Creamos las colecciones

```

curso> db.coches.drop()
true

curso> db.coches.insertOne(
{
    matricula: "XXXXXX1",
    marca: "Ford",
    modelo: "Mustang",
    caballos: 300

```

```

    })
    {
      acknowledged: true,
      insertedId: ObjectId("649b028ef1d57e1e92c29aed")
    }

Coches> db.conductores.insertOne(
  {
    dni: "eeeeeeee",
    nombre: "Pedro",
    apellidos: "PErez",
    edad: 40,
    cod_matricula: "XXXXXX1"
  }
)

{
  acknowledged: true,
  insertedId: ObjectId("649b029bf1d57e1e92c29aee")
}

```

- Sacar la marca, modelo y nombre del conductor mediante un lookup

```

db.coches.aggregate( [
  {
    $lookup:
    {
      from: "conductores",
      localField: "matricula",
      foreignField: "cod_matricula",
      as: "vehiculos"
    },
    {$project:{_id:0,  marca: 1,modelo: 1, "vehiculos.nombre":
1}}
  ] )

```

```
[
  {
    marca: 'Ford',
    modelo: 'Mustang',
    vehiculos: [ { nombre: 'Pedro' } ]
  }
]
```

• Uno a Muchos de tipo Embebido

- Crear un modelo Uno a Muchos de tipo referencia con los siguientes datos. Vamos a usar la colección conductores, con lo que tendremos que borrarla y recrearla.

CONDUCTORES

DNI
Nombre
Apellidos
Edad
Cod_matricula

VIAJES

Cod_viaje
Origen
Destino

kilometros

- Crear la colección CONDUCTORES con un par de viajes embebidos.

```
db.conductores.insertOne(
{
  dni: "eeeeeee",
  nombre: "Pedro",
  apellidos: "PErez",
  edad: 40,
  cod_matricula: "XXXXXX1",
  viajes:
  [
    {
      cod_viaje: 1,
      origen: "punto A",
      destino: "punto B",
      kilometros: 430
    },
    {
      cod_viaje: 2,
      origen: "punto B",
      destino: "punto C",
      kilometros: 430
    }
  ]
}
)
```

- Visualizar el nombre del conductor y los kilómetros

```
db.conductores.find({}, {nombre: 1, 'viajes.kilometros': 1})
[
  {
```

```
  _id: ObjectId("649b0cc0f1d57e1e92c29af1"),
  nombre: 'Pedro',
  viajes: [ { kilometros: 430 }, { kilometros: 430 } ]
}
```

Uno a Muchos de tipo Embebido

- Borramos la colección CONDUCTORES y la convertimos en dos colecciones. La colección CONDUCTOR tiene como clave el DNI, mientras que la colección VIAJES tiene cod_conductor como columna de referencia.
- Cargamos un par de viajes

```
db.conductores.drop()
true
```

- Colección conductores

```
db.conductores.insertOne(
  {
    dni: "eeeeeee",
    nombre: "Pedro",
    apellidos: "PErez",
    edad: 40,
    cod_matricula: "XXXXXX1"}
)
{
  acknowledged: true,
  insertedId: ObjectId("649b1535f1d57e1e92c29af2")
}
```

- Colección Viajes. Insertamos un par de filas

```
db.viajes.insertOne(
{
  cod_viaje: 1,
  origen: "punto B",
  destino: "punto C",
```

```

        kilometros: 430,
        cod_conductor: "eeeeeee"

    }
)

{
  acknowledged: true,
  insertedId: ObjectId("649b157cf1d57e1e92c29af3")
}

db.viajes.insertOne(
{
    cod_viaje: 2,
    origen: "punto B",
    destino: "punto C",
    kilometros: 430,
    cod_conductor: "eeeeeee"

}
)

{
  acknowledged: true,
  insertedId: ObjectId("649b16cbf1d57e1e92c29af4")
}

```

- Visualizar el nombre del conductor y los kilómetros mediante lookup

```

db.conductores.aggregate( [
  {
    $lookup:
    {
      from: "viajes",
      localField: "dni",
      foreignField: "cod_conductor",
      as: "viajes"
    }
  }
]
)

```

```

    }},
    {$project:{_id:0, nombre: 1, "viajes.kilometros": 1}}
  ] )
[
  {
    nombre: 'Pedro',
    viajes: [ { kilometros: 430 }, { kilometros: 430 } ]
  }
]

```

- Crear una vista llamada "datos_viajes" con la consulta anterior.

```

db.createView
(
  "datos_viajes",
  "conductores",
  [
    {
      $lookup:
      {
        from: "conductores",
        localField: "matricula",
        foreignField: "cod_matricula",
        as: "vehiculos"
      },
      $lookup:
      {
        from: "viajes",
        localField: "dni",
        foreignField: "cod_conductor",
        as: "viajes"
      },
      {$project:{_id:0, nombre: 1, "viajes.kilometros": 1}}
    }
  ] )

```



```
{ ok: 1 }
```

- Comprobar que la vista se ha creado

```
show collections
coches
conductores
datos_viajes      [view]
personas
productos
productos2
viajes
system.views
```

- Hacer una consulta con la vista

```
db.datos_viajes.find()
[
  {
    nombre: 'Pedro',
    viajes: [ { kilometros: 430 }, { kilometros: 430 } ]
  }
]
```

-