**Mongodb Queries Online Shopping**

// 1. Find High-Spending Users

db.users.aggregate([

{

$lookup: {

from: "orders",

localField: "userId",

foreignField: "userId",

as: "orders"

}

},

{

$unwind: "$orders"

},

{

$group: {

\_id: "$userId",

name: { $first: "$name" },

totalSpent: { $sum: "$orders.totalAmount" }

}

},

{

$match: {

totalSpent: { $gt: 500 }

}

},

{

$project: {

\_id: 0,

userId: "$\_id",

name: 1,

totalSpent: 1

}

}

]);

// 2. List Popular Products by Average Rating

db.products.aggregate([

{

$unwind: "$ratings"

},

{

$group: {

\_id: "$productId",

name: { $first: "$name" },

avgRating: { $avg: "$ratings.rating" }

}

},

{

$match: {

avgRating: { $gte: 4 }

}

},

{

$project: {

\_id: 0,

productId: "$\_id",

name: 1,

avgRating: 1

}

}

]);

// 3. Search for Orders in a Specific Time Range

db.orders.aggregate([

{

$match: {

orderDate: {

$gte: new Date("2024-12-01T00:00:00Z"),

$lte: new Date("2024-12-31T23:59:59Z")

}

}

},

{

$lookup: {

from: "users",

localField: "userId",

foreignField: "userId",

as: "user"

}

},

{

$unwind: "$user"

},

{

$project: {

\_id: 0,

orderId: 1,

orderDate: 1,

status: 1,

totalAmount: 1,

userName: "$user.name"

}

}

]);

// 4. Update Stock After Order Completion

db.orders.find({ orderId: "ORD001" }).forEach(order => {

order.items.forEach(item => {

db.products.updateOne(

{ productId: item.productId },

{ $inc: { stock: -item.quantity } }

);

});

});

// 5. Find Nearest Warehouse

db.warehouses.aggregate([

{

$geoNear: {

near: { type: "Point", coordinates: [-74.006, 40.7128] }, // User's location

distanceField: "distance",

maxDistance: 50000, // 50 km in meters

spherical: true,

query: { products: "P001" }

}

},

{

$project: {

\_id: 0,

warehouseId: 1,

distance: 1,

products: 1

}

}

]);