db.createCollection('customers', {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "customer\_numb" ],

properties: {

customer\_numb: {

bsonType: "int",

description: "Customer identifier"

},

customer\_first\_name: {

bsonType: "string",

description: "First Name"

},

customer\_last\_name: {

bsonType: "string",

description: "Last Name"

},

customer\_street: {

bsonType: "string",

description: "Street"

},

customer\_city: {

bsonType: "string",

description: "City"

},

customer\_state: {

bsonType: "string",

description: "State"

},

customer\_zip: {

bsonType: "int",

description: "Zipcode"

},

customer\_phone: {

bsonType: "int",

description: "Phone",

minimum : 100000000,

maximum : 999999999

},

customer\_orders: {

bsonType: ["array"],

uniqueItems: true,

description: "Customer Orders",

items: {

bsonType: ["object"],

required: ["order\_number", "order\_date"],

properties: {

order\_number: {

bsonType: "int",

description: "Order Number"

},

customer\_numb: {

bsonType: "int",

description: "Customer Number"

},

order\_date: {

bsonType: "date",

description: "Order Date"

},

cc\_number: {

bsonType: "int",

description: "CC Number"

},

cc\_exp\_date:{

bsonType: "int",

description: "CC Date",

},

order\_complete:{

bsonType: "bool",

description: "Order completion status",

},

pickup\_or\_ship:{

enum: ['pickup', 'ship'],

description: "Pickup or Ship",

}

}

}

}

}

}

}

});

// Insert 2 customers with 3 orders each (sub-document orders)

db.customers.insertMany(

[

{

\_id: 1,

customer\_numb: 1,

customer\_first\_name: 'Sumeet',

customer\_last\_name: 'Duddagi',

customer\_street: '1026 W 24 Street',

customer\_city: 'Los Angeles',

customer\_state: 'California',

customer\_zip: 90007,

customer\_orders: [

{

order\_number: 101,

customer\_numb: 1,

order\_date: new Date('2020-08-10'),

},

{

order\_number: 102,

customer\_numb: 1,

order\_date: new Date('2021-09-21'),

},

{

order\_number: 103,

customer\_numb: 1,

order\_date: new Date('2022-01-17'),

},

]

},

{

\_id: 2,

customer\_numb: 2,

customer\_first\_name: 'Rohit',

customer\_last\_name: 'Duddagi',

customer\_street: '1027 W 24 Street',

customer\_city: 'London',

customer\_state: 'United Kingdom',

customer\_zip: 5211,

customer\_orders: [

{

order\_number: 201,

customer\_numb: 2,

order\_date: new Date('2020-10-10'),

},

{

order\_number: 202,

customer\_numb: 2,

order\_date: new Date('2022-01-02'),

},

{

order\_number: 203,

customer\_numb: 2,

order\_date: new Date('2021-12-17'),

},

]

}

]

)

// create orders collection with customer number required

db.createCollection('orders', {

validator: {

$jsonSchema: {

bsonType: "object",

properties: {

order\_number: {

bsonType: "int",

description: "Order Number"

},

customer\_numb: {

bsonType: "int",

description: "Customer Number"

},

order\_date: {

bsonType: "date",

description: "Order Date"

},

cc\_number: {

bsonType: "number",

description: "CC Number"

},

cc\_exp\_date:{

bsonType: "string",

description: "CC Date",

},

order\_complete:{

bsonType: "bool",

description: "Order completion status",

},

pickup\_or\_ship:{

enum: ['pickup', 'ship', null],

description: "Pickup or Ship",

}

}

}

}

})

// Insert new customers and orders separately in their own collections

db.customers.insertMany(

[

{

\_id: 3,

customer\_numb: 3,

customer\_first\_name: 'Shamanth',

customer\_last\_name: 'Prabhu',

customer\_street: '4220 President G Bush',

customer\_city: 'Dallas',

customer\_state: 'Texas',

customer\_zip: 75216,

},

{

\_id: 4,

customer\_numb: 4,

customer\_first\_name: 'Shylesh',

customer\_last\_name: 'Pala',

customer\_street: '2020 W. 91th Street',

customer\_city: 'hartford',

customer\_state: 'Connecticut',

customer\_zip: 06101,

}

])

db.orders.insertMany(

[

{

order\_number: 301,

customer\_numb: 3,

order\_date: new Date('2021-05-25'),

cc\_number: 12345678,

cc\_exp\_date: '04/26',

order\_complete: false,

pickup\_or\_ship: 'pickup',

},

{

order\_number: 302,

customer\_numb: 3,

order\_date: new Date('2020-02-03'),

cc\_number: 12345678,

cc\_exp\_date: '04/26',

order\_complete: true,

pickup\_or\_ship: 'ship',

},

{

order\_number: 303,

customer\_numb: 3,

order\_date: new Date('2029-12-10'),

cc\_number: 12345678,

cc\_exp\_date: '04/26',

order\_complete: true,

pickup\_or\_ship: 'pickup',

},

{

order\_number: 304,

customer\_numb: 4,

order\_date: new Date('2022-01-17'),

cc\_number: 12345678,

cc\_exp\_date: '04/26',

order\_complete: false,

pickup\_or\_ship: 'ship',

},

{

order\_number: 305,

customer\_numb: 4,

order\_date: new Date('2021-12-29'),

cc\_number: 12345678,

cc\_exp\_date: '04/26',

order\_complete: false,

pickup\_or\_ship: 'pickup',

},

{

order\_number: 306,

customer\_numb: 4,

order\_date: new Date('2021-11-11'),

cc\_number: 12345678,

cc\_exp\_date: '04/26',

order\_complete: false,

pickup\_or\_ship: 'ship',

}

]

)

// aggregating both collections in a single query

Print(db.customers.aggregate( [

{

$lookup: {

from: 'orders',

localField: 'customer\_numb',

foreignField: 'customer\_numb',

as: 'orders'

}

}

]))