MongoDB ASSIGNMENT 3

1. Total sales per product.

```
Assignment> db.Sales.aggregate([{$group:{_id:"$product" , Total_Sales:{$sum:{$multiply:['$quantity' , '$price']} }}} , {$sort:{Total_Sales:-1}}])
[
  { _id: 'Phone', Total_Sales: 6000 },
  { _id: 'Laptop', Total_Sales: 4000 },
  { _id: 'TV', Total_Sales: 3600 },
  { _id: 'Watch', Total_Sales: 1050 },
  { _id: 'Shoes', Total_Sales: 1000 }
]
```

2. Total revenue per product.

```
Assignment> db.Sales.aggregate([{$group:{_id:"$product" , Total_Rev:{$sum:{$multiply:['$quantity' , '$price']} }}} , {$sort:{Total_Rev:-1}}])
[
  { _id: 'Phone', Total_Rev: 6000 },
  { _id: 'Laptop', Total_Rev: 4000 },
  { _id: 'TV', Total_Rev: 3600 },
  { _id: 'Watch', Total_Rev: 1050 },
  { _id: 'Shoes', Total_Rev: 1000 }
]
```

3. Total revenue per category.

```
Assignment> db.Sales.aggregate([{$group:{_id:"$category" , Total_Rev:{$sum:{$multiply:['$quantity' , '$price']} }}} , {$sort:{Total_Rev:-1}}])
[
  { _id: 'Electronics', Total_Rev: 13600 },
  { _id: 'Fashion', Total_Rev: 2050 }
]
```

4. Count of products per category.

```
Assignment> db.Sales.aggregate([{$group:{_id:"$category" , Total_Product:{$count: {}}}} , {$sort:{Total_Product:-1}}])
[
  { _id: 'Electronics', Total_Product: 3 },
  { _id: 'Fashion', Total_Product: 2 }
]
```

5. Store-wise total sales.

```
Assignment> db.Sales.aggregate([{$group:{_id:"$store" , TotalSalesByStore:{$sum:{$multiply:['$quantity' , '$price']}}}} , {$sort:{TotalSalesByStore:-1}}])
[
  { _id: 'A', TotalSalesByStore: 7600 },
  { _id: 'B', TotalSalesByStore: 7050 },
  { _id: 'C', TotalSalesByStore: 1000 }
]
```

6. Average price of products per category.

```
Assignment> db.Sales.aggregate([{$group:{_id:"$category" , Avg_Price:{$avg:'$price'}}} , {$sort:{Avg_Price:-1}}])
[
  { _id: 'Electronics', Avg_Price: 866.6666666666666 },
  { _id: 'Fashion', Avg_Price: 100 }
]
```

7. Top-selling product.

```
Assignment> db.Sales.aggregate([{$group:{_id:"$product" , TopSelling:{$max:"$quantity"}}} , {$sort:{TopSelling:-1}} , {$limit:1}])
[ { _id: 'Shoes', TopSelling: 20 } ]
```

8. Total sales for Electronics category.

```
Assignment> db.Sales.aggregate([{$match:{category:"Electronics"}} , {$group:{_id:'Electronics' , TotalSales:{$sum:{$multiply:['$quantity' , '$price']}}}}])
[ { _id: 'Electronics', TotalSales: 13600 } ]
```

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

```
Assignment> db.Sales.aggregate([{$group:{_id:{$dateToString:{format:"%Y-%m-%d" , date:"$date"}} , MonthlySales:{$sum:{$multiply:['$quantity' , '$price']}}}}
])
[
  { _id: '2024-03-03', MonthlySales: 3600 },
  { _id: '2024-03-02', MonthlySales: 6000 },
  { _id: '2024-03-04', MonthlySales: 1000 },
  { _id: '2024-03-05', MonthlySales: 1050 },
  { _id: '2024-03-01', MonthlySales: 4000 }
]
```

10. Highest revenue-generating product.

```
Assignment> db.Sales.aggregate([{$group:{_id:'$product' , Highest_Rev:{$sum:{$multiply:['$quantity' , '$price']}}}} , {$sort:{Highest_Rev:-1}} , {$limit:1}]
)
[ { _id: 'Phone', Highest_Rev: 6000 } ]
```

11. Average revenue per sale.

```
Assignment> db.Sales.aggregate([{$project:{Sales:{$sum:{$multiply:['$quantity' , '$price']}}}} , {$group:{_id:'category' , AvgSales:{$avg:'$Sales'}}}])
[ { _id: 'category', AvgSales: 3130 } ]
```

12. Sales performance per store.

```
Assignment> db.Sales.aggregate([{$group:{_id:'$store' , SalesPerformence:{$sum:{$multiply:['$quantity' , '$price']}}}}])
[
  { _id: 'C', SalesPerformence: 1000 },
  { _id: 'A', SalesPerformence: 7600 },
  { _id: 'B', SalesPerformence: 7050 }
]
```

13. Products sold more than 5 times.

```
Assignment> db.Sales.find({quantity:{$gt:5}})
[
  {
    _id: 2,
    product: 'Phone',
    category: 'Electronics',
    price: 600,
    quantity: 10,
    date: ISODate('2024-03-02T12:00:00.000Z'),
    store: 'B'
  },
  {
    _id: 4,
    product: 'Shoes',
    category: 'Fashion',
    price: 50,
    quantity: 20,
    date: ISODate('2024-03-04T16:00:00.000Z'),
    store: 'C'
  },
  {
    _id: 5,
    product: 'Watch',
    category: 'Fashion',
    price: 150,
    quantity: 7,
    date: ISODate('2024-03-05T18:00:00.000Z'),
    store: 'B'
  }
]
```

14. Least sold product.

```
Assignment> db.Sales.aggregate([{$group:{_id:"$product" , TotalSales:{$sum:"$quantity"}}}  , {$sort:{"quantity":1}} , {$limit:1}])
[ { _id: 'TV', TotalSales: 3 } ]
```

15. Monthly sales summary.

```
Assignment> db.Sales.aggregate([{$group:{_id:{$dateToString:{format:"%Y-%m" , date:"$date"}} , MonthlySales:{$sum:{$multiply:['$quantity' , '$price']}}}}])
[ { _id: '2024-03', MonthlySales: 15650 } ]
```

16. Number of unique products sold.

```
Assignment> db.Sales.distinct("product").length
5
```

17. Maximum and minimum priced product.

```
Assignment> db.Sales.aggregate([{$group:{_id:'$product' , MaxPriced:{$max:"$price"} , MinPriced:{$min:"$price"}}}])
[
  { _id: 'TV', MaxPriced: 1200, MinPriced: 1200 },
  { _id: 'Watch', MaxPriced: 150, MinPriced: 150 },
  { _id: 'Phone', MaxPriced: 600, MinPriced: 600 },
  { _id: 'Shoes', MaxPriced: 50, MinPriced: 50 },
  { _id: 'Laptop', MaxPriced: 800, MinPriced: 800 }
]
```

18. Total revenue per product in descending order.

```
Assignment> db.Sales.aggregate([{$group:{_id:"$product" , Total_Rev:{$sum:{$multiply:['$quantity' , '$price']}}}} , {$sort:{Total_Rev:-1}}])
[
  { _id: 'Phone', Total_Rev: 6000 },
  { _id: 'Laptop', Total_Rev: 4000 },
  { _id: 'TV', Total_Rev: 3600 },
  { _id: 'Watch', Total_Rev: 1050 },
  { _id: 'Shoes', Total_Rev: 1000 }
]
```

19. Revenue generated per store per category.

```
Assignment> db.Sales.aggregate([{$group:{_id:{Store:"$store" , Category:"$category"} , Total_Rev:{$sum:{$multiply:['$quantity' , '$price']}}}} , {$sort:{Tot
al_Rev:-1}}])
[
  { _id: { Store: 'A', Category: 'Electronics' }, Total_Rev: 7600 },
  { _id: { Store: 'B', Category: 'Electronics' }, Total_Rev: 6000 },
  { _id: { Store: 'B', Category: 'Fashion' }, Total_Rev: 1050 },
  { _id: { Store: 'C', Category: 'Fashion' }, Total_Rev: 1000 }
]
```

20. Products contributing more than 50% revenue.