

# Trabajo Práctico 2

## Base de datos II

### #Ejercicio 1:

```
local> use empresa
switched to db empresa
empresa> db.empleados.insertMany([{\nombre:"Ramiro", edad:27, puesto:"presidente"},{\nombre:"Pedro", edad:40, puesto:"CEO"},{\nombre:"Julian", edad:30, puesto:"pasante"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68117b3bce2228ca3dd861e6'),
    '1': ObjectId('68117b3bce2228ca3dd861e7'),
    '2': ObjectId('68117b3bce2228ca3dd861e8')
  }
}
empresa> db.empleados.updateOne({\nombre:"Ramiro"}, {\$set: {edad:33}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
empresa> db.empleados.deleteOne({\puesto:"pasante"})
{ acknowledged: true, deletedCount: 1 }
empresa> []
```

### #Ejercicio 2:

```
empresa> db.empleados.find({\edad: {\$gte: 25, \$lte:40}})
[
  {
    _id: ObjectId('68117b3bce2228ca3dd861e6'),
    nombre: 'Ramiro',
    edad: 33,
    puesto: 'presidente'
  },
  {
    _id: ObjectId('68117b3bce2228ca3dd861e7'),
    nombre: 'Pedro',
    edad: 40,
    puesto: 'CEO'
  }
]
empresa> []
```

### #Ejercicio 3:

```
empresa> db.empleados.find({}, {\nombre: 1, puesto: 1, _id:0});
[
  { nombre: 'Ramiro', puesto: 'presidente' },
  { nombre: 'Pedro', puesto: 'CEO' }
]
empresa> []
```

## Ejercicio 4:

```
empresa> db.empleados.updateOne({nombre: "Ramiro"}, {$set: {direccion: {calle: "balbin", ciudad:"puan", codigo_postal:"8180"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

## Output:

```
empresa> db.empleados.find()
[
  {
    _id: ObjectId('68117b3bce2228ca3dd861e6'),
    nombre: 'Ramiro',
    edad: 33,
    puesto: 'presidente',
    direccion: { calle: 'balbin', ciudad: 'puan', codigo_postal: '8180' }
  },
]
```

## Ejercicio 5:

```
db.ventas.insertMany([
  {producto: "Laptop", cantidad: 2, precio_unitario: 1200},
  {producto: "Mouse", cantidad: 10, precio_unitario: 25},
  {producto: "Teclado", cantidad: 5, precio_unitario: 45},
  {producto: "Monitor", cantidad: 3, precio_unitario: 300},
  {producto: "Impresora", cantidad: 1, precio_unitario: 150},
  {producto: "Laptop", cantidad: 1, precio_unitario: 1000},
  {producto: "Mouse", cantidad: 5, precio_unitario: 75},
  {producto: "Teclado", cantidad: 3, precio_unitario: 100},
  {producto: "Monitor", cantidad: 1, precio_unitario: 700},
  {producto: "Impresora", cantidad: 7, precio_unitario: 950}
])

db.ventas.aggregate([{$group: {_id: "$producto", totalVentas: {$sum: "$cantidad"}}}])
```

## Output:

```
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("68115b94a2df312225e38c18"),
    ObjectId("68115b94a2df312225e38c19"),
    ObjectId("68115b94a2df312225e38c1a"),
    ObjectId("68115b94a2df312225e38c1b"),
    ObjectId("68115b94a2df312225e38c1c"),
    ObjectId("68115b94a2df312225e38c1d"),
    ObjectId("68115b94a2df312225e38c1e"),
    ObjectId("68115b94a2df312225e38c1f"),
    ObjectId("68115b94a2df312225e38c20"),
    ObjectId("68115b94a2df312225e38c21")
  ]
}
{ "_id" : "Laptop", "totalVentas" : 3 }
{ "_id" : "Mouse", "totalVentas" : 15 }
{ "_id" : "Teclado", "totalVentas" : 8 }
{ "_id" : "Monitor", "totalVentas" : 4 }
{ "_id" : "Impresora", "totalVentas" : 8 }
```

### Ejercicio 6:

```
db.clientes.insertMany([
  {nombre: "Mateo", apellido: "Rodriguez"},
  {nombre: "Ana", apellido: "Soteldo"},
  {nombre: "Franco", apellido: "Villaruel"},
  {nombre: "Matias", apellido: "Viñedo"},
  {nombre: "Sergio", apellido: "Cacha"}
])
db.clientes.createIndex({apellido: 1, nombre: 1})
```

### Ejercicio 7:

```
db.cursos.insertMany([
  {_id: 1, nombre: "Matematica"},
  {_id: 2, nombre: "Literatura"},
  {_id: 3, nombre: "Biologia"},
  {_id: 4, nombre: "Quimica"},
  {_id: 5, nombre: "Fisica"}
])
db.alumnos.insertMany([
  {nombre: "Ana", edad: 15, id_cursos: [1, 4, 5]},
  {nombre: "Alejo", edad: 14, id_cursos: [2, 3, 4]},
  {nombre: "Sergio", edad: 16, id_cursos: [5, 4, 1]}
])
```

### Ejercicio 8:

```
db.cursos.insertMany([
  {_id: 1, nombre: "Matematica"},
  {_id: 2, nombre: "Literatura"},
  {_id: 3, nombre: "Biologia"},
  {_id: 4, nombre: "Quimica"},
  {_id: 5, nombre: "Fisica"}
])
db.alumnos.insertMany([
  {nombre: "Ana", edad: 15, id_cursos: [1, 4, 5]},
  {nombre: "Alejo", edad: 14, id_cursos: [2, 3, 4]},
  {nombre: "Sergio", edad: 16, id_cursos: [5, 4, 1]}
])

db.alumnos.aggregate([
  [
    { $lookup: {
      from: "cursos",
      localField: "id_cursos",
      foreignField: "_id",
      as: "id_cursos"
    } }
  ]
])
```

## Output:

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4, 5 ] }
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6811642174b64f23576d222d"),
    ObjectId("6811642174b64f23576d222e"),
    ObjectId("6811642174b64f23576d222f")
  ]
}
{ "_id" : ObjectId("6811642174b64f23576d222d"), "nombre" : "Ana", "edad" : 15, "id_cursos" : [ { "_id" : 1, "nombre" : "Matematica" }, { "_id" : 4, "nombre" : "Quimica" }, { "_id" : 5, "nombre" : "Fisica" } ] }
{ "_id" : ObjectId("6811642174b64f23576d222e"), "nombre" : "Alejo", "edad" : 14, "id_cursos" : [ { "_id" : 2, "nombre" : "Literatura" }, { "_id" : 3, "nombre" : "Biologia" }, { "_id" : 4, "nombre" : "Quimica" } ] }
{ "_id" : ObjectId("6811642174b64f23576d222f"), "nombre" : "Sergio", "edad" : 16, "id_cursos" : [ { "_id" : 1, "nombre" : "Matematica" }, { "_id" : 4, "nombre" : "Quimica" }, { "_id" : 5, "nombre" : "Fisica" } ] }
```

## Ejercicio 9:

Las ventajas de utilizar “Replica set” es que nos permite que a la hora de que el nodo primario falle el servidor pueda ser redirigido al nodo secundario, previniendo pérdidas de datos, errores.

Y los beneficios que aporta el “Sharding” en una base de datos de alto volumen, es que nos permite declarar que datos son más importantes que otros y distribuirlos en varios rangos, así nos facilita a la hora de manipular dichos datos.

## Ejercicio 10:

Para crear un usuario con con permisos de lectura y escritura, primero utilizamos la función “db.createUser” para crear el usuario, le damos un nombre, una contraseña, y roles, en este caso, le damos el rol de “readWriter”, que le permite al usuario leer y escribir.

```
db.createUser({ user: "tester", pwd: "12345", roles: ["readWrite"] })
```

Y los comandos necesarios para hacer backup y restauración de una base de datos son: “mongodump” y “mongorestore”.