1. Find the total revenue (price × quantity) for each item, sorted from highest to lowest.

ANS:

```
db.sales.aggregate([
    {
        $group: {
            _id: "$item",  // Group by item
            totalRevenue: { $sum: { $multiply: ["$price", "$quantity"] } }
        }
    },
    {
        $sort: { totalRevenue: -1 }
    }
]);
```

2. Calculate the total quantity sold per month in 2022.

```
db.sales.aggregate([
    {
        $match: {
            date: {
                $gte: ISODate("2022-01-01T00:00:00Z"),
                $lt: ISODate("2023-01-01T00:00:00Z")
            }
        }
    },
    {
        $group: {
            _id: {
                year: { $year: "$date" },
                month: { $month: "$date" }
            },
            totalQuantity: { $sum: "$quantity" }

        }
    },
    {
        $sort: { "_id.year": 1, "_id.month": 1 }
    }
]);
```

3) Find all items where price is greater than 10 and size is not 'Short'.

```
db.sales.find({
    price: { $gt: 10 },
    size: { $ne: 'Short' }  });
```

4. Get all Cappuccino sales with quantity between 10 and 20.

```
db.sales.find({
    item: "Cappuccino",
    quantity: { $gte: 10, $lte: 20 }
});
```

5. Query to find items where the item name starts with "A".

```
db.sales.find({
    item: { $regex: /^A/, $options: 'i' }
});
```

6. Find all records that do not have the field size.

```
db.sales.find({
    size: { $exists: false }
});
```

7. List all items sold in February 2022.

```
db.sales.find({
    date: {
        $gte: ISODate("2022-02-01T00:00:00Z"),
        $lt: ISODate("2022-03-01T00:00:00Z")
    }
});
```

8. Find all sales that are either "Grande" or "Tall" but not "Americanos".

```
db.sales.find({
    size: { $in: ["Grande", "Tall"] },
    item: { $ne: "Americanos" }
});
```

9. Find sales where the quantity is more than twice the price.

```
db.sales.find({
    quantity: { $gt: { $multiply: ["$price", 2] } }
});
```

10. Find all sales where the price is greater than the average price of their respective size.

```
db.sales.aggregate([
    {
        $group: {
            _id: "$size",                    // Group by size
            averagePrice: { $avg: "$price" }
        }
    },
    {
        $lookup: {
            from: "sales",
            localField: "_id",
            foreignField: "size",
            as: "salesData"
        }
    },
    {
        $unwind: "$salesData"
    },
    {
        $match: {
            $expr: {
                $gt: ["$salesData.price", "$averagePrice"]
            }
        }
    },
    {
        $replaceRoot: { newRoot: "$salesData" }
    }
]);
```

11. Filter sales where the total revenue is even and exceeds 100.

```
db.sales.find({
  $where: function() {
    const total = this.price * this.quantity;
    return total > 100 && total % 2 === 0;
  }
})
```

11. Find Sales Where the Day of Week Matches Quantity's Last Digit

```
db.sales.find({
    $expr: {
        $eq: [
            { $dayOfWeek: "$date" },
            { $mod: [{ $last: { $toString: "$quantity" } }, 10] }
        ]
    }
});
```

12. Find Sales Where the Month is Prime and Quantity is Odd

```
db.sales.find({
    $expr: {
        $and: [
            { $in: [{ $month: "$date" }, [2, 3, 5, 7, 11]] },
            { $eq: [{ $mod: ["$quantity", 2] }, 1] }
        ]
    }
});
```

13. Find Sales with "Suspicious Quantities" (Divisible by 5 or 7)

```
db.sales.find({
    $or: [
        { $expr: { $eq: [{ $mod: ["$quantity", 5] }, 0] } },  // Quantity divisible by 5
        { $expr: { $eq: [{ $mod: ["$quantity", 7] }, 0] } }   // Quantity divisible by 7
    ]
});
```