NoSQL - Illustration

DB / Collection

_id	Item	Price	Size	Quantity	Date
1	Americanos	5	Short	22	2022-01-15T08:00:00Z
2	Cappuccino	6	Short	12	2022-01-16T09:00:00Z
3	Lattes	15	Grande	25	2022-01-16T09:05:00Z
4	Mochas	25	Tall	11	2022-02-17T08:00:00Z
5	Americanos	10	Grande	12	2022-02-18T21:06:00Z
6	Cappuccino	7	Tall	20	2022-02-20T10:07:00Z
7	Lattes	25	Tall	30	2022-02-21T10:08:00Z
8	Americanos	10	Grande	21	2022-02-22T14:09:00Z
9	Cappuccino	10	Grande	17	2022-02-23T14:09:00Z
10	Americanos	8	Tall	15	2022-02-25T14:09:00Z

```
db.sales.insertMany([
   { "_id" : 1, "item" : "Americanos", "price" : 5, "size": "Short", "quantity" : 22, "date" : ISODate("2022-01-15T08:00:00Z") },
   { "_id" : 2, "item" : "Cappuccino", "price" : 6, "size": "Short","quantity" : 12, "date" : ISODate("2022-01-16T09:00:00Z") },
   { "_id" : 3, "item" : "Lattes", "price" : 15, "size": "Grande","quantity" : 25, "date" : ISODate("2022-01-16T09:05:00Z") },
   { "_id" : 4, "item" : "Mochas", "price" : 25,"size": "Tall", "quantity" : 11, "date" : ISODate("2022-02-17T08:00:00Z") },
   { "_id" : 5, "item" : "Americanos", "price" : 10, "size": "Grande","quantity" : 12, "date" : IS
ODate("2022-02-18T21:06:00Z") },
  { "_id" : 6, "item" : "Cappuccino", "price" : 7, "size": "Tall","quantity" : 20, "date" : ISODat
e("2022-02-20T10:07:00Z") },
  { "_id" : 7, "item" : "Lattes", "price" : 25,"size": "Tall", "quantity" : 30, "date" : ISODate("2
022-02-21T10:08:00Z") },
  { "_id" : 8, "item" : "Americanos", "price" : 10, "size": "Grande","quantity" : 21, "date" : IS
ODate("2022-02-22T14:09:00Z") },
  { "_id" : 9, "item" : "Cappuccino", "price" : 10, "size": "Grande","quantity" : 17, "date" : IS
ODate("2022-02-23T14:09:00Z") },
  { "_id" : 10, "item" : "Americanos", "price" : 8, "size": "Tall","quantity" : 15, "date" : ISODat
e("2022-02-25T14:09:00Z")}
]);
```

CRUD

1. Create (Insert)

```
// Insert a new sale record
db.sales.insertOne({
  item: "Espresso",
  price: 4,
  size: "Short",
  quantity: 10,
  date: ISODate("2022-03-01T09:00:00Z")
})
```

---

2. Read (Find)

```
// Find all sales of Americanos
db.sales.find({ item: "Americanos" })

// Find sales with quantity > 20
db.sales.find({ quantity: { $gt: 20 } })

// Find only Cappuccino sales, show item & quantity only
db.sales.find(
  { item: "Cappuccino" },
  { _id: 0, item: 1, quantity: 1 }
)
```

3. Update

```
// Update price of Short Cappuccino to 8
db.sales.updateOne(
  { item: "Cappuccino", size: "Short" },
  { $set: { price: 8 } }
)

// Increase all Tall drink prices by 2
db.sales.updateMany(
  { size: "Tall" },
  { $inc: { price: 2 } }
)
```

4. Delete

```
// Delete one sale of Lattes with price 25
db.sales.deleteOne({ item: "Lattes", price: 25 })

// Delete all sales before Feb 1, 2022
db.sales.deleteMany({ date: { $lt: ISODate("2022-02-01") } })
```

Aggregation

$match , $group, $sort

```
db.sales.aggregate([
   {
      $match: { item: "Americanos" }
   },
   {
        $group: {
            _id: "$size",
            totalQty: {$sum: "$quantity"}
        }
    },
    {
        $sort: { totalQty : -1}
    }
]);
```

$count

```
db.sales.aggregate([
  {
    $group: {
      _id: '$item',
      itemCount: { $count: {} },
    },
  },
])

db.sales.aggregate([
  {
    $group: {
      _id: '$item',
      itemCount: { $count: {} },
    },
  },
  {
    $match: { itemCount: { $gt: 2 } },
  },
]);
```

$limit

```
db.sales.aggregate([
  { $limit: 3 }
]);
```

$lookup

Create a new collection:

```
db.suppliers.insertMany([
  { _id: 1, item: "Americanos", supplier: "Starbucks" },
  { _id: 2, item: "Cappuccino", supplier: "Cafe Coffee Day" },
  { _id: 3, item: "Lattes", supplier: "Costa" },
  { _id: 4, item: "Mochas", supplier: "Blue Tokai" }
]);
```

Query:

```
db.sales.aggregate([
  {
    $lookup: {
      from: "suppliers",        // other collection
      localField: "item",       // field in sales
      foreignField: "item",     // field in suppliers
      as: "supplierInfo"
    }
  }
]);
```

$project

```
db.sales.aggregate([
  {
    $project: {
      _id: 0,
      item: 1,
      quantity: 1,
      totalPrice: { $multiply: ["$price", "$quantity"] }
    }
  }
]);
```

Retrieve the top 3 best-performing coffee items from February 2022 with their supplier information and calculated metrics. (combined example)

```
db.sales.aggregate([
  // $match: Filter sales from February 2022 only
  {
    $match: {
      date: {
        $gte: ISODate("2022-02-01T00:00:00Z"),
        $lt: ISODate("2022-03-01T00:00:00Z")
      }
    }
  },

  // $lookup: Join with suppliers collection to get supplier information
  {
    $lookup: {
      from: "suppliers",
      localField: "item",
      foreignField: "item",
      as: "supplier_info"
    }
  },

  // $group: Group by item and calculate total quantity and revenue
  {
    $group: {
      _id: "$item",
      total_quantity: { $sum: "$quantity" },
      total_revenue: { $sum: { $multiply: ["$price", "$quantity"] } },
      avg_price: { $avg: "$price" },
      supplier: { $first: { $arrayElemAt: ["$supplier_info.supplier", 0] } }
    }
  },
  // $project: Shape the output and add calculated fields
  {
    $project: {
      _id: 0,
      item_name: "$_id",
      total_quantity: 1,
      total_revenue: 1,
      avg_price: { $round: ["$avg_price", 2] },
      supplier: 1,
      revenue_per_unit: {
        $round: [{ $divide: ["$total_revenue", "$total_quantity"] }, 2]
      }
    }
  },

  // $sort: Sort by total revenue in descending order
  {
    $sort: {
      total_revenue: -1
    }
  },

  // $limit: Get only top 3 performing items
  {
    $limit: 3
  },

]);
```