CASE STUDY

DATABASE SYSTEM

NAME:SHIBAM BANIK

REG NO:23BKT0108

TOPIC: **E-Commerce Database System using MongoDB.**

INTRODUCTION:

An **e-commerce system** is a digital platform that enables businesses to sell products or services online. It allows users to browse items, add them to a cart, place orders, and make payments electronically. In addition, e-commerce systems often manage product catalogs, track inventory, process orders, and collect customer feedback, providing a seamless shopping experience.

Modern e-commerce systems typically include:

- **Product management:** Adding, updating, and categorizing products.
- **User management:** Registering, authenticating, and managing customers.
- **Order management:** Tracking purchases, payments, and delivery.
- **Reviews and ratings:** Allowing users to provide feedback.
- **Inventory management:** Keeping track of stock levels and availability.

NoSQL databases like **MongoDB** are particularly suitable for e-commerce systems due to the following reasons:

1. **Flexibility:** MongoDB stores data in JSON-like documents, allowing different products or users to have varied attributes without needing a fixed schema. For example, a T-shirt might have size and color options, while a laptop might have RAM and storage specifications.
2. **Scalability:** MongoDB supports horizontal scaling, which is essential for e-commerce platforms expecting high traffic and growing datasets.
3. **High Performance:** Its document-based structure enables fast read and write operations, which is crucial for handling real-time orders and user interactions.
4. **Rich Querying and Indexing:** MongoDB supports complex queries, filtering, and aggregation, making it easy to analyze orders, users, and product data efficiently.

Integrated Handling of Complex Data: Features like nested documents and arrays are perfect for storing products with multiple variants, order histories, or user reviews.

Collections in MongoDB for E-Commerce

1. **Products Collection**
   Stores information about items available for sale:
   - Name, description, price, category
   - Product images
   - Variants (size, color)
   - Ratings or average review
2. **Users Collection**
   Stores user information:
   - Name, email, password (hashed), contact details
   - Shipping addresses
   - Order history and wishlist
3. **Orders Collection**
   Tracks all purchase transactions:
   - User ID, products purchased, quantities
   - Total amount, payment status
   - Order status (pending, shipped, delivered)
4. **Reviews Collection**
   Stores user feedback on products:
   - Product ID, User ID
   - Rating (1–5 stars), review text
   - Timestamp for tracking feedback over time
5. **Inventory Collection**
   Tracks stock levels of products:
   - Product ID, quantity available
   - Warehouse location

CONTENT

DATABASE DESIGN:

ER-Style Diagram

You can generate an ER diagram using the following relationships:

- **Users** ↔ **Orders**: One user can place multiple orders.
- **Products** ↔ **Orders**: Each order contains one or more products.
- **Products** ↔ **Reviews**: Users can submit reviews for products.
- **Products** ↔ **Inventory**: Each product has inventory data to track stock levels.

Users (user_id) 1 --- * Orders (order_id, user_id)

Products (product_id) 1 --- * Orders.products

Products (product_id) 1 --- * Reviews (review_id, product_id, user_id)

Products (product_id) 1 --- 1 Inventory (inventory_id, product_id)

Users ↔ Orders

- The `orders` collection stores a field `user_id` referencing the `_id` of a user in the `users` collection.
- This allows fetching all orders placed by a specific user.

Orders ↔ Products

- Each order contains an array of products with `product_id` and quantity.

Products ↔ Reviews

- The `reviews` collection links each review to both a product and a user via `product_id` and `user_id`.
- This allows fetching all reviews for a product and also the reviews submitted by a user.

Products ↔ Inventory

- Each product has a corresponding document in the `inventory` collection.
- This tracks the quantity available and can include warehouse location or restock thresholds.

Design Benefits

- **Modularity:** Each collection focuses on a single type of data (users, products, orders, etc.), making updates and queries easier.
- **Scalability:** Collections can grow independently without impacting the structure of others.
- **Ease of Aggregation:** MongoDB allows joining collections using `$lookup`, enabling flexible reporting and analytics.

EXECUTION OF THE QUERIES:

SIMPLE QUERIES:

1. Find all products in the “Electronics” category:

```
db.products.find({ category: "Electronics" })
```

2. Count total number of users:

```
db.users.countDocuments()
```

3.Find orders placed by a specific user:

```
db.orders.find({ user_id: "U0005" })
```

4.Find top 5 most expensive products:

```
db.products.find().sort({ price: -1 }).limit(5)
```

5.Show all reviews with a 5-star rating:

```
db.reviews.find({ rating: 5 })
```

COMPLEX QUERIES:

1. Find average rating per product:

→

```
db.reviews.aggregate([
  { $group: { _id: "$product_id", avgRating: { $avg: "$rating" } } },
  { $sort: { avgRating: -1 } }
])
```

2.Find total number of orders and revenue per payment method:

→

```
db.orders.aggregate([
  { $group: { _id: "$payment_method", totalOrders: { $sum: 1 },
totalRevenue: { $sum: "$total" } } },
  { $sort: { totalRevenue: -1 } }
])
```

3.Find users from Delhi who purchased more than 3 orders:

→

```
db.orders.aggregate([
  { $group: { _id: "$user_id", ordersCount: { $sum: 1 } } },
  { $match: { ordersCount: { $gt: 3 } } },
  { $lookup: { from: "users", localField: "_id", foreignField: "user_id", as: "user_info" } },
  { $unwind: "$user_info" },
  { $match: { "user_info.location": "Delhi" } },
  { $project: { _id: 0, user: "$user_info.name", location: "$user_info.location", ordersCount: 1 } }
])
```

4.Find products that are out of stock:

→ `db.inventory.find({ stock: { $lte: 0 } })`

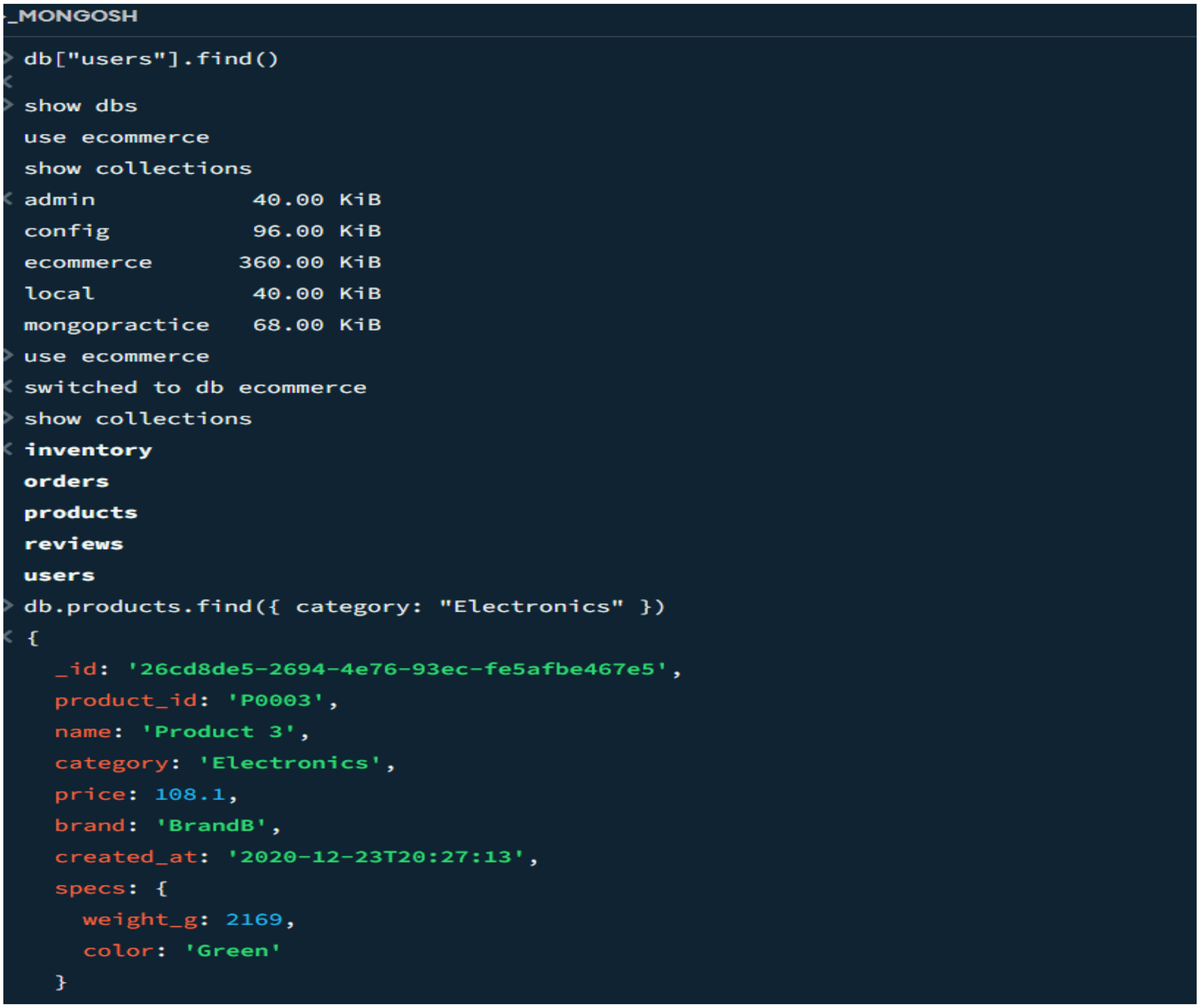
5. Find total items sold per product:

→

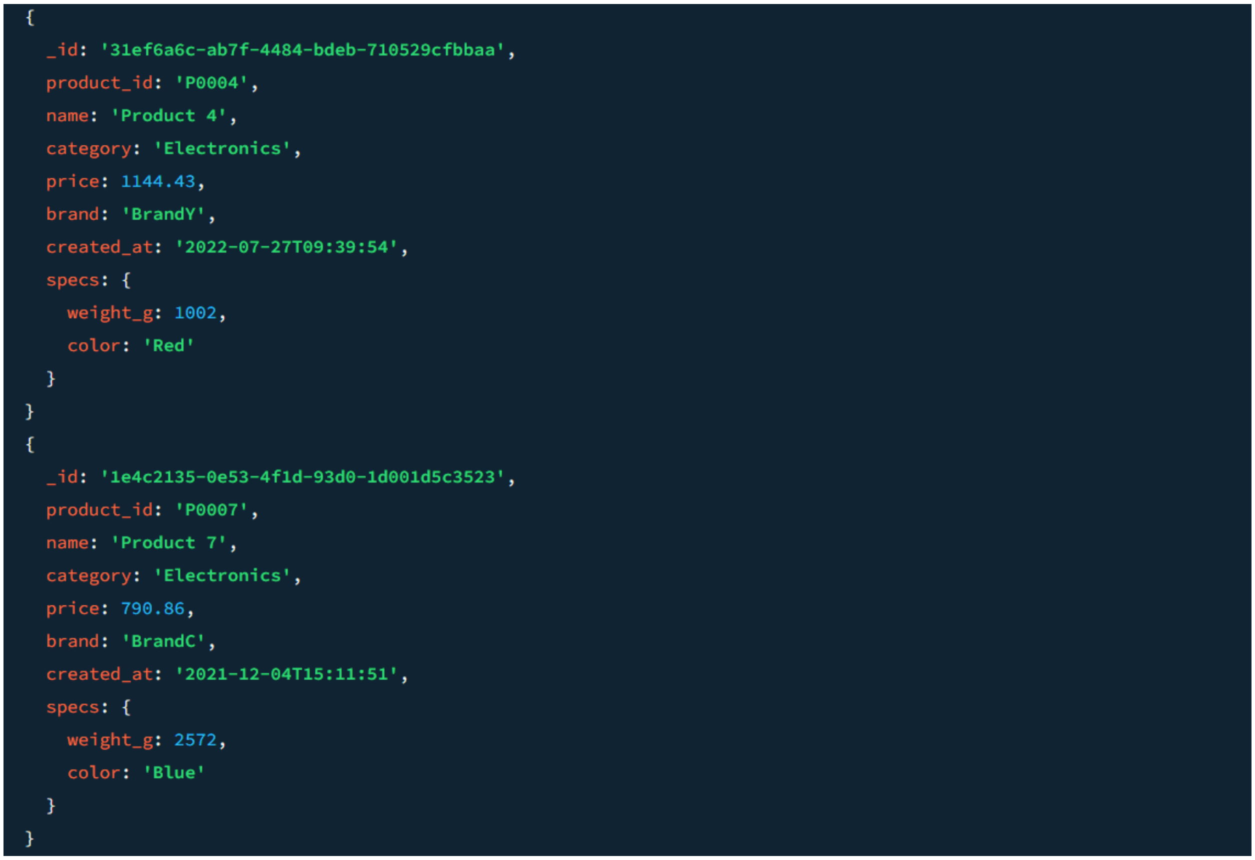
```
db.orders.aggregate([
  { $unwind: "$items" },
  { $group: { _id: "$items.product_id", totalSold: { $sum: "$items.quantity" } } },
  { $sort: { totalSold: -1 } }
])
```

EXECUTION SCREENSHOT:

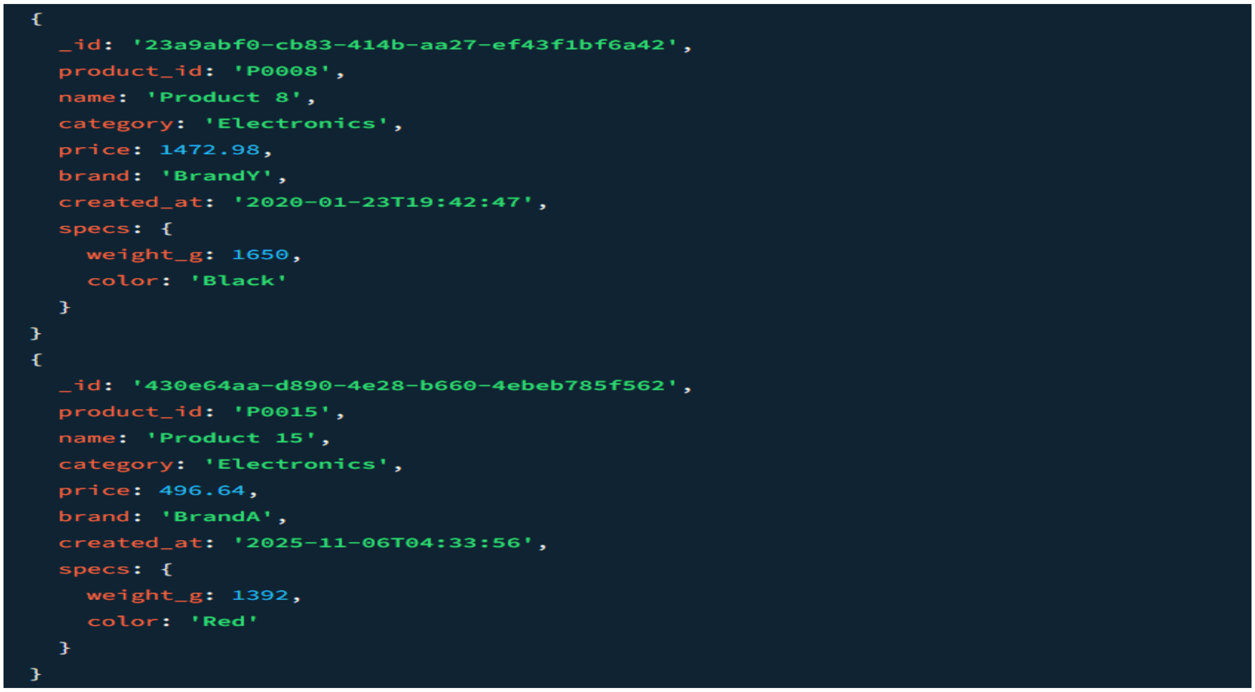
1.



```
> db["users"].find()
<
> show dbs
  use ecommerce
  show collections
< admin           40.00 KiB
  config          96.00 KiB
  ecommerce      360.00 KiB
  local           40.00 KiB
  mongopractice   68.00 KiB
> use ecommerce
< switched to db ecommerce
> show collections
< inventory
  orders
  products
  reviews
  users
> db.products.find({ category: "Electronics" })
< {
    _id: '26cd8de5-2694-4e76-93ec-fe5afbe467e5',
    product_id: 'P0003',
    name: 'Product 3',
    category: 'Electronics',
    price: 108.1,
    brand: 'BrandB',
    created_at: '2020-12-23T20:27:13',
    specs: {
      weight_g: 2169,
      color: 'Green'
    }
```

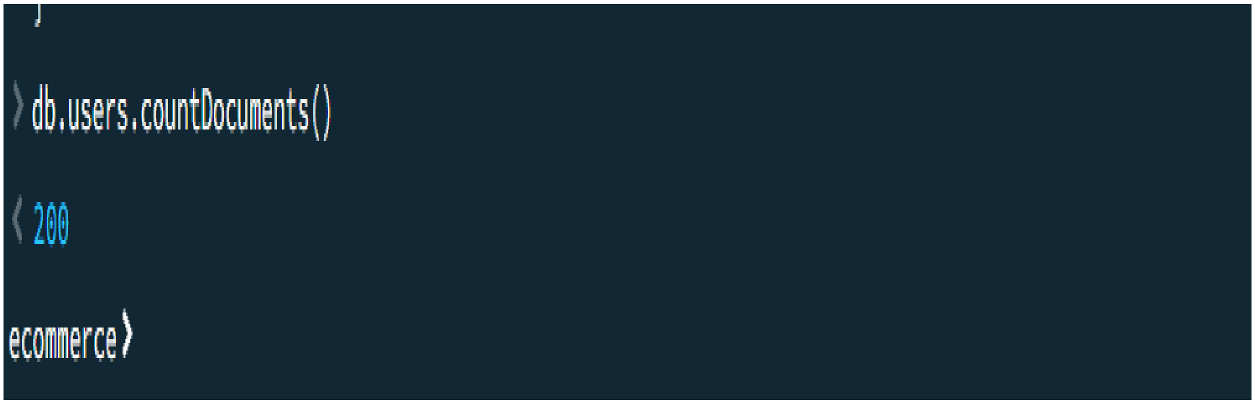


```
{
  _id: '31ef6a6c-ab7f-4484-bdeb-710529cfbbaa',
  product_id: 'P0004',
  name: 'Product 4',
  category: 'Electronics',
  price: 1144.43,
  brand: 'BrandY',
  created_at: '2022-07-27T09:39:54',
  specs: {
    weight_g: 1002,
    color: 'Red'
  }
}
{
  _id: '1e4c2135-0e53-4f1d-93d0-1d001d5c3523',
  product_id: 'P0007',
  name: 'Product 7',
  category: 'Electronics',
  price: 790.86,
  brand: 'BrandC',
  created_at: '2021-12-04T15:11:51',
  specs: {
    weight_g: 2572,
    color: 'Blue'
  }
}
```



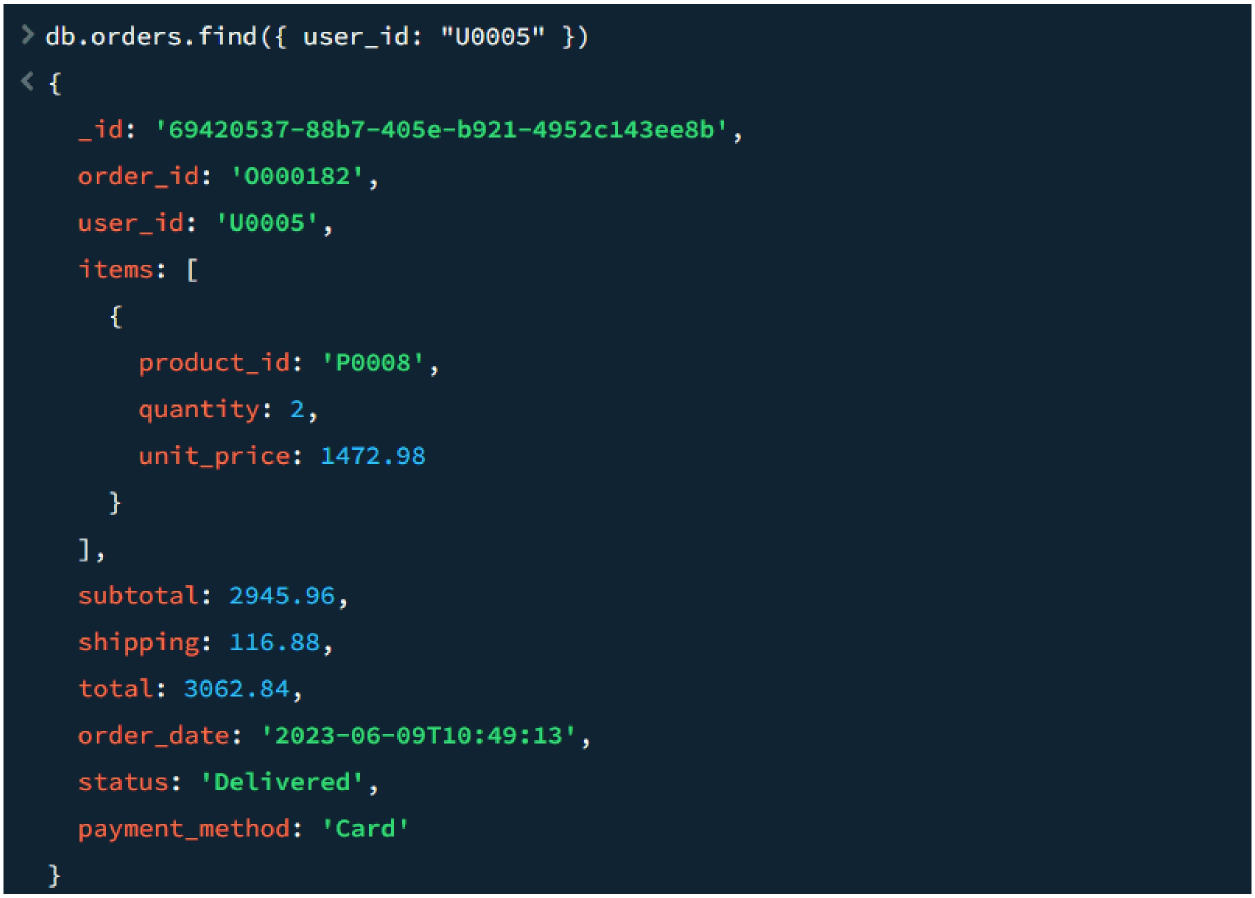
```
{
  _id: '23a9abf0-cb83-414b-aa27-ef43f1bf6a42',
  product_id: 'P0008',
  name: 'Product 8',
  category: 'Electronics',
  price: 1472.98,
  brand: 'BrandY',
  created_at: '2020-01-23T19:42:47',
  specs: {
    weight_g: 1650,
    color: 'Black'
  }
}
{
  _id: '430e64aa-d890-4e28-b660-4ebeb785f562',
  product_id: 'P0015',
  name: 'Product 15',
  category: 'Electronics',
  price: 496.64,
  brand: 'BrandA',
  created_at: '2025-11-06T04:33:56',
  specs: {
    weight_g: 1392,
    color: 'Red'
  }
}
```

2.

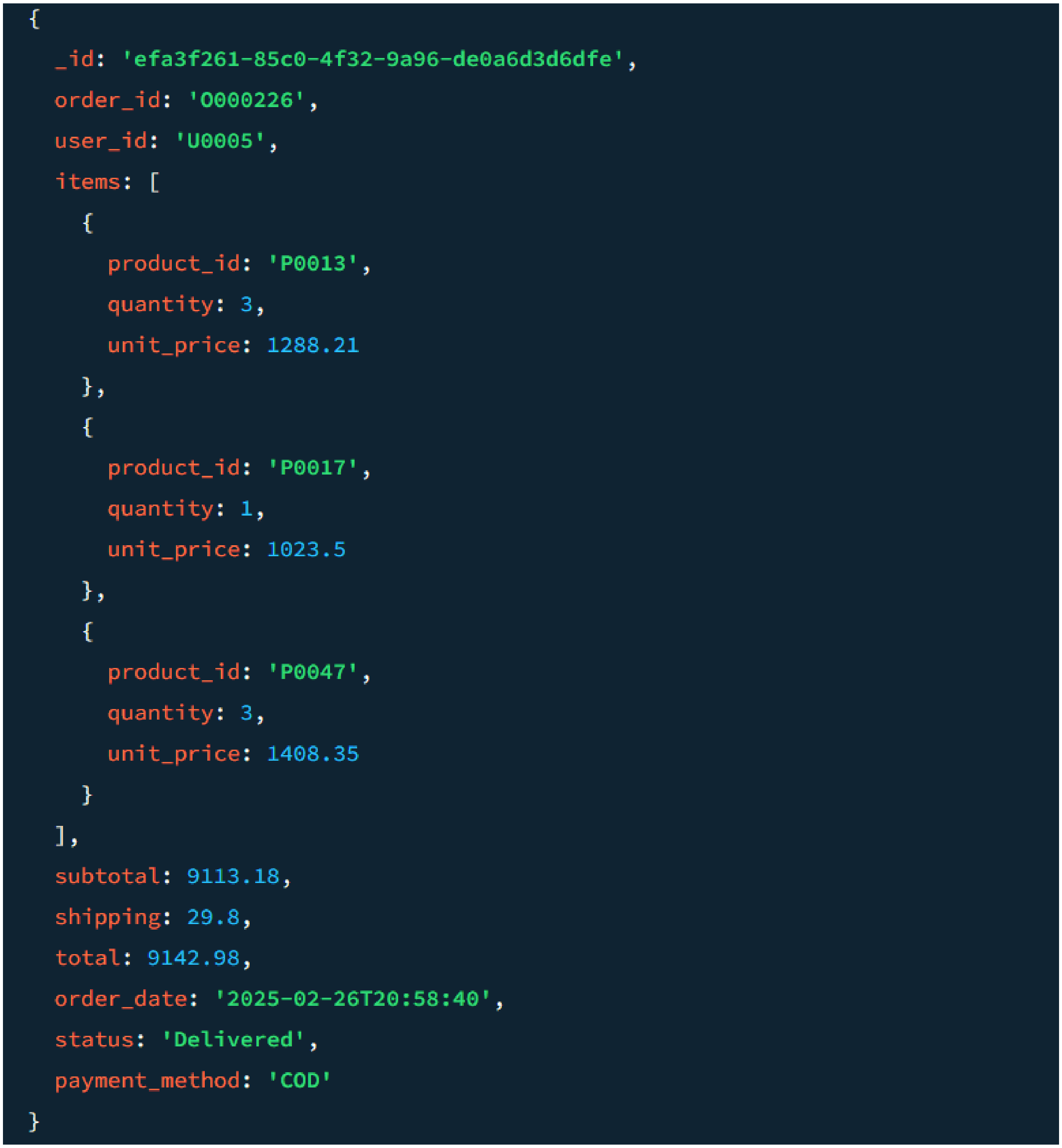


```
  }
> db.users.countDocuments()
< 200
ecommerce>
```

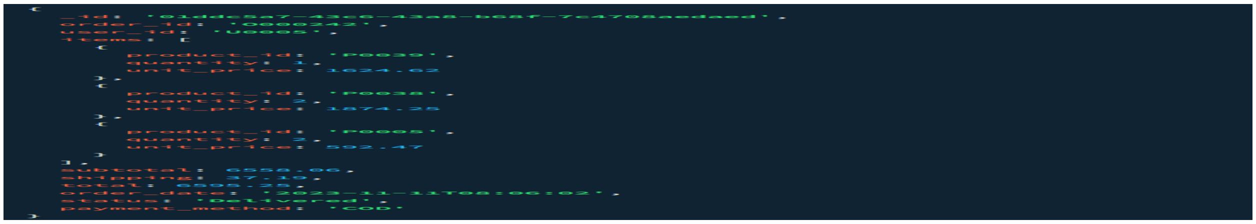
3.



```
> db.orders.find({ user_id: "U0005" })
< {
    _id: '69420537-88b7-405e-b921-4952c143ee8b',
    order_id: 'O000182',
    user_id: 'U0005',
    items: [
      {
        product_id: 'P0008',
        quantity: 2,
        unit_price: 1472.98
      }
    ],
    subtotal: 2945.96,
    shipping: 116.88,
    total: 3062.84,
    order_date: '2023-06-09T10:49:13',
    status: 'Delivered',
    payment_method: 'Card'
  }
```

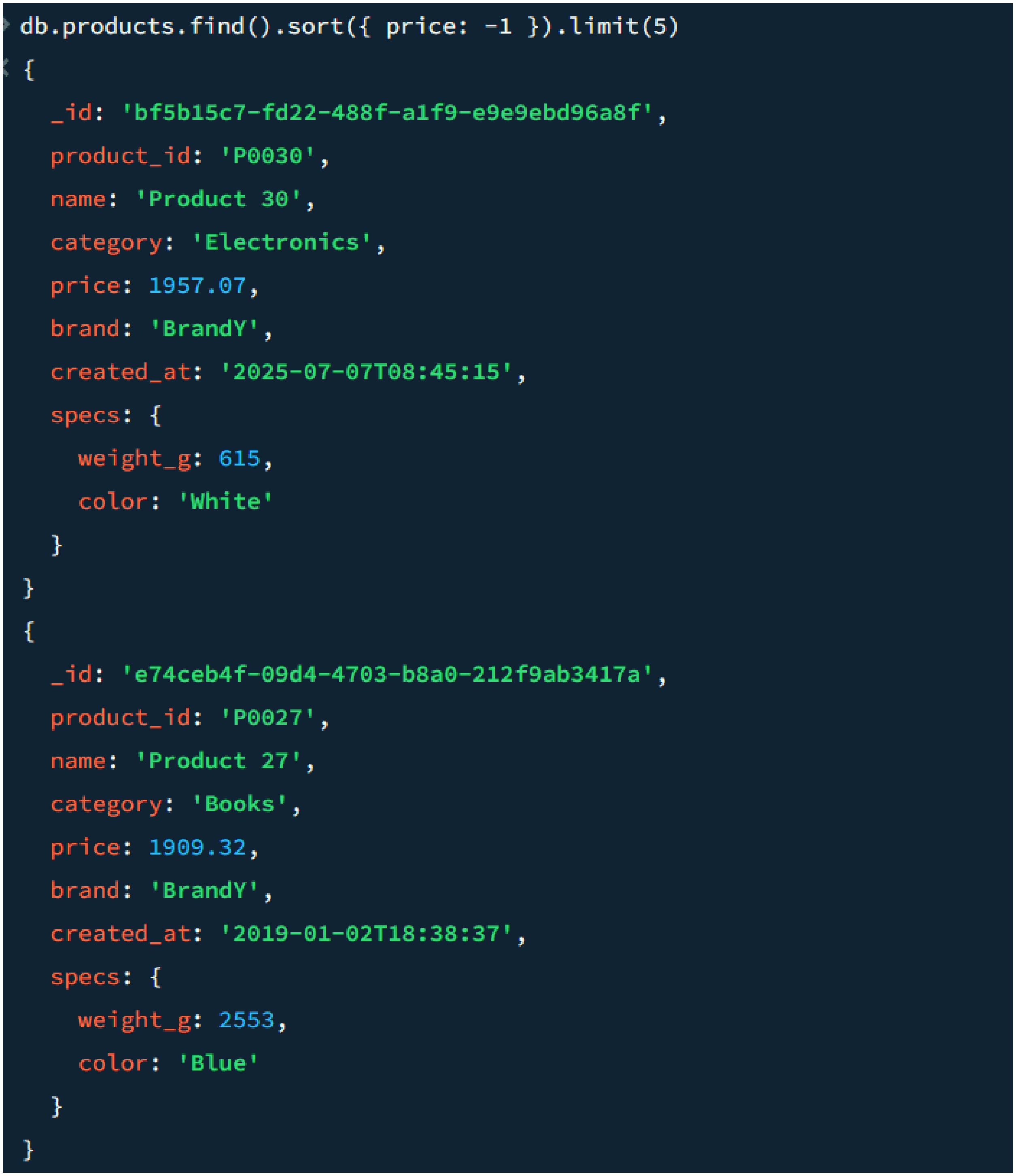


```
{
  _id: 'efa3f261-85c0-4f32-9a96-de0a6d3d6dfe',
  order_id: 'O000226',
  user_id: 'U0005',
  items: [
    {
      product_id: 'P0013',
      quantity: 3,
      unit_price: 1288.21
    },
    {
      product_id: 'P0017',
      quantity: 1,
      unit_price: 1023.5
    },
    {
      product_id: 'P0047',
      quantity: 3,
      unit_price: 1408.35
    }
  ],
  subtotal: 9113.18,
  shipping: 29.8,
  total: 9142.98,
  order_date: '2025-02-26T20:58:40',
  status: 'Delivered',
  payment_method: 'COD'
}
```

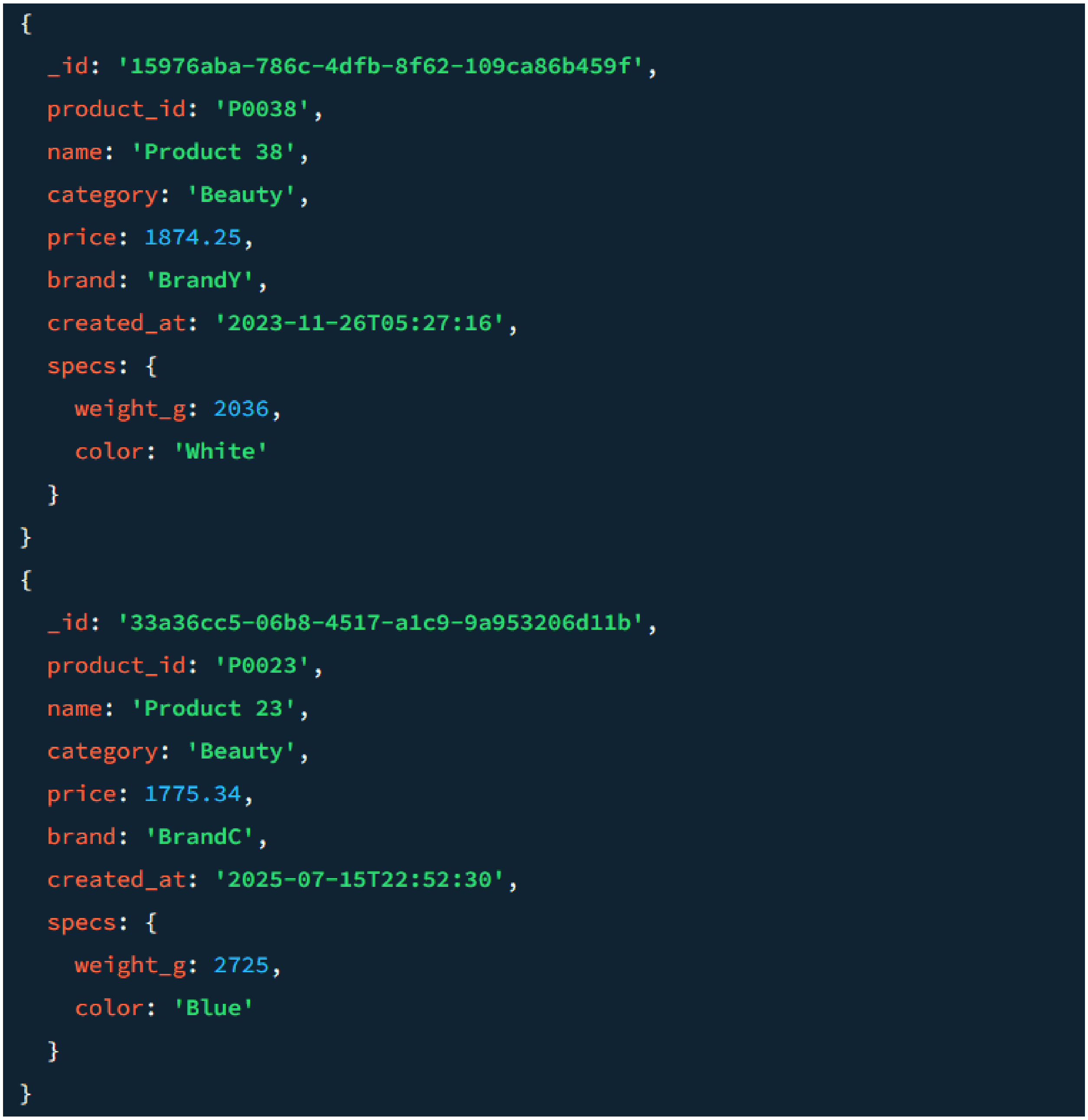


```
{
  _id: '01ddc5a7-43c6-43a8-b68f-7c4708aedaed',
  order_id: 'O000242',
  user_id: 'U0005',
  items: [
    {
      product_id: 'P0039',
      quantity: 1,
      unit_price: 1624.62
    },
    {
      product_id: 'P0038',
      quantity: 2,
      unit_price: 1874.25
    },
    {
      product_id: 'P0005',
      quantity: 2,
      unit_price: 592.47
    }
  ],
  subtotal: 6558.06,
  shipping: 37.19,
  total: 6595.25,
  order_date: '2023-11-11T08:06:02',
  status: 'Delivered',
  payment_method: 'COD'
}
```

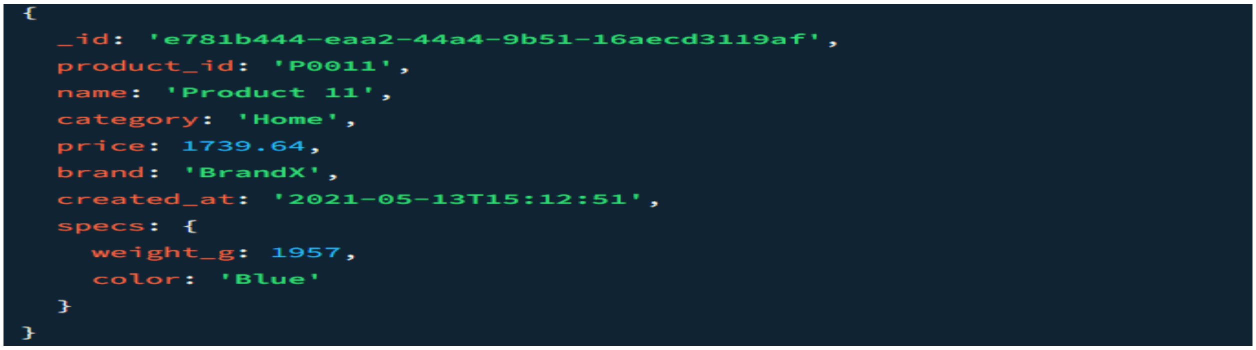
4.



```
db.products.find().sort({ price: -1 }).limit(5)
{
  _id: 'bf5b15c7-fd22-488f-a1f9-e9e9ebd96a8f',
  product_id: 'P0030',
  name: 'Product 30',
  category: 'Electronics',
  price: 1957.07,
  brand: 'BrandY',
  created_at: '2025-07-07T08:45:15',
  specs: {
    weight_g: 615,
    color: 'White'
  }
}
{
  _id: 'e74ceb4f-09d4-4703-b8a0-212f9ab3417a',
  product_id: 'P0027',
  name: 'Product 27',
  category: 'Books',
  price: 1909.32,
  brand: 'BrandY',
  created_at: '2019-01-02T18:38:37',
  specs: {
    weight_g: 2553,
    color: 'Blue'
  }
}
```

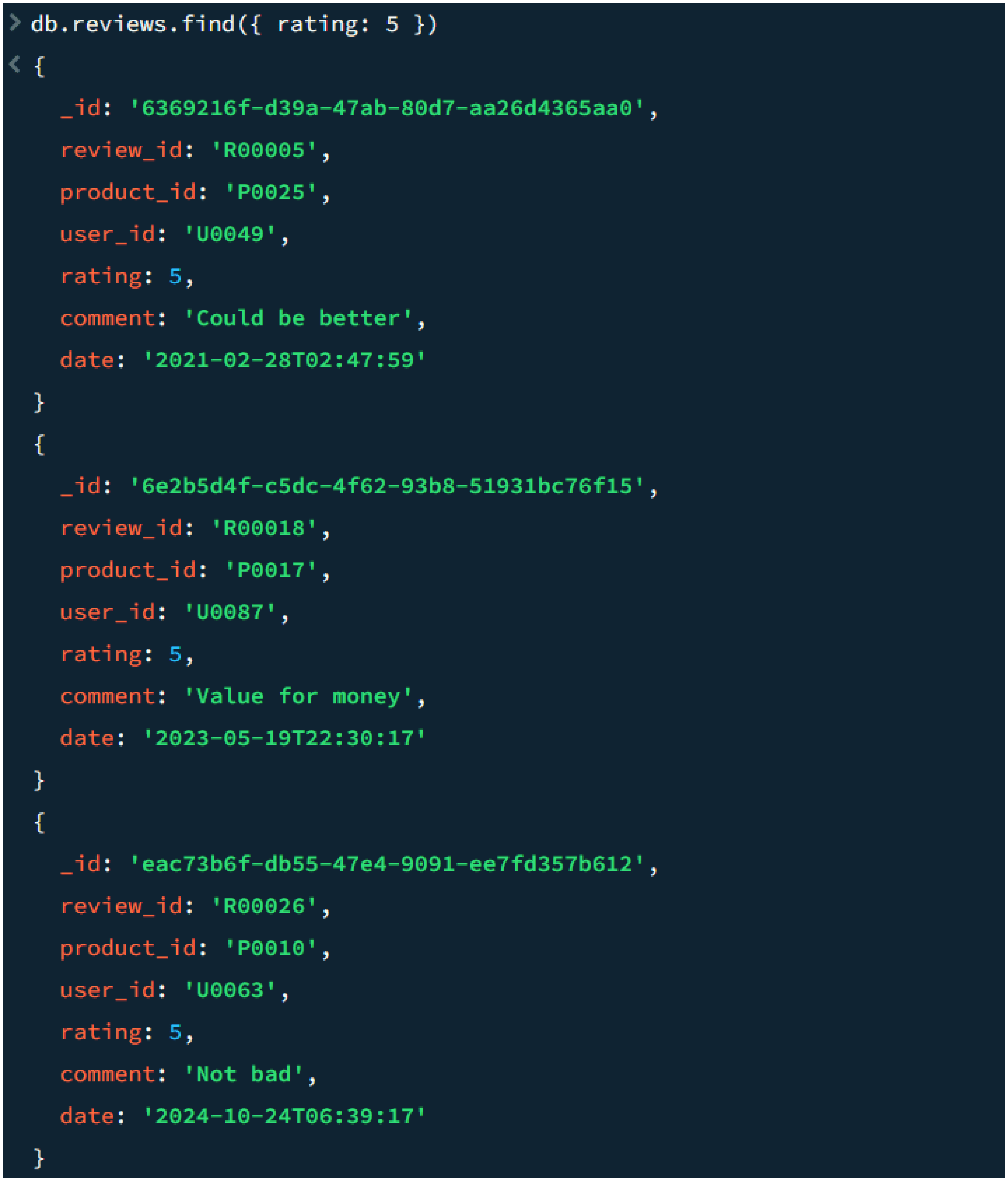


```
{
  _id: '15976aba-786c-4dfb-8f62-109ca86b459f',
  product_id: 'P0038',
  name: 'Product 38',
  category: 'Beauty',
  price: 1874.25,
  brand: 'BrandY',
  created_at: '2023-11-26T05:27:16',
  specs: {
    weight_g: 2036,
    color: 'White'
  }
}
{
  _id: '33a36cc5-06b8-4517-a1c9-9a953206d11b',
  product_id: 'P0023',
  name: 'Product 23',
  category: 'Beauty',
  price: 1775.34,
  brand: 'BrandC',
  created_at: '2025-07-15T22:52:30',
  specs: {
    weight_g: 2725,
    color: 'Blue'
  }
}
```

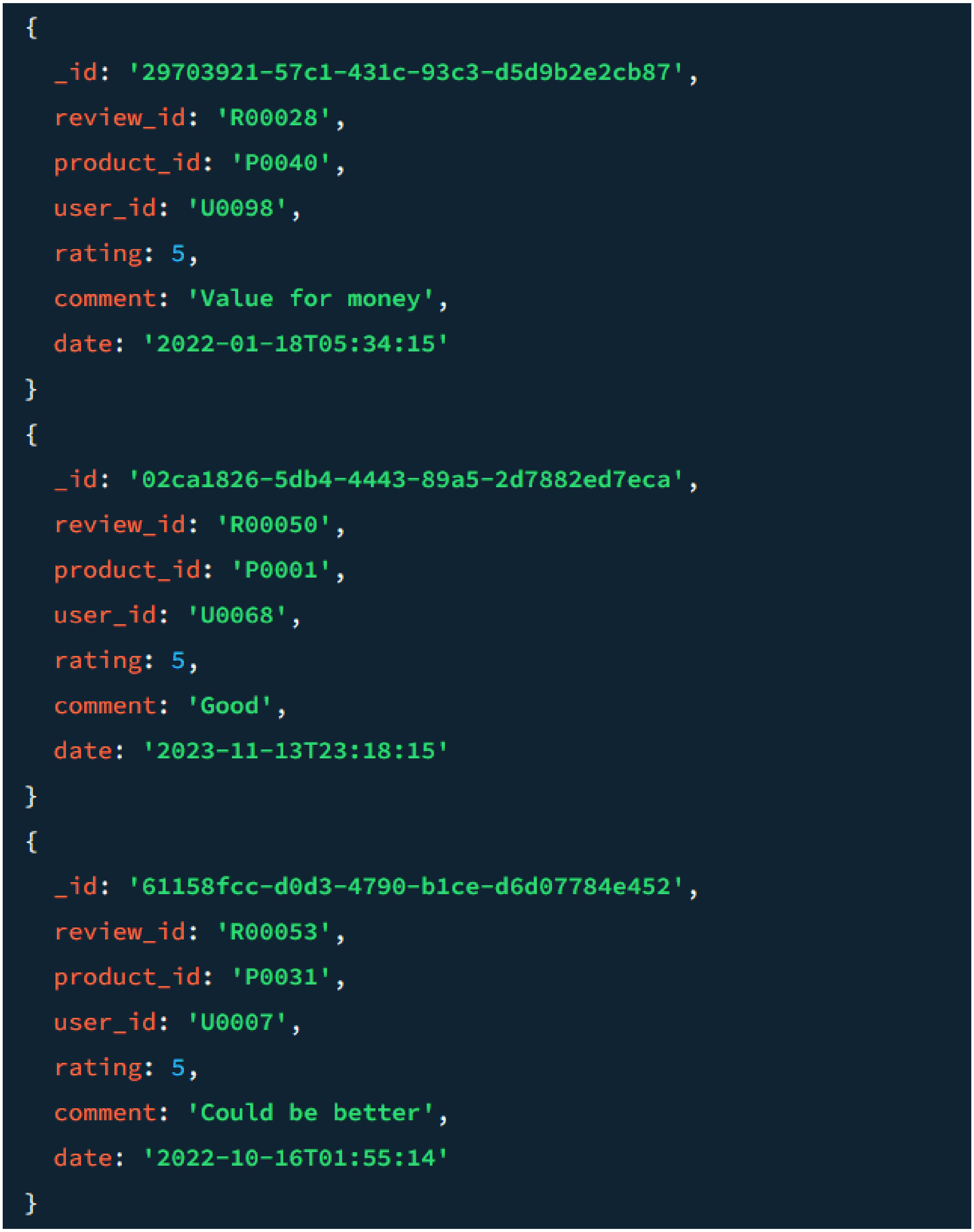


```
{
  _id: 'e781b444-eaa2-44a4-9b51-16aecd3119af',
  product_id: 'P0011',
  name: 'Product 11',
  category: 'Home',
  price: 1739.64,
  brand: 'BrandX',
  created_at: '2021-05-13T15:12:51',
  specs: {
    weight_g: 1957,
    color: 'Blue'
  }
}
```

5.



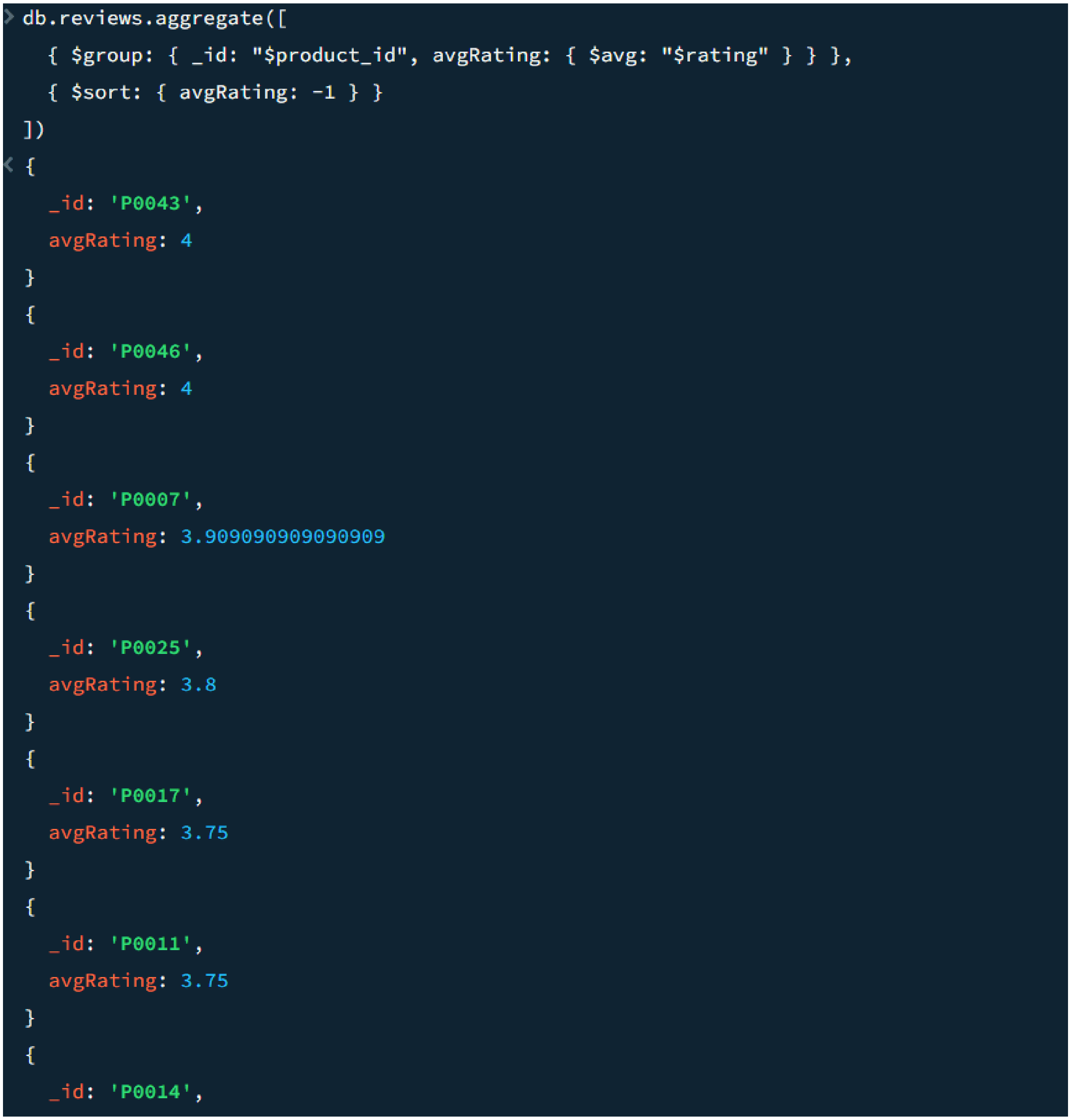
```
> db.reviews.find({ rating: 5 })
< {
    _id: '6369216f-d39a-47ab-80d7-aa26d4365aa0',
    review_id: 'R00005',
    product_id: 'P0025',
    user_id: 'U0049',
    rating: 5,
    comment: 'Could be better',
    date: '2021-02-28T02:47:59'
  }
  {
    _id: '6e2b5d4f-c5dc-4f62-93b8-51931bc76f15',
    review_id: 'R00018',
    product_id: 'P0017',
    user_id: 'U0087',
    rating: 5,
    comment: 'Value for money',
    date: '2023-05-19T22:30:17'
  }
  {
    _id: 'eac73b6f-db55-47e4-9091-ee7fd357b612',
    review_id: 'R00026',
    product_id: 'P0010',
    user_id: 'U0063',
    rating: 5,
    comment: 'Not bad',
    date: '2024-10-24T06:39:17'
  }
```



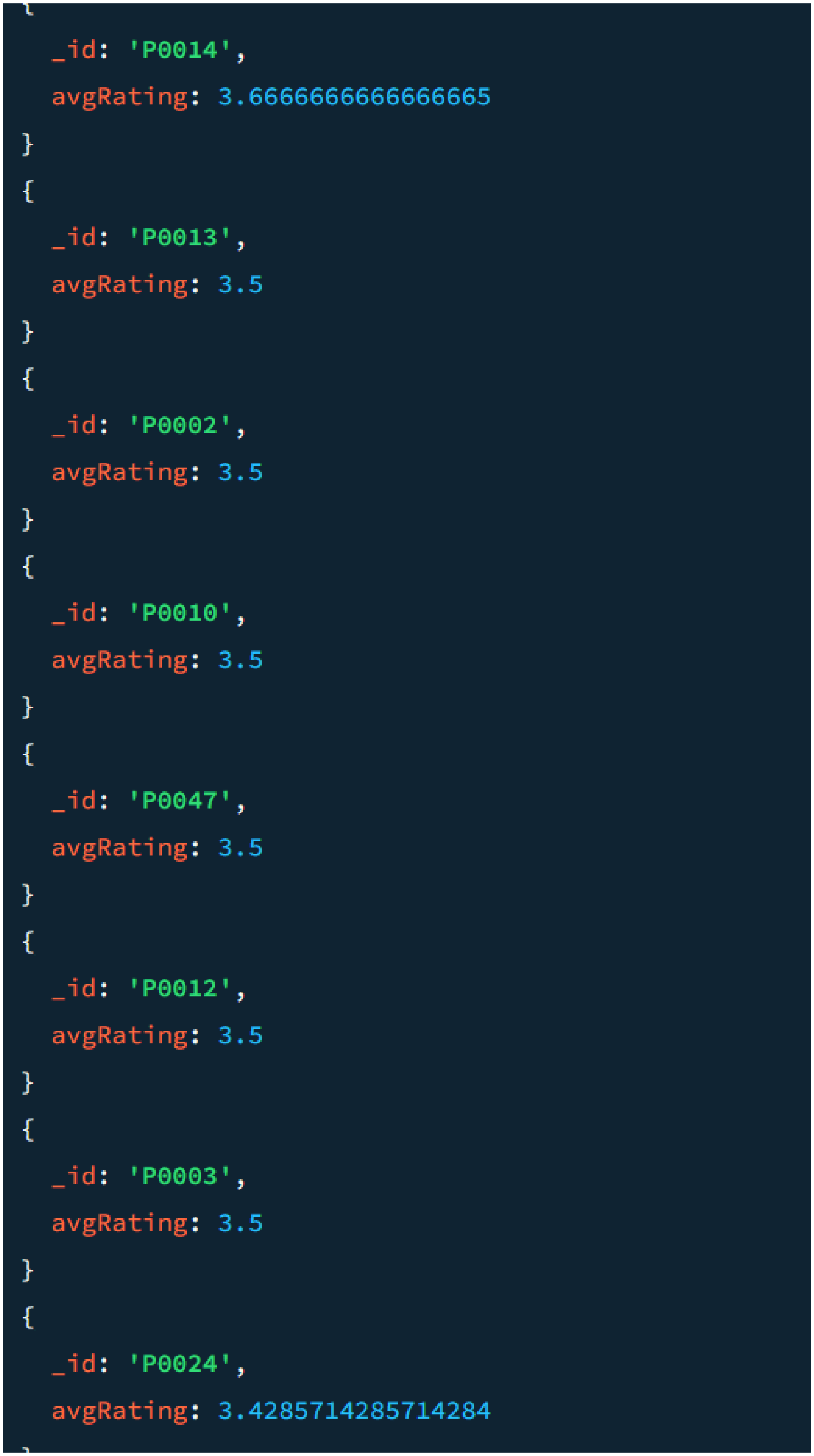
```
{
  _id: '29703921-57c1-431c-93c3-d5d9b2e2cb87',
  review_id: 'R00028',
  product_id: 'P0040',
  user_id: 'U0098',
  rating: 5,
  comment: 'Value for money',
  date: '2022-01-18T05:34:15'
}
{
  _id: '02ca1826-5db4-4443-89a5-2d7882ed7eca',
  review_id: 'R00050',
  product_id: 'P0001',
  user_id: 'U0068',
  rating: 5,
  comment: 'Good',
  date: '2023-11-13T23:18:15'
}
{
  _id: '61158fcc-d0d3-4790-b1ce-d6d07784e452',
  review_id: 'R00053',
  product_id: 'P0031',
  user_id: 'U0007',
  rating: 5,
  comment: 'Could be better',
  date: '2022-10-16T01:55:14'
}
```

Complex queries:

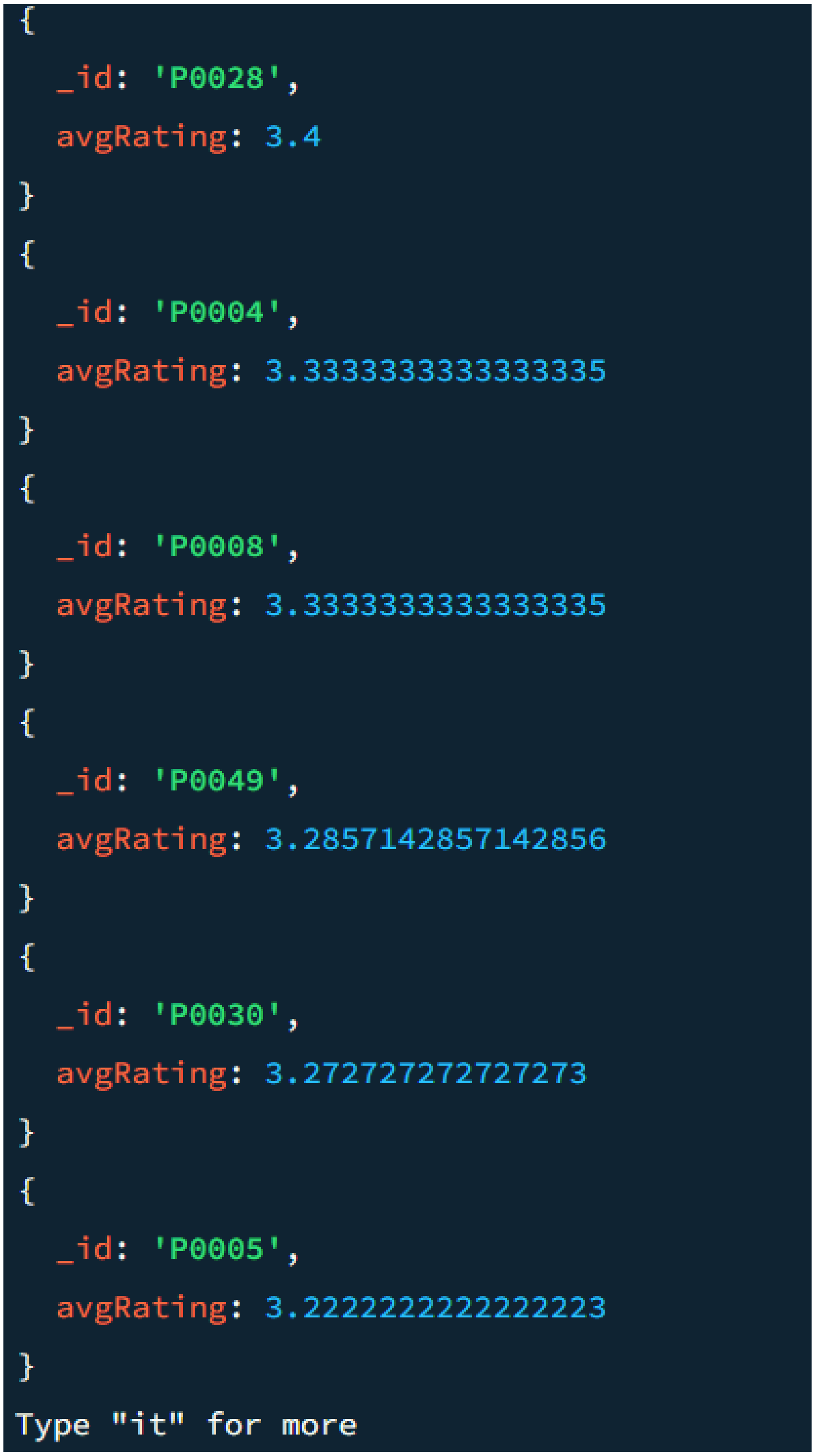
1.



```
> db.reviews.aggregate([
    { $group: { _id: "$product_id", avgRating: { $avg: "$rating" } } },
    { $sort: { avgRating: -1 } }
  ])
< {
    _id: 'P0043',
    avgRating: 4
  }
  {
    _id: 'P0046',
    avgRating: 4
  }
  {
    _id: 'P0007',
    avgRating: 3.909090909090909
  }
  {
    _id: 'P0025',
    avgRating: 3.8
  }
  {
    _id: 'P0017',
    avgRating: 3.75
  }
  {
    _id: 'P0011',
    avgRating: 3.75
  }
  {
    _id: 'P0014',
```

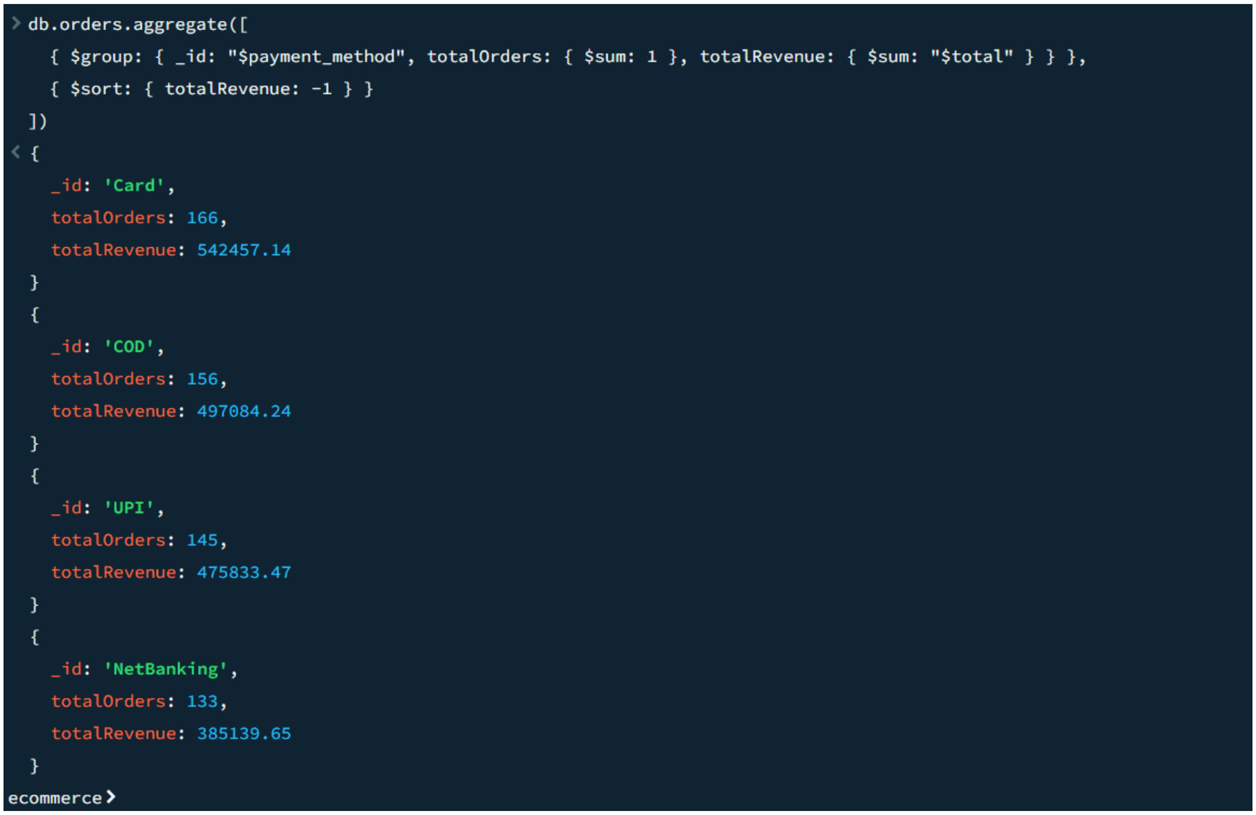


```
{
  _id: 'P0014',
  avgRating: 3.6666666666666665
}
{
  _id: 'P0013',
  avgRating: 3.5
}
{
  _id: 'P0002',
  avgRating: 3.5
}
{
  _id: 'P0010',
  avgRating: 3.5
}
{
  _id: 'P0047',
  avgRating: 3.5
}
{
  _id: 'P0012',
  avgRating: 3.5
}
{
  _id: 'P0003',
  avgRating: 3.5
}
{
  _id: 'P0024',
  avgRating: 3.4285714285714284
}
```



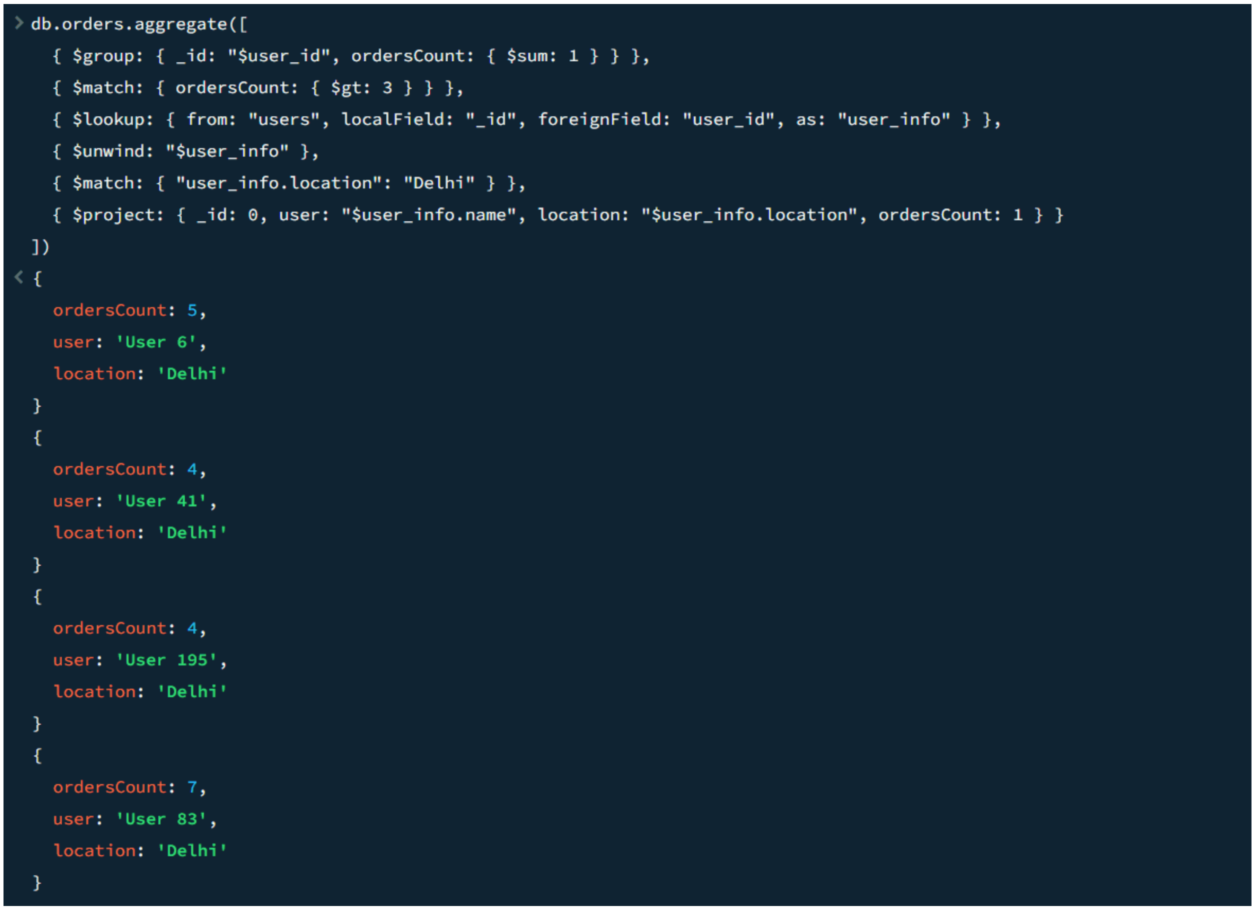
```
{
  _id: 'P0028',
  avgRating: 3.4
}
{
  _id: 'P0004',
  avgRating: 3.3333333333333335
}
{
  _id: 'P0008',
  avgRating: 3.3333333333333335
}
{
  _id: 'P0049',
  avgRating: 3.2857142857142856
}
{
  _id: 'P0030',
  avgRating: 3.272727272727273
}
{
  _id: 'P0005',
  avgRating: 3.2222222222222223
}
Type "it" for more
```

2.

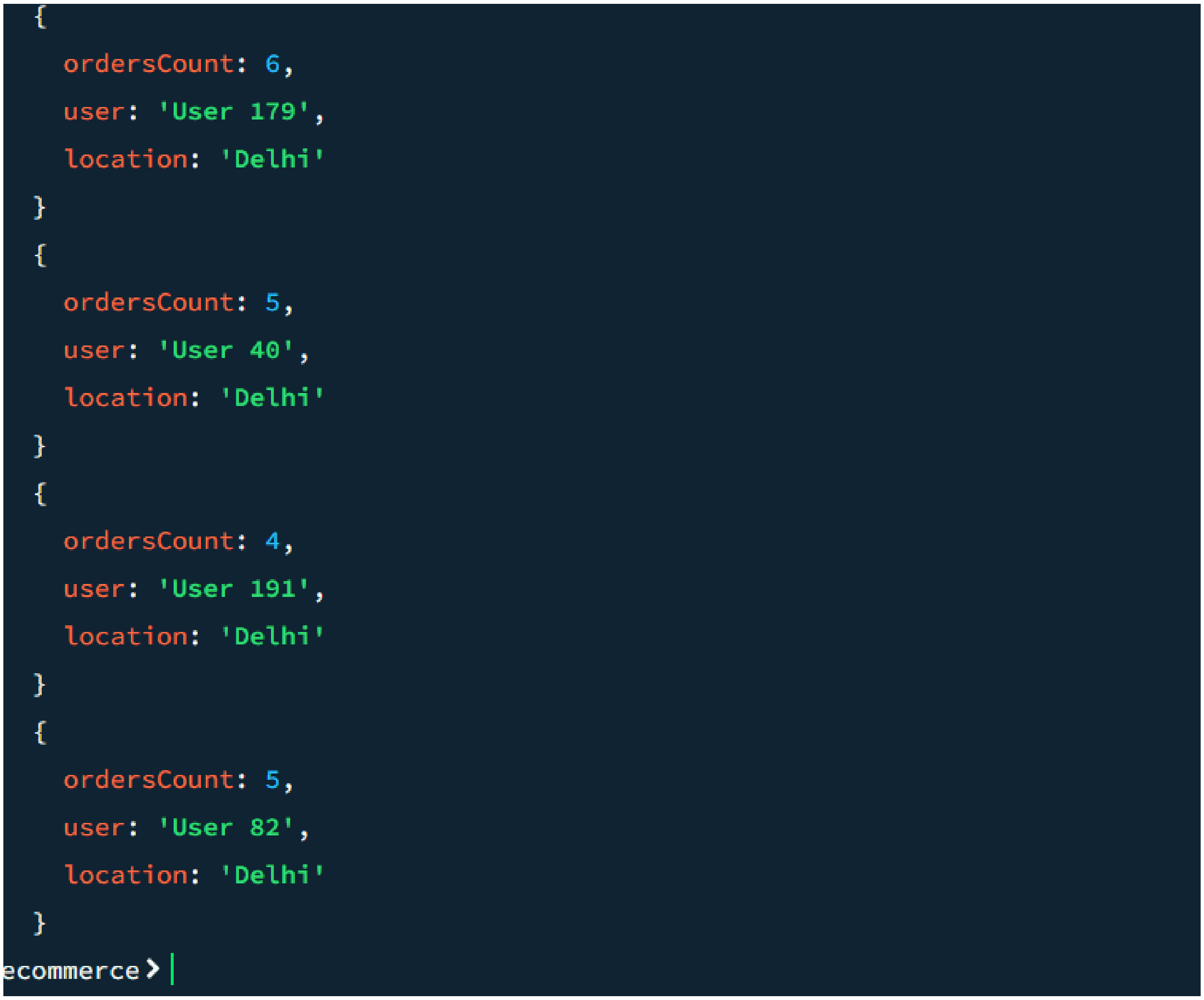


```
> db.orders.aggregate([
    { $group: { _id: "$payment_method", totalOrders: { $sum: 1 }, totalRevenue: { $sum: "$total" } } },
    { $sort: { totalRevenue: -1 } }
  ])
< {
    _id: 'Card',
    totalOrders: 166,
    totalRevenue: 542457.14
  }
  {
    _id: 'COD',
    totalOrders: 156,
    totalRevenue: 497084.24
  }
  {
    _id: 'UPI',
    totalOrders: 145,
    totalRevenue: 475833.47
  }
  {
    _id: 'NetBanking',
    totalOrders: 133,
    totalRevenue: 385139.65
  }
ecommerce>
```

3.



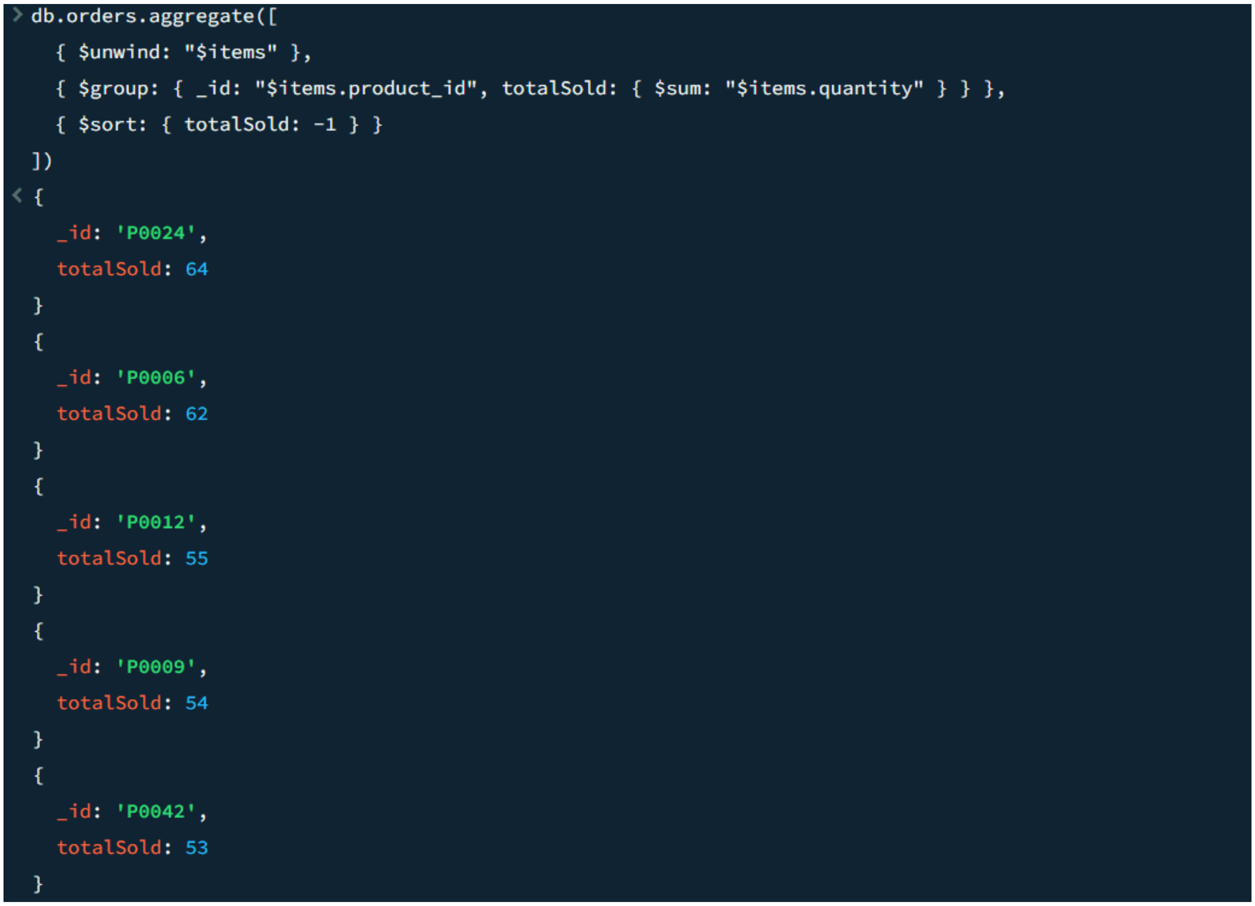
```
> db.orders.aggregate([
    { $group: { _id: "$user_id", ordersCount: { $sum: 1 } } },
    { $match: { ordersCount: { $gt: 3 } } },
    { $lookup: { from: "users", localField: "_id", foreignField: "user_id", as: "user_info" } },
    { $unwind: "$user_info" },
    { $match: { "user_info.location": "Delhi" } },
    { $project: { _id: 0, user: "$user_info.name", location: "$user_info.location", ordersCount: 1 } }
  ])
< {
    ordersCount: 5,
    user: 'User 6',
    location: 'Delhi'
  }
  {
    ordersCount: 4,
    user: 'User 41',
    location: 'Delhi'
  }
  {
    ordersCount: 4,
    user: 'User 195',
    location: 'Delhi'
  }
  {
    ordersCount: 7,
    user: 'User 83',
    location: 'Delhi'
  }
```



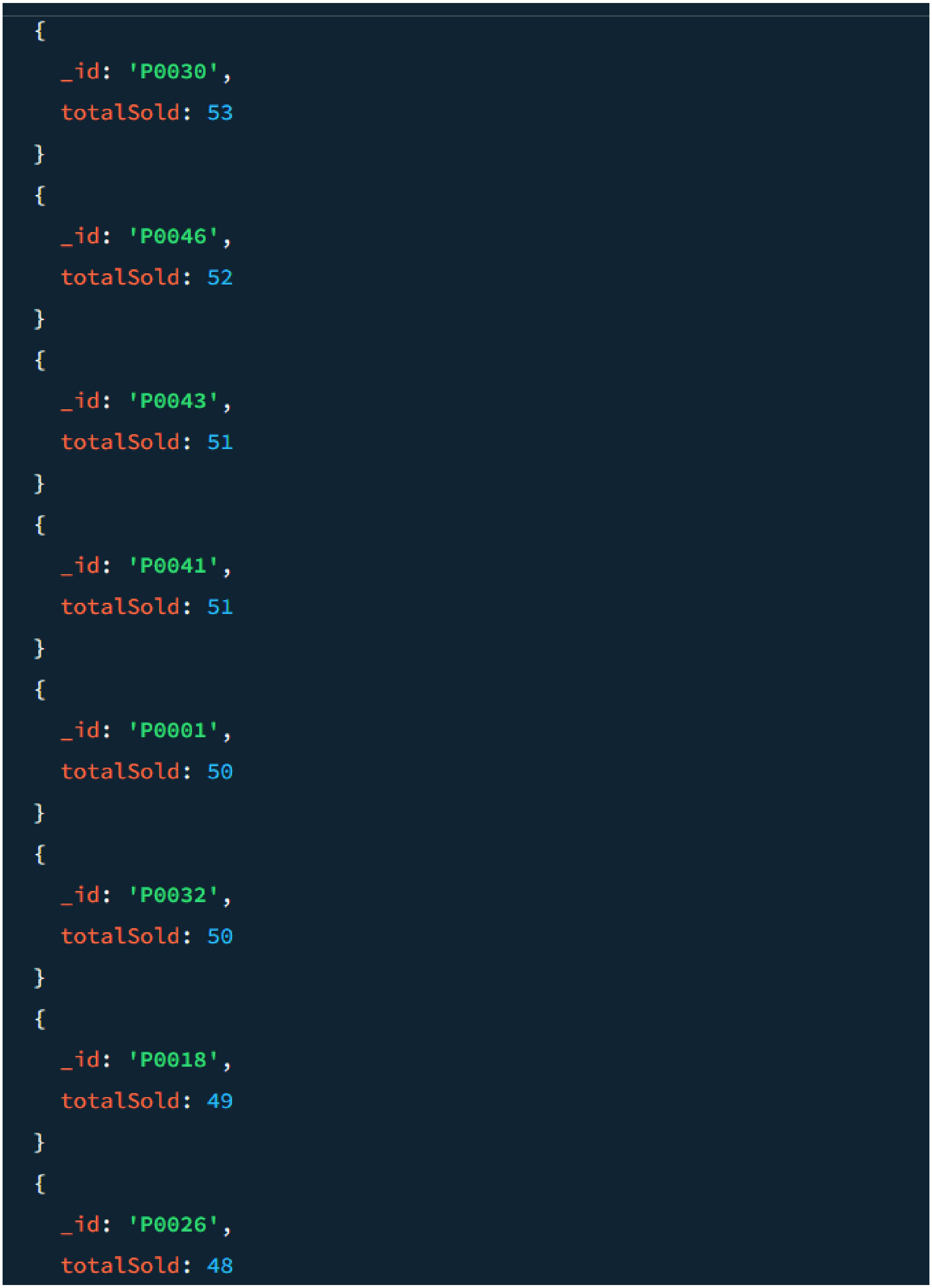
```
  {
    ordersCount: 6,
    user: 'User 179',
    location: 'Delhi'
  }
  {
    ordersCount: 5,
    user: 'User 40',
    location: 'Delhi'
  }
  {
    ordersCount: 4,
    user: 'User 191',
    location: 'Delhi'
  }
  {
    ordersCount: 5,
    user: 'User 82',
    location: 'Delhi'
  }
ecommerce>
```

4.

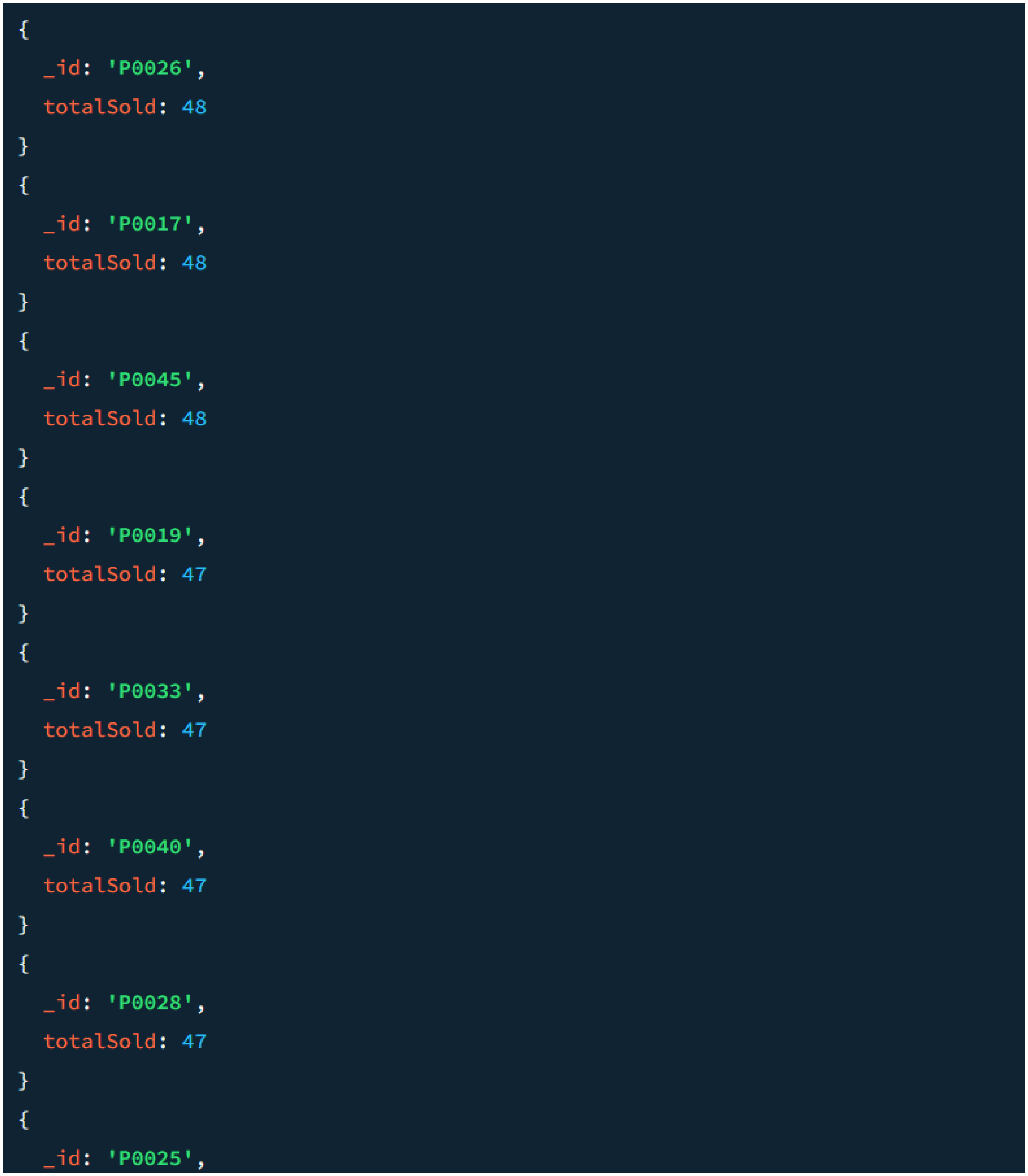
5.



```
> db.orders.aggregate([
    { $unwind: "$items" },
    { $group: { _id: "$items.product_id", totalSold: { $sum: "$items.quantity" } } },
    { $sort: { totalSold: -1 } }
  ])
< {
    _id: 'P0024',
    totalSold: 64
  }
  {
    _id: 'P0006',
    totalSold: 62
  }
  {
    _id: 'P0012',
    totalSold: 55
  }
  {
    _id: 'P0009',
    totalSold: 54
  }
  {
    _id: 'P0042',
    totalSold: 53
  }
```



```
{
  _id: 'P0030',
  totalSold: 53
}
{
  _id: 'P0046',
  totalSold: 52
}
{
  _id: 'P0043',
  totalSold: 51
}
{
  _id: 'P0041',
  totalSold: 51
}
{
  _id: 'P0001',
  totalSold: 50
}
{
  _id: 'P0032',
  totalSold: 50
}
{
  _id: 'P0018',
  totalSold: 49
}
{
  _id: 'P0026',
  totalSold: 48
```



```
{
  _id: 'P0026',
  totalSold: 48
}
{
  _id: 'P0017',
  totalSold: 48
}
{
  _id: 'P0045',
  totalSold: 48
}
{
  _id: 'P0019',
  totalSold: 47
}
{
  _id: 'P0033',
  totalSold: 47
}
{
  _id: 'P0040',
  totalSold: 47
}
{
  _id: 'P0028',
  totalSold: 47
}
{
  _id: 'P0025',
```

FUTURE PROSPECTS:

The e-commerce system you have designed is robust and scalable, but there are numerous opportunities to expand its functionality and improve the user experience. Some promising future prospects include:

1. Integration of Artificial Intelligence (AI)

- **Personalized Recommendations:** Use machine learning algorithms to suggest products based on users' browsing history, purchase patterns, and reviews.
- **Chatbots for Customer Support:** AI-powered chatbots can provide instant responses, track orders, and assist in product selection.

Demand Forecasting: Predict future sales and optimize inventory using AI-driven analytics.

2. Advanced Analytics and Reporting

- **Customer Insights:** Analyze purchase behavior, preferences, and feedback to improve marketing strategies.
- **Sales Trend Analysis:** Track product performance over time to identify seasonal trends and top-selling products.
- **Inventory Optimization:** Monitor stock levels in real time to prevent shortages or overstocking.

3. Mobile and Cross-Platform Expansion

- Developing **mobile apps** for iOS and Android can improve accessibility and increase user engagement.
- Progressive Web Apps (PWAs) can provide a native-app-like experience without needing downloads.

4. Enhanced Security and Payment Options

- Implement **multi-factor authentication** and advanced encryption to protect user data.
- Integrate multiple **digital payment gateways** and options like UPI, wallets, and buy-now-pay-later schemes.

5. Multi-Language and Multi-Currency Support

- Expanding the platform for international markets with **multi-language interfaces** and **currency conversion**.
- Helps target global customers and boosts cross-border sales.

6. Integration with Supply Chain and Logistics

- Real-time tracking of shipments and inventory across multiple warehouses.
- Automated order routing to the nearest warehouse for faster delivery.

7. Incorporating Blockchain Technology

- **Transparent Supply Chains:** Track product origins and authenticity for quality assurance.

- **Secure Transactions:** Smart contracts can ensure secure and automated payment processing.

8. Social Commerce and User Engagement

- Integration with social media platforms to allow direct purchases from posts or ads.
- Gamification, loyalty programs, and interactive reviews to enhance user retention.

CONCLUSION:

A well-designed database is the backbone of any e-commerce system. By leveraging **MongoDB and a NoSQL approach**, the system achieves **flexibility, scalability, and high performance**, handling diverse product types, large user bases, and complex order data efficiently. The use of separate but interconnected collections—**users, products, orders, reviews, and inventory**—ensures modularity, easy maintenance, and fast query execution.

The database design also supports **advanced analytics, personalized recommendations, and inventory management**, making it capable of adapting to future growth. Properly structured relationships between collections allow seamless aggregation and reporting, which are essential for operational efficiency, customer satisfaction, and informed decision-making. Overall, an effective e-commerce database management system not only powers daily transactions but also forms the foundation for long-term business scalability and innovation.

REFERENCES:

- MongoDB, Inc. *MongoDB Manual*. https://docs.mongodb.com/manual/
- Heller, K., & Heller, J. (2020). *MongoDB Basics: A Practical Guide to Building NoSQL Databases*. Packt Publishing.
- Elmasri, R., & Navathe, S. (2016). *Fundamentals of Database Systems* (7th Edition). Pearson.
- Laudon, K. C., & Traver, C. G. (2020). *E-Commerce 2020: Business, Technology, Society* (16th Edition). Pearson.