# MongoDB Functions for Data Analysts

## 1. Basic CRUD Operations

### 1.1. Insert Documents

* Insert a Single Document:

db.stocks.insertOne({  
 "symbol": "AAPL",  
 "price": 175.50,  
 "volume": 1000000,  
 "date": ISODate("2024-03-27")  
})

* Insert Multiple Documents:

db.stocks.insertMany([  
 { "symbol": "GOOG", "price": 2800, "volume": 500000, "date": ISODate("2024-03-27") },  
 { "symbol": "MSFT", "price": 310, "volume": 700000, "date": ISODate("2024-03-27") }  
])

### 1.2. Read (Retrieve) Data

* Retrieve All Documents:

db.stocks.find()

* Retrieve Documents with a Filter:

db.stocks.find({ "symbol": "AAPL" })

* Retrieve Specific Fields:

db.stocks.find({ "symbol": "AAPL" }, { "price": 1, "volume": 1, "\_id": 0 })

* Retrieve Sorted Data (Descending Order):

db.stocks.find().sort({ "price": -1 })

* Limit the Number of Results:

db.stocks.find().limit(5)

### 1.3. Update Documents

* Update a Single Document:

db.stocks.updateOne(  
 { "symbol": "AAPL" },  
 { $set: { "price": 180.00 } }  
)

* Update Multiple Documents:

db.stocks.updateMany(  
 { "symbol": "GOOG" },  
 { $set: { "price": 2850 } }  
)

### 1.4. Delete Documents

* Delete a Single Document:

db.stocks.deleteOne({ "symbol": "AAPL" })

* Delete Multiple Documents:

db.stocks.deleteMany({ "symbol": "GOOG" })

## 2. Aggregation & Data Analysis

### 2.1. Aggregation Pipeline

* Group by Symbol and Calculate Total Volume:

db.stocks.aggregate([  
 { $group: { \_id: "$symbol", totalVolume: { $sum: "$volume" } } }  
])

* Calculate Average Stock Price:

db.stocks.aggregate([  
 { $group: { \_id: null, avgPrice: { $avg: "$price" } } }  
])

* Filter and Then Group:

db.stocks.aggregate([  
 { $match: { "price": { $gt: 200 } } },  
 { $group: { \_id: "$symbol", avgPrice: { $avg: "$price" } } }  
])

## 3. Indexing & Performance Optimization

### 3.1. Create Index

* Create an Index on the 'symbol' Field:

db.stocks.createIndex({ "symbol": 1 })

* Create a Compound Index:

db.stocks.createIndex({ "symbol": 1, "price": -1 })

* View All Indexes:

db.stocks.getIndexes()

* Remove an Index:

db.stocks.dropIndex("symbol\_1")