NoSQL - Supermarket by Pranil Gurung

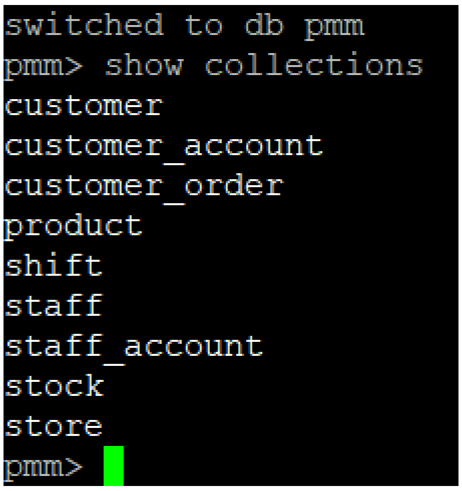
Database solution: Develop a relational database solution in NoSQL for PMM Grocery Supermarket to store all details and every element within the organisation.

Table of Contents

NoSQL Database Overview:

```
connect to database
use pmm
```

```
// staff Collection
db.createCollection("staff")

// staff account Collection
db.createCollection("staff_account")

// Products Collection
db.createCollection("product")

// Customer Collection
db.createCollection("customer")

// Customer account Collection
db.createCollection("customer_account")

// customer order Collection
db.createCollection("customer_order")

// store Collection
db.createCollection("store")

// stock Collection
db.createCollection("stock")

// shift Collection
db.createCollection("shift")
```

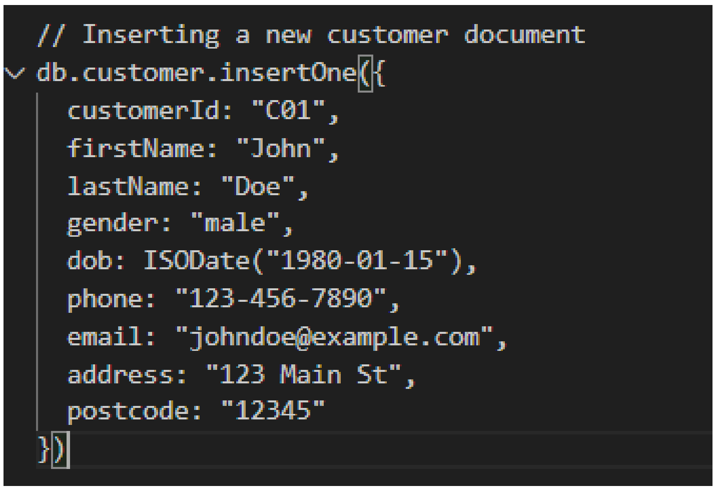


```
switched to db pmm
pmm> show collections
customer
customer_account
customer_order
product
shift
staff
staff_account
stock
store
pmm>
```

Task T1: Database Development

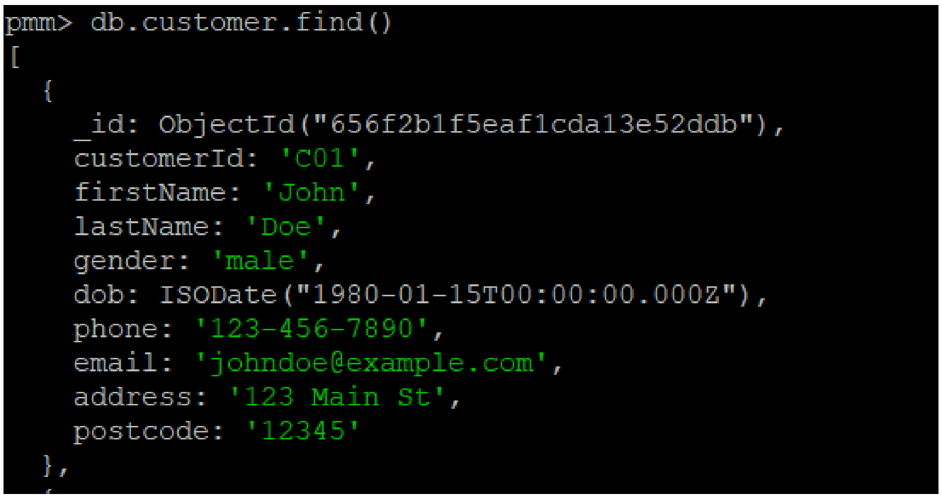
Customer document:

Inserts:

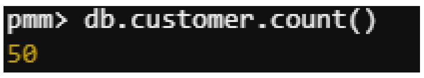


```
// Inserting a new customer document
db.customer.insertOne({
    customerId: "C01",
    firstName: "John",
    lastName: "Doe",
    gender: "male",
    dob: ISODate("1980-01-15"),
    phone: "123-456-7890",
    email: "johndoe@example.com",
    address: "123 Main St",
    postcode: "12345"
})
```

Screenshot:



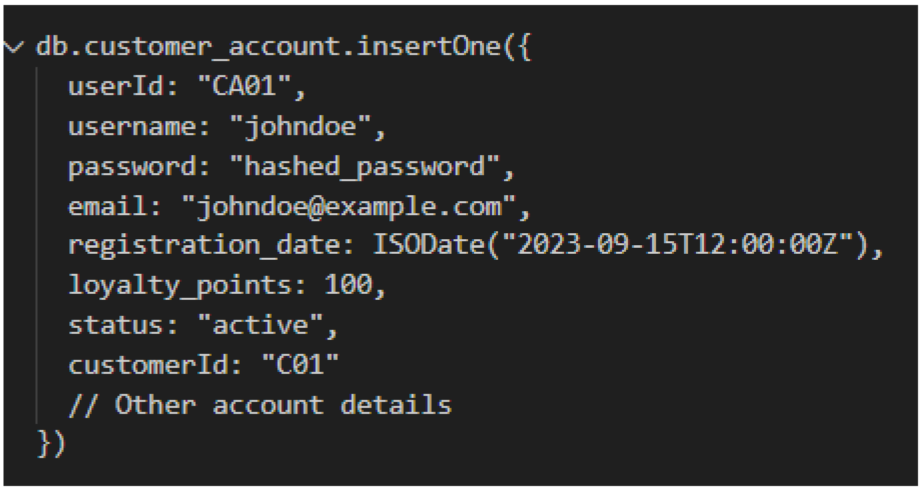
```
pmm> db.customer.find()
[
  {
    _id: ObjectId("656f2b1f5eaf1cda13e52ddb"),
    customerId: 'C01',
    firstName: 'John',
    lastName: 'Doe',
    gender: 'male',
    dob: ISODate("1980-01-15T00:00:00.000Z"),
    phone: '123-456-7890',
    email: 'johndoe@example.com',
    address: '123 Main St',
    postcode: '12345'
  },
```



```
pmm> db.customer.count()
50
```

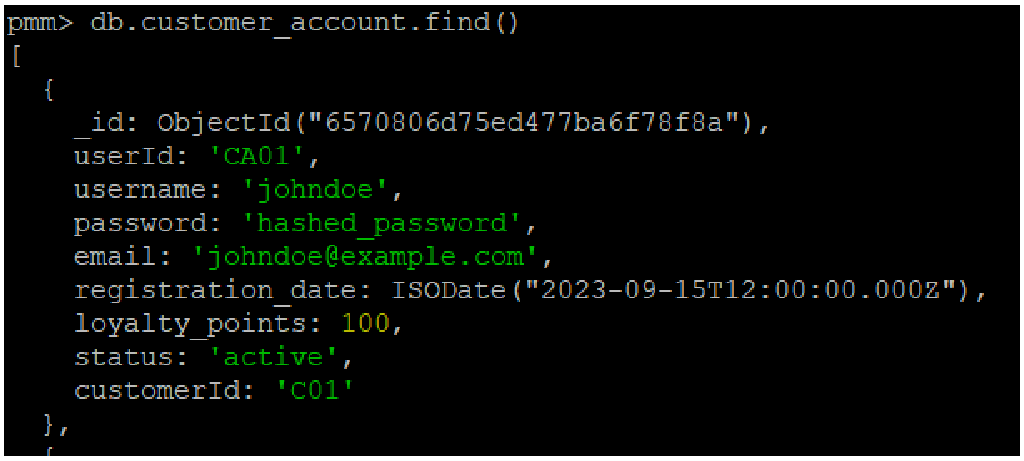
Customer_account document:

Insert:

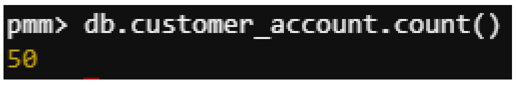


```
db.customer_account.insertOne({
  userId: "CA01",
  username: "johndoe",
  password: "hashed_password",
  email: "johndoe@example.com",
  registration_date: ISODate("2023-09-15T12:00:00Z"),
  loyalty_points: 100,
  status: "active",
  customerId: "C01"
  // Other account details
})
```

Screenshot:



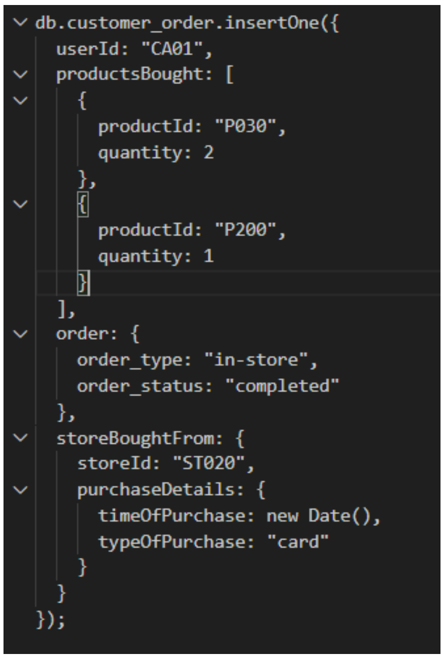
```
pmm> db.customer_account.find()
[
  {
    _id: ObjectId("6570806d75ed477ba6f78f8a"),
    userId: 'CA01',
    username: 'johndoe',
    password: 'hashed_password',
    email: 'johndoe@example.com',
    registration_date: ISODate("2023-09-15T12:00:00.000Z"),
    loyalty_points: 100,
    status: 'active',
    customerId: 'C01'
  },
```



```
pmm> db.customer_account.count()
50
```

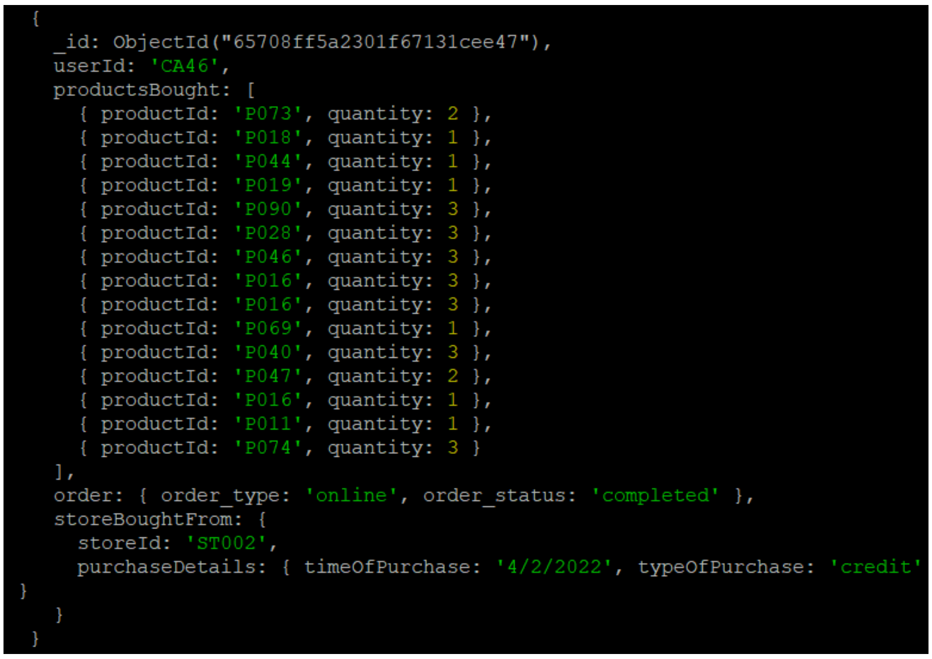
Customer_order document:

Insert:

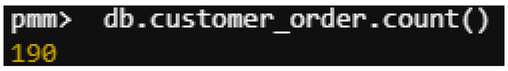


```
db.customer_order.insertOne({
  userId: "CA01",
  productsBought: [
    {
      productId: "P030",
      quantity: 2
    },
    {
      productId: "P200",
      quantity: 1
    }
  ],
  order: {
    order_type: "in-store",
    order_status: "completed"
  },
  storeBoughtFrom: {
    storeId: "ST020",
    purchaseDetails: {
      timeOfPurchase: new Date(),
      typeOfPurchase: "card"
    }
  }
});
```

Screenshot:



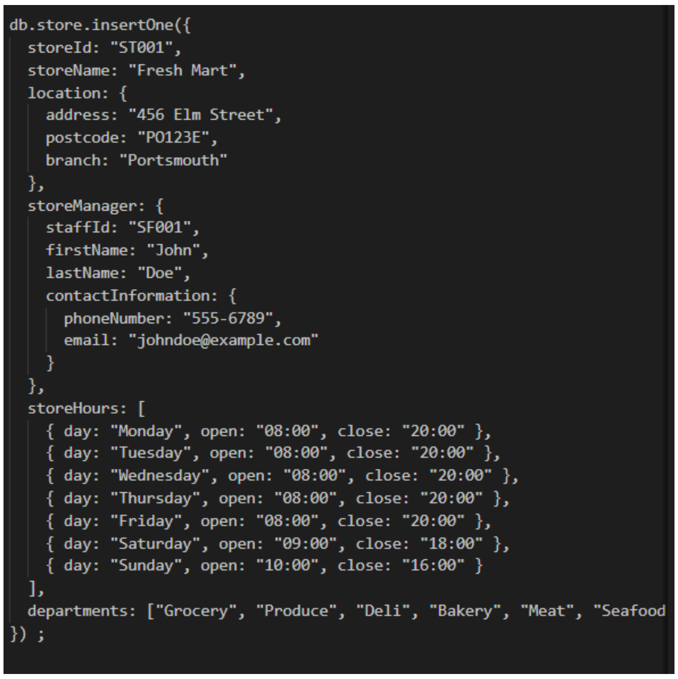
```
{
  _id: ObjectId("65708ff5a2301f67131cee47"),
  userId: 'CA46',
  productsBought: [
    { productId: 'P073', quantity: 2 },
    { productId: 'P018', quantity: 1 },
    { productId: 'P044', quantity: 1 },
    { productId: 'P019', quantity: 1 },
    { productId: 'P090', quantity: 3 },
    { productId: 'P028', quantity: 3 },
    { productId: 'P046', quantity: 3 },
    { productId: 'P016', quantity: 3 },
    { productId: 'P016', quantity: 3 },
    { productId: 'P069', quantity: 1 },
    { productId: 'P040', quantity: 3 },
    { productId: 'P047', quantity: 2 },
    { productId: 'P016', quantity: 1 },
    { productId: 'P011', quantity: 1 },
    { productId: 'P074', quantity: 3 }
  ],
  order: { order_type: 'online', order_status: 'completed' },
  storeBoughtFrom: {
    storeId: 'ST002',
    purchaseDetails: { timeOfPurchase: '4/2/2022', typeOfPurchase: 'credit'
}
    }
  }
```



```
pmm> db.customer_order.count()
190
```

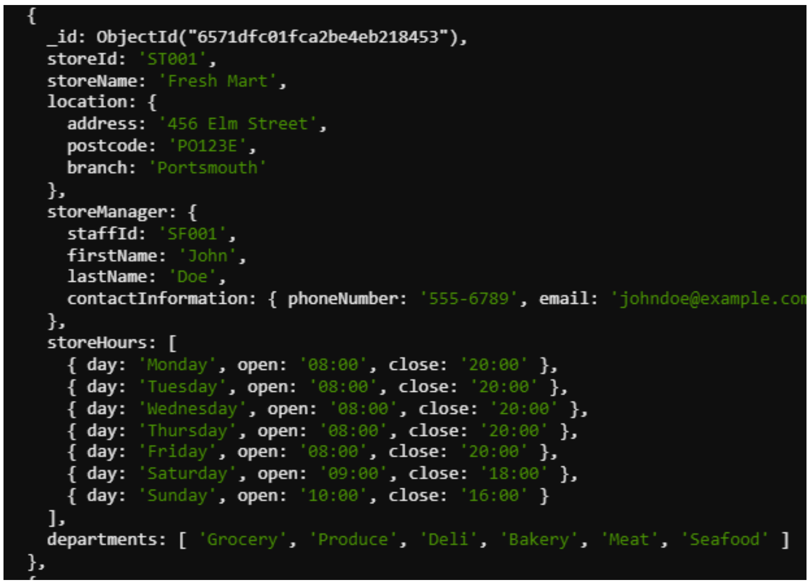
Store document:

Insert:

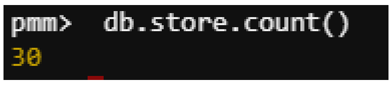


```
db.store.insertOne({
  storeId: "ST001",
  storeName: "Fresh Mart",
  location: {
    address: "456 Elm Street",
    postcode: "PO123E",
    branch: "Portsmouth"
  },
  storeManager: {
    staffId: "SF001",
    firstName: "John",
    lastName: "Doe",
    contactInformation: {
      phoneNumber: "555-6789",
      email: "johndoe@example.com"
    }
  },
  storeHours: [
    { day: "Monday", open: "08:00", close: "20:00" },
    { day: "Tuesday", open: "08:00", close: "20:00" },
    { day: "Wednesday", open: "08:00", close: "20:00" },
    { day: "Thursday", open: "08:00", close: "20:00" },
    { day: "Friday", open: "08:00", close: "20:00" },
    { day: "Saturday", open: "09:00", close: "18:00" },
    { day: "Sunday", open: "10:00", close: "16:00" }
  ],
  departments: ["Grocery", "Produce", "Deli", "Bakery", "Meat", "Seafood
}) ;
```

Screenshot:



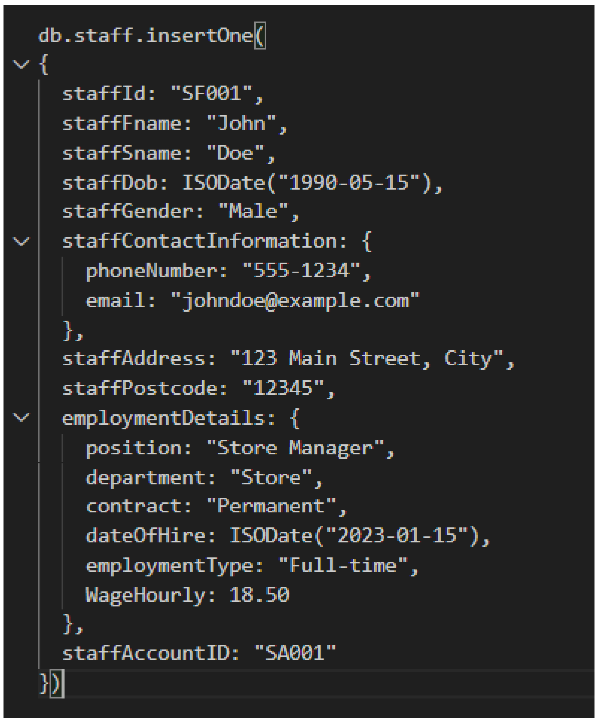
```
  {
    _id: ObjectId("6571dfc01fca2be4eb218453"),
    storeId: 'ST001',
    storeName: 'Fresh Mart',
    location: {
      address: '456 Elm Street',
      postcode: 'PO123E',
      branch: 'Portsmouth'
    },
    storeManager: {
      staffId: 'SF001',
      firstName: 'John',
      lastName: 'Doe',
      contactInformation: { phoneNumber: '555-6789', email: 'johndoe@example.co
    },
    storeHours: [
      { day: 'Monday', open: '08:00', close: '20:00' },
      { day: 'Tuesday', open: '08:00', close: '20:00' },
      { day: 'Wednesday', open: '08:00', close: '20:00' },
      { day: 'Thursday', open: '08:00', close: '20:00' },
      { day: 'Friday', open: '08:00', close: '20:00' },
      { day: 'Saturday', open: '09:00', close: '18:00' },
      { day: 'Sunday', open: '10:00', close: '16:00' }
    ],
    departments: [ 'Grocery', 'Produce', 'Deli', 'Bakery', 'Meat', 'Seafood' ]
  },
```



```
pmm>  db.store.count()
30
```

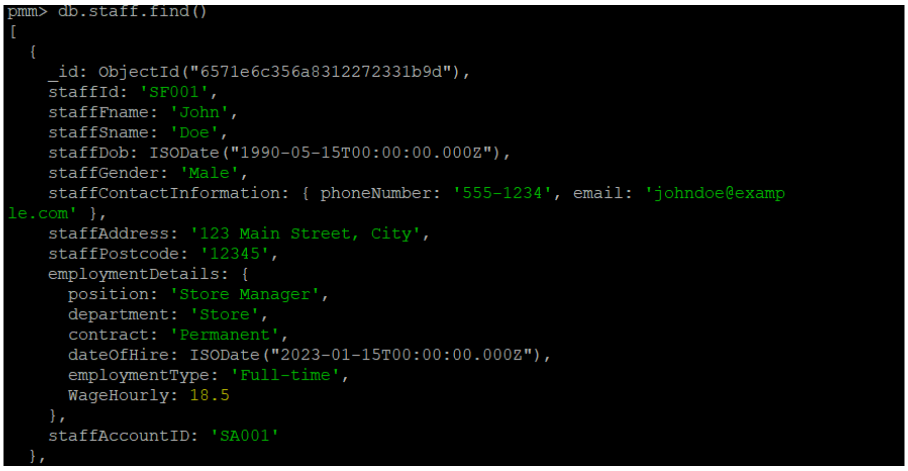
Staff document:

Insert:

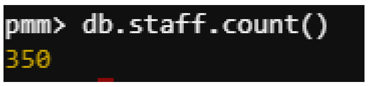


```
db.staff.insertOne(
{
  staffId: "SF001",
  staffFname: "John",
  staffSname: "Doe",
  staffDob: ISODate("1990-05-15"),
  staffGender: "Male",
  staffContactInformation: {
    phoneNumber: "555-1234",
    email: "johndoe@example.com"
  },
  staffAddress: "123 Main Street, City",
  staffPostcode: "12345",
  employmentDetails: {
    position: "Store Manager",
    department: "Store",
    contract: "Permanent",
    dateOfHire: ISODate("2023-01-15"),
    employmentType: "Full-time",
    WageHourly: 18.50
  },
  staffAccountID: "SA001"
})
```

Screenshot:



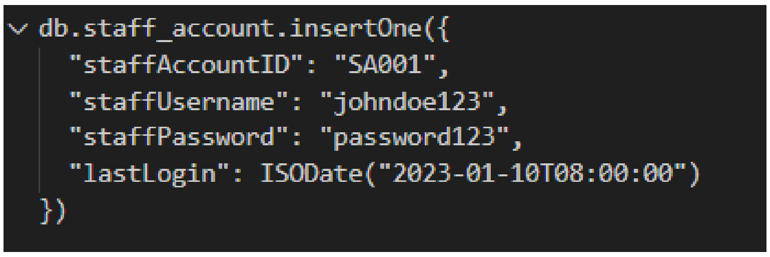
```
pmm> db.staff.find()
[
  {
    _id: ObjectId("6571e6c356a8312272331b9d"),
    staffId: 'SF001',
    staffFname: 'John',
    staffSname: 'Doe',
    staffDob: ISODate("1990-05-15T00:00:00.000Z"),
    staffGender: 'Male',
    staffContactInformation: { phoneNumber: '555-1234', email: 'johndoe@examp
le.com' },
    staffAddress: '123 Main Street, City',
    staffPostcode: '12345',
    employmentDetails: {
      position: 'Store Manager',
      department: 'Store',
      contract: 'Permanent',
      dateOfHire: ISODate("2023-01-15T00:00:00.000Z"),
      employmentType: 'Full-time',
      WageHourly: 18.5
    },
    staffAccountID: 'SA001'
  },
```



```
pmm> db.staff.count()
350
```

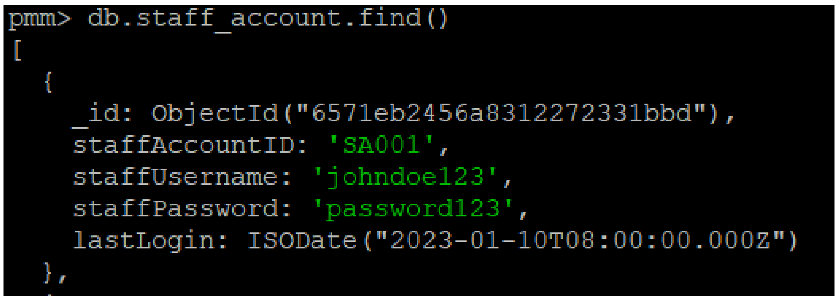
Staff_account document:

Insert:

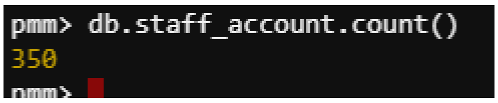


```
db.staff_account.insertOne({
  "staffAccountID": "SA001",
  "staffUsername": "johndoe123",
  "staffPassword": "password123",
  "lastLogin": ISODate("2023-01-10T08:00:00")
})
```

Screenshot:



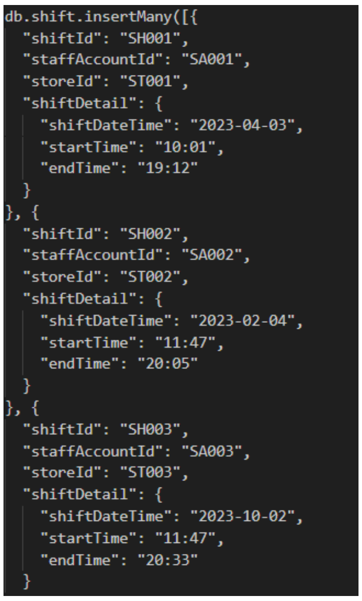
```
pmm> db.staff_account.find()
[
  {
    _id: ObjectId("6571eb2456a8312272331bbd"),
    staffAccountID: 'SA001',
    staffUsername: 'johndoe123',
    staffPassword: 'password123',
    lastLogin: ISODate("2023-01-10T08:00:00.000Z")
  },
```



```
pmm> db.staff_account.count()
350
pmm>
```

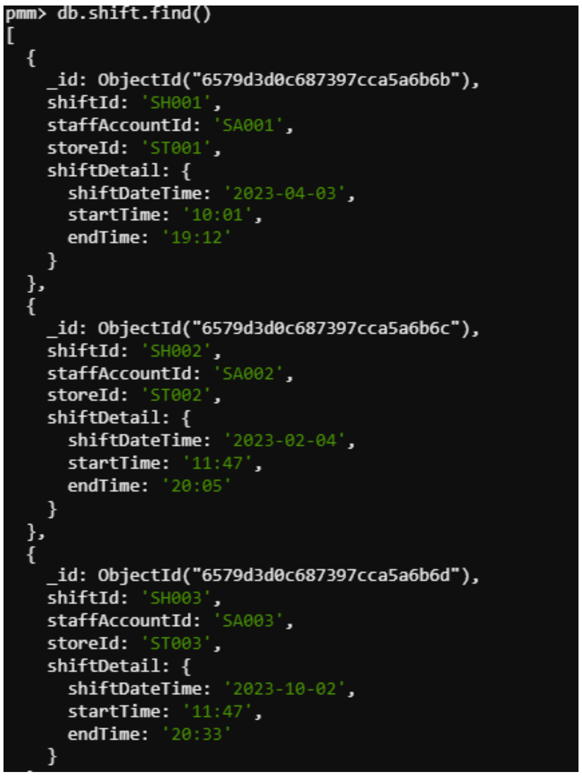
Shift document:

Insert:

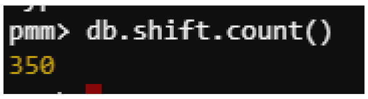


```
db.shift.insertMany([{
  "shiftId": "SH001",
  "staffAccountId": "SA001",
  "storeId": "ST001",
  "shiftDetail": {
    "shiftDateTime": "2023-04-03",
    "startTime": "10:01",
    "endTime": "19:12"
  }
}, {
  "shiftId": "SH002",
  "staffAccountId": "SA002",
  "storeId": "ST002",
  "shiftDetail": {
    "shiftDateTime": "2023-02-04",
    "startTime": "11:47",
    "endTime": "20:05"
  }
}, {
  "shiftId": "SH003",
  "staffAccountId": "SA003",
  "storeId": "ST003",
  "shiftDetail": {
    "shiftDateTime": "2023-10-02",
    "startTime": "11:47",
    "endTime": "20:33"
  }
```

Screenshot:



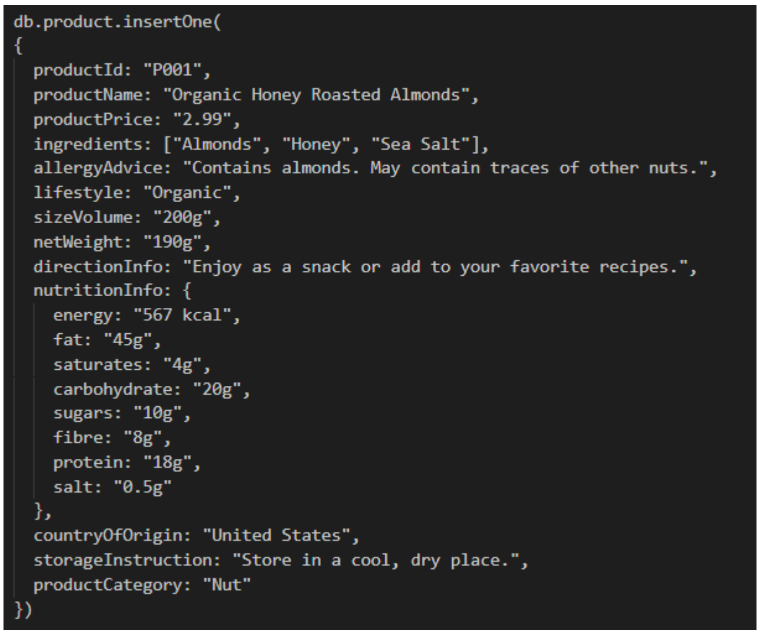
```
pmm> db.shift.find()
[
  {
    _id: ObjectId("6579d3d0c687397cca5a6b6b"),
    shiftId: 'SH001',
    staffAccountId: 'SA001',
    storeId: 'ST001',
    shiftDetail: {
      shiftDateTime: '2023-04-03',
      startTime: '10:01',
      endTime: '19:12'
    }
  },
  {
    _id: ObjectId("6579d3d0c687397cca5a6b6c"),
    shiftId: 'SH002',
    staffAccountId: 'SA002',
    storeId: 'ST002',
    shiftDetail: {
      shiftDateTime: '2023-02-04',
      startTime: '11:47',
      endTime: '20:05'
    }
  },
  {
    _id: ObjectId("6579d3d0c687397cca5a6b6d"),
    shiftId: 'SH003',
    staffAccountId: 'SA003',
    storeId: 'ST003',
    shiftDetail: {
      shiftDateTime: '2023-10-02',
      startTime: '11:47',
      endTime: '20:33'
    }
```



```
pmm> db.shift.count()
350
```

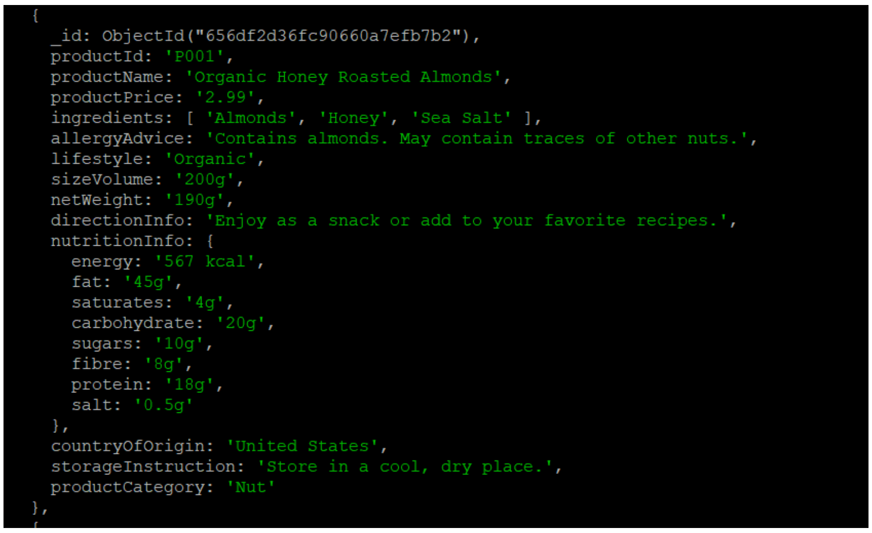
Product document:

Insert:

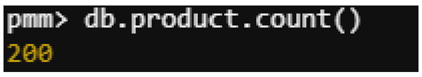


```
db.product.insertOne(
{
  productId: "P001",
  productName: "Organic Honey Roasted Almonds",
  productPrice: "2.99",
  ingredients: ["Almonds", "Honey", "Sea Salt"],
  allergyAdvice: "Contains almonds. May contain traces of other nuts.",
  lifestyle: "Organic",
  sizeVolume: "200g",
  netWeight: "190g",
  directionInfo: "Enjoy as a snack or add to your favorite recipes.",
  nutritionInfo: {
    energy: "567 kcal",
    fat: "45g",
    saturates: "4g",
    carbohydrate: "20g",
    sugars: "10g",
    fibre: "8g",
    protein: "18g",
    salt: "0.5g"
  },
  countryOfOrigin: "United States",
  storageInstruction: "Store in a cool, dry place.",
  productCategory: "Nut"
})
```

Screenshot:



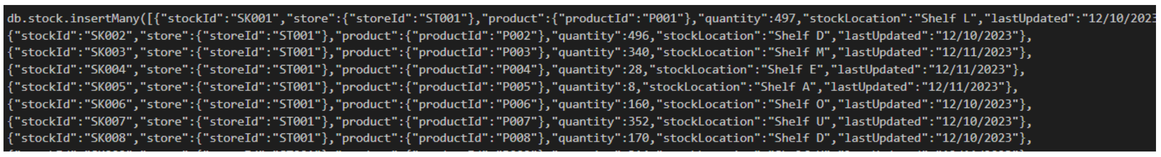
```
  {
    _id: ObjectId("656df2d36fc90660a7efb7b2"),
    productId: 'P001',
    productName: 'Organic Honey Roasted Almonds',
    productPrice: '2.99',
    ingredients: [ 'Almonds', 'Honey', 'Sea Salt' ],
    allergyAdvice: 'Contains almonds. May contain traces of other nuts.',
    lifestyle: 'Organic',
    sizeVolume: '200g',
    netWeight: '190g',
    directionInfo: 'Enjoy as a snack or add to your favorite recipes.',
    nutritionInfo: {
      energy: '567 kcal',
      fat: '45g',
      saturates: '4g',
      carbohydrate: '20g',
      sugars: '10g',
      fibre: '8g',
      protein: '18g',
      salt: '0.5g'
    },
    countryOfOrigin: 'United States',
    storageInstruction: 'Store in a cool, dry place.',
    productCategory: 'Nut'
  },
```



```
pmm> db.product.count()
200
```

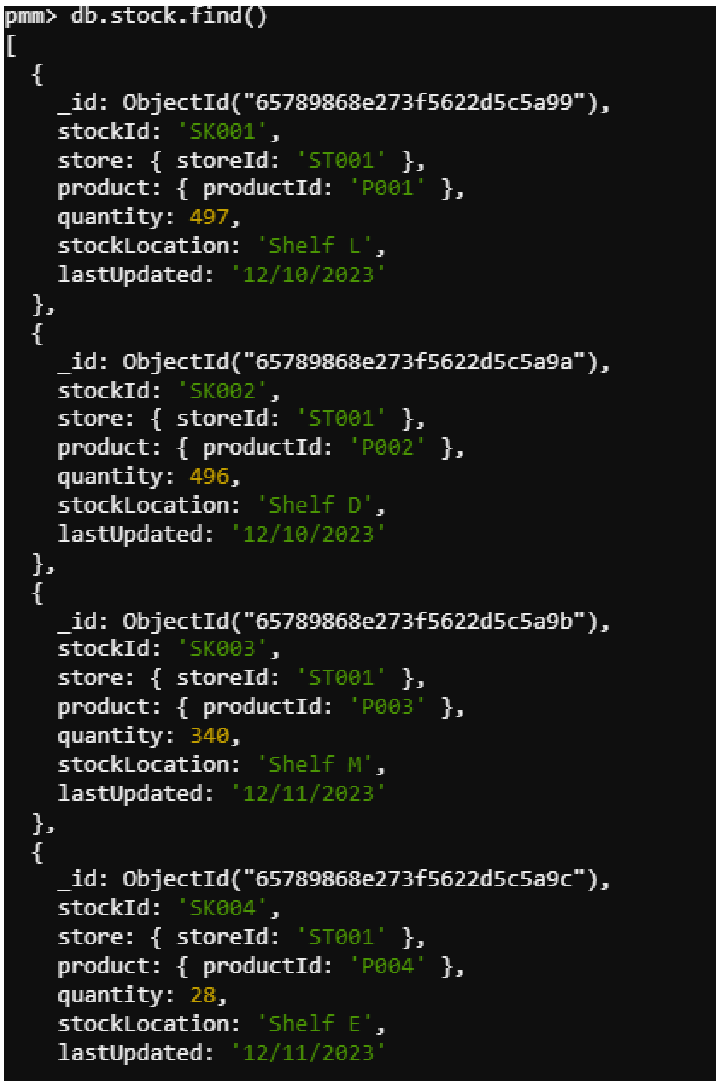
Stock document:

Insert:

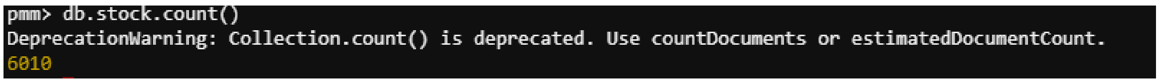


```
db.stock.insertMany([{"stockId":"SK001","store":{"storeId":"ST001"},"product":{"productId":"P001"},"quantity":497,"stockLocation":"Shelf L","lastUpdated":"12/10/2023
{"stockId":"SK002","store":{"storeId":"ST001"},"product":{"productId":"P002"},"quantity":496,"stockLocation":"Shelf D","lastUpdated":"12/10/2023"},
{"stockId":"SK003","store":{"storeId":"ST001"},"product":{"productId":"P003"},"quantity":340,"stockLocation":"Shelf M","lastUpdated":"12/11/2023"},
{"stockId":"SK004","store":{"storeId":"ST001"},"product":{"productId":"P004"},"quantity":28,"stockLocation":"Shelf E","lastUpdated":"12/11/2023"},
{"stockId":"SK005","store":{"storeId":"ST001"},"product":{"productId":"P005"},"quantity":8,"stockLocation":"Shelf A","lastUpdated":"12/11/2023"},
{"stockId":"SK006","store":{"storeId":"ST001"},"product":{"productId":"P006"},"quantity":160,"stockLocation":"Shelf O","lastUpdated":"12/10/2023"},
{"stockId":"SK007","store":{"storeId":"ST001"},"product":{"productId":"P007"},"quantity":352,"stockLocation":"Shelf U","lastUpdated":"12/10/2023"},
{"stockId":"SK008","store":{"storeId":"ST001"},"product":{"productId":"P008"},"quantity":170,"stockLocation":"Shelf D","lastUpdated":"12/10/2023"},
```

Screenshot:



```
pmm> db.stock.find()
[
  {
    _id: ObjectId("65789868e273f5622d5c5a99"),
    stockId: 'SK001',
    store: { storeId: 'ST001' },
    product: { productId: 'P001' },
    quantity: 497,
    stockLocation: 'Shelf L',
    lastUpdated: '12/10/2023'
  },
  {
    _id: ObjectId("65789868e273f5622d5c5a9a"),
    stockId: 'SK002',
    store: { storeId: 'ST001' },
    product: { productId: 'P002' },
    quantity: 496,
    stockLocation: 'Shelf D',
    lastUpdated: '12/10/2023'
  },
  {
    _id: ObjectId("65789868e273f5622d5c5a9b"),
    stockId: 'SK003',
    store: { storeId: 'ST001' },
    product: { productId: 'P003' },
    quantity: 340,
    stockLocation: 'Shelf M',
    lastUpdated: '12/11/2023'
  },
  {
    _id: ObjectId("65789868e273f5622d5c5a9c"),
    stockId: 'SK004',
    store: { storeId: 'ST001' },
    product: { productId: 'P004' },
    quantity: 28,
    stockLocation: 'Shelf E',
    lastUpdated: '12/11/2023'
```



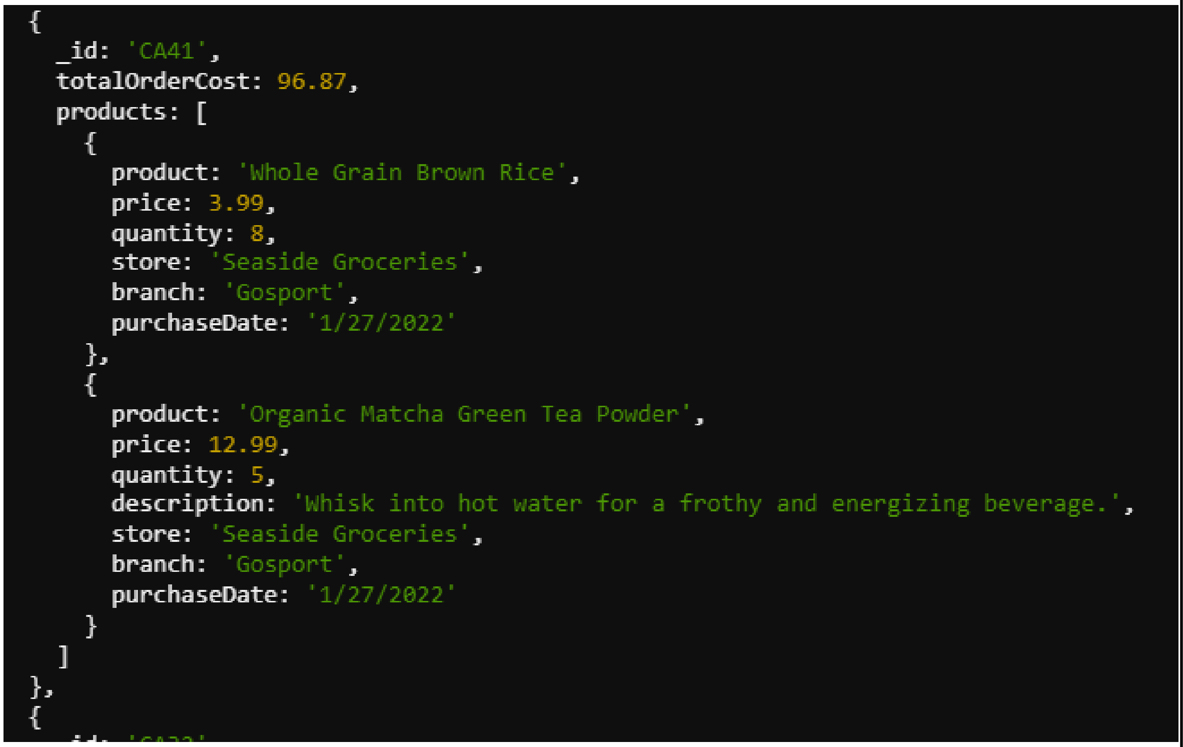
```
pmm> db.stock.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
6010
```

Task T2: Queries

Query 1

```
db.customer_order.aggregate([
  { $unwind: "$productsBought" },
  { $lookup: {
    from: "product",
    localField: "productsBought.productId",
    foreignField: "productId",
    as: "productDetails"
 } },
  { $unwind: "$productDetails" },
  { $lookup: {
    from: "store",
    localField: "storeBoughtFrom.storeId",
    foreignField: "storeId",
    as: "storeInfo"
} },
  { $unwind: "$storeInfo" },
  { $addFields: {
    "productPrice": { $toDouble: "$productDetails.productPrice" },
    "totalCost": { $multiply: ["$productsBought.quantity", { $toDouble:
     "$productDetails.productPrice" }] },
    "storeName": "$storeInfo.storeName"
} },
  {
    $group: {
      _id: "$userId",
      totalOrderCost: { $sum: "$totalCost" },
      products: {
        $push: {
          product: "$productDetails.productName",
          price: "$productPrice",
          quantity: "$productsBought.quantity",
          description: "$productDetails.directionInfo",
          store: "$storeInfo.storeName",
          branch: "$storeInfo.storeLocation.branch",
          purchaseDate: "$storeBoughtFrom.purchaseDetails.timeOfPurchase",

        }
      }
    }
  },
  {
    $addFields: {
      totalOrderCost: { $round: ["$totalOrderCost", 2] }
    }
  }
])
```

The first query is a simple search of past purchases made from each unique customer registered within the database. It shows how much a user has spent and displays all the purchased products with all relevant information. For example, the name of the product, price, quantity, where it was bought and when they have made the transaction. This may provide usefulness as the marketing or sales department from PMM might require information regarding purchasing history to boost favourable products or products that do not sell as much. Which as a business can limit or discontinue a line to maintain break-even.



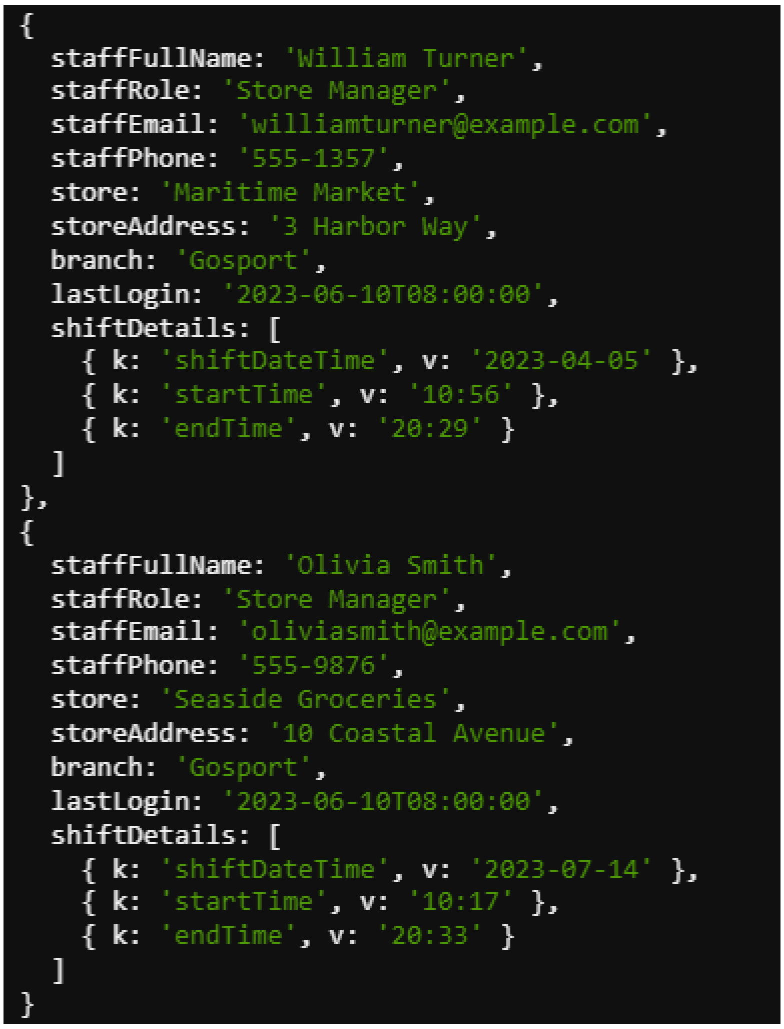
```
{
  _id: 'CA41',
  totalOrderCost: 96.87,
  products: [
    {
      product: 'Whole Grain Brown Rice',
      price: 3.99,
      quantity: 8,
      store: 'Seaside Groceries',
      branch: 'Gosport',
      purchaseDate: '1/27/2022'
    },
    {
      product: 'Organic Matcha Green Tea Powder',
      price: 12.99,
      quantity: 5,
      description: 'Whisk into hot water for a frothy and energizing beverage.',
      store: 'Seaside Groceries',
      branch: 'Gosport',
      purchaseDate: '1/27/2022'
    }
  ]
},
{
```

One of the results is taken as a screenshot. It displays the _id which are the customers. With a totalordercost of the entire purchasing history with pmm. And the list of products with all its relevant information.

Query 2

```
db.shift.aggregate([
  {
    $lookup: {
      from: "staff_account",
      localField: "staffAccountId",
      foreignField: "staffAccountId",
      as: "staffAccountDetails"
    }
  },
  {
    $unwind: "$staffAccountDetails"
  },
  {
    $lookup: {
      from: "staff",
      localField: "staffAccountDetails.staffAccountId",
      foreignField: "staffAccountId",
      as: "staffDetails"
    }
  },
  {
    $unwind: "$staffDetails"
  },
  {
    $lookup: {
      from: "store",
      localField: "storeId",
      foreignField: "storeId",
      as: "storeDetails"
    }
  },
  {
    $unwind: "$storeDetails"
  },
  {
    $project: {
      staffFullName: {
        $concat: ["$staffDetails.staffFname", " ", "$staffDetails.staffSname"]
      },
      staffRole: "$staffDetails.employmentDetails.position",
      staffEmail: "$staffDetails.staffContactInformation.email",
      staffPhone: "$staffDetails.staffContactInformation.phoneNumber",
      store: "$storeDetails.storeName",
      storeAddress: "$storeDetails.storeLocation.address",
      branch: "$storeDetails.storeLocation.branch",
      lastLogin: "$staffAccountDetails.lastLogin",
      shiftDetails: {
        $objectToArray: "$shiftDetail"
      },
      _id: 0
    }
  }
])
```

This query can be used to gather information regarding details of staff shifts as well as information regarding the staff. Managers can utilise this to view which store/branch they will be working at, as well as shift details like shift date, start time and end time of each shift.



```
{
  staffFullName: 'William Turner',
  staffRole: 'Store Manager',
  staffEmail: 'williamturner@example.com',
  staffPhone: '555-1357',
  store: 'Maritime Market',
  storeAddress: '3 Harbor Way',
  branch: 'Gosport',
  lastLogin: '2023-06-10T08:00:00',
  shiftDetails: [
    { k: 'shiftDateTime', v: '2023-04-05' },
    { k: 'startTime', v: '10:56' },
    { k: 'endTime', v: '20:29' }
  ]
},
{
  staffFullName: 'Olivia Smith',
  staffRole: 'Store Manager',
  staffEmail: 'oliviasmith@example.com',
  staffPhone: '555-9876',
  store: 'Seaside Groceries',
  storeAddress: '10 Coastal Avenue',
  branch: 'Gosport',
  lastLogin: '2023-06-10T08:00:00',
  shiftDetails: [
    { k: 'shiftDateTime', v: '2023-07-14' },
    { k: 'startTime', v: '10:17' },
    { k: 'endTime', v: '20:33' }
  ]
}
```

The screenshot shows the results when performing the query. It lists all the staff details including their full name as well as which store they will be working at. And a nested document on shift details.

Query 3

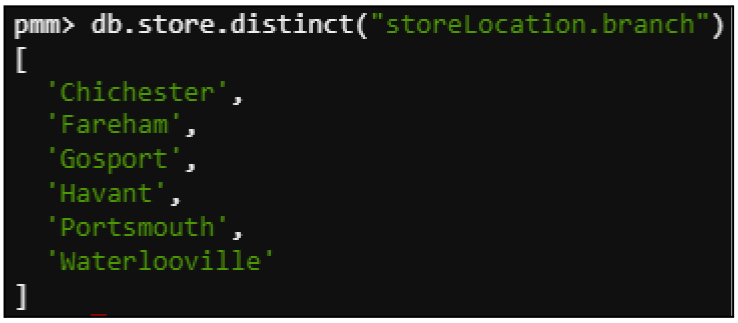
```
db.customer_order.aggregate([
  {
   $lookup: {
     from: "store",
     localField: "storeBoughtFrom.storeId",
     foreignField: "storeId",
     as: "store"
   }
  },
  {
   $unwind: "$store"
  },
  {
   $unwind: "$productsBought"
  },
  {
   $lookup: {
     from: "product",
     localField: "productsBought.productId",
     foreignField: "productId",
     as: "product"
   }
  },
  {
   $unwind: "$product"
  },
  {
   $addFields: {
     totalPrice: { $multiply: [{ $toDouble: "$product.productPrice" }, "$productsBought.quantity"] },
     purchaseDate: { $toDate: "$storeBoughtFrom.purchaseDetails.timeOfPurchase" }
   }
  },
  {
   $addFields: {
     quarter: {
       $concat: [
         "Q",
         { $toString: { $add: [1, { $trunc: { $divide: [{ $month: "$purchaseDate" }, 3] } }] } }
       ]
     }
   }
  },
  {
   $addFields: {
     year: { $year: "$purchaseDate" }
   }
  },
  {
   $match: { "quarter": { $in: ["Q1", "Q2", "Q3", "Q4"] } }
  },
   {
   $match: { "store.storeLocation.branch": "Portsmouth" } // To pick one branch to check fiscal report
  },
  {
   $group: {
     _id: { branch: "$store.storeLocation.branch", year: "$year", quarter: "$quarter"},
     TotalSales: { $sum: "$totalPrice" }
   }
  },
  {
   $addFields: {
     TotalRevenue: { $round: ["$TotalSales", 2] }
   }
  },
  {
   $sort: { "_id.year": 1, "_id.quarter": 1 }
  },
  {
   $project: {
     _id: 0,
     branch: "$_id.branch",
     year: "$_id.year",
     quarter: "$_id.quarter",
     TotalRevenue: 1
   }
  }
 ])
```

This query can be used as a fiscal report for each branch. Managers or shareholders can implement this query to view the performance and total revenue of each branch. Indicating which branch is over or underperforming depending on the output. Each quarterly is represented from Q1 (Jan, Feb, Mar) to Q4 (Oct, Nov, Dec)

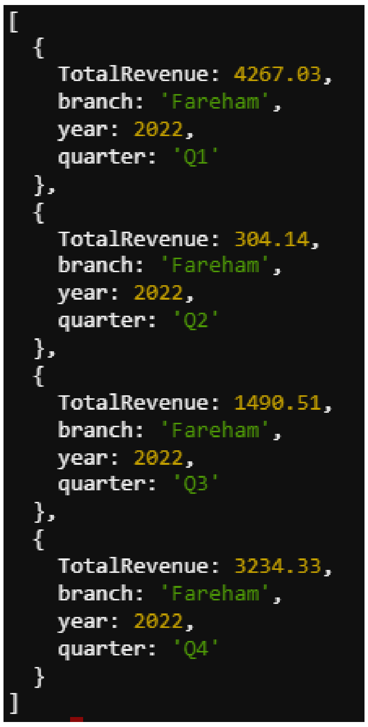
Additional information:

```
{
  $match: { "store.storeLocation.branch": "Fareham" } // To pick one branch to check fiscal report
 },
```

You can either remove this line of code to gather all the reports for all the branches instead of viewing each separately to view differences in performances and sales of each branch throughout the fiscal year. Or you can manually alter the branch name to any dedicated branches:

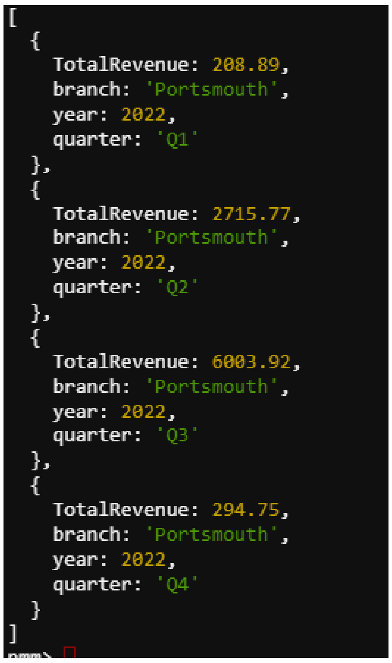


```
pmm> db.store.distinct("storeLocation.branch")
[
  'Chichester',
  'Fareham',
  'Gosport',
  'Havant',
  'Portsmouth',
  'Waterlooville'
]
```



```
[
  {
    TotalRevenue: 4267.03,
    branch: 'Fareham',
    year: 2022,
    quarter: 'Q1'
  },
  {
    TotalRevenue: 304.14,
    branch: 'Fareham',
    year: 2022,
    quarter: 'Q2'
  },
  {
    TotalRevenue: 1490.51,
    branch: 'Fareham',
    year: 2022,
    quarter: 'Q3'
  },
  {
    TotalRevenue: 3234.33,
    branch: 'Fareham',
    year: 2022,
    quarter: 'Q4'
  }
]
```

The results demonstrate the query for the Fareham branch and the total revenue generated per quarter in 2022. Furthermore, With a simple alternation on the code mentioned above you can view other branches as well. For example, for Portsmouth.



```
[
  {
    TotalRevenue: 208.89,
    branch: 'Portsmouth',
    year: 2022,
    quarter: 'Q1'
  },
  {
    TotalRevenue: 2715.77,
    branch: 'Portsmouth',
    year: 2022,
    quarter: 'Q2'
  },
  {
    TotalRevenue: 6003.92,
    branch: 'Portsmouth',
    year: 2022,
    quarter: 'Q3'
  },
  {
    TotalRevenue: 294.75,
    branch: 'Portsmouth',
    year: 2022,
    quarter: 'Q4'
  }
]
pmm>
```