

IE6700 Data Management for Analytics

Milestone: Implementation in NoSQL

Group 5

Student 1: Aliya LNU

Student 2: Nixon Lobo

6315389069 (Tel of Student 1)

8573209201 (Tel of Student 2)

Lnu.ali@northeastern.edu

Lobo.ni@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Aliya LNU

Signature of Student 2: Nixon Lobo

Submission Date: 04/14/2024

NoSQL implementation

Query:

Find all orders with a specific order status (e.g., "pending"):

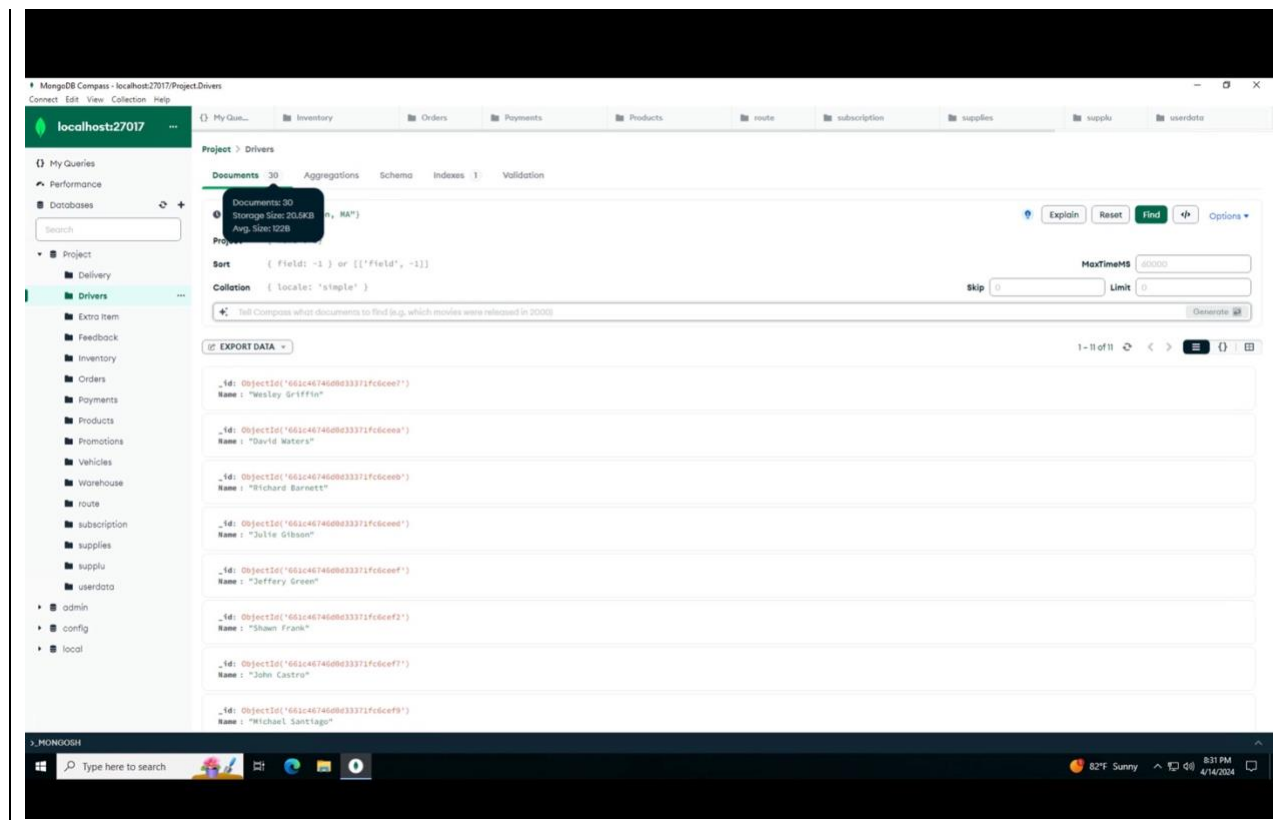
```
db.OORDERS.find({ orderStatus: "processing", "shipped" })
```

The screenshot shows the MongoDB Compass application running on localhost:27017. The database is 'Project' and the collection is 'Orders'. A query is entered in the search bar: `{ "orderStatus": { "$in": ["processing", "shipped"] } }`. The results are displayed in a table with 4 rows. Each row shows the document structure with fields like `_id`, `OrderID`, `ProductID`, `Amount`, `OrderStatus`, `OrderDate`, `RouteID`, `WarehouseID`, and `PromotionID`.

Document
<pre>{ "_id": "603c73f104204990a3806f", "OrderID": "000001", "ProductID": "P1004", "Amount": 5, "OrderStatus": "shipped", "OrderDate": "2023-10-19T00:00:00.000+00:00", "RouteID": "R0110", "WarehouseID": "W014", "PromotionID": "P003053" }</pre>
<pre>{ "_id": "603c73f104204990a38060", "OrderID": "000002", "ProductID": "P1002", "Amount": 2, "OrderStatus": "processing", "OrderDate": "2023-05-15T00:00:00.000+00:00", "RouteID": "R0020", "WarehouseID": "W011", "PromotionID": "P002920" }</pre>
<pre>{ "_id": "603c73f104204990a38061", "OrderID": "000003", "ProductID": "P1002", "Amount": 6, "OrderStatus": "shipped", "OrderDate": "2023-08-03T00:00:00.000+00:00", "RouteID": "R0040", "WarehouseID": "W020", "PromotionID": "P003052" }</pre>
<pre>{ "_id": "603c73f104204990a38062", "OrderID": "000004", "ProductID": "P1005", "Amount": 3, "OrderStatus": "processing" }</pre>

Find all Drivers that are Currently in Boston:

```
db.drivers.find({ "City": "Boston, Ma" })
```



Find the total quantity of a specific product in stock across all warehouses:

```
db.inventory.aggregate([
  {
    $match: {
      ProductID: "P44725"
    }
  },
  {
    $group: {
      _id: "$ProductID",
      totalQuantity: {
        $sum: "$Amount"
      }
    }
  }
])
```

}

])

The screenshot shows the MongoDB Compass interface. The left sidebar lists the following collections: Project, Delivery, Drivers, Extra item, Feedback, Inventory (selected), Orders, Payments, Products, Promotions, Vehicles, Warehouse, route, subscription, supplies, supply, and userdata. The main panel displays the 'Inventory' collection with a query editor. The query is: `{ 'field': 'value' }`. Below the query editor, a list of documents is shown, each containing an '_id' and a 'ProductID'. The documents are:

- `{ "_id": "601c73e5d426499d438114", "ProductID": "P44275" }`
- `{ "_id": "601c73e5d426499d438114", "ProductID": "P44275" }`
- `{ "_id": "601c73e5d426499d438114", "ProductID": "P44275" }`
- `{ "_id": "601c73e5d426499d438114", "ProductID": "P44275" }`
- `{ "_id": "601c73e5d426499d438114", "ProductID": "P44275" }`
- `{ "_id": "601c73e5d426499d438114", "ProductID": "P44275" }`
- `{ "_id": "601c73e5d426499d438114", "ProductID": "P44275" }`
- `{ "_id": "601c73e5d426499d438114", "ProductID": "P44275" }`
- `{ "_id": "601c73e5d426499d438114", "ProductID": "P44275" }`

Find the products that cost less than \$50

```
db.products.find({Price: {$lt: 50}})
```

MongoDB Compass - localhost:27017/ProjectPayments

ConnectEditViewCollectionHelp

localhost:27017

My QueriesPerformanceDatabasesProjectDatabaseInventoryOrdersPaymentsProductsrouteSubscriptionSuppliesSuppluUserdata

My QueriesPerformanceDatabasesProjectDatabaseInventoryOrdersPaymentsProductsrouteSubscriptionSuppliesSuppluUserdata

ProjectDatabase

ProjectDatabaseInventoryOrdersPaymentsProductsrouteSubscriptionSuppliesSuppluUserdata

Project > Payments

Documents200AggregationsSchemaIndexes1Validation

Q{"amount":{"\$lt":50}}

Project{"amount":1}

Sort{field:-1} or [{"field",-1}]

Collection{local:"staple"}

Find

MaxTimeMS60000

Skip0Limit0

Generate

EXPORT DATA

1 - 20 of 25

_id:ObjectId("661c74881d426499dc38f6ce")
amount:10.24

_id:ObjectId("661c74881d426499dc38f6d2")
amount:11.45

_id:ObjectId("661c74881d426499dc38ff9b")
amount:41.13

_id:ObjectId("661c74881d426499dc38ff6e")
amount:25.45

_id:ObjectId("661c74881d426499dc31006c")
amount:29.09

_id:ObjectId("661c74881d426499dc31014c")
amount:41.13

_id:ObjectId("661c74881d426499dc31017f")
amount:16.5

_id:ObjectId("661c74881d426499dc31029c")
amount:43.3