```
// ============================
// ✅ USE DATABASE
// ============================
use CollegeDB


// ============================
// ✅ SAMPLE COLLECTION
// ============================
db.sales.drop()
db.sales.insertMany([
    { _id: 1, item: "Laptop", qty: 5, price: 50000, category: "Electronics", city: "Pune" },
    { _id: 2, item: "Mouse", qty: 20, price: 500, category: "Electronics", city: "Mumbai" },
    { _id: 3, item: "Keyboard", qty: 15, price: 800, category: "Electronics", city: "Pune" },
    { _id: 4, item: "Chair", qty: 10, price: 2500, category: "Furniture", city: "Delhi" },
    { _id: 5, item: "Table", qty: 8, price: 5000, category: "Furniture", city: "Pune" }
])


// ============================
// ✅ 1. SIMPLE AGGREGATION (SUM)
// ============================
// Total quantity of all products
db.sales.aggregate([
    { $group: { _id: null, totalQty: { $sum: "$qty" } } }
])
```

```
// ==============================
// ✅ 2. GROUP BY CATEGORY (AVG + SUM)
// ==============================
db.sales.aggregate([
  {
    $group: {
      _id: "$category",
      totalQty: { $sum: "$qty" },
      avgPrice: { $avg: "$price" }
    }
  }
])




// ==============================
// ✅ 3. MATCH + GROUP
// ==============================
// Only Electronics category
db.sales.aggregate([
  { $match: { category: "Electronics" } },
  { $group: { _id: "$item", totalQty: { $sum: "$qty" } } }
])
```

```
// ==============================
// ✅ 4. SORT + LIMIT
// ==============================
db.sales.aggregate([
   { $sort: { price: -1 } },
   { $limit: 3 }
])




// ==============================
// ✅ 5. PROJECT (Select specific fields)
// ==============================
db.sales.aggregate([
   {
     $project: {
        item: 1,
        totalValue: { $multiply: ["$qty", "$price"] },
        _id: 0
     }
   }
])




// ==============================
// ✅ 6. FACET (Multiple pipelines in one)
// ==============================
```

```
db.sales.aggregate([

  {

    $facet: {

      "Top Items": [

        { $sort: { qty: -1 } },

        { $limit: 2 }

      ],

      "City Count": [

        { $group: { _id: "$city", count: { $sum: 1 } } }

      ]

    }

  }

])




// ==============================
// ✅ 7. LOOKUP (JOIN Operation)
// ==============================

db.suppliers.drop()
db.suppliers.insertMany([

  { item: "Laptop", supplier: "Dell" },

  { item: "Mouse", supplier: "Logitech" },

  { item: "Chair", supplier: "Godrej" }

])
```

```
// JOIN sales with suppliers

db.sales.aggregate([

   {

      $lookup: {

         from: "suppliers",

         localField: "item",

         foreignField: "item",

         as: "SupplierDetails"

      }

   }

])
```

// ==============================
// ✅ 8. INDEXING
// ==============================

```
// Create index on item

db.sales.createIndex({ item: 1 })


// Create compound index

db.sales.createIndex({ category: 1, price: -1 })
```

// ==============================
// ✅ 9. CHECK INDEXES

```
// ===============================

db.sales.getIndexes()




// ===============================

// ✅ 10. CHECK QUERY SPEED (Using explain)

// ===============================

db.sales.find({ item: "Laptop" }).explain("executionStats")




// ===============================

// ✅ 11. DROP INDEX

// ===============================

db.sales.dropIndex({ item: 1 })
```