**Exercise** 1.

1.Create 'customers' and 'orders' collections and insert data into them

db.orders.insertMany([

{

"order\_id": "ord1",

"customer\_id": "raj@gmail.com",

"order\_date": ISODate("2024-05-15T14:00:00Z"),

"status": "shipped",

"items": [

{ "product\_name": "Laptop", "quantity": 1, "price": 1500 },

{ "product\_name": "Mouse", "quantity": 2, "price": 25 }

],

"total\_value": 1550

},

{

"order\_id": "ord2",

"customer\_id": "ram@gmail.com",

"order\_date": ISODate("2024-06-20T15:30:00Z"),

"status": "delivered",

"items": [

{ "product\_name": "Tablet", "quantity": 1, "price": 300 },

{ "product\_name": "Keyboard", "quantity": 1, "price": 50 }

],

"total\_value": 350

},

{

"order\_id": "ord3",

"customer\_id": "mohan@gmail.com",

"order\_date": ISODate("2024-07-10T09:45:00Z"),

"status": "processing",

"items": [

{ "product\_name": "Smartphone", "quantity": 1, "price": 800 }

],

"total\_value": 800

},

{

"order\_id": "ord4",

"customer\_id": "sita@gmail.com",

"order\_date": ISODate("2024-08-05T11:20:00Z"),

"status": "shipped",

"items": [

{ "product\_name": "Monitor", "quantity": 2, "price": 200 },

{ "product\_name": "HDMI Cable", "quantity": 3, "price": 15 }

],

"total\_value": 445

},

{

"order\_id": "ord5",

"customer\_id": "geeta@gmail.com",

"order\_date": ISODate("2024-09-01T17:00:00Z"),

"status": "cancelled",

"items": [

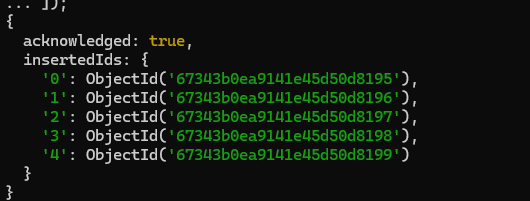
{ "product\_name": "Printer", "quantity": 1, "price": 200 }

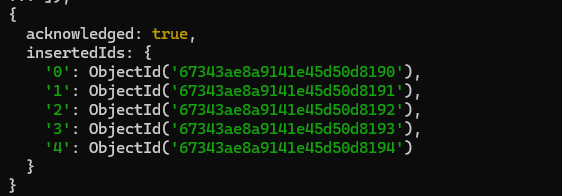
],

"total\_value": 200

}

]);





2. Insert 5 orders into the 'orders' collection

db.orders.insertMany([

{

"order\_id": "ord1",

"customer\_id": "raj@gmail.com",

"order\_date": ISODate("2024-11-05T10:30:00Z"),

"status": "shipped",

"items": [

{ "product\_name": "Laptop", "quantity": 1, "price": 1500 }

],

"total\_value": 1500

},

{

"order\_id": "ord2",

"customer\_id": "ram@gmail.com",

"order\_date": ISODate("2024-11-08T13:45:00Z"),

"status": "delivered",

"items": [

{ "product\_name": "Tablet", "quantity": 1, "price": 300 }

],

"total\_value": 300

},

{

"order\_id": "ord3",

"customer\_id": "mohan@gmail.com",

"order\_date": ISODate("2024-11-10T16:00:00Z"),

"status": "processing",

"items": [

{ "product\_name": "Smartphone", "quantity": 1, "price": 800 }

],

"total\_value": 800

},

{

"order\_id": "ord4",

"customer\_id": "sita@gmail.com",

"order\_date": ISODate("2024-11-11T11:20:00Z"),

"status": "shipped",

"items": [

{ "product\_name": "Monitor", "quantity": 1, "price": 200 }

],

"total\_value": 200

},

{

"order\_id": "ord5",

"customer\_id": "geeta@gmail.com",

"order\_date": ISODate("2024-11-12T09:15:00Z"),

"status": "cancelled",

"items": [

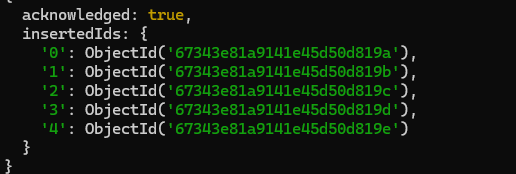
{ "product\_name": "Printer", "quantity": 1, "price": 200 }

],

"total\_value": 200

}

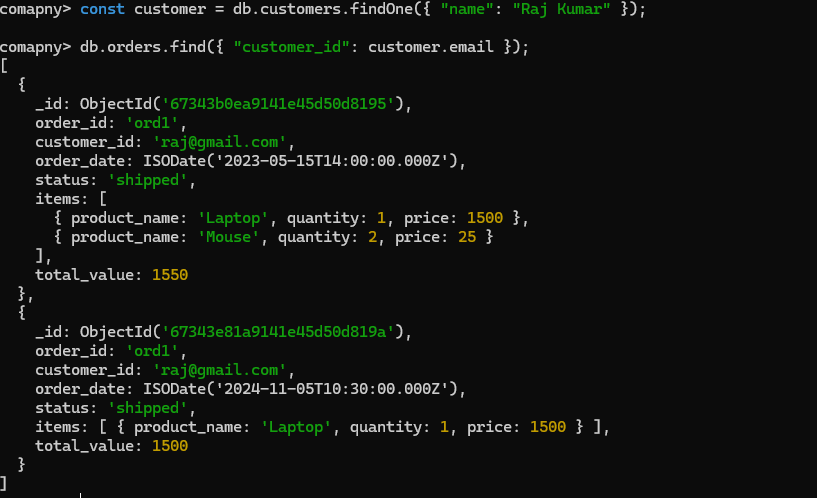
]);



3. Find all orders placed by the customer "Raj Kumar"

const customer = db.customers.findOne({ "name": "Raj Kumar" });

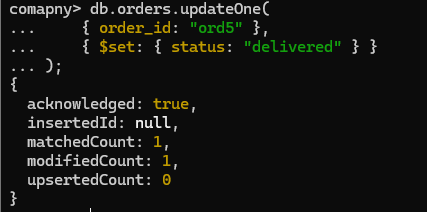
db.orders.find({ "customer\_id": customer.email });



4. Find the customer information for the order with order\_id "ord1"

const order = db.orders.findOne({ order\_id: "ord1 });

const customerForOrder = db.customers.findOne({ \_id: order.customer\_id });

t

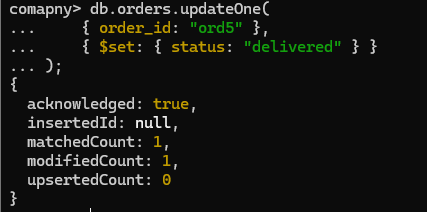
5. Update the status of the order with order\_id "ord5" to "delivered"

db.orders.updateOne(

{ order\_id: "ord5" },

{ $set: { status: "delivered" } }

);



6. Delete the order with order\_id "ord5"

db.orders.deleteOne({ order\_id: "ord5" });



Exercise 2.

1. Calculate Total Value of All Orders by Customer

db.orders.aggregate([

{

$group: {

\_id: "$customer\_id",

total\_order\_value: { $sum: "$total\_value" }

}

},

{

$lookup: {

from: "customers",

localField: "\_id",

foreignField: "email",

as: "customer\_info"

}

},

{

$unwind: "$customerdata"

},

{

$project: {

"customerdata.name": 1,

"customerdata.email": 1,

"total\_order\_value": 1

}

}

]);



2. Group Orders by Status

db.orders.aggregate([

{

$group: {

\_id: "$status",

total\_orders: { $sum: 1 },

total\_value: { $sum: "$total\_value" }

}

},

{

$project: {

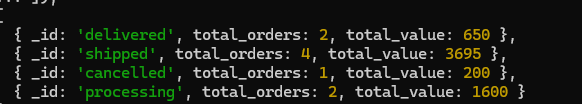
total\_orders: 1,

total\_value: 1

}

}

]);



3. List Customers with Their Recent Orders

db.orders.aggregate([

{

$lookup: {

from: "customers",

localField: "customer\_id",

foreignField: "customer\_id",

as: "customer\_info"

}

},

{

$sort: { "order\_date": -1 }

},

{

$group: {

\_id: "$customer\_id",

most\_recent\_order: { $first: "$$ROOT" },

customer\_info: { $first: "$customer\_info" }

}

},

{

$project: {

"customer\_info.name": 1,

"customer\_info.email": 1,

"most\_recent\_order.total\_value": 1,

"most\_recent\_order.order\_date": 1

}

}

]);

4. Find the Most Expensive Order by Customer

db.orders.aggregate([

{

$sort: { "total\_value": -1 }

},

{

$group: {

\_id: "$customer\_id",

most\_expensive\_order\_id: { $first: "$order\_id" },

most\_expensive\_order\_date: { $first: "$order\_date" },

most\_expensive\_order\_value: { $first: "$total\_value" }

}

},

{

$lookup: {

from: "customers",

localField: "\_id",

foreignField: "email",

as: "customer\_data"

}

},

{

$unwind: "$customer\_data"

},

{

$project: {

"customer\_data.name": 1,

"customer\_data.email": 1,

"most\_expensive\_order\_id": 1,

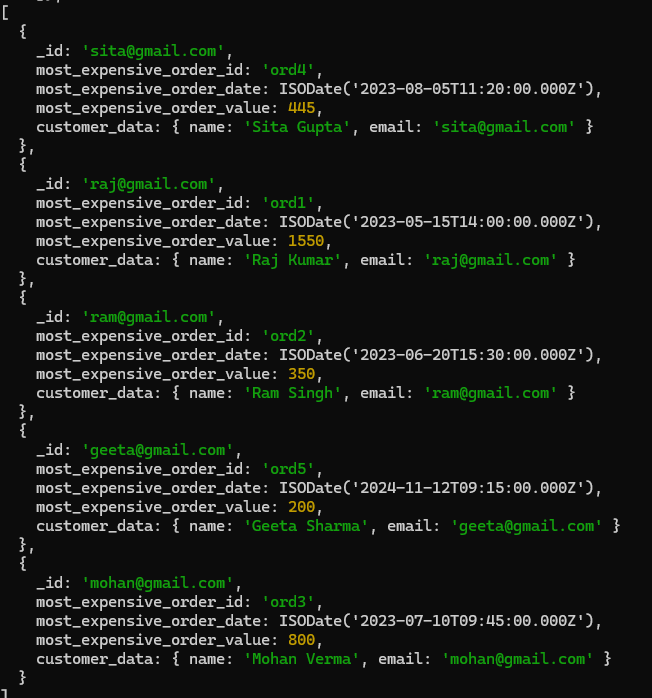
"most\_expensive\_order\_value": 1,

"most\_expensive\_order\_date": 1

}

}

]);



Exercise 3.

1. Find All Customers Who Placed Orders in the Last Month

const lastMonth = new Date();

lastMonth.setMonth(lastMonth.getMonth() - 1);

db.orders.aggregate([

{

$match: { "order\_date": { $gte: lastMonth } }

},

{

$lookup: {

from: "customers",

localField: "customer\_id",

foreignField: "email",

as: "customer\_data"

}

},

{

$unwind: "$customer\_data"

},

{

$group: {

\_id: "$customer\_id",

customer\_name: { $first: "$customer\_data.name" },

customer\_email: { $first: "$customer\_data.email" }

}

}

]);

2. Find All Products Ordered by a Specific Customer

db.orders.aggregate([

{

$match: { customer\_id: 'ram@gmail.com' }

},

{

$unwind: "$items"

},

{

$project: {

"items.product\_name": 1,

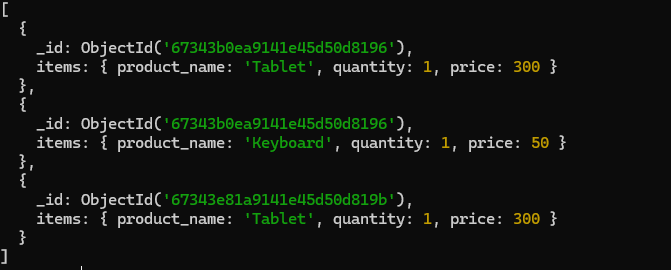
"items.quantity": 1,

"items.price": 1

}

}

];



3. Find the Top 3 Customers with the Most Expensive Total Orders

db.orders.aggregate([

{

$group: {

\_id: "$customer\_id",

total\_spent: { $sum: "$total\_value" }

}

},

{

$sort: { total\_spent: -1 }

},

{

$limit: 3

},

{

$lookup: {

from: "customers",

localField: "\_id",

foreignField: "email",

as: "customer\_data"

}

},

{

$unwind: "$customer\_data"

},

{

$project: {

"customer\_data.name": 1,

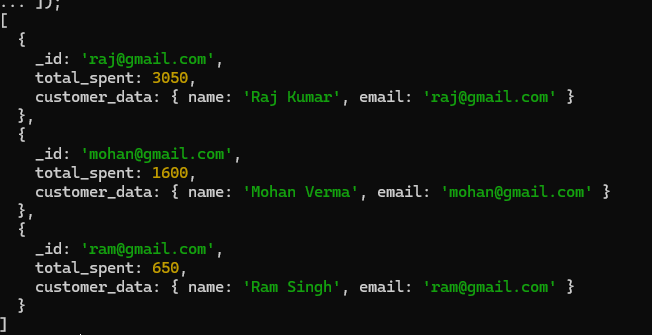
"customer\_data.email": 1,

"total\_spent": 1

}

}

]);



4. Add a New Order for an Existing Customer

db.orders.insertOne({

order\_id: "ORD6",

customer\_id: customer.email,

order\_date: new Date("2024-11-13T10:00:00Z"),

status: "processing",

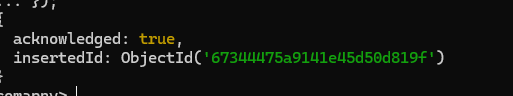
items: [

{ product\_name: "Keyboard", quantity: 1, price: 50 }

],

total\_value: 50

});



Excercise 4.

1. Find Customers Who Have Not Placed Orders

db.customers.aggregate([

{

$lookup: {

from: "orders",

localField: "\_id",

foreignField: "customer\_id",

as: "orders"

}

},

{

$match: { orders: { $size: 0 } }

},

{

$project: {

name: 1,

email: 1

}

}

]);



2. Calculate the Average Number of Items Ordered per Order

db.orders.aggregate([

{

$project: {

item\_count: { $size: "$items" }

}

},

{

$group: {

\_id: null,

average\_items\_per\_order: { $avg: "$item\_count" }

}

}

]);



3. Join Customer and Order Data Using $lookup

db.orders.aggregate([

{

$lookup: {

from: "customers",

localField: "customer\_id",

foreignField: "email",

as: "customer\_data"

}

},

{

$unwind: "$customer\_data"

},

{

$project: {

order\_id: 1,

order\_date: 1,

status: 1,

items: 1,

total\_value: 1,

"customer\_data.name": 1,

"customer\_data.email": 1,

"customer\_data.phone": 1,

"customer\_data.address": 1

}

}

]);

