NoSQL Implementation:

The database and collections was created in MongoDB and the following queries were performed:

Use Cases:

1) Retrieving Order Details where orders are supplied from the Inventory id = 2

```
> show dbs;
< admin                         40.00 KiB
  config                        72.00 KiB
  local                         72.00 KiB
  optimized_supply_chain   200.00 KiB
> use optimized_supply_chain
<    'switched to db optimized_supply_chain'
> db.orders.find({inventory_id:2})
<    {
        _id: ObjectId("643c042a05722608ffdee882"),
        order_id: 1,
        inventory_id: 2,
        order_date: '2022-07-02 13:34:56',
        product_id: 89,
        selling_price: 20.74,
        quantity_ordered: 3
     }
```

2) Finding the Total sales based on Order ID of different customers:

```
> db.orders.aggregate([
    {
      $group: {
        _id: "$order_id",
        totalSales: { $sum: { $multiply: ["$selling_price"
      }
    },
    {
      $project: {
        order_id: "$_id",
        totalSales: 1,
        _id: 0
      }
    }
  ])
<    {
        totalSales: 56.16,
        order_id: 15
```

3) Retrieve Data from collection inventory_stock and sort in ascending order in basis of quantity ordered:

```
optimized_supply_chain> db.inventory_stock.find().sort({quantity:1})
[
  {
    _id: ObjectId("643c056105722608ffdeee7b"),
    inventory_id: 3,
    product_id: 21,
    quantity: 1
  },
  {
    _id: ObjectId("643c056105722608ffdeeea1"),
    inventory_id: 10,
    product_id: 13,
    quantity: 1
  },
  {
    id: ObjectId("643c056105722608ffdeeeb1")
```

4) Retrieving data from collection CUSTOMERS whose location is greater than 500 i.e.: in the Western region.

```
optimized_supply_chain> db.customers.find({location_id:{$gt:500}})
[
  {
    _id: ObjectId("643c051c05722608ffdeec6b"),
    customer_id: 1,
    customer_name: 'Benditt',
    location_id: 540
  },
  {
    _id: ObjectId("643c051c05722608ffdeec7d"),
    customer_id: 19,
    customer_name: 'Warkup',
    location_id: 547
  },
```