Name: Omkar.U.Hulawale

Class: TE-A

Roll no: 13165

Batch: A3

Aim : Implement Map reduce operation with suitable example using MongoDB.

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\hulaw> mongosh

Current Mongosh Log ID: 671004523ff533183186b01c

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName= mongosh+2.3.2

Using MongoDB: 8.0.1

Using Mongosh: 2.3.2

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

  The server generated these startup warnings when booting

  2024-10-16T20:29:40.756+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

------

test> show dbs

admin   40.00 KiB

bill   152.00 KiB

book   40.00 KiB

config   108.00 KiB

customer 108.00 KiB

local   40.00 KiB

test> use bill

switched to db bill

*Insert Data:

bill> db.pay.insertMany([

...   {Cust_ID: "A123", Product: "Milk", Amount: 40, Status: "P"},

...   {Cust_ID: "A123", Product: "Parle_G", Amount: 50, Status: "NP"},

...   {Cust_ID: "A123", Product: "Lays Chips", Amount: 40, Status: "P"},

...   {Cust_ID: "B123", Product: "Mentos", Amount: 10, Status: "P"},

...   {Cust_ID: "B123", Product: "Maggie", Amount: 60, Status: "NP"}

...]);

{

  acknowledged: true,

  insertedIds: {

   '0': ObjectId('6710045c3ff533183186b01d'),

   '1': ObjectId('6710045c3ff533183186b01e'),

   '2': ObjectId('6710045c3ff533183186b01f'),

   '3': ObjectId('6710045c3ff533183186b020'),

   '4': ObjectId('6710045c3ff533183186b021')

  }

}

*Aggregation Pipeline:

bill> db.pay.aggregate([

...   { $match: { Status: "NP" } }, // Filter documents with Status "NP"

...   { $group: { _id: "$Cust_ID", totalAmount: { $sum: "$Amount" } } } // Group by Cust_ID and sum amounts

...]);

[

  { _id: 'B123', totalAmount: 240 },

{ _id: 'A123', totalAmount: 200 }

]

*Store Results (Optional):

bill> db.pay.aggregate([

...   { $match: { Status: "NP" } },

...   { $group: { _id: "$Cust_ID", totalAmount: { $sum: "$Amount" } } },

...   { $out: "Bill_Amount" } // Output the results to the Bill_Amount collection

...]);

*Retrieve Results:

bill> db.Bill_Amount.find().pretty(); // If you used the $out stage

[

  { _id: 'A123', totalAmount: 200 },

  { _id: 'B123', totalAmount: 240 }

]