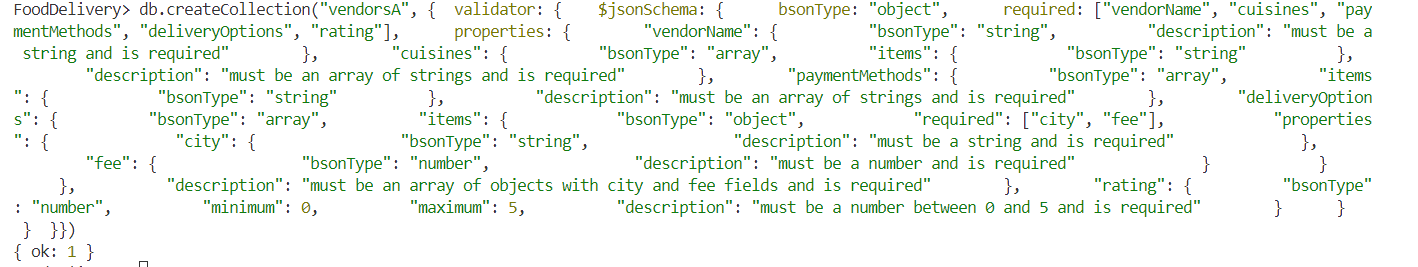
**Classroom Activity: Write multiple examples of Evaluation operators in MongoDB collection.**

**Code: element $jsonSchema operator in MongoDB.**

FoodDelivery> db.createCollection("vendorsA", { validator: { $jsonSchema: { bsonType: "object", required: ["vendorName", "cuisines", "paymentMethods", "deliveryOptions", "rating"], properties: { "vendorName": { "bsonType": "string", "description": "must be a string and is required" }, "cuisines": { "bsonType": "array", "items": { "bsonType": "string" }, "description": "must be an array of strings and is required" }, "paymentMethods": { "bsonType": "array", "items": { "bsonType": "string" }, "description": "must be an array of strings and is required" }, "deliveryOptions": { "bsonType": "array", "items": { "bsonType": "object", "required": ["city", "fee"], "properties": { "city": { "bsonType": "string", "description": "must be a string and is required" }, "fee": { "bsonType": "number", "description": "must be a number and is required" } } }, "description": "must be an array of objects with city and fee fields and is required" }, "rating": { "bsonType": "number", "minimum": 0, "maximum": 5, "description": "must be a number between 0 and 5 and is required" } } } }})

**OUTPUT:**



**Code: element $mod operator in MongoDB.**

FoodDelivery> // Find all documents where any delivery fee is divisible by 2

FoodDelivery> db.vendors.find({ "deliveryOptions.fee": { $mod: [2, 0] } })

**OUTPUT:**

****

**Code: element $regex operator in MongoDB.**

|  |
| --- |
| FoodDelivery> // Find all documents where the vendorName contains the substring "Foods"  FoodDelivery> db.vendors.find({ "vendorName": { $regex: /Foods/, $options: 'i' } }) |

**Output:**

****

**Code: element $text operator in MongoDB.**

|  |
| --- |
| FoodDelivery> // Create a text index on the vendorName field  FoodDelivery> db.vendors.createIndex({ "vendorName": "text" })  vendorName\_text  FoodDelivery> // Perform a text search query to find documents where vendorName contains "Healthy"  FoodDelivery> db.vendors.find({ $text: { $search: "Healthy" } }) |

**Output:**

****

**Code: element $text operator in MongoDB.**

|  |
| --- |
| FoodDelivery> // Find all documents where any delivery option has a fee less than 4  FoodDelivery> db.vendors.find({ $where: function() { return this.deliveryOptions.some(option => option.fee < 4); }}) |

**Output:**

****

**Execution Steps:**

1. Install MongoDB compass and MongoDB shell
2. Create a new connection in MongoDB Compass
3. Create a new Database eg.)Food Delivery
4. Go to Visual Studio Code
5. Install the MongoDB for VS code extension
6. Click on the MongoDB icon on the navigation pane
7. Right click on the localhost:27017 connection
8. Select Launch MongoDB shell
9. Finally execute the above code.