

1. Inserción Datos

```
Inventario> db.Producto.insertOne({sku:000001,genero:'camisa',talla:'L',precio:100,existencia:[{tienda:'SE',cantidad:8},{tienda:'CA',cantidad:12}],tipo:'hombre'})
{
  acknowledged: true,
  insertedId: ObjectId("63765766c82c4a2f43de2876")
}
Inventario> db.Producto.insertOne({sku:000002,genero:'camisa',talla:'M',precio:50,existencia:[{tienda:'SE',cantidad:15},{tienda:'CA',cantidad:10}],tipo:'mujer'})
{
  acknowledged: true,
  insertedId: ObjectId("637657c1c82c4a2f43de2877")
}
Inventario> db.Producto.insertOne({sku:000003,genero:'traje',talla:'M',precio:450,existencia:[{tienda:'SE',cantidad:5},{tienda:'CA',cantidad:15}],tipo:'mujer'})
{
  acknowledged: true,
  insertedId: ObjectId("637657ecc82c4a2f43de2878")
}
Inventario> db.Producto.insertOne({sku:000004,genero:'pantalón',talla:'L',precio:50,existencia:[{tienda:'SE',cantidad:100},{tienda:'CA',cantidad:50}],tipo:'hombre'})
{
  acknowledged: true,
  insertedId: ObjectId("6376581ac82c4a2f43de2879")
}
```

```
Inventario> db.Producto.find()
[
  {
    _id: ObjectId("63765766c82c4a2f43de2876"),
    sku: 1,
    genero: 'camisa',
    talla: 'L',
    precio: 100,
    existencia: [ { tienda: 'SE', cantidad: 8 }, { tienda: 'CA', cantidad: 12 } ],
    tipo: 'hombre'
  },
  {
    _id: ObjectId("637657c1c82c4a2f43de2877"),
    sku: 2,
    genero: 'camisa',
    talla: 'M',
    precio: 50,
    existencia: [ { tienda: 'SE', cantidad: 15 }, { tienda: 'CA', cantidad: 10 } ],
    tipo: 'mujer'
  },
  {
    _id: ObjectId("637657ecc82c4a2f43de2878"),
    sku: 3,
    genero: 'traje',
    talla: 'M',
    precio: 450,
    existencia: [ { tienda: 'SE', cantidad: 5 }, { tienda: 'CA', cantidad: 15 } ],
    tipo: 'mujer'
  },
  {
    _id: ObjectId("6376581ac82c4a2f43de2879"),
    sku: 4,
    genero: 'pantalón',
    talla: 'L',
    precio: 50,
    existencia: [ { tienda: 'SE', cantidad: 100 }, { tienda: 'CA', cantidad: 50 } ],
    tipo: 'hombre'
  }
]
Inventario>
```

2. Abrimos una nueva tienda “CO”, donde solo se va a vender ropa de hombre y va a comenzar con un stock (cantidad) de cada producto de 10 prendas. Actualiza la BD.

```
Inventario> db.Producto.updateOne({tipo:"hombre"},{$push:{"existencia":{"tienda":"CO",cantidad:10}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Inventario> db.Producto.find()
```

```

Inventario> db.Producto.find()
[
  {
    _id: ObjectId("637657c1c82c4a2f43de2877"),
    sku: '000011',
    genero: 'camisa',
    talla: 'M',
    precio: 50,
    existencia: [ { tienda: 'SE', cantidad: 15 }, { tienda: 'CA', cantidad: 10 } ],
    tipo: 'mujer'
  },
  {
    _id: ObjectId("637657ecc82c4a2f43de2878"),
    sku: '000002',
    genero: 'traje',
    talla: 'M',
    precio: 450,
    existencia: [ { tienda: 'SE', cantidad: 5 }, { tienda: 'CA', cantidad: 15 } ],
    tipo: 'mujer'
  },
  {
    _id: ObjectId("6376581ac82c4a2f43de2879"),
    sku: '000009',
    genero: 'pantalon',
    talla: 'L',
    precio: 50,
    existencia: [
      { tienda: 'SE', cantidad: 100 },
      { tienda: 'CA', cantidad: 50 },
      { tienda: 'CO', cantidad: 10 }
    ],
    tipo: 'hombre'
  },
  {
    _id: ObjectId("63765d5ec82c4a2f43de287a"),
    sku: 1,
    genero: 'camisa',
    talla: 'L',
    precio: 100,
    existencia: [ { tienda: 'SE', cantidad: 8 }, { tienda: 'CA', cantidad: 12 } ],
    tipo: 'hombre'
  }
]
Inventario>

```

3. Consulta la información de los dos productos de menor precio, muestra el sku, género y precio.

```

Inventario> db.Producto.find({}, {sku:1, genero:1, precio:1}).limit(2).sort({precio:1})
[
  {
    _id: ObjectId("6376581ac82c4a2f43de2879"),
    sku: '000009',
    genero: 'pantalon',
    precio: 50
  },
  {
    _id: ObjectId("637657c1c82c4a2f43de2877"),
    sku: '000011',
    genero: 'camisa',
    precio: 50
  }
]
Inventario>

```

4. Mostrar los datos sku y precio de todos los productos que no se venden en la tienda "CO" o cuyo precio sea superior a 50, ordenados por precio descendente.

```
Inventario> db.Producto.find({$or:[{$nor:[{"existencia.tienda":"CO"}]},{precio:{$gt:50}}]},{sku:1,precio:1}).sort({precio:1})
[
  {
    _id: ObjectId("637657c1c82c4a2f43de2877"),
    sku: '000011',
    precio: 50
  },
  {
    _id: ObjectId("63765d5ec82c4a2f43de287a"),
    sku: '000001',
    precio: 100
  },
  {
    _id: ObjectId("637657ecc82c4a2f43de2878"),
    sku: '000002',
    precio: 450
  }
]
Inventario>
```

5. Mostrar los datos sku, genero, talla y tipo de aquellos productos que tengan un stock inferior a 10 prendas en alguna tienda y su precio sea mayor de 100, ordenados por talla y sku.

```
Inventario> db.Producto.find({$and:[{"existencia.cantidad":{$lt:10}},{precio:{$gt:100}}]},{sku:1,genero:1,talla:1,tip:1})
[
  {
    _id: ObjectId("637657ecc82c4a2f43de2878"),
    sku: '000002',
    genero: 'traje',
    talla: 'M',
    tipo: 'mujer'
  }
]
Inventario>
```

DISEÑO

1. Añadir en la BD los datos sobre las ventas (tienda, sku, cantidad y fecha) qué diseño elegirías? razónalo y modifica la BD con un ejemplo.

```
Inventario> db.Ventas.insertOne({tienda:"CO",stock:[{Producto:ObjectId("6376581ac82c4a2f43de2879"),sku:"000009",stock:10},{Producto:ObjectId("6376581ac82c4a2f43de2879"),sku:"000009",stock:10}],fecha: new Date(2022-11-21)})
```

1. SUBCONSULTAS

```
Inventario> db.Ventas.aggregate([{$lookup:{from:"Producto", localField:"stock.Producto", foreignField:"_id",as:"tablaPadre"}},{ $project:{'stock.stock':1}}])
[
  {
    _id: ObjectId("637b63ee5b981c06eb29c5f5"),
    stock: [ { stock: 10 }, { stock: 10 } ]
  }
]
Inventario> db.Ventas.aggregate([{$lookup:{from:"Producto", localField:"stock.Producto", foreignField:"_id",as:"tablaPadre"}},{ $project:{'stock.stock':1}}])
```

From: Tabla Padre

localField: columna donde esta el id de la foranea (tabla donde esta la id foranea)

foreignField: columna de la tabla padre donde esta el id que vamos a referenciar.

as: nombre que le ponemos por defecto para que salga la subconsulta

\$project: es el select para mostrar o no lo que queríamos

2. Qué índices crearías, suponiendo que el volumen de productos es muy grande (999999), Crea los índices en la BD y razona el motivo por el que has creado cada uno de ellos.
¿Cómo comprobarías si ha mejorado el rendimiento de la BD?

```
Inventario> db.Ventas.createIndex({fecha:1})
fecha_1
Inventario>
```

```
Inventario> db.Ventas.find({fecha:{$gte:1970}}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'Inventario.Ventas',
    indexFilterSet: false,
    parsedQuery: { fecha: { '$gte': 1970 } },
    queryHash: '3787E495',
    planCacheKey: '006D6C5C',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { fecha: 1 },
        indexName: 'fecha_1',
        isMultiKey: false,
        multiKeyPaths: { fecha: [ ] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { fecha: [ '[1970, inf.0]' ] }
      }
    },
    rejectedPlans: [ ]
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 0,
    executionTimeMillis: 6,
    totalKeysExamined: 0,
    totalDocsExamined: 0,
    executionStages: {
      stage: 'FETCH',
      nReturned: 0,
      executionTimeMillisEstimate: 0,

```

3. Cuántos productos hay por tipo, su precio máximo y mínimo

```
Inventario> db.Producto.aggregate([{$group:{_id:"$tipo",maximo:{$max:"$precio"},Minimo:{$min:"$precio"}}}])
[
  { _id: 'mujer', maximo: 450, Minimo: 50 },
  { _id: 'hombre', maximo: 100, Minimo: 50 }
]
Inventario>
```

4. Cuántos productos hay de cada género y su precio medio.

```
Inventario> db.Producto.aggregate([{$group:{_id:"$genero",generos:{$sum:1},media:{$avg:"$precio"}}}])
[
  { _id: 'camisa', generos: 2, media: 75 },
  { _id: 'traje', generos: 1, media: 450 },
  { _id: 'pantalon', generos: 1, media: 50 }
]
Inventario>
```

5. Muestra el stock total existente (entre todas las tiendas) de cada producto.

```
Inventario> db.Producto.aggregate([{$unwind:"$existencia"},{$group:{_id:{"id":"$sku"},sumaTotal:{$sum:{"$existencia.cant*$ad"}}}}])
[
  { _id: { id: '000001' }, sumaTotal: 20 },
  { _id: { id: '000009' }, sumaTotal: 160 },
  { _id: { id: '000011' }, sumaTotal: 25 },
  { _id: { id: '000002' }, sumaTotal: 20 }
]
Inventario>
```

6. Visualización, de mayor a menor por tienda y producto, de la valoración de las existencias.