

Week – 7 MongoDB Assignment

Piyush Kumar Yadav

To perform:

Store and Query JSON Data in MongoDB

Goal:

Explore NoSQL by storing and querying semi-structured sales data in MongoDB.

Tasks:

Store order data as nested JSON documents (customer, product, region).

Run queries to:

Retrieve all orders in a given date range.

Group sales by region or category using the aggregation pipeline.

Compare flexibility vs. relational SQL model.

Tech: MongoDB, Python (PyMongo)

Deliverables: .ipynb or .py with CRUD + aggregation examples + screenshots (Mongo shell/Compass query outputs)

Link to github: <https://github.com/friedcheesee/blendweek7allstar>

Compare flexibility vs. relational SQL model

MongoDB and relational SQL databases differ fundamentally in how they model, store, and query data.

MongoDB: MongoDB uses a document-oriented data model with no fixed schema, allowing related data - such as customer, product, and region information - to be nested within a single document.

Fields are optional in mongoDB, so csv imports and streaming data can contain missing fields or attributes, and still get inserted successfully. Hence, mongoDB is well suited for semi-structured or evolving datasets.

Relational SQL: Relational databases require data to be normalized across multiple tables (e.g., orders, customers, products, regions). While this enforces strong consistency, it increases schema complexity.

Schema changes - such as adding new attributes or restructuring relationships- can be expensive, and will require major updates/ downtime to change the schema. Queries typically rely on JOIN operations across multiple tables. As data volume grows, these joins can become performance bottlenecks and increase query complexity.

Property	MongoDB	SQL
Data model	Document based (JSON like)	Table based (rows and columns)
Schema	Schema-less	Fixed
Handling missing fields	Allowed without errors	Must be defined
Data relationships	Nested docs	Normalized across tables
Schema evolution	Low cost	Expensive
Analytics	Aggregation pipeline	JOIN- heavy SQL queries
Best suited for	Semi-structured data, evolving data.	Highly structured, transactional data

Inserted documents (Compass):

The screenshot shows the MongoDB Compass interface. The top navigation bar includes 'MongoDB Compass - cluster0.pj6njea.mongodb.net/sales_db.orders', 'Connections', 'Edit', 'View', 'Collection', 'Help', and a maximize/minimize/collapse button. Below the navigation is a toolbar with 'Welcome', 'config', 'local', 'cluster0.pj6njea.mongodb.net', 'orders' (selected), 'oplog.rs', and a '+' button. The left sidebar has sections for 'Compass', 'My Queries', 'Data Modeling', and 'CONNECTIONS'. Under 'CONNECTIONS', there's a tree view for 'cluster0.pj6njea.mongodb.net' with branches for 'admin', 'config', 'local', and 'sales_db'. The 'orders' collection is selected under 'sales_db'. The main area is titled 'Documents 78K' and contains tabs for 'Aggregations', 'Schema', 'Indexes (1)', and 'Validation'. A search bar says 'Type a query: { field: "value" } or Generate query'. Below it are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. A preview pane shows three document snippets:

```
_id: ObjectId('6958c9649f3668d00bd1620a')
order_id: "405-8078784-5731545"
date: 2022-04-30T00:00:00.000+00:00
status: "Cancelled"
fulfillment: "Merchant"
sales_channel: "Amazon.in"
customer: Object
product: Object
payment: Object

_id: ObjectId('6958c9649f3668d00bd1620b')
order_id: "171-9198151-1101146"
date: 2022-04-30T00:00:00.000+00:00
status: "Shipped - Delivered to Buyer"
fulfillment: "Merchant"
sales_channel: "Amazon.in"
customer: Object
product: Object
payment: Object

_id: ObjectId('6958c9649f3668d00bd1620c')
order_id: "404-0687676-7273146"
date: 2022-04-30T00:00:00.000+00:00
status: "Shipped"
fulfillment: "Amazon"
sales_channel: "Amazon.in"
customer: Object
```

Queries:

The screenshot shows a terminal window with a dark theme running the MongoDB shell. The command `use sales_db` is run, followed by `db.orders.find().limit(5)`. The output displays five documents from the 'orders' collection:

```
>_MONGOISH
> use sales_db
< switched to db sales_db
> db.orders.find().limit(5)
< {
  "_id": ObjectId("6958c9649f3668d00bd1620a"),
  "order_id": "405-8078784-5731545",
  "date": 2022-04-30T00:00:00.000Z,
  "status": "Cancelled",
  "fulfillment": "Merchant",
  "sales_channel": "Amazon.in",
  "customer": {
    "b2b": false,
    "region": {
      "city": "Mumbai",
      "state": "Maharashtra",
      "postal_code": "400001",
      "country": "IN"
    }
  },
  "product": {
    "sku": "SET389-KR-NP-S",
    "style": "SET389",
    "category": "Set"
  },
  "payment": {
    "currency": "INR",
    "amount": 647.62,
    "promotion_ids": "No Promotion"
  }
}
```

Orders in given date range:

```
>_MONGOSH
>
> db.orders.find({
  date: {
    $gte: ISODate("2022-04-01"),
    $lte: ISODate("2022-04-30")
  }
})
< [
  {
    _id: ObjectId('6958c9649f3668d00bd1620a'),
    order_id: '405-8078784-5731545',
    date: 2022-04-30T00:00:00.000Z,
    status: 'Cancelled',
    fulfilment: 'Merchant',
    sales_channel: 'Amazon.in',
    customer: {
      b2b: false,
      region: {
        city: 'MUMBAI',
        state: 'MAHARASHTRA',
        postal_code: '400081',
        country: 'IN'
      }
    },
    product: {
      sku: 'SET389-KR-NP-S',
      style: 'SET389',
      category: 'Set'
    },
    payment: {
      currency: 'INR'
    }
  }
]
```

Group sales by state:

```
>_MONGOSH
>
> db.orders.aggregate([
  {
    $group: {
      _id: "$customer.region.state",
      totalSales: { $sum: "$payment.amount" },
      orderCount: { $sum: 1 }
    }
  },
  {
    $sort: { totalSales: -1 }
  }
])
< [
  {
    _id: 'MAHARASHTRA',
    totalSales: 13335534.14,
    orderCount: 21073
  },
  {
    _id: 'KARNATAKA',
    totalSales: 10481114.37,
    orderCount: 16394
  },
  {
    _id: 'TELANGANA',
    totalSales: 6916615.65,
    orderCount: 10637
  },
  {
    _id: 'UTTAR_PRADESH',
    totalSales: 6816615.65,
    orderCount: 10637
  }
]
```

Group by category:

```
> db.orders.aggregate([
  {
    $group: {
      _id: "$product.category",
      totalSales: { $sum: "$payment.amount" }
    }
  }
])
< [
  {
    _id: 'Saree',
    totalSales: 123933.76
  },
  {
    _id: 'Western Dress',
    totalSales: 11215337.69
  },
  {
    _id: 'kurta',
    totalSales: 21291538.7
  },
  {
    _id: 'Blouse',
    totalSales: 458408.18
  },
  {
    _id: 'Ethnic Dress',
    totalSales: 791217.66
  }
]
```