db.inventory.drop()

db.createCollection("inventory")

db.createCollection("orders")

db.inventory.insertMany([

{ prodId: 101, price: 12, quantity: 200 },

{ prodId: 102, price: 18, quantity: 300 },

{ prodId: 103, price: 22, quantity: 150 },

{ prodId: 104, price: 25, quantity: 100 },

{ prodId: 105, price: 30, quantity: 80 },

{ prodId: 108, price: 40, quantity: 70 },

{ prodId: 109, price: 45, quantity: 60 },

{ prodId: 110, price: 50, quantity: 50 },

{ prodId: 112, price: 60, quantity: 30 },

{ prodId: 113, price: 65, quantity: 20 },

{ prodId: 114, price: 70, quantity: 10 },

{ prodId: 115, price: 75, quantity: 5 },

{ prodId: 117, price: 85, quantity: 15 },

{ prodId: 118, price: 90, quantity: 8 },

{ prodId: 119, price: 95, quantity: 12 }

])

db.orders.insertMany([

{ orderId: 216, custid: 310, prodId: 101, numPurchased: 10 },

{ orderId: 217, custid: 311, prodId: 102, numPurchased: 5 },

{ orderId: 218, custid: 312, prodId: 103, numPurchased: 7 },

{ orderId: 219, custid: 314, prodId: 103, numPurchased: 3 },

{ orderId: 220, custid: 314, prodId: 105, numPurchased: 8 },

{ orderId: 221, custid: 315, prodId: 106, numPurchased: 12 },

{ orderId: 222, custid: 316, prodId: 107, numPurchased: 4 },

{ orderId: 223, custid: 317, prodId: 108, numPurchased: 2 },

{ orderId: 224, custid: 318, prodId: 109, numPurchased: 5 },

{ orderId: 225, custid: 319, prodId: 107, numPurchased: 11 },

{ orderId: 226, custid: 320, prodId: 111, numPurchased: 3 },

{ orderId: 227, custid: 321, prodId: 112, numPurchased: 8 },

{ orderId: 228, custid: 322, prodId: 113, numPurchased: 1 },

{ orderId: 229, custid: 323, prodId: 114, numPurchased: 4 },

{ orderId: 231, custid: 325, prodId: 116, numPurchased: 9 },

{ orderId: 232, custid: 326, prodId: 117, numPurchased: 5 },

{ orderId: 233, custid: 327, prodId: 118, numPurchased: 6 },

{ orderId: 234, custid: 328, prodId: 119, numPurchased: 3 },

{ orderId: 235, custid: 329, prodId: 120, numPurchased: 4 },

{ orderId: 236, custid: 330, prodId: 121, numPurchased: 1 },

{ orderId: 237, custid: 331, prodId: 122, numPurchased: 5 }

])

db.inventory.findOne()

db.orders.findOne()

– Inner Join

db.orders.aggregate([

{$lookup: {

from: "inventory",

localField: "prodId", // Field from "orders" collection

foreignField: "prodId", // Field from "inventory" collection

as: "productDetails"

}},

{$unwind:"$productDetails"} // Flatten array to get a single document

])

**OR**

db.orders.aggregate([

{

$lookup: {

from: "inventory",

localField: "prodId",

foreignField: "prodId",

as: "inventoryDetails"

}

},

{

$unwind: "$inventoryDetails"

},

{

$project: {

\_id: 0,

orderId: 1,

custid: 1,

prodId: 1,

numPurchased: 1,

price: "$inventoryDetails.price",

quantity: "$inventoryDetails.quantity"

}

}

])

– Left Join

db.orders.aggregate([

{$lookup: {

from: "inventory",

localField: "prodId", // Field from "orders" collection

foreignField: "prodId", // Field from "inventory" collection

as: "productDetails"

}}

])

**OR**

db.orders.aggregate([

{

$lookup: {

from: "inventory",

localField: "prodId",

foreignField: "prodId",

as: "inventoryDetails"

}

},

{

$unwind: {

path: "$inventoryDetails",

preserveNullAndEmptyArrays: true

}

},

{

$project: {

\_id: 0,

orderId: 1,

custid: 1,

prodId: 1,

numPurchased: 1,

price: { $ifNull: ["$inventoryDetails.price", null] },

quantity: { $ifNull: ["$inventoryDetails.quantity", null] }

}

}

])

– Right Join

db.inventory.aggregate([

{

$lookup: {

from: "orders",

localField: "prodId",

foreignField: "prodId",

as: "ordersDetails"

}

},

{

$unwind: {

path: "$ordersDetails",

preserveNullAndEmptyArrays: true

}

},

{

$project: {

\_id: 0,

prodId:1,

price: 1,

quantity: 1,

orderId: { $ifNull: ["$ordersDetails.orderId", null] },

custid: { $ifNull: ["$ordersDetails.custid", null] },

numPurchased: { $ifNull: ["$ordersDetails.numPurchased", null] }

}

}

])

– Union: Concatenation between two collection documents.

db.orders.aggregate([

{

$unionWith: {

coll: "inventory"

}

}

])

– Union (Full Outer Join)

db.orders.aggregate([

// First stage: Lookup inventory details for each order

{

$lookup: {

from: "inventory",

localField: "prodId",

foreignField: "prodId",

as: "inventoryDetails"

}

},

{

$unwind: {

path: "$inventoryDetails",

preserveNullAndEmptyArrays: true

}

},

{

$project: {

\_id: 0,

orderId: 1,

custid: 1,

prodId: 1,

numPurchased: 1,

price: { $ifNull: ["$inventoryDetails.price", null] },

quantity: { $ifNull: ["$inventoryDetails.quantity", null] }

}

},

// Union with inventory collection to include unmatched inventory records

{

$unionWith: {

coll: "inventory",

pipeline: [

{

$lookup: {

from: "orders",

localField: "prodId",

foreignField: "prodId",

as: "orderDetails"

}

},

{

$unwind: {

path: "$orderDetails",

preserveNullAndEmptyArrays: true

}

},

{

$project: {

\_id: 0,

prodId: 1,

price: 1,

quantity: 1,

orderId: { $ifNull: ["$orderDetails.orderId", null] },

custid: { $ifNull: ["$orderDetails.custid", null] },

numPurchased: { $ifNull: ["$orderDetails.numPurchased", null] }

}

}

]

}

}

])