1. **Write Mongo query to retrieve documents from the orders in ascending order by total.**

ANS : db.orders.aggregate([

{ $sort: { total: 1 } }

])

2. **Write Mongo query to retrieve the oldest paymentMethod from the payments collection as "_id".**

ANS : db.payments.aggregate([

{ $sort: { payment_date: 1 } },

{ $limit: 1 },

{ $project: { _id: "$paymentMethod" } }

])

3. **Write Mongo query to retrieve 2nd and 3rd buyers from the buyers collection.**

ANS : db.buyers.aggregate([

{ $sort: { _id: 1 } },

{ $skip: 1 },

{ $limit: 2 }

])

4. **Write Mongo query to retrieve the most Expensive product from order_details.**

ANS : db.order_details.aggregate([

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

{ $limit: 1 }

])

5. **Write Mongo query to retrieve the first order from the orders as per the order_date.**

ANS : db.orders.aggregate([

{ $sort: { order_date: 1 } },

{ $limit: 1 }

])

6. **Write Mongo query to retrieve the last 3 orders from the orders collection who have less total amount.**

ANS : db.orders.aggregate([

{ $sort: { total: 1 } },

```
 { $limit: 3 }
])
```

7. **Write Mongo query to retrieve the most recent shipped order from the orders collection.**

ANS :
```
db.orders.aggregate([
 { $match: { status: "shipped" } },
 { $sort: { order_date: -1 } },
 { $limit: 1 }
])
```

8. **Write Mongo query to get the total revenue from all orders**

ANS :
```
db.orders.aggregate([
 { $group: { _id: null, totalRevenue: { $sum: "$total" } } },
 { $project: { _id: 0, totalRevenue: 1 } }
])
```

9. **Write Mongo query to retrieve all the orders that shipped before 2022-05-26**

ANS :
```
db.orders.aggregate([
 { $match: { ship_date: { $lt: new Date("2022-05-26") } } }
])
```

10. **Write Mongo query to find the maximum price as maxPrice of products and their names as name for each category.**

ANS :
```
db.products.aggregate([
 { $group: { _id: "$category_id", maxPrice: { $max: "$price" }, names: { $addToSet: "$name" } } },
 { $project: { category_id: "$_id", maxPrice: 1, names: 1, _id: 0 } }
])
```

11. **Write Mongo query to find Most used payment Method as paymentMethod and the number of time it is used as count.**

ANS :
```
db.payments.aggregate([
 { $group: { _id: "$paymentMethod", count: { $sum: 1 } } },
 { $sort: { count: -1 } },
 { $limit: 1 },
 { $project: { paymentMethod: "$_id", count: 1, _id: 0 } }
```

])

12. Write Mongo query to find the total count of orders by status.

ANS : db.orders.aggregate([

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

])

13. Write Mongo query to retrieve the orders grouped by customer_id with the max total.

ANS : db.orders.aggregate([

{ $group: { _id: "$customer_id", maxTotal: { $max: "$total" } } }

])

14.Assess the impact of RAM capacity on laptop prices and ratings. Group laptops by RAM capacity and analyze the average price and rating for each group. Identify any significant trends or outliers.

ANS : db.laptops.aggregate([

{ $group: { _id: "$RAM", avgPrice: { $avg: "$price" }, avgRating: { $avg: "$rating" } } },

{ $project: { RAM: "$_id", avgPrice: 1, avgRating: 1, _id: 0 } },

{ $sort: { RAM: 1 } }

])

15.Investigate the price and rating distribution for gaming laptops. Identify which brands are leading in the gaming laptop market by comparing the average price, rating, GPU type, and RAM capacity for laptops categorized as gaming.

ANS : db.laptops.aggregate([

{ $match: { category: "gaming" } },

{ $group: { _id: "$brand", avgPrice: { $avg: "$price" }, avgRating: { $avg: "$rating" }, avgRAM: { $avg: "$RAM" }, GPUtypes: { $addToSet: "$GPU" } } },

{ $project: { brand: "$_id", avgPrice: 1, avgRating: 1, avgRAM: 1, GPUtypes: 1, _id: 0 } },

{ $sort: { avgRating: -1 } }

])

16.Analyze the warranty periods offered by different brands and their correlation with laptop prices and ratings. Identify any patterns or insights regarding how warranty periods influence consumer ratings and pricing strategies.

ANS : db.laptops.aggregate([

{ $group: { _id: "$brand", avgWarranty: { $avg: "$warranty_period" }, avgPrice: { $avg: "$price" }, avgRating: { $avg: "$rating" } } },

```
 { $project: { brand: "$_id", avgWarranty: 1, avgPrice: 1, avgRating: 1, _id: 0 } },
 { $sort: { avgWarranty: -1 } }
])
```

17.Examine the relationship between processor brand (Intel, AMD, Apple) and laptop price, rating, and primary storage capacity. Group the laptops by processor brand and compare their average price, rating, and storage statistics to identify key differences.

ANS :

```
db.laptops.aggregate([
 { $group: { _id: "$processor_brand", avgPrice: { $avg: "$price" }, avgRating: { $avg: "$rating" }, avgStorage: { $avg: "$storage" } } },
 { $project: { processorBrand: "$_id", avgPrice: 1, avgRating: 1, avgStorage: 1, _id: 0 } },
 { $sort: { avgRating: -1 } }
])
```