

```

// lookup -- left outer join
// syntax
{
  $lookup: {
    from: "customers", // other collection
    localField: "customerId", // field in orders
    foreignField: "customerId", // field in customers
    as: "customerInfo" // output array field
  }
}

// Examples
// 1. join orders with customers(display orderid, sales , customerName, city)
// Thinking
// Start from orders
// Need customer data → $lookup
// Array output → $unwind
// Select fields → $project
db.orders.aggregate([
  {
    $lookup: {
      from: "customers",
      localField: "customerId",
      foreignField: "customerId",
      as: "customerInfo"
    }
  },
  { $unwind: "$customerInfo" },
  {
    $project: {
      _id: 0,
      orderId: 1,
      sales: 1,
      customerName: "$customerInfo.customerName",
      city: "$customerInfo.city"
    }
  }
]);

//2. basic filter after lookup
// show orders placedby customers from dallas
// Thinking
// Need customer city → lookup first
// Then filter
db.orders.aggregate([
  {
    $lookup: {
      from: "customers",
      localField: "customerId",
      foreignField: "customerId",
      as: "cust"
    }
  }
]);

```

```

    }
  },
  { $unwind: "$cust" },
  { $match: { "cust.city": "Dallas" } },
  {
    $project: {
      _id: 0,
      orderId: 1,
      sales: 1,
      city: "$cust.city"
    }
  }
]);

// customers with or without orders
// Display all customers, even those with no orders
// Thinking
// Start from customers
// Lookup orders
// Do NOT unwind
db.customers.aggregate([
  {
    $lookup: {
      from: "orders",
      localField: "customerId",
      foreignField: "customerId",
      as: "orders"
    }
  },
  {
    $project: {
      _id: 0,
      customerName: 1,
      totalOrders: { $size: "$orders" }
    }
  }
]);

// Medium -- total sales per customer, display customerName and their total sales
// Thinking
// Customers → lookup orders
// Unwind orders
// Group by customer
db.customers.aggregate([
  {
    $lookup: {
      from: "orders",
      localField: "customerId",
      foreignField: "customerId",
      as: "orders"
    }
  },
  { $unwind: "$orders" },
  {

```

```

        $group: {
          _id: "$customerName",
          totalSales: { $sum: "$orders.sales" }
        }
      }
    });

// list customers who never placed an order
// Thinking
// Lookup
// orders array empty → $size: 0
db.customers.aggregate([
  {
    $lookup: {
      from: "orders",
      localField: "customerId",
      foreignField: "customerId",
      as: "orders"
    }
  },
  {
    $match: {
      orders: { $eq: [] }
    }
  },
  {
    $project: {
      _id: 0,
      customerName: 1
    }
  }
]);

// display orderId, productName, category, qty( orders with product details)
// Thinking
// Orders → lookup products
// Unwind
// Project
db.orders.aggregate([
  {
    $lookup: {
      from: "products",
      localField: "productId",
      foreignField: "productId",
      as: "productInfo"
    }
  },
  { $unwind: "$productInfo" },
  {
    $project: {
      _id: 0,
      orderId: 1,
      qty: 1,
      productName: "$productInfo.productName",

```

```
        category: "$productInfo.category"
      }
    }
  });

// find total qty sold for each product category
// Thinking
// Orders → lookup products
// Unwind
// Group by category
db.orders.aggregate([
  {
    $lookup: {
      from: "products",
      localField: "productId",
      foreignField: "productId",
      as: "prod"
    }
  },
  { $unwind: "$prod" },
  {
    $group: {
      _id: "$prod.category",
      totalQty: { $sum: "$qty" }
    }
  }
]);

// top customer by total sales

// Find customer who spent the most
// Thinking
// Customers → orders
// Group → sort → limit
db.customers.aggregate([
  {
    $lookup: {
      from: "orders",
      localField: "customerId",
      foreignField: "customerId",
      as: "orders"
    }
  },
  { $unwind: "$orders" },
  {
    $group: {
      _id: "$customerName",
      totalSales: { $sum: "$orders.sales" }
    }
  },
  { $sort: { totalSales: -1 } },
  { $limit: 1 }
]);
```

