AGGREGATE PIPELINE IN MONGODB

Create Database **“kk-db”** and collection **“Customers”**

```
test> show dbs
admin   40.00 KiB
config  48.00 KiB
kk-db   72.00 KiB
local   40.00 KiB
test> use kk-db
switched to db kk-db
kk-db> show collections
Students
kk-db> db.createCollection("Customers")
{ ok: 1 }
kk-db>
```

Insert the documents

```
kk-db> db.Customers.insertMany([
...   {
...     "order_id": 1,
...     "customer_name": "Ramesh Chaudhary",
...     "items": [
...       { "product": "Laptop", "quantity": 1, "price": 1200 },
...       { "product": "Mouse", "quantity": 2, "price": 25 }
...     ],
...     "order_date": ISODate("2023-10-01"),
...     "total_amount": 1250,
...     "status": "completed"
...   },
...   {
...     "order_id": 2,
...     "customer_name": "Sahil Shah",
...     "items": [
...       { "product": "Keyboard", "quantity": 1, "price": 50 },
...       { "product": "Monitor", "quantity": 1, "price": 300 }
...     ],
...     "order_date": ISODate("2023-10-02"),
...     "total_amount": 350,
...     "status": "pending"
...   },
...   {
...     "order_id": 3,
...     "customer_name": "Binita Mukarmi",
...     "items": [
...       { "product": "Printer", "quantity": 1, "price": 200 },
...       { "product": "Paper", "quantity": 5, "price": 10 }
...     ],
...     "order_date": ISODate("2023-10-03"),
...     "total_amount": 250,
...     "status": "completed"
...   },
...   {
...     "order_id": 4,
...     "customer_name": "Dinesh Khatri",
...     "items": [
...       { "product": "Tablet", "quantity": 1, "price": 500 },
...       { "product": "Case", "quantity": 1, "price": 20 }
...     ],
...     "order_date": ISODate("2023-10-04"),
...     "total_amount": 520,
...     "status": "cancelled"
...   }
... ])
{
  acknowledged: true,
```

Aggregation Pipeline

structure

```
db.collection.aggregate([
    {stage1},
    {stage2},
    {stage3},
])
```

$match – Filters documents to pass only those that match the given condition(s).
$group – Groups documents by some field(s) and performs aggregation operations like sum, avg, min, max, etc.
$sort - Sorts the documents by specified fields
$project – Reshapes each document by adding or removing fields
$limit – Limits the number of documents passed to the next stage.
$skip – Skips the specified number of documents
$unwind – Deconstructs an array field from the input documents to output a document for each element of the array
$lookup – Joins documents from another collection (like SQL JOIN).

Calculate the total revenue generated from all completed orders

[Hint: Find completed orders and sum the amounts to get total revenue]

```
db.Customers.aggregate([
        { $match: { status: "completed" } },
        { $group: { _id: null, total_revenue: { $sum: "$total_amount" }}}
    ])
```

Find the average order value for all completed orders.

[Hint: Find completed orders and average the amount]

```
db.Customers.aggregate([
        { $match: { status: "completed" }},
        { $group: { _id: null, avg_order: { $avg: "$total_amount" } }}
])
```

```
kk-db> db.Customers.aggregate([
...                     {
...                             $match: { status: "completed" }
...                     },
...                     {
...                             $group: {
...                                     _id: null,
...                                     avg_order: { $avg: "$total_amount" }                    }
...                     }
...                     ])
[ { _id: null, avg_order: 750 } ]
kk-db>
```

Identify the customer who has spent the most across all their orders.

[Hint: Group Customers by their spending, sort in descending order and get top record]

```
db.Customers.aggregate([
        {$group: {
                _id: "$customer_name",
                total_spent: { $sum: "$total_amount" } }
        },
        { $sort: { total_spent: -1 } },
        { $limit: 1 }
])
```

```
kk-db> db.Customers.aggregate([
...                     {
...                             $group: {
...                                     _id: "$customer_name",
...                                     total_spent: { $sum: "$total_amount" }
...                             }
...                     },
...                     {
...                             $sort: { total_spent: -1 }
...                     },
...                     {
...                             $limit: 1
...                     }
...                     ])
[ { _id: 'Ramesh Chaudhary', total_spent: 1250 } ]
kk-db>
```

Determine the most popular product based on the total quantity sold.

[Hint: Separate with items, group with total quantities, sort in descending order and select top record]

```
db.Customers.aggregate([
        { $unwind: "$items" },
        {$group: {
                _id: "$items.product",
                total_quantity_sold: { $sum: "$items.quantity" }}
        },
        { $sort: { total_quantity_sold: -1 } },
        { $limit: 1 }
        ])
```

```
kk-db> db.Customers.aggregate([
...                     { $unwind: "$items" },
...                     {
...                             $group: {
...                                     _id: "$items.product",
...                                     total_quantity_sold: { $sum: "$items.quantity" }
...                             }
...                     },
...                     { $sort: { total_quantity_sold: -1 } },
...                     { $limit: 1 }
...                     ])
[ { _id: 'Paper', total_quantity_sold: 5 } ]
kk-db>
```

Count the number of orders in each status.

[Hint: Group by customer status and count]

```
db.Customers.aggregate([
        {$group: {
                _id: "$status",
                order_count: { $sum: 1 }}}
        ])
```

```
kk-db> db.Customers.aggregate([
...                     {
...                             $group: {
...                                     _id: "$status",
...                                     order_count: { $sum: 1 }
...                             }
...                     }
...                     ])
[
  { _id: 'cancelled', order_count: 1 },
  { _id: 'completed', order_count: 2 },
  { _id: 'pending', order_count: 1 }
]
kk-db>
```

Calculate the total revenue generated by each customer.

[Hint: Group customer by their total revenue]

```
db.Customers.aggregate([
    {$group: {
        _id: "$customer_name",
        total_revenue: { $sum: "$total_amount" } }}
    ])
```

```
kk-db> db.Customers.aggregate([
...                     {
...                         $group: {
...                             _id: "$customer_name",
...                             total_revenue: { $sum: "$total_amount" }
...                         }
...                     }
...                     ])
[
  { _id: 'Dinesh Khatri', total_revenue: 520 },
  { _id: 'Ramesh Chaudhary', total_revenue: 1250 },
  { _id: 'Binita Mukarmi', total_revenue: 250 },
  { _id: 'Sahil Shah', total_revenue: 350 }
]
kk-db>
```

Find all orders placed between October 1,2023 and October 1,2023

[Hint: Filter order with dates]

```
db.Customers.aggregate([
    {$match: {
        order_date: {
        $gte: ISODate("2023-10-01T00:00:00Z"),
        $lte: ISODate("2023-10-03T23:59:59Z")
        }}}
    ])
```

```
kk-db> db.Customers.aggregate([
...                     {
...                                     $match: {
...                                             order_date: {
...                                             $gte: ISODate("2023-10-01T00:00:00Z"),
...                                             $lte: ISODate("2023-10-03T23:59:59Z")
...                                             }
...                                     }
...                     }
...                     ])
...
[
  {
    _id: ObjectId('67ce8ccf6fd5d51ae6fa4218'),
    order_id: 1,
    customer_name: 'Ramesh Chaudhary',
    items: [
      { product: 'Laptop', quantity: 1, price: 1200 },
      { product: 'Mouse', quantity: 2, price: 25 }
    ],
    order_date: ISODate('2023-10-01T00:00:00.000Z'),
    total_amount: 1250,
    status: 'completed'
  },
  {
    _id: ObjectId('67ce8ccf6fd5d51ae6fa4219'),
    order_id: 2,
    customer_name: 'Sahil Shah',
    items: [
      { product: 'Keyboard', quantity: 1, price: 50 },
      { product: 'Monitor', quantity: 1, price: 300 }
    ],
    order_date: ISODate('2023-10-02T00:00:00.000Z'),
    total_amount: 350,
    status: 'pending'
  },
  {
    _id: ObjectId('67ce8ccf6fd5d51ae6fa421a'),
    order_id: 3,
    customer_name: 'Binita Mukarmi',
    items: [
      { product: 'Printer', quantity: 1, price: 200 },
      { product: 'Paper', quantity: 5, price: 10 }
    ],
    order_date: ISODate('2023-10-03T00:00:00.000Z'),
    total_amount: 250,
    status: 'completed'
  }
]
kk-db>
```

Calculate the total quantity sold for each product

[Hint: separates with item and group item with total quantity sold]

```
db.Customers.aggregate([
        {$unwind: "$items" },
        {
            $group: {
                _id: "$items.product",
                total_quantity_sold: { $sum: "$items.quantity" } }
        },
        {$sort: { total_quantity_sold: -1 }}
    ])
```

```
kk-db> db.Customers.aggregate([
...                     {
...                             $unwind: "$items"
...                     },
...                     {
...                             $group: {
...                                     _id: "$items.product",
...                                     total_quantity_sold: { $sum: "$items.quantity" }
...                             }
...                     },
...                     {
...                             $sort: { total_quantity_sold: -1 }
...                     }
...                     ])
[
  { _id: 'Paper', total_quantity_sold: 5 },
  { _id: 'Mouse', total_quantity_sold: 2 },
  { _id: 'Printer', total_quantity_sold: 1 },
  { _id: 'Tablet', total_quantity_sold: 1 },
  { _id: 'Laptop', total_quantity_sold: 1 },
  { _id: 'Monitor', total_quantity_sold: 1 },
  { _id: 'Case', total_quantity_sold: 1 },
  { _id: 'Keyboard', total_quantity_sold: 1 }
]
kk-db>
```