MongoDB Assignment 3

```
use Sales_Data

db.sales.insertMany{

  "_id": 1, "product": "Laptop", "category": "Electronics", "price": 800, "quantity": 5, "date": ISODate("2024-03-01T10:00:00Z"), "store": "A"

},

{

  "_id": 2, "product": "Phone", "category": "Electronics", "price": 600, "quantity": 10, "date": ISODate("2024-03-02T12:00:00Z"), "store": "B"

},

{

  "_id": 3, "product": "TV", "category": "Electronics", "price": 1200, "quantity": 3, "date": ISODate("2024-03-03T15:00:00Z"), "store": "A"

},

{

  "_id": 4, "product": "Shoes", "category": "Fashion", "price": 50, "quantity": 20, "date": ISODate("2024-03-04T16:00:00Z"), "store": "C"

},

{

  "_id": 5, "product": "Watch", "category": "Fashion", "price": 150, "quantity": 7, "date": ISODate("2024-03-05T18:00:00Z"), "store": "B"

}
```

1. Total sales per product.

db.sales.aggregate([

{ $group:{_id:"$product",quantity:{$push:"$quantity"} } }

])

```
mycompiler_mongodb> ... ... [
  { _id: 'Laptop', quantity: [ 5 ] },
  { _id: 'Phone', quantity: [ 10 ] },
  { _id: 'Shoes', quantity: [ 20 ] },
  { _id: 'Watch', quantity: [ 7 ] },
  { _id: 'TV', quantity: [ 3 ] }
]
```

2. Total revenue per product.

db.sales.aggregate([
{ $group: { _id: "$product", totalRevenue: {$sum: {$multiply:
["$quantity", "$price"] } } } }
])

```
mycompiler_mongodb> ... ... ... ... ... [
  { _id: 'Watch', totalRevenue: 1050 },
  { _id: 'TV', totalRevenue: 3600 },
  { _id: 'Laptop', totalRevenue: 4000 },
  { _id: 'Phone', totalRevenue: 6000 },
  { _id: 'Shoes', totalRevenue: 1000 }
]
```

3. Total revenue per category.

db.students.aggregate([
{ $group: { _id: "$category", totalRevenue: {$sum: {$multiply:
["$quantity", "$price"] } } } }
])

```
mycompiler_mongodb> ... ... [
  { _id: 'Fashion', totalRevenue: 2050 },
  { _id: 'Electronics', totalRevenue: 13600 }
]
```

4. Count of products per category.

```
db.sales.aggregate([
  { $group: { _id: "$category", productCount: {$sum: 1} } }
])
```

```
mycompiler_mongodb> [
  { _id: 'Electronics', productCount: 3 },
  { _id: 'Fashion', productCount: 2 }
]
```

5. Store-wise total sales.

```
db.sales.aggregate([
  { $group:{_id:"$store",totalSale:{$sum:{$multiply:["$price","$quantity"]}
   } } }
])
```

```
mycompiler_mongodb> ... ... [
  { _id: 'A', totalSale: 7600 },
  { _id: 'B', totalSale: 7050 },
  { _id: 'C', totalSale: 1000 }
]
```

6. Average price of products per category.

```
db.sales.aggregate([
  {$group:{_id:{category: "$category", product:
   "$product"},avgPrice:{$avg:"$price"} } }
])
```

```
mycompiler_mongodb> ... ... [
  {
    _id: { category: 'Electronics', product: 'Laptop' },
    avgPrice: 800
  },
  { _id: { category: 'Electronics', product: 'TV' }, avgPrice: 1200 },
  { _id: { category: 'Fashion', product: 'Shoes' }, avgPrice: 50 },
  { _id: { category: 'Electronics', product: 'Phone' }, avgPrice: 600 },
  { _id: { category: 'Fashion', product: 'Watch' }, avgPrice: 150 }
]
```

7. Top-selling product.

```
db.sales.aggregate([
    {$group:{_id:"$product",maximum_sale:{$max:{$sum:"$quantity"}}}},{$
    sort:{maximum_sale:-1}},{$limit:1}
])
```

```
[ { _id: 'Shoes', maximum_sale: 20 } ]
```

8. Total sales for Electronics category.

```
db.sales.aggregate([
    {$match:{category:"Electronics"}},{$group:{_id:null,total_sale:{$sum:"$q
    uantity"}}}
])
```

```
[ { _id: null, total_sale: 18 } ]
```

9. Sales trend over time (day-wise total sales).

```
db.sales.aggregate([
   {$group:{_id:{$dateToString:{format:"%Y-%m-%d",date:"$date"}},
    totalSales: {$sum: { $multiply: ["$price", "$quantity"] } } } },
   { $sort: { _id: 1 } }
])
```

```
mycompiler_mongodb> ... ... [
  { _id: '2024-03-01', totalSales: 4000 },
  { _id: '2024-03-02', totalSales: 6000 },
  { _id: '2024-03-03', totalSales: 3600 },
  { _id: '2024-03-04', totalSales: 1000 },
  { _id: '2024-03-05', totalSales: 1050 }
]
```

10. Highest revenue-generating product.

```
db.sales.aggregate([
    {$group:{_id:"$product",totalRevenue:{$sum:{$multiply:["$quantity","$
    price"]}}}},{$sort:{totalRevenue:-1}},{$limit:1}
])
```

```
[ { _id: 'Phone', totalRevenue: 6000 } ]
```

11. Average revenue per sale.

db.sales.aggregate([

{$group:{_id:null,avgRevenue:{ $avg:{ $sum:{ $multiply: ["$price","$quantity"] } } } } }

])

```
[ { _id: null, avgRevenue: 3130 } ]
```

12. Sales performance per store.

db.sales.aggregate([
{$group:{_id:"$store",sales:{$sum:{$multiply:["$price","$quantity"]}}}}
])

```
mycompiler_mongodb> ... ... [
  { _id: 'B', sales: 7050 },
  { _id: 'C', sales: 1000 },
  { _id: 'A', sales: 7600 }
]
```

13. Products sold more than 5 times.

db.sales.aggregate([
{$group:{_id:"$product",quantity:{$sum:"$quantity"}}},
{$match:{quantity:{$gt:5}}}
])

```
mycompiler_mongodb> ... ... ... [
  { _id: 'Shoes', quantity: 20 },
  { _id: 'Watch', quantity: 7 },
  { _id: 'Phone', quantity: 10 }
]
```

14. Least sold product.

```
db.sales.aggregate([
   {$group:{_id:"$product",quantity:{$sum:"$quantity"}}},
   {$sort:{quantity:1}},
   {$limit:1}
])
```

```
[ { _id: 'TV', quantity: 3 } ]
```

15. Monthly sales summary.

```
db.sales.aggregate([
    {$group:{_id:{year:{$year:"$date"},month:{$month:"$date"}},
    sales:{$sum:{$multiply:["$quantity","$price"]}}}},
   {$sort:{sales:1}}
])
```

```
[ { _id: { year: 2024, month: 3 }, sales: 15650 } ]
```

16. Number of unique products sold.

```
db.sales.aggregate([{$count:"product"}])
```

```
[ { product: 5 } ]
```

17. Maximum and minimum priced product.

```
db.sales.aggregate([
   {$group:{_id:"$product",minPrice:{$min:"$price"}}},
   {$sort:{minPrice:1}},
   {$limit:1}
])
db.sales.aggregate([
   {$group:{_id:"$product",maxPrice:{$max:"$price"}}},
   {$sort:{maxPrice:-1}},
   {$limit:1}
])
```

```
[ { _id: 'Shoes', minPrice: 50 } ]
[ { _id: 'TV', maxPrice: 1200 } ]
```

18. Total revenue per product in descending order.

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

```
mycompiler_mongodb> ... ... ... [
  { _id: 'Phone', totalRevenue: 6000 },
  { _id: 'Laptop', totalRevenue: 4000 },
  { _id: 'TV', totalRevenue: 3600 },
  { _id: 'Watch', totalRevenue: 1050 },
  { _id: 'Shoes', totalRevenue: 1000 }
]
```

19. Revenue generated per store per category.

```
db.sales.aggregate([
    {$group:{_id:{store:"$store",category:"$category"},revenue:{$sum:{$multiply:["$price","$quantity"]}}}}
])
```

```
mycompiler_mongodb> ... ... [
  { _id: { store: 'B', category: 'Electronics' }, revenue: 6000 },
  { _id: { store: 'C', category: 'Fashion' }, revenue: 1000 },
  { _id: { store: 'B', category: 'Fashion' }, revenue: 1050 },
  { _id: { store: 'A', category: 'Electronics' }, revenue: 7600 }
]
```