**MYNTRA MP ORDERS CODE:**

CODE 1:

db.Myntra\_MP\_Orders.aggregate([

{ "$addFields": { "order\_release\_id": { "$toString": "$order\_release\_id" } } },

{ "$project": { "concatenated": { "$concat": ["$order\_release\_id", "$seller\_order\_id"] } } },

{"$group": { "\_id": "$concatenated", "dups": { $push: "$\_id" }, "count": { "$sum": 1 } } },

{ "$match": { "\_id": { "$ne": null }, "count": { "$gt": 1 } } }],

{ allowDiskUse: true })

.forEach(function (doc) {

doc.dups.shift();

db.Myntra\_MP\_Orders.remove({ \_id: { $in: doc.dups } });

});

CODE 2:

db.Myntra\_MP\_Orders.aggregate([

{"$addFields":{"quantity": 1} },

{ "$project": { "\_id": 0 } },

{ "$out": "Final\_MP\_Orders\_File" }

]);

*NOTE: First Run Code 1, comment Code 1, and then run Code 2*

**MYNTRA MP RETURNS CODE:**

CODE 1:

db.Myntra\_MP\_Returns.createIndex({order\_id: 1})

db.Myntra\_MP\_Orders.createIndex({order\_release\_id: 1})

db.Myntra\_MP\_Returns.aggregate([

{ "$addFields": { "order\_id": { "$toString": "$order\_id" } } },

{"$group": { "\_id": "$order\_id", "dups": { $push: "$\_id" }, "count": { "$sum": 1 } } },

{ "$match": { "\_id": { "$ne": null }, "count": { "$gt": 1 } } }],

{ allowDiskUse: true })

.forEach(function (doc) {

doc.dups.shift();

db.Myntra\_MP\_Returns.remove({ \_id: { $in: doc.dups } });

});

CODE 2:

db.Myntra\_MP\_Returns.aggregate([

{

$lookup: {

from: "Myntra\_MP\_Orders",

localField: "order\_id",

foreignField: "order\_release\_id",

as: "order\_docs"} },

{

$unwind: "$order\_docs"},

{ $addFields: {returns\_final\_amount: "$order\_docs.final\_amount"}},

{$project: {\_id: 0}},

{$out: "Final\_MP\_Returns\_File"} ]);

*NOTE: First Run Code 1, comment Code 1, and then run Code 2.*

*Also, Out the File as Final\_MP\_Returns\_File\_2*

**MYNTRA\_FYND\_ORDERS:**

CODE:

db.Myntra\_fynd\_Orders.createIndex({affiliate\_order\_id: 1})

db.Myntra\_MP\_Orders.createIndex({seller\_order\_id: 1})

db.Myntra\_fynd\_Orders.aggregate(["

{

$lookup: {

from: "Myntra\_MP\_Orders",

localField: "affiliate\_order\_id",

foreignField: "seller\_order\_id",

as: "matched\_orders"

}},

{$unwind: "$matched\_orders"},

{$addFields: {fynd\_order\_release\_id: "$matched\_orders.order\_release\_id"} },

{$project: {\_id: 0} },

{$out: "Final\_Fynd\_Orders"}

]);

**MYNTRA\_FYND\_RETURNS:**

CODE:

db.Myntra\_fynd\_Returns.createIndex({affiliate\_order\_id: 1})

db.Myntra\_MP\_Orders.createIndex({seller\_order\_id: 1})

db.Myntra\_fynd\_Returns.aggregate([

{

$lookup: {

from: "Myntra\_MP\_Orders",

localField: "affiliate\_order\_id",

foreignField: "seller\_order\_id",

as: "matched\_release"

}},

{$unwind: "$matched\_release"},

{$addFields: {fynd\_return\_release\_id: "$matched\_release.order\_release\_id"} },

{$project: {\_id: 0} },

{$out: "Final\_Fynd\_Returns"}

]);

**MYNTRA\_SAP\_ORDERS:**

CODE:

db.Myntra\_SAP\_Orders.createIndex({tender\_item\_number: 1})

db.Myntra\_fynd\_Orders.createIndex({bag\_id: 1})

db.Myntra\_fynd\_Returns.createIndex({bag\_id: 1})

db.Myntra\_SAP\_Orders.aggregate([

{ "$addFields": { "tender\_item\_number": { "$toInt": "$tender\_item\_number" } } },

{

"$lookup": {

"from": "Myntra\_fynd\_Orders",

"localField": "tender\_item\_number",

"foreignField": "bag\_id",

"as": "matched\_saporders"} },

{

"$addFields": {

"sap\_order\_release\_id": {

"$cond": {

"if": { "$gt": [{ "$size": "$matched\_saporders" }, 0] },

"then": { "$arrayElemAt": ["$matched\_saporders.affiliate\_order\_id", 0] },

"else": null } } } },

{

"$lookup": {

"from": "Myntra\_fynd\_Returns",

"localField": "tender\_item\_number",

"foreignField": "bag\_id",

"as": "matched\_sapreturns"} },

{

"$addFields": {

"sap\_return\_release\_id": {

"$cond": {

"if": { "$gt": [{ "$size": "$matched\_sapreturns" }, 0] },

"then": { "$arrayElemAt": ["$matched\_sapreturns.affiliate\_order\_id", 0] },

"else": null } } } },

{

"$addFields": {

"new\_sap\_order\_release\_id": {

"$cond": {

"if": { "$eq": ["$sap\_order\_release\_id", null] },

"then": "$sap\_return\_release\_id",

"else": "$sap\_order\_release\_id"} } } },

{ "$addFields": { "tender\_value": { "$toInt": "$tender\_value" } } },

{

$group: { \_id:

{

new\_sap\_order\_release\_id: "$new\_sap\_order\_release\_id",

department: "$department",

tender\_item\_number: "$tender\_item\_number",

store: "$store",

ship\_sales: "$ship\_sales"

},

"total\_tender\_value": { $sum: "$tender\_value" },

"data": { "$first": "$$ROOT" }

} },

{

"$replaceRoot": { "newRoot": { "$mergeObjects": ["$data", { "total\_tender\_value": "$total\_tender\_value" }] } }

},

{ "$project": { "new\_sap\_order\_release\_id": 1,"total\_tender\_value": 1, "department": 1, "tender\_item\_number": 1, "store": 1, "ship\_sales": 1 } },

{ "$out": "Final\_Sap\_File" }

]);

**FINAL\_MP\_ORDERS\_FILE:**

CODE:

db.Final\_MP\_Orders\_File.createIndex({ order\_release\_id: 1 });

db.Final\_MP\_Returns\_File.createIndex({ order\_id: 1 });

db.Final\_MP\_Returns\_File\_2.createIndex({ order\_id: 1 });

db.Final\_Fynd\_Orders.createIndex({ fynd\_order\_release\_id: 1 });

db.Final\_Fynd\_Returns.createIndex({ fynd\_return\_release\_id: 1 });

db.Final\_Sap\_File.createIndex({ department: 1 });

db.Final\_Sap\_File.createIndex({ new\_sap\_order\_release\_id: 1 });

var reservoirSize = 27515;

// Desired number of documents in the sample & the value has to be changed incrementally as data increases

var reservoir = [];

db.Final\_MP\_Orders\_File.aggregate([

{

$lookup: {

from: "Final\_MP\_Returns\_File",

let: { order\_release\_id: "$order\_release\_id" },

pipeline: [

{

$match: {

$expr: { $eq: ["$order\_id", "$$order\_release\_id"] },

"lmdo\_status": { $in: ["DELIVERED", "CLOSED"] }

} },

{

$addFields: {

mp\_return\_qty: "$quantity",

mp\_return\_amt: "$returns\_final\_amount",

mp\_return\_status: "$lmdo\_status",

mp\_return\_initiated: "$return\_created\_date",

mp\_return\_master\_bag\_id: "$master\_bag\_id"

} } ],

as: "mp\_returns\_vl"

} },

{

$addFields: {

mp\_returns\_vl: { $arrayElemAt: ["$mp\_returns\_vl", 0] }

} },

{

$addFields: {

mp\_returns\_vl: {

$cond: {

if: { $eq: [{ $type: "$mp\_returns\_vl" }, "missing"] },

then: {},

else: "$mp\_returns\_vl"

} } } },

{

$addFields: {

mp\_return\_qty: { $ifNull: ["$mp\_returns\_vl.quantity", 0] },

mp\_return\_amt: { $ifNull: ["$mp\_returns\_vl.returns\_final\_amount", 0] },

mp\_return\_status: { $ifNull: ["$mp\_returns\_vl.lmdo\_status", 0] },

mp\_return\_initiated\_date: { $ifNull: ["$mp\_returns\_vl.return\_created\_date", 0] },

mp\_return\_master\_bag\_id:{ $ifNull: ["$mp\_returns\_vl.master\_bag\_id", 0] }

} },

{

$lookup: {

from: "Final\_MP\_Returns\_File\_2",

let: { order\_release\_id: "$order\_release\_id" },

pipeline: [

{

$match: {

$expr: { $eq: ["$order\_id", "$$order\_release\_id"] },

"lmdo\_status": { $nin: ["DELIVERED", "CLOSED"] },

"return\_status": "DLS"

} },

{

$addFields: {

mp\_return\_qty\_2: "$quantity",

mp\_return\_amt\_2: "$returns\_final\_amount",

mp\_hub\_return\_status: "$return\_status",

mp\_return\_initiated\_2: "$return\_created\_date"

} } ],

as: "mp\_returns\_vl\_2"

} },

{

$addFields: {

mp\_returns\_vl\_2: { $arrayElemAt: ["$mp\_returns\_vl\_2", 0] }

} },

{

$addFields: {

mp\_returns\_vl\_2: {

$cond: {

if: { $eq: [{ $type: "$mp\_returns\_vl\_2" }, "missing"] },

then: {},

else: "$mp\_returns\_vl\_2"

} } } },

{

$addFields: {

mp\_return\_qty\_2: { $ifNull: ["$mp\_returns\_vl\_2.quantity", 0] },

mp\_return\_amt\_2: { $ifNull: ["$mp\_returns\_vl\_2.returns\_final\_amount", 0] },

mp\_hub\_return\_status: { $ifNull: ["$mp\_returns\_vl\_2.return\_status", 0] },

mp\_return\_initiated\_date\_2: { $ifNull: ["$mp\_returns\_vl\_2.return\_created\_date", 0] }

} },

{$addFields: {final\_mp\_ret\_qty: {$sum: ["$mp\_return\_qty", "$mp\_return\_qty\_2"]}}},

{$addFields: {final\_mp\_ret\_amt: {$sum: ["$mp\_return\_amt", "$mp\_return\_amt\_2"]}}},

{

$set: {

mp\_return\_initiated\_date: {

$cond: [

{ $or: [

{ $eq: ["$mp\_return\_initiated\_date", 0] },

{ $eq: [{ $type: "$mp\_return\_initiated\_date" }, "number"] },

{ $eq: ["$mp\_return\_initiated\_date", NaN] }

] },

NaN,

"$mp\_return\_initiated\_date"

] },

mp\_return\_initiated\_date\_2: {

$cond: [

{ $or: [

{ $eq: ["$mp\_return\_initiated\_date\_2", 0] },

{ $eq: [{ $type: "$mp\_return\_initiated\_date\_2" }, "number"] },

{ $eq: ["$mp\_return\_initiated\_date\_2", NaN] }

] },

NaN,

"$mp\_return\_initiated\_date\_2"

] } } },

{

$set: {

mp\_return\_initiated\_date: {

$cond: [

{ $eq: ["$mp\_return\_initiated\_date", ""] },

NaN,

"$mp\_return\_initiated\_date"

] },

mp\_return\_initiated\_date\_2: {

$cond: [

{ $eq: ["$mp\_return\_initiated\_date\_2", ""] },

NaN,

"$mp\_return\_initiated\_date\_2"

] } } },

{ $addFields: {

final\_mp\_ret\_created\_date: {

$cond: {

if: {

$and: [

{ $ne: ["$mp\_return\_initiated\_date", ""] },

{ $ne: ["$mp\_return\_initiated\_date", NaN] },

{ $ne: ["$mp\_return\_initiated\_date", 0] }

] },

then: "$mp\_return\_initiated\_date",

else: {

$cond: {

if: {

$and: [

{

$or: [

{ $eq: ["$mp\_return\_initiated\_date", ""] },

{ $eq: ["$mp\_return\_initiated\_date", NaN] },

{ $eq: ["$mp\_return\_initiated\_date", 0] }

] },

{ $ne: ["$mp\_return\_initiated\_date\_2", ""] }

] },

then: "$mp\_return\_initiated\_date\_2",

else: "$mp\_return\_initiated\_date"

} } } } } },

{

$lookup: {

from: "Final\_Fynd\_Orders",

let: { order\_release\_id: "$order\_release\_id" },

pipeline: [

{

$match: {

$expr: { $eq: ["$fynd\_order\_release\_id", "$$order\_release\_id"] },

} },

{

$addFields: {

fynd\_sale\_bag\_id: "$bag\_id",

fynd\_sale\_shipment\_id: "$forward\_\_shipment",

fynd\_sale\_cur\_status: "$cur\_status",

fynd\_sale\_store\_code: "$store\_code"

} } ],

as: "fynd\_order\_vl"

} },

{

$addFields: {

fynd\_order\_vl: { $arrayElemAt: ["$fynd\_order\_vl", 0] }

} },

{

$addFields: {

fynd\_order\_vl: {

$cond: {

if: { $eq: [{ $type: "$fynd\_order\_vl" }, "missing"] },

then: {},

else: "$fynd\_order\_vl"

} } } },

{

$addFields: {

fynd\_sale\_bag\_id: { $ifNull: ["$fynd\_order\_vl.bag\_id", 0] },

fynd\_sale\_shipment\_id: { $ifNull: ["$fynd\_order\_vl.forward\_\_shipment", 0] },

fynd\_sale\_cur\_status: { $ifNull: ["$fynd\_order\_vl.cur\_status", 0] },

fynd\_sale\_store\_code: { $ifNull: ["$fynd\_order\_vl.store\_code", 0] },

} },

{

$set: {

fynd\_sale\_bag\_id: { $cond: [{ $eq: ["$fynd\_sale\_bag\_id", 0] }, NaN, "$fynd\_sale\_bag\_id"] },

fynd\_sale\_shipment\_id: { $cond: [{ $eq: ["$fynd\_sale\_shipment\_id", 0] }, NaN,

"$fynd\_sale\_shipment\_id"] },

fynd\_sale\_cur\_status: { $cond: [{ $eq: ["$fynd\_sale\_cur\_status", 0] }, NaN, "$fynd\_sale\_cur\_status"] },

fynd\_sale\_store\_code: { $cond: [{ $eq: ["$fynd\_sale\_store\_code", 0] }, NaN, "$fynd\_sale\_store\_code"] }

} },

{

$lookup: {

from: "Final\_Fynd\_Returns",

let: { order\_release\_id: "$order\_release\_id" },

pipeline: [

{

$match: {

$expr: { $eq: ["$fynd\_return\_release\_id", "$$order\_release\_id"] },

} },

{

$addFields: {

fynd\_return\_bag\_id: "$bag\_id",

fynd\_return\_shipment\_id: "$state\_\_shipment",

fynd\_return\_cur\_status: "$cur\_status",

fynd\_return\_store\_code: "$store\_code"

} } ],

as: "fynd\_return\_vl"

} },

{

$addFields: {

fynd\_return\_vl: { $arrayElemAt: ["$fynd\_return\_vl", 0] } } },

{

$addFields: {

fynd\_return\_vl: {

$cond: {

if: { $eq: [{ $type: "$fynd\_return\_vl" }, "missing"] },

then: {},

else: "$fynd\_return\_vl"

} } } },

{

$addFields: {

fynd\_return\_bag\_id: { $ifNull: ["$fynd\_return\_vl.bag\_id", 0] },

fynd\_return\_shipment\_id: { $ifNull: ["$fynd\_return\_vl.state\_\_shipment", 0] },

fynd\_return\_cur\_status: { $ifNull: ["$fynd\_return\_vl.cur\_status", 0] },

fynd\_return\_store\_code: { $ifNull: ["$fynd\_return\_vl.store\_code", 0] },

} },

{

$set: {

fynd\_return\_bag\_id: { $cond: [{ $eq: ["$fynd\_return\_bag\_id", 0] }, NaN, "$fynd\_return\_bag\_id"] },

fynd\_return\_shipment\_id: { $cond: [{ $eq: ["$fynd\_return\_shipment\_id", 0] }, NaN, "$fynd\_return\_shipment\_id"] },

fynd\_return\_cur\_status: { $cond: [{ $eq: ["$fynd\_return\_cur\_status", 0] }, NaN, "$fynd\_return\_cur\_status"] },

fynd\_return\_store\_code: { $cond: [{ $eq: ["$fynd\_return\_store\_code", 0] }, NaN, "$fynd\_return\_store\_code"] }

} },

{

$addFields: {

final\_bag\_id: {

$cond: {

if: {

$ne: ["$fynd\_sale\_bag\_id", NaN]

},

then: "$fynd\_sale\_bag\_id",

else: "$fynd\_return\_bag\_id"

} } } },

{

$addFields: {

final\_store\_code: {

$cond: {

if: {

$ne: ["$fynd\_sale\_store\_code", NaN]

},

then: "$fynd\_sale\_store\_code",

else: "$fynd\_return\_store\_code"

} } } },

{

$lookup: {

from: "Final\_Sap\_File",

let: { seller\_order\_id: "$seller\_order\_id" },

pipeline: [

{

$match: {

$expr: { $eq: ["$new\_sap\_order\_release\_id", "$$seller\_order\_id"] },

"department": "RETURNS"

} },

{

$addFields: {

sap\_return\_amt: { $abs: { $toDouble: "$total\_tender\_value" } }

} } ],

as: "sap\_returns\_data"

} },

{

$lookup: {

from: "Final\_Sap\_File",

let: { seller\_order\_id: "$seller\_order\_id" },

pipeline: [

{

$match: {

$expr: { $eq: ["$new\_sap\_order\_release\_id", "$$seller\_order\_id"] },

"department": "INVOICE"

} },

{

$addFields: {

sap\_sales\_amt: { $abs: { $toDouble: "$total\_tender\_value" } }

} } ],

as: "sap\_sales\_data"

} },

{

$addFields: {

sap\_sales\_amt: { $ifNull: [{ $arrayElemAt: ["$sap\_sales\_data.sap\_sales\_amt", 0] }, 0] },

sap\_return\_amt: { $ifNull: [{ $arrayElemAt: ["$sap\_returns\_data.sap\_return\_amt", 0] }, 0] }

} },

{

$project: {

\_id: 0,

mp\_release\_id: "$order\_release\_id",

mp\_master\_id: "$seller\_order\_id",

sku\_code: "$seller\_sku\_code",

sku\_id: "$sku\_id",

sales\_order\_date: "$created\_on",

shipped\_date: "$shipped\_on",

fmpu\_date: "$fmpu\_date",

inscanned\_on: "$inscanned\_on",

mp\_return\_initiated\_date: "$mp\_returns\_vl.mp\_return\_initiated",

mp\_sales\_qty: "$quantity",

mp\_sales\_amt: "$final\_amount",

order\_status\_temp: "$order\_status",

mp\_return\_qty: "$mp\_return\_qty",

mp\_return\_amt: "$mp\_return\_amt",

sap\_sales\_amt: 1,

sap\_return\_amt: 1,

return\_status: "$mp\_return\_status",

mp\_return\_qty\_2: 1,

mp\_return\_amt\_2: 1,

mp\_hub\_return\_status: 1,

mp\_return\_initiated\_date\_2: 1,

final\_mp\_return\_qty: "$final\_mp\_ret\_qty",

final\_mp\_return\_amt: "$final\_mp\_ret\_amt",

return\_created\_on: "$final\_mp\_ret\_created\_date",

mp\_return\_master\_bag\_id: "$mp\_return\_master\_bag\_id",

fynd\_sale\_bag\_id: 1,

fynd\_sale\_shipment\_id: 1,

fynd\_sale\_cur\_status: 1,

fynd\_sale\_store\_code: 1,

fynd\_return\_bag\_id: 1,

fynd\_return\_shipment\_id: 1,

fynd\_return\_cur\_status: 1,

fynd\_return\_store\_code: 1,

final\_bag\_id: 1,

final\_store\_code: 1

} },

{

$sample: { size: reservoirSize }

},

{

$group: {

\_id: null,

docs: { $push: "$$ROOT" }

} },

{

$addFields: {

sampledDocs: {

$slice: ["$docs", reservoirSize]

} } },

{

$unwind: "$sampledDocs"

},

{

$replaceRoot: { newRoot: "$sampledDocs" }

},

{$out: "RK\_Myntra\_Recon\_EXP"}

]);

**RK\_MYNTRA\_RECON\_EXP:**

CODE:

db.RK\_Myntra\_Recon\_EXP.aggregate([

{

"$addFields" : {

"order\_status" : {

"$switch" : {

"branches" : [

{

"case" : {

"$in" : [

"$order\_status\_temp",

[ "C", "DL", "RTO", "SH", "PK", "L", "WP"] },

"then" : {

"$switch" : {

"branches" : [

{

"case" : {"$eq" : ["$order\_status\_temp","C" ] },

"then" : "Completed"

},

{

"case" : {"$eq" : ["$order\_status\_temp","DL"] },

"then" : "Delivered"

},

{

"case" : {"$eq" : ["$order\_status\_temp","RTO"] },

"then" : "Return to Origin"

},

{

"case" : {"$eq" : ["$order\_status\_temp","SH"] },

"then" : "Shipped"

},

{

"case" : {"$eq" : ["$order\_status\_temp","PK"] },

"then" : "Packed"

},

{

"case" : {"$eq" : ["$order\_status\_temp","L"] },

"then" : "Lost"

},

{

"case" : {"$eq" : ["$order\_status\_temp","WP"] },

"then" : "Awaiting Pickup"}

],

"default" : "blank"

} } },

{

"case" : {"$eq" : ["$order\_status\_temp","F"] },

"then" : {

"$cond" : {

"if" : {

"$and" : [

{"$eq":["$fmpu\_date",NaN]},

{"$eq":["$inscanned\_on",NaN]},

{"$eq":["$shipped\_date",NaN]}

] },

"then" : "Failed before pickup",

"else" : "Failed after pickup",

} } } ] } } } },

{

"$set": {

"final\_mp\_return\_qty": {

"$cond": {

"if": {

"$and": [

{ "$eq": ["$order\_status", "Failed before pickup"] },

{ "$gt": ["$mp\_sales\_qty", 0] },

{ "$gt": ["$mp\_sales\_amt", 0] },

{ "$eq": ["$final\_mp\_return\_qty", 0] }

] },

"then": "$mp\_sales\_qty",

"else": "$final\_mp\_return\_qty"

} },

"final\_mp\_return\_amt": {

"$cond": {

"if": {

"$and": [

{ "$eq": ["$order\_status", "Failed before pickup"] },

{ "$gt": ["$mp\_sales\_qty", 0] },

{ "$gt": ["$mp\_sales\_amt", 0] },

{ "$eq": ["$final\_mp\_return\_amt", 0] }

] },

"then": "$mp\_sales\_amt",

"else": "$final\_mp\_return\_amt"

} } } },

{

"$set": {

"final\_mp\_return\_qty": {

"$cond": {

"if": {

"$and": [

{ "$eq": ["$fynd\_return\_cur\_status", "Credit Note Generated"] },

{ "$gt": ["$mp\_sales\_qty", 0] },

{ "$gt": ["$mp\_sales\_amt", 0] },

{ "$eq": ["$final\_mp\_return\_qty", 0] }

] },

"then": "$mp\_sales\_qty",

"else": "$final\_mp\_return\_qty"

} },

"final\_mp\_return\_amt": {

"$cond": {

"if": {

"$and": [

{ "$eq": ["$fynd\_return\_cur\_status", "Credit Note Generated"] },

{ "$gt": ["$mp\_sales\_qty", 0] },

{ "$gt": ["$mp\_sales\_amt", 0] },

{ "$eq": ["$final\_mp\_return\_amt", 0] }

] },

"then": "$mp\_sales\_amt",

"else": "$final\_mp\_return\_amt"

} } } },

{

"$addFields": {

"sales\_diff": { "$subtract": ["$mp\_sales\_amt", "$sap\_sales\_amt"] }

} },

{

"$addFields": {

"returns\_diff": { "$subtract": ["$final\_mp\_return\_amt", "$sap\_return\_amt"] }

} },

{

"$addFields": {

"net\_diff": { "$subtract": ["$sales\_diff", "$returns\_diff"] }

} },

{

$addFields: {

exception\_description\_sales: {

$cond: {

if: {

$and: [

{ $eq: ["$sap\_sales\_amt", 0] },

{ $eq: ["$sap\_return\_amt", 0] },

{ $eq: ["$order\_status", "Failed before pickup"] }

]},

then: "CANCELLED",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$sap\_sales\_amt", 0] },

{ $in: ["$order\_status", ["Packed", "Awaiting Pickup"]] }

]},

then: "UNSHIPPED",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$mp\_sales\_amt", "$sap\_sales\_amt"] },

{ $eq: ["$sales\_diff", 0] }

]},

then: "EXACT POSTED",

else: {

$cond: {

if: {

$and: [

{ $gt: ["$sales\_diff", 0] },

{ $lt: ["$sales\_diff", 101] },

{ $gt: ["$mp\_sales\_amt", 0] },

{ $gt: ["$sap\_sales\_amt", 0] }

]},

then: "ONLY VALUE SHORT POSTED",

else: {

$cond: {

if: {

$and: [

{ $lt: ["$sales\_diff", 0] },

{ $gt: ["$sales\_diff", -101] },

{ $gt: ["$mp\_sales\_amt", 0] },

{ $gt: ["$sap\_sales\_amt", 0] }

]},

then: "ONLY VALUE EXCESS POSTED",

else: {

$cond: {

if: {

$and: [

{ $gte: [{ $multiply: [-1, "$sales\_diff"] }, { $divide: ["$mp\_sales\_amt", "$mp\_sale\_qty"] }] },

{ $lt: ["$sales\_diff", 0] }

]},

then: "EXCESS POSTED",

else: {

$cond: {

if: {

$and: [

{ $lte: ["$sales\_diff", 0] },

{ $eq: ["$mp\_sales\_amt", 0] }

] },

then: "NOT REFLECTING IN MKT REPORT",

else: {

$cond: {

if: {

$and: [

{ $gte: [{ $multiply: [-1, "$sales\_diff"] }, { $divide: ["$mp\_sales\_amt", "$mp\_sale\_qty"] }] },

{ $gt: ["$sap\_sales\_amt", 0] }

] },

then: "SHORT POSTED",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$sap\_sales\_amt", 0] },

{ $gte: ["$mp\_sales\_amt", 0] }

] },

then: "NOT REFLECTING IN RRL REPORT",

else: "CHECK"

} } } } } } } } } } } } } } } } } } } },

{

$addFields: {

sales\_actionables: {

$cond: {

if: { $eq: ["$exception\_description\_sales", "EXACT POSTED"] },

then: "NO ACTION REQUIRED",

else: {

$cond: {

if: { $eq: ["$exception\_description\_sales", "UNSHIPPED"] },

then: "NO ACTION REQUIRED",

else: {

$cond: {

if: { $eq: ["$exception\_description\_sales", "CANCELLED"] },

then: "CANCELLED",

else: {

$cond: {

if: { $eq: ["$exception\_description\_sales", "SHORT POSTED"] },

then: "POST AS SALE",

else: {

$cond: {

if: { $eq: ["$exception\_description\_sales", "EXCESS POSTED"] },

then: "POST AS RETURN",

else: {

$cond: {

if: { $eq: ["$exception\_description\_sales", "ONLY VALUE SHORT POSTED"] },

then: "MARK SALE JV",

else: {

$cond: {

if: { $eq: ["$exception\_description\_sales", "ONLY VALUE EXCESS POSTED"] },

then: "MARK RETURN JV",

else: {

$cond: {

if: { $eq: ["$exception\_description\_sales", "NOT REFLECTING IN MKT REPORT"] },

then: "CHECK MKT RPRT",

else: {

$cond: {

if: { $eq: ["$exception\_description\_sales", "NOT REFLECTING IN RRL REPORT"] },

then: "POST AS SALE",

else: "CHECK"}}}}}}}}}}}}}}}}}}}},

{

$addFields: {

exception\_description\_returns: {

$cond: {

if: {

$and: [

{ $eq: ["$sap\_return\_amt", 0] },

{ $eq: ["$sap\_sales\_amt", 0] },

{ $eq: ["$exception\_description\_sales", "CANCELLED"] },

{ $gt: ["$returns\_diff", 0] }

]},

then: "CANCELLED",

else: {

$cond: {

if: {

$and: [

{ $gt: ["$returns\_diff", 0] },

{ $lt: ["$returns\_diff", 101] },

{ $gt: ["$final\_mp\_return\_amt", 0] },

{ $gt: ["$sap\_return\_amt", 0] }

]},

then: "ONLY VALUE SHORT POSTED",

else: {

$cond: {

if: {

$and: [

{ $lt: ["$returns\_diff", 0] },

{ $gt: ["$returns\_diff", -101] },

{ $gt: ["$final\_mp\_return\_amt", 0] },

{ $gt: ["$sap\_return\_amt", 0] }

]},

then: "ONLY VALUE EXCESS POSTED",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$order\_status", "Failed Before Pickup"] },

{ $eq: ["$sap\_sales\_amt", 0] },

{ $gt: ["$sap\_return\_amt", 0] },

{ $gt: ["$returns\_diff", 0] },

{ $lt: ["$returns\_diff", 100] }

]},

then: "ONLY VALUE SHORT POSTED",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$order\_status", "Failed Before Pickup"] },

{ $eq: ["$sap\_sales\_amt", 0] },

{ $gt: ["$sap\_return\_amt", 0] },

{ $lt: ["$returns\_diff", 0] },

{ $gt: ["$returns\_diff", -100] }

] },

then: "ONLY VALUE EXCESS POSTED",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$sap\_return\_amt", 0] },

{ $eq: ["$order\_status", "Cancelled"] },

{ $gt: ["$returns\_diff", 0] }

]},

then: "NOT REFLECTING IN RRL REPORT",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$sap\_return\_amt", 0] },

{ $eq: ["$order\_status", "Awaiting Pickup"] }

]},

then: "UNSHIPPED",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$final\_mp\_return\_amt", "$sap\_return\_amt"] },

{ $eq: ["$returns\_diff", 0] }

]},

then: "EXACT POSTED",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$final\_mp\_return\_amt", 0] },

{ $gt: ["$sap\_return\_amt", 0] }

]},

then: "EXCESS POSTED",

else: {

$cond: {

if: {

$and: [

{ $gte: [{ $multiply: [-1, "$returns\_diff"] }, { $divide: ["$final\_mp\_return\_amt", { $cond: [{ $eq:

["$final\_mp\_return\_qty", 0] }, 1, "$final\_mp\_return\_qty"] }] }] },

{ $lt: ["$returns\_diff", 0] }

]},

then: "EXCESS POSTED",

else: {

$cond: {

if: {

$and: [

{ $lte: ["$returns\_diff", 0] },

{ $eq: ["$final\_mp\_return\_amt", 0] }

]},

then: "NOT REFLECTING IN MKT REPORT",

else: {

$cond: {

if: {

$and: [

{ $eq: [{ $multiply: [-1, "$returns\_diff"] }, { $divide: ["$final\_mp\_return\_amt",

"$final\_mp\_return\_qty"] }] },

{ $lt: ["$returns\_diff", 0] }

]},

then: "EXCESS POSTED",

else: {

$cond: {

if: {

$and: [

{ $eq: ["$sap\_return\_amt", 0] },

{ $gte: ["$final\_mp\_return\_amt", 0] }

]},

then: "NOT REFLECTING IN RRL REPORT",

else: "CHECK"

} } } } } } } } } } } } } } } } } } } } } } } } } } } },

{

$addFields: {

returns\_actionables: {

$cond: {

if: { $eq: ["$exception\_description\_returns", "EXACT POSTED"] },

then: "NO ACTION REQUIRED",

else: {

$cond: {

if: { $eq: ["$exception\_description\_returns", "UNSHIPPED"] },

then: "NO ACTION REQUIRED",

else: {

$cond: {

if: { $eq: ["$exception\_description\_returns", "CANCELLED"] },

then: "CANCELLED",

else: {

$cond: {

if: { $eq: ["$exception\_description\_returns", "SHORT POSTED"] },

then: "POST AS RETURN",

else: {

$cond: {

if: { $eq: ["$exception\_description\_returns", "EXCESS POSTED"] },

then: "POST AS SALE",

else: {

$cond: {

if: { $eq: ["$exception\_description\_returns", "ONLY VALUE SHORT POSTED"] },

then: "MARK RETURN JV",

else: {

$cond: {

if: { $eq: ["$exception\_description\_returns", "ONLY VALUE EXCESS POSTED"] },

then: "MARK SALE JV",

else: {

$cond: {

if: { $eq: ["$exception\_description\_returns", "NOT REFLECTING IN MKT REPORT"] },

then: "CHECK MKT RPRT",

else: {

$cond: {

if: { $eq: ["$exception\_description\_returns", "NOT REFLECTING IN RRL REPORT"] },

then: "POST AS RETURN",

else: "CHECK"

}}}}}}}}}}}}}}}}}}}},

{

$addFields: {

exception\_description\_net: {

$concat: ["SALE ", "$exception\_description\_sales"," & RETURN ", "$exception\_description\_returns"

]}}},

{

$addFields: {

actionables\_net: {

$switch: {

branches: [

{ case: { $eq: ["$exception\_description\_net", "SALE EXACT POSTED & RETURN EXACT POSTED"] }, then: "NO ACTION REQUIRED" },

{ case: { $eq: ["$exception\_description\_net", "SALE ONLY VALUE EXCESS POSTED & RETURN EXACT POSTED"] }, then: "MARK RETURN JV" },

{ case: { $eq: ["$exception\_description\_net", "SALE ONLY VALUE SHORT POSTED & RETURN EXACT POSTED"] }, then: "MARK SALE JV" },

{ case: { $eq: ["$exception\_description\_net", "SALE NOT REFLECTING IN RRL REPORT & RETURN NOT REFLECTING IN RRL REPORT"] }, then: "NO ACTION REQUIRED" },

{ case: { $eq: ["$exception\_description\_net", "SALE ONLY VALUE SHORT POSTED & RETURN NOT REFLECTING IN RRL REPORT"] }, then: "MARK SALE JV & POST AS RETURN" },

{ case: { $eq: ["$exception\_description\_net", "SALE ONLY VALUE EXCESS POSTED & RETURN ONLY VALUE EXCESS POSTED"] }, then: "NO ACTION REQUIRED" },

{ case: { $eq: ["$exception\_description\_net", "SALE ONLY VALUE EXCESS POSTED & RETURN NOT REFLECTING IN RRL REPORT"] }, then: "MARK RETURN JV & POST AS RETURN" },

{ case: { $eq: ["$exception\_description\_net", "SALE NOT REFLECTING IN RRL REPORT & RETURN ONLY VALUE SHORT POSTED"] }, then: "POST AS SALE & MARK RETURN JV" },

{ case: { $eq: ["$exception\_description\_net", "SALE ONLY VALUE SHORT POSTED & RETURN ONLY VALUE SHORT POSTED"] }, then: "NO ACTION REQUIRED" },

{ case: { $eq: ["$exception\_description\_net", "SALE NOT REFLECTING IN RRL REPORT & RETURN ONLY VALUE EXCESS POSTED"] }, then: "POST AS SALE & MARK SALE JV" },

{ case: { $eq: ["$exception\_description\_net", "SALE EXCESS POSTED & RETURN NOT REFLECTING IN RRL REPORT"] }, then: "POST AS RETURN" },

{ case: { $eq: ["$exception\_description\_net", "SALE EXCESS POSTED & RETURN EXACT POSTED"] }, then: "POST AS RETURN" },

{ case: { $and: [{ $eq: ["$exception\_description\_net", "SALE EXCESS POSTED & RETURN EXCESS POSTED"] }, { $gt: ["$net\_diff", 0] }] }, then: "POST AS SALE" },

{ case: { $and: [{ $eq: ["$exception\_description\_net", "SALE EXCESS POSTED & RETURN EXCESS POSTED"] }, { $lt: ["$net\_diff", 0] }] }, then: "POST AS RETURN" },

{ case: { $and: [{ $eq: ["$exception\_description\_net", "SALE EXCESS POSTED & RETURN EXCESS POSTED"] }, { $eq: ["$net\_diff", 0] }] }, then: "NO ACTION REQUIRED" },

{ case: { $eq: ["$exception\_description\_net", "SALE NOT REFLECTING IN RRL REPORT & RETURN EXCESS POSTED"] }, then: "POST AS SALE" },

{ case: { $eq: ["$exception\_description\_net", "SALE CANCELLED & RETURN CANCELLED"] }, then: "CANCELLED" },

{ case: { $eq: ["$exception\_description\_net", "SALE EXACT POSTED & RETURN ONLY VALUE EXCESS POSTED"] }, then: "MARK SALE JV" },

{ case: { $eq: ["$exception\_description\_net", "SALE EXACT POSTED & RETURN EXCESS POSTED"] }, then: "POST AS SALE" },

{ case: { $eq: ["$exception\_description\_net", "SALE EXCESS POSTED & RETURN ONLY VALUE EXCESS POSTED"] }, then: "POST AS RETURN & MARK SALE JV" },

{ case: { $and: [{ $eq: ["$exception\_description\_net", "SALE ONLY VALUE SHORT POSTED & RETURN ONLY VALUE EXCESS POSTED"] }, { $eq: ["$net\_diff", 0] }] }, then: "NO ACTION REQUIRED" },

{ case: { $and: [{ $eq: ["$exception\_description\_net", "SALE ONLY VALUE EXCESS POSTED & RETURN EXCESS POSTED"] }, { $gt: ["$net\_diff", 0] }] }, then: "POST AS SALE" },

{ case: { $eq: ["$exception\_description\_net", "SALE EXACT POSTED & RETURN ONLY VALUE SHORT POSTED"] }, then: "MARK RETURN JV" },

{case: { $eq: ["$exception\_description\_net", "SALE EXACT POSTED & RETURN NOT REFLECTING IN RRL REPORT"] }, then: "POST AS RETURN" },

{case: { $eq: ["$exception\_description\_net", "SALE NOT REFLECTING IN RRL REPORT & RETURN EXACT POSTED"] }, then: "POST AS SALE" },

],

default: "CHECK"

} } } },

{ "$out": "RK\_Myntra\_Recon\_Report" }

]);