// 1. Basic Lookup with Matching Criteria

db.students.aggregate([

{

$lookup: {

from: "courses",

localField: "course\_id",

foreignField: "\_id",

as: "course"

}

},

{

$project: {

"student\_name": "$name",

"course\_name": { $arrayElemAt: ["$course.course\_name", 0] }

}

}

])

// 2. Filtering with Match

db.students.aggregate([

{

$match: {

scores: { $gt: 75 }

}

}

])

// 3. Lookup with Multiple Conditions

db.orders.aggregate([

{

$match: {

amount: { $gt: 500 }

}

},

{

$lookup: {

from: "customers",

localField: "customerId",

foreignField: "\_id",

as: "customer"

}

},

{

$match: {

"customer.location": "New York"

}

}

])

// 4. Multi-Stage Aggregation

db.sales.aggregate([

{

$lookup: {

from: "products",

localField: "productId",

foreignField: "\_id",

as: "product"

}

},

{

$match: {

"product.price": { $gt: 20 }

}

},

{

$group: {

\_id: "$productId",

totalRevenue: {

$sum: { $multiply: ["$quantity", { $arrayElemAt: ["$product.price", 0] }] }

}

}

}

])

// 5. Using $in Operator

db.employees.aggregate([

{

$match: {

department: {

$in: ["HR", "Engineering", "Marketing"]

}

}

}

])

// 6. Advanced Filtering with $or Operator

db.transactions.aggregate([

{

$match: {

$or: [

{ type: "credit" },

{ amount: { $gt: 1000 } }

]

}

}

])

// 7. Combining $lookup and $ne Operator

db.orders.aggregate([

{

$lookup: {

from: "customers",

localField: "customerId",

foreignField: "\_id",

as: "customer"

}

},

{

$match: {

"customer.membership": { $ne: "premium" }

}

}

])

// 8. Multi-Level Lookup

db.students.aggregate([

{

$lookup: {

from: "courses",

localField: "courseId",

foreignField: "\_id",

as: "course"

}

},

{

$unwind: "$course"

},

{

$lookup: {

from: "instructors",

localField: "course.instructorId",

foreignField: "\_id",

as: "instructor"

}

},

{

$project: {

student\_name: "$name",

course\_name: "$course.course\_name",

instructor\_name: { $arrayElemAt: ["$instructor.name", 0] }

}

}

])

// 9. Conditional Lookup with $gte

db.purchases.aggregate([

{

$lookup: {

from: "customers",

localField: "customerId",

foreignField: "\_id",

as: "customer"

}

},

{

$match: {

"customer.loyaltyPoints": { $gte: 1000 }

}

}

])

// 10. Using $and in a Complex Condition

db.products.aggregate([

{

$match: {

$and: [

{ category: "electronics" },

{ price: { $lt: 500 } },

{ stock: { $gt: 10 } }

]

}

}

])