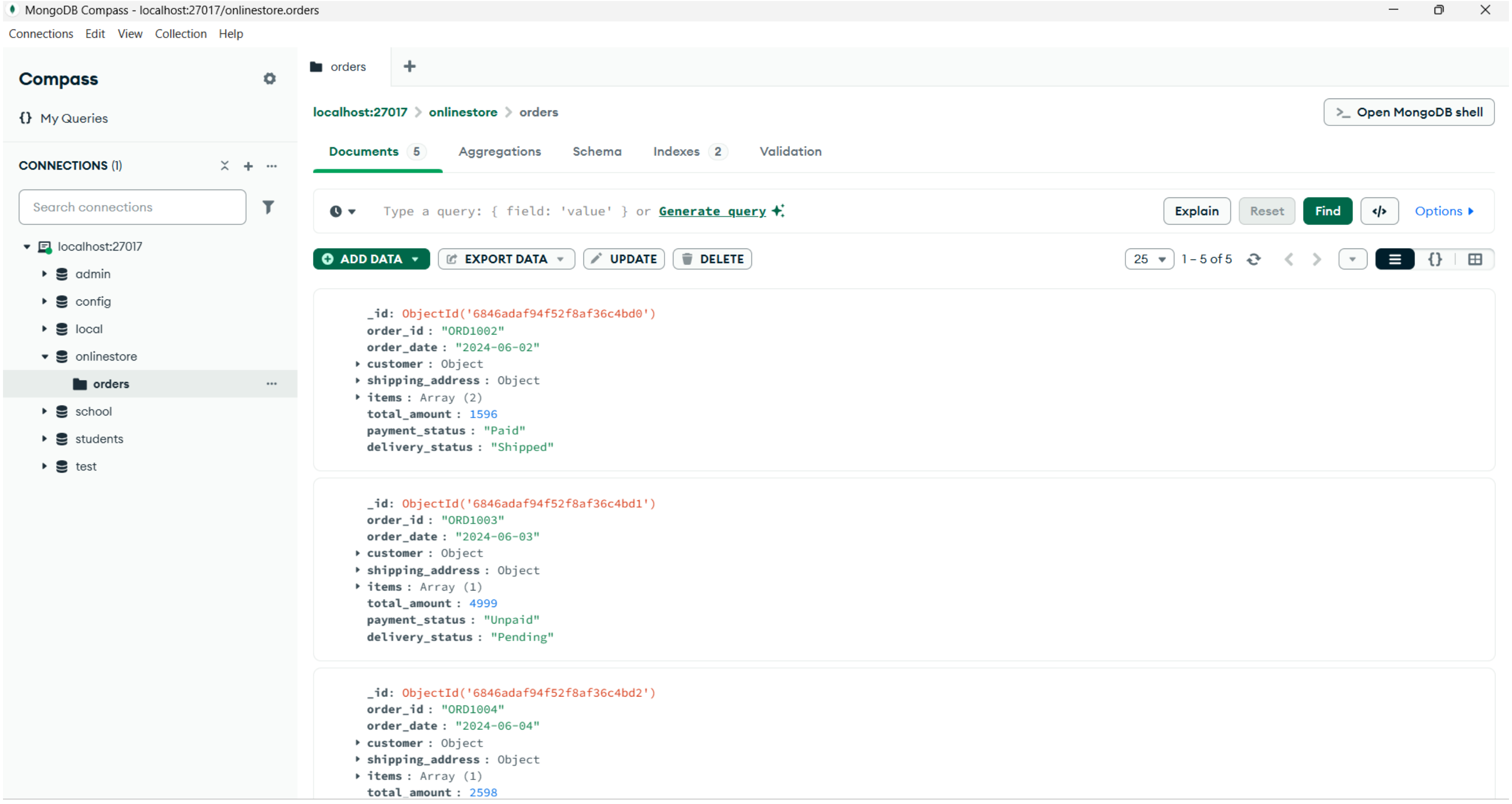
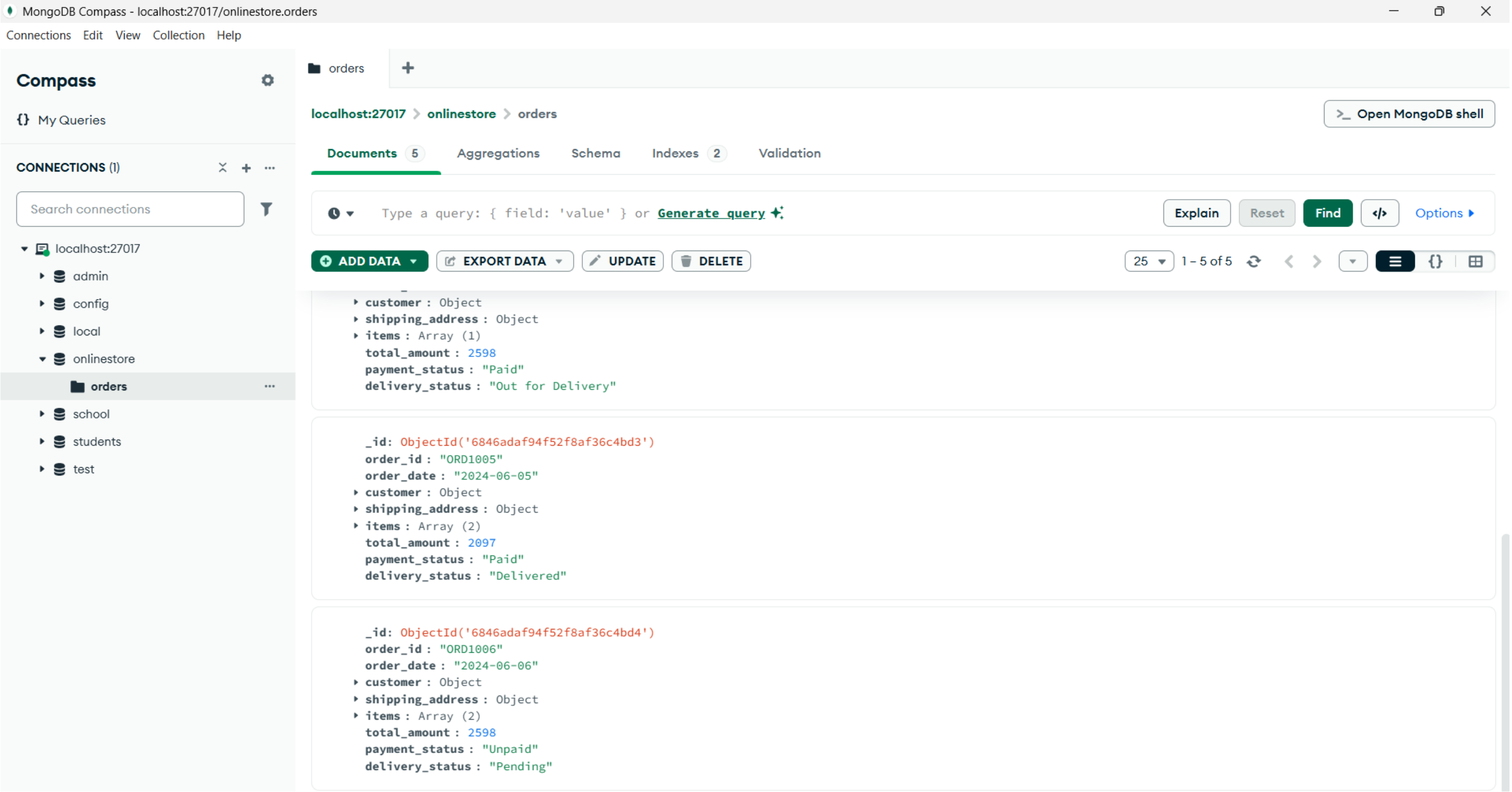
Project 2 – Online Store Management System Using MongoDB



```
_id: ObjectId('6846adaf94f52f8af36c4bd0')
order_id : "ORD1002"
order_date : "2024-06-02"
customer : Object
shipping_address : Object
items : Array (2)
total_amount : 1596
payment_status : "Paid"
delivery_status : "Shipped"

_id: ObjectId('6846adaf94f52f8af36c4bd1')
order_id : "ORD1003"
order_date : "2024-06-03"
customer : Object
shipping_address : Object
items : Array (1)
total_amount : 4999
payment_status : "Unpaid"
delivery_status : "Pending"

_id: ObjectId('6846adaf94f52f8af36c4bd2')
order_id : "ORD1004"
order_date : "2024-06-04"
customer : Object
shipping_address : Object
items : Array (1)
total_amount : 2598
```



```
customer : Object
shipping_address : Object
items : Array (1)
total_amount : 2598
payment_status : "Paid"
delivery_status : "Out for Delivery"

_id: ObjectId('6846adaf94f52f8af36c4bd3')
order_id : "ORD1005"
order_date : "2024-06-05"
customer : Object
shipping_address : Object
items : Array (2)
total_amount : 2097
payment_status : "Paid"
delivery_status : "Delivered"

_id: ObjectId('6846adaf94f52f8af36c4bd4')
order_id : "ORD1006"
order_date : "2024-06-06"
customer : Object
shipping_address : Object
items : Array (2)
total_amount : 2598
payment_status : "Unpaid"
delivery_status : "Pending"
```

```
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sakshi Rathi>mongosh
test> use onlinestore
switched to db onlinestore
onlinestore> db.createCollection("orders")
{ ok: 1 }
```

Q1. Insert 5 different orders

```
onlinestore> db.orders.insertMany([
...  {
...    "order_id": "ORD1002",
...    "order_date": "2024-06-02",
...    "customer": {
...      "name": "Raj Mehta",
...      "email": "raj@example.com",
...      "phone": "9876500001"
...    },
...    "shipping_address": {
...      "street": "456 Hill View",
...      "city": "Delhi",
...      "state": "DL",
...      "pincode": "110001"
...    },
...   "items": [
...     { "product": "Laptop Bag", "price": 1299, "quantity": 1 },
...     { "product": "Notebook", "price": 99, "quantity": 3 }
...   ],
...   "total_amount": 1596,
...   "payment_status": "Paid",
...   "delivery_status": "Shipped"
... },
... {
...   "order_id": "ORD1003",
...   "order_date": "2024-06-03",
...   "customer": {
...     "name": "Anjali Verma",
...     "email": "anjali@example.com",
...     "phone": "9876500002"
...   },
...   "shipping_address": {
...     "street": "789 Rose Villa",
...     "city": "Bangalore",
...     "state": "KA",
...     "pincode": "560001"
...   },
...   "items": [
...     { "product": "Monitor", "price": 4999, "quantity": 1 }
...   ],
...   "total_amount": 4999,
...   "payment_status": "Unpaid",
...   "delivery_status": "Pending"
... },
... {
...   "order_id": "ORD1004",
...   "order_date": "2024-06-04",
...   "customer": {
...     "name": "Farhan Ali",
...     "email": "farhan@example.com",
...     "phone": "9876500003"
...   },
...   "shipping_address": {
...     "street": "22 Sky Tower",
...     "city": "Hyderabad",
...     "state": "TS",
...     "pincode": "500001"
...   },
...   "items": [
...     { "product": "Power Bank", "price": 1299, "quantity": 2 }
...   ],
...   "total_amount": 2598,
...   "payment_status": "Paid",
...   "delivery_status": "Out for Delivery"
... },
...  {
...    "order_id": "ORD1005",
...    "order_date": "2024-06-05",
...    "customer": {
...      "name": "Nina Das",
...      "email": "nina@example.com",
...      "phone": "9876500004"
...    },
...    "shipping_address": {
...      "street": "500 River Side",
...      "city": "Chennai",
...      "state": "TN",
...      "pincode": "600001"
...    },
...    "items": [
...      { "product": "USB Hub", "price": 799, "quantity": 2 },
...      { "product": "HDMI Cable", "price": 499, "quantity": 1 }
...    ],
...    "total_amount": 2097,
...    "payment_status": "Paid",
...    "delivery_status": "Delivered"
...  },
...  {
...    "order_id": "ORD1006",
...    "order_date": "2024-06-06",
...     "customer": {
...       "name": "Arjun Sen",
...       "email": "arjun@example.com",
...       "phone": "9876500005"
...     },
...     "shipping_address": {
...       "street": "88 Market Road",
...       "city": "Kolkata",
...       "state": "WB",
...       "pincode": "700001"
...     },
...     "items": [
...       { "product": "Webcam", "price": 1599, "quantity": 1 },
...       { "product": "Tripod", "price": 999, "quantity": 1 }
...     ],
...     "total_amount": 2598,
...     "payment_status": "Unpaid",
...     "delivery_status": "Pending"
...   }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6846adaf94f52f8af36c4bd0'),
    '1': ObjectId('6846adaf94f52f8af36c4bd1'),
    '2': ObjectId('6846adaf94f52f8af36c4bd2'),
    '3': ObjectId('6846adaf94f52f8af36c4bd3'),
    '4': ObjectId('6846adaf94f52f8af36c4bd4')
  }
}
```

Q2. Find all orders where payment_status is "Paid" but delivery_status is not "Delivered"

```
onlinestore> db.orders.find({
...   payment_status: "Paid",
...   delivery_status: { $ne: "Delivered" }
... })
[
  {
    _id: ObjectId('6846adaf94f52f8af36c4bd0'),
    order_id: 'ORD1002',
    order_date: '2024-06-02',
    customer: {
      name: 'Raj Mehta',
      email: 'raj@example.com',
      phone: '9876500001'
    },
    shipping_address: {
      street: '456 Hill View',
      city: 'Delhi',
      state: 'DL',
      pincode: '110001'
    },
    items: [
      { product: 'Laptop Bag', price: 1299, quantity: 1 },
      { product: 'Notebook', price: 99, quantity: 3 }
    ],
    total_amount: 1596,
    payment_status: 'Paid',
    delivery_status: 'Shipped'
  },
  {
    _id: ObjectId('6846adaf94f52f8af36c4bd2'),
    order_id: 'ORD1004',
    order_date: '2024-06-04',
    customer: {
      name: 'Farhan Ali',
      email: 'farhan@example.com',
      phone: '9876500003'
    },
    shipping_address: {
      street: '22 Sky Tower',
      city: 'Hyderabad',
      state: 'TS',
     pincode: '500001'
    },
    items: [ { product: 'Power Bank', price: 1299, quantity: 2 } ],
    total_amount: 2598,
    payment_status: 'Paid',
    delivery_status: 'Out for Delivery'
   } ]
```

Q3. List all orders that contain at least one product priced more than 1000

```
onlinestore> db.orders.find({
...   "items.price": { $gt: 1000 }
... })
...
[
 {
  _id: ObjectId('6846adaf94f52f8af36c4bd0'),
  order_id: 'ORD1002',
  order_date: '2024-06-02',
  customer: {
   name: 'Raj Mehta',
   email: 'raj@example.com',
   phone: '9876500001'
  },
  shipping_address: {
      street: '456 Hill View',
      city: 'Delhi',
      state: 'DL',
      pincode: '110001'
    },
    items: [
      { product: 'Laptop Bag', price: 1299, quantity: 1 },
      { product: 'Notebook', price: 99, quantity: 3 }
    ],
    total_amount: 1596,
    payment_status: 'Paid',
    delivery_status: 'Shipped'
  },
  {
    _id: ObjectId('6846adaf94f52f8af36c4bd1'),
    order_id: 'ORD1003',
    order_date: '2024-06-03',
    customer: {
      name: 'Anjali Verma',
      email: 'anjali@example.com',
      phone: '9876500002'
    },
    shipping_address: {
      street: '789 Rose Villa',
      city: 'Bangalore',
    state: 'KA',
    pincode: '560001'
  },
  items: [ { product: 'Monitor', price: 4999, quantity: 1 } ],
  total_amount: 4999,
  payment_status: 'Unpaid',
  delivery_status: 'Pending'
},
{
  _id: ObjectId('6846adaf94f52f8af36c4bd2'),
  order_id: 'ORD1004',
  order_date: '2024-06-04',
  customer: {
    name: 'Farhan Ali',
    email: 'farhan@example.com',
    phone: '9876500003'
  },
  shipping_address: {
    street: '22 Sky Tower',
    city: 'Hyderabad',
    state: 'TS',
    pincode: '500001'
  },
  items: [ { product: 'Power Bank', price: 1299, quantity: 2 } ],
  total_amount: 2598,
  payment_status: 'Paid',
  delivery_status: 'Out for Delivery'
},
{
  _id: ObjectId('6846adaf94f52f8af36c4bd4'),
  order_id: 'ORD1006',
  order_date: '2024-06-06',
  customer: {
    name: 'Arjun Sen',
    email: 'arjun@example.com',
    phone: '9876500005'
  },
  shipping_address: {
    street: '88 Market Road',
    city: 'Kolkata',
    state: 'WB',
    pincode: '700001'
  },
  items: [
    { product: 'Webcam', price: 1599, quantity: 1 },
    { product: 'Tripod', price: 999, quantity: 1 }
  ],
  total_amount: 2598,
  payment_status: 'Unpaid',
  delivery_status: 'Pending'   }]
```

Q4. Get a list of orders shipped to “Mumbai” or “Delhi” sorted by most recent date

```
onlinestore> db.orders.find(
...   { "shipping_address.city": { $in: ["Mumbai", "Delhi"] } },
...   { order_id: 1, "shipping_address.city": 1, order_date: 1, _id: 0 }
... ).sort({ order_date: -1 })
...
[
  {
    order_id: 'ORD1002',
    order_date: '2024-06-02',
    shipping_address: { city: 'Delhi' }
  }
]
```

Q5. Group orders by payment_status and count how many orders fall under each category

```
onlinestore> db.orders.aggregate([
...   { $group: { _id: "$payment_status", total_orders: { $sum: 1 } } }
... ])
...
[
  { _id: 'Paid', total_orders: 3 },
  { _id: 'Unpaid', total_orders: 2 }
]
```

Q6. Calculate the total revenue (sum of total_amount) from all "Delivered" orders

```
onlinestore> db.orders.aggregate([

...  { $match: { delivery_status: "Delivered" } },

...  { $group: { _id: null, total_revenue: { $sum: "$total_amount" } } }

... ])

...

[ { _id: null, total_revenue: 2097 } ]
```

Q7. Add a new field "priority": "High" to all orders that have more than 3 items

```
onlinestore> db.orders.updateMany(

...  { $expr: { $gt: [ { $size: "$items" }, 3 ] } },

...  { $set: { priority: "High" } }

... )

...

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 0,

  modifiedCount: 0,

  upsertedCount: 0

}
```

Q8. Count how many times each product was ordered (regardless of order)

```
onlinestore> db.orders.aggregate([
...   { $unwind: "$items" },
...   { $group: {
...       _id: "$items.product",
...       total_quantity: { $sum: "$items.quantity" }
...   }}
... ])
...
[
  { _id: 'Webcam', total_quantity: 1 },
  { _id: 'USB Hub', total_quantity: 2 },
  { _id: 'Laptop Bag', total_quantity: 1 },
  { _id: 'Tripod', total_quantity: 1 },
  { _id: 'HDMI Cable', total_quantity: 1 },
  { _id: 'Monitor', total_quantity: 1 },
  { _id: 'Notebook', total_quantity: 3 },
  { _id: 'Power Bank', total_quantity: 2 }
]
```

Q9. Add a new item to an existing order (e.g., add "USB Cable" to order_id ORD1001)

```
onlinestore> db.orders.updateOne(
...   { order_id: "ORD1001" },
...   { $push: {
```

```
...    items: { product: "USB Cable", price: 299, quantity: 1 }
...   }}
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

Q10. Rename field "phone" inside customer object to "mobile_number" across all documents

```
onlinestore> db.orders.updateMany(
...   {},
...   { $rename: { "customer.phone": "customer.mobile_number" } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0  }
```

Q11. Create an index on the customer email to speed up lookups.

```
onlinestore> db.orders.createIndex({ "customer.email": 1 })

customer.email_1
```

Q12. Export the data using ` mongoexport ` for reporting purposes.

```
onlinestore> mongoexport --db onlinestore --collection orders --out orders_export.json

... 2025-06-09T15:37:12.345+0530  connected to: mongodb://localhost:27017/

...2025-06-09T15:37:12.678+0530    exported 5 records

onlinestore>
```