HW2

Jinglei “Lesly” Liu

**Q1**

**Embeded:**

db.createCollection("customer", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "customer\_numb", "customer\_first\_name", "customer\_last\_name", "customer\_street", "customer\_city", "customer\_state", "customer\_zip", "customer\_phone", "order" ],

properties: {

customer\_numb: {

bsonType: "int",

description: "must be an integer and is required"

},

customer\_first\_name: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_last\_name: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_street: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_city: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_state: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_zip: {

bsonType: "int",

description: "must be an integer and is required"

},

customer\_phone: {

bsonType: "long",

description: "must be an integer and is required"

},

order:{

bsonType: "array",

required: [ "order\_numb", "customer\_numb", "order\_date", "credit\_card\_numb", "credit\_card\_exp\_date", "order\_complete?", "pickup\_or\_ship?" ],

properties: {

order\_numb: {

bsonType: "int",

description: "must be an integer and is required"

},

customer\_numb: {

bsonType: "int",

description: "must be an integer and is required"

},

order\_date: {

bsonType: "date",

description: "must be a date and is required"

},

credit\_card\_numb: {

bsonType: "long",

description: "must be an integer and is required"

},

credit\_card\_exp\_date: {

bsonType: "date",

description: "must be a date and is required"

},

"order\_complete?": {

bsonType: "string",

description: "must be a string and is required"

},

"pickup\_or\_ship?": {

bsonType: "string",

description: "must be a string and is required"

}

}

}

}

}

}

})

**insert 2 customers and 3 orders for each customer:**

db.customer.insertMany([

{

\_id: 121,

customer\_numb: NumberInt(121),

customer\_first\_name: "Jim",

customer\_last\_name: "L",

customer\_street: "Flower",

customer\_city: "Carlisle",

customer\_state: "PA",

customer\_zip: NumberInt(17013),

customer\_phone: NumberLong("2111111234"),

order: [{

\_id: 10,

order\_numb: NumberInt(10),

customer\_numb: NumberInt(121),

order\_date: ISODate("2020-12-10"),

credit\_card\_numb: NumberLong("1111111799987855"),

credit\_card\_exp\_date: ISODate("2026-10-10"),

"order\_complete?": "complete",

"pickup\_or\_ship?": "ship"

},

{

\_id: 31,

order\_numb: NumberInt(31),

customer\_numb: NumberInt(121),

order\_date: ISODate("2021-01-11"),

credit\_card\_numb: NumberLong("1111111799987855"),

credit\_card\_exp\_date: ISODate("2026-10-10"),

"order\_complete?": "incomplete",

"pickup\_or\_ship?": "ship"

},

{

\_id: 54,

order\_numb: NumberInt(54),

customer\_numb: NumberInt(121),

order\_date: ISODate("2021-06-11"),

credit\_card\_numb: NumberLong("1111111799987855"),

credit\_card\_exp\_date: ISODate("2026-10-10"),

"order\_complete?": "complete",

"pickup\_or\_ship?": "pickup"

}

]},

{

\_id: 989,

customer\_numb: NumberInt(989),

customer\_first\_name: "Tim",

customer\_last\_name: "J",

customer\_street: "Sun",

customer\_city: "San Jose",

customer\_state: "CA",

customer\_zip: NumberInt(91078),

customer\_phone: NumberLong("4516117399"),

order: [

{

\_id: 03,

order\_numb: NumberInt(03),

customer\_numb: NumberInt(989),

order\_date: ISODate("2019-04-10"),

credit\_card\_numb: NumberLong("5113111979982855"),

credit\_card\_exp\_date: ISODate("2024-08-10"),

"order\_complete?": "incomplete",

"pickup\_or\_ship?": "ship"

},

{

\_id: 65,

order\_numb: NumberInt(65),

customer\_numb: NumberInt(989),

order\_date: ISODate("2020-11-11"),

credit\_card\_numb: NumberLong("5113111979982855"),

credit\_card\_exp\_date: ISODate("2024-08-10"),

"order\_complete?": "incomplete",

"pickup\_or\_ship?": "pickup"

},

{

\_id: 73,

order\_numb: NumberInt(73),

customer\_numb: NumberInt(989),

order\_date: ISODate("2021-08-11"),

credit\_card\_numb: NumberLong("5113111979982855"),

credit\_card\_exp\_date: ISODate("2024-08-10"),

"order\_complete?": "complete",

"pickup\_or\_ship?": "ship"

}

]

}

])

**Link:**

**customer collection:**

db.createCollection("customer", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "customer\_numb", "customer\_first\_name", "customer\_last\_name", "customer\_street", "customer\_city", "customer\_state", "customer\_zip", "customer\_phone" ],

properties: {

customer\_numb: {

bsonType: "int",

description: "must be an integer and is required"

},

customer\_first\_name: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_last\_name: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_street: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_city: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_state: {

bsonType: "string",

description: "must be a string and is required"

},

customer\_zip: {

bsonType: "int",

description: "must be an integer and is required"

},

customer\_phone: {

bsonType: "long",

description: "must be an integer and is required"

}

}

}

}

} )

**Order collection:**

db.createCollection("order", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "order\_numb", "customer\_numb", "order\_date", "credit\_card\_numb", "credit\_card\_exp\_date", "order\_complete?", "pickup\_or\_ship?" ],

properties: {

order\_numb: {

bsonType: "int",

description: "must be an integer and is required"

},

customer\_numb: {

bsonType: "int",

description: "must be an integer and is required"

},

order\_date: {

bsonType: "date",

description: "must be a date and is required"

},

credit\_card\_numb: {

bsonType: "long",

description: "must be an integer and is required"

},

credit\_card\_exp\_date: {

bsonType: "date",

description: "must be a date and is required"

},

"order\_complete?": {

bsonType: "string",

description: "must be a string and is required"

},

"pickup\_or\_ship?": {

bsonType: "string",

description: "must be a string and is required"

}

}

}

}

} )

**insert 2 customers:**

db.customer.insertMany([

{

\_id: 121,

customer\_numb: NumberInt(121),

customer\_first\_name: "Jim",

customer\_last\_name: "L",

customer\_street: "Flower",

customer\_city: "Carlisle",

customer\_state: "PA",

customer\_zip: NumberInt(17013),

customer\_phone: NumberLong("2111111234")

} ,

{

\_id: 989,

customer\_numb: NumberInt(989),

customer\_first\_name: "Tim",

customer\_last\_name: "J",

customer\_street: "Sun",

customer\_city: "San Jose",

customer\_state: "CA",

customer\_zip: NumberInt(91078),

customer\_phone: NumberLong("4516117399")

}

])

**insert 3 orders for each customer:**

db.order.insertMany([

{

\_id: 10,

order\_numb: NumberInt(10),

customer\_numb: NumberInt(121),

order\_date: ISODate("2020-12-10"),

credit\_card\_numb: NumberLong("1111111799987855"),

credit\_card\_exp\_date: ISODate("2026-10-10"),

"order\_complete?": "complete",

"pickup\_or\_ship?": "ship"

},

{

\_id: 31,

order\_numb: NumberInt(31),

customer\_numb: NumberInt(121),

order\_date: ISODate("2021-01-11"),

credit\_card\_numb: NumberLong("1111111799987855"),

credit\_card\_exp\_date: ISODate("2026-10-10"),

"order\_complete?": "incomplete",

"pickup\_or\_ship?": "ship"

},

{

\_id: 54,

order\_numb: NumberInt(54),

customer\_numb: NumberInt(121),

order\_date: ISODate("2021-06-11"),

credit\_card\_numb: NumberLong("1111111799987855"),

credit\_card\_exp\_date: ISODate("2026-10-10"),

"order\_complete?": "complete",

"pickup\_or\_ship?": "pickup"

},

{

\_id: 03,

order\_numb: NumberInt(03),

customer\_numb: NumberInt(989),

order\_date: ISODate("2019-04-10"),

credit\_card\_numb: NumberLong("5113111979982855"),

credit\_card\_exp\_date: ISODate("2024-08-10"),

"order\_complete?": "incomplete",

"pickup\_or\_ship?": "ship"

},

{

\_id: 65,

order\_numb: NumberInt(65),

customer\_numb: NumberInt(989),

order\_date: ISODate("2020-11-11"),

credit\_card\_numb: NumberLong("5113111979982855"),

credit\_card\_exp\_date: ISODate("2024-08-10"),

"order\_complete?": "incomplete",

"pickup\_or\_ship?": "pickup"

},

{

\_id: 73,

order\_numb: NumberInt(73),

customer\_numb: NumberInt(989),

order\_date: ISODate("2021-08-11"),

credit\_card\_numb: NumberLong("5113111979982855"),

credit\_card\_exp\_date: ISODate("2024-08-10"),

"order\_complete?": "complete",

"pickup\_or\_ship?": "ship"

}

])

**Q2**

**1). Build collection:**

db.createCollection("student", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "student\_name", "address", "phone\_number"],

properties: {

student\_name: {

bsonType: "string",

description: "must be a string and is required"

},

address: {

bsonType: "object",

required: [ "street", "city", "state", "zip" ],

properties: {

street: {

bsonType: "string",

description: "must be a and is required"

},

city: {

bsonType: "string",

"description": "must be a string and is required"

},

state: {

bsonType: "string",

"description": "must be a string and is required"

},

zip: {

bsonType: "int",

"description": "must be an integer and is required"

}

}

},

phone\_number: {

bsonType: "long",

"description": "must be an integer and is required"

}

}

}

}

})

**4). Insert 10 records:**

db.student.insertMany([

{

student\_name: "Jim Y",

address: {

street: "Hoover",

city: "LA",

state: "CA",

zip: NumberInt(90037)

},

phone\_number: NumberLong("3145791300")

},

{

student\_name: "Song J",

address: {

street: "Denver",

city: "LA",

state: "CA",

zip: NumberInt(90030)

},

phone\_number: NumberLong("3909791380")

},

{

student\_name: "Kim S",

address: {

street: "Field",

city: "LA",

state: "CA",

zip: NumberInt(90017)

},

phone\_number: NumberLong("2149708300")

},

{

student\_name: "Zoe Y",

address: {

street: "Calm",

city: "Carlisle",

state: "PA",

zip: NumberInt(17013)

},

phone\_number: NumberLong("5374791060")

},

{

student\_name: "Nam L",

address: {

street: "Flower",

city: "LA",

state: "CA",

zip: NumberInt(90023)

},

phone\_number: NumberLong("3940790810")

},

{

student\_name: "Peet N",

address: {

street: "College",

city: "Carlisle",

state: "PA",

zip: NumberInt(17017)

},

phone\_number: NumberLong("2740791342")

},

{

student\_name: "Kris Y",

address: {

street: "Park",

city: "LA",

state: "CA",

zip: NumberInt(90106)

},

phone\_number: NumberLong("4723791070")

},

{

student\_name: "Mon Y",

address: {

street: "Unit",

city: "SF",

state: "CA",

zip: NumberInt(91827)

},

phone\_number: NumberLong("3142094355")

},

{

student\_name: "Ern S",

address: {

street: "Down",

city: "LA",

state: "CA",

zip: NumberInt(90106)

},

phone\_number: NumberLong("5374791060")

},

{

student\_name: "Jay Y",

address: {

street: "Luke",

city: "A",

state: "CA",

zip: NumberInt(90037)

},

phone\_number: NumberLong("3167898177")

}

])

5). I would recommend setting zip code and phone number as indexes because combing these two will make sure the uniqueness to each student. Adding two indexes will ensure that we can find the right student, avoiding situations such as some students have the same zip code or accidentally the same phone number. Uniqueness would be beneficial for searching and result in high selectivity.

**6). Create indexes**

db.student.createIndex( { "address.zip": 1} )

db.student.createIndex( { "phone\_number": 1 } )

**7). Run the search queries**

db.student.find({ "address.zip": 90106, "phone\_number": 5374791060})

**8) Prove indexes are used.**

db.student.find({ "address.zip": 90106, "phone\_number": 5374791060}).explain("executionStats")

A picture containing text

Description automatically generated