```
// grouping multiple columns
// use("practice_db");
// 1. for each region+segment find total orders,total profits

db.orders.find();
db.orders.aggregate([
    {
        $group: {
            _id: {
                region: "$customer.region",
                segment: "$customer.segment"
            },
            totalOrders: { $sum: 1 },
            totalProfit: { $sum: "$profit" }
        }
    },
    {
        $project:
        {
            _id: 0,
            region: "$_id.region",
            segment: "$_id.segment",
            totalOrders: 1,
            totalProfit: 1
        }
    }
]);

// 2. for each customer+region find
// total profit, only include customers whose total profit>1000

db.orders.aggregate([
    {
        $group: {
            _id: {
                name: "$customer.name",
                region: "$customer.region"
            },
            totalProfit: { $sum: "$profit" }
        }
    },
    { $match: { totalProfit: { $gt: 1000 } } },
    {
        $project: {
            _id: 0,
            name: "$_id.name",
            region: "$_id.region",
            totalProfit: 1
        }
    }
]);
```

```
// 3. for each month+region calculate
// average profit(sort by average profit descending)

db.orders.aggregate([
    {
        $group: {
            _id: {
                month: { $month: "$orderDate" },
                region: "$customer.region"
            },
            avgProfit: { $avg: "$profit" }
        }
    },
    { $sort: { avgProfit: -1 } },
    {
        $project: {
            _id: 0,
            month: "$_id.month",
            region: "$_id.region",
            avgProfit: 1
        }
    }
]);

db.orders.find();
// 4. For each category + region, find:
// number of orders
//     (One order counts once even if it has multiple items of same category)

// here we need to count unique orders per category+region
// so first unwind, then group by region+category+orderid -- this ensures one
order is counted once
// next group again by category+region
db.orders.aggregate([
    { $unwind: "$items" },
    {
        $group: {
            _id: {
                category: "$items.category",
                region: "$customer.region",
                orderId: "$orderId"
            }
        }
    },

    // count orders per category+region
    {
        $group: {
            _id: {
                category: "$_id.category",
                region: "$_id.region"
            },
            orderCount: { $sum: 1 }
        }
```

```
    },
    {
        $project: {
            _id: 0,
            category: "$_id.category",
            region: "$_id.region",
            orderCount: 1
        }
    }
]);


db.orders.find();
// 5. for each region+segment+category find
// total profit,only include electronics and furniture
db.orders.aggregate([
    { $unwind: "$items" },
    {
        $match: {
            "items.category": { $in:["Electronics","Furniture"]}
    }},
    {
        $group: {
            _id: {
                region: "$customer.region",
                segment: "$customer.segment",
                category: "$items.category"
            },
            totalProfit: { $sum: "$profit" }
        }
    },
    {
        $project: {
            _id: 0,
            region: "$_id.region",
            segment: "$_id.segment",
            category: "$_id.category",
            totalProfit: 1
        }
    }
]);


// 6. For each year + month + region, find:
// total orders
// Sort by year, then month.

db.orders.aggregate([
    {
        $group: {
            _id: {
                year: { $year: "$orderDate" },
                month: { $month: "$orderDate" },
                region: "$customer.region"
```

```
            },
            totalOrders: { $sum: 1 }
        }
    },
    {
        $sort: {
            "_id.year": 1,
            "_id.month":1
         },
    },
    {
        $project: {
            _id: 0, year: "$_id.year", month: "$_id.month", region: "$_id.region",
totalOrders: 1
        }
    }
]);

// 7. For each customer + category + subCategory, find:total number of items
purchased
db.orders.aggregate([
    { $unwind: "$items" },
    {
        $group: {
            _id: {
                customerName: "$customer.name",
                category: "$items.category",
                subCategory: "$items.subCategory"
            },
            totalitems: { $sum: "$items.quantity" }
        }
    },
    {
        $project: {
            _id: 0,
            customerName: "$_id.customerName",
            category: "$_id.category",
            subCategory: "$_id.subCategory",
            totalitems: 1
        }
    }
]);


// 8.From Corporate segment, find top 3 region + category combinations by total
profit.
db.orders.aggregate([
    { $unwind: "$items" },
    { $match: { "customer.segment": "Corporate" } },
    {
        $group: {
            _id: {
                region: "$customer.region",
                category: "$items.category"
```

```
            },
            totalProfit: { $sum: "$profit" }
        }
    },
    {$sort:{totalProfit:-1}},
    { $limit: 3 }
]);

// 9. Find top 5 customers by total profit, but:only for Electronics
// only for orders after 2023-01-01

db.orders.aggregate([
    { $unwind: "$items" },
    {
        $match: {
            $and: [{ "items.category": "Electronics" }, { orderDate: { $gt:
ISODate("2023-01-01") } }]
        }
    },
    {
        $group: {
            _id: "$customer.name",
            totalProfit: { $sum: "$profit" }
        }
    },
    { $sort: { totalProfit: -1 } },
    { $limit: 5 }
]);

// Q10 For each region + month, find:
// total profit
// Then return only the top month per region.

db.orders.aggregate([
    {
        $group: {
            _id: {
                region: "$customer.region",
                month: { $month: "$orderDate" }
            },
            totalProfit: { $sum: "$profit" }
        }
    },
    // Sort months by profit (highest first) per region
    {
        $sort: {
            "_id.region": 1,
            totalProfit: -1
        }
    },
    // Pick top month per region
    {
        $group: {
            _id: "$_id.region",
```

```
            topMonth: { $first: "$_id.month" },
            maxProfit: { $first: "$totalProfit" }
        }
    },
    {
        $project: {
            _id: 0,
            region: "$_id",
            topMonth: 1,
            maxProfit: 1
        }
    }
]);

db.orders.find();
// 11. Count how many unique customers exist per:
// region
// segment

db.orders.aggregate([
    {
        $group: {
            _id: {
                region: "$customer.region",
                segment: "$customer.segment"
            },
            customers: { $addToSet: "$customer.id" }
        }
    },
    {
        $project: {
            _id: 0, region: "$_id.region", segment: "$_id.segment",
            totalUnique: { $size: "$customers" }
        }
    }
]);

// 12.For each category + subCategory, count:
// number of distinct orders
db.orders.aggregate([
    { $unwind: "$items" },
    {
        $group: {
            _id: {
                cat: "$items.category",
                subcat: "$items.subCategory"
            },
            uniqueOrders: { $addToSet: "$orderId" }
        }
    },
    {
        $project: {
            _id: 0, category: "$_id.cat", subCategory: "$_id.subcat",
            distinctOrders: { $size: "$uniqueOrders" }
```

```
        }
    }
]);

// 14. Find regions where:
// number of orders > 50
// AND average profit > 500
db.orders.aggregate([
    { $group: { _id: "$customer.region", numOrders: { $sum: 1 }, avgProfit: {
$avg: "$profit" } } },
    {
        $match: {
            $and: [{ "numOrders": { $gt: 50 } },
            { "avgProfit": { $gt: 500 } }]
        }
    },
    {
        $project: {
            _id: 0, region: "$_id", numOrders: 1, avgProfit: 1
        }
    }
]);
```