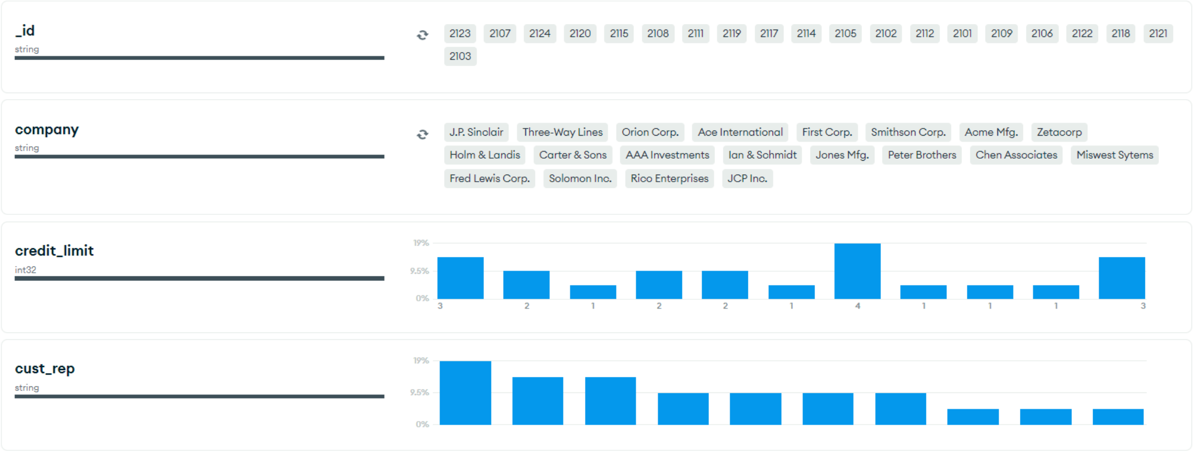
Week 2 Assignment MongoDB Shreyank Saksena

UML Diagrams/Schemas

Customers Table Schema



_id
string

2123 2107 2124 2120 2115 2108 2111 2119 2117 2114 2105 2102 2112 2101 2109 2106 2122 2118 2121 2103

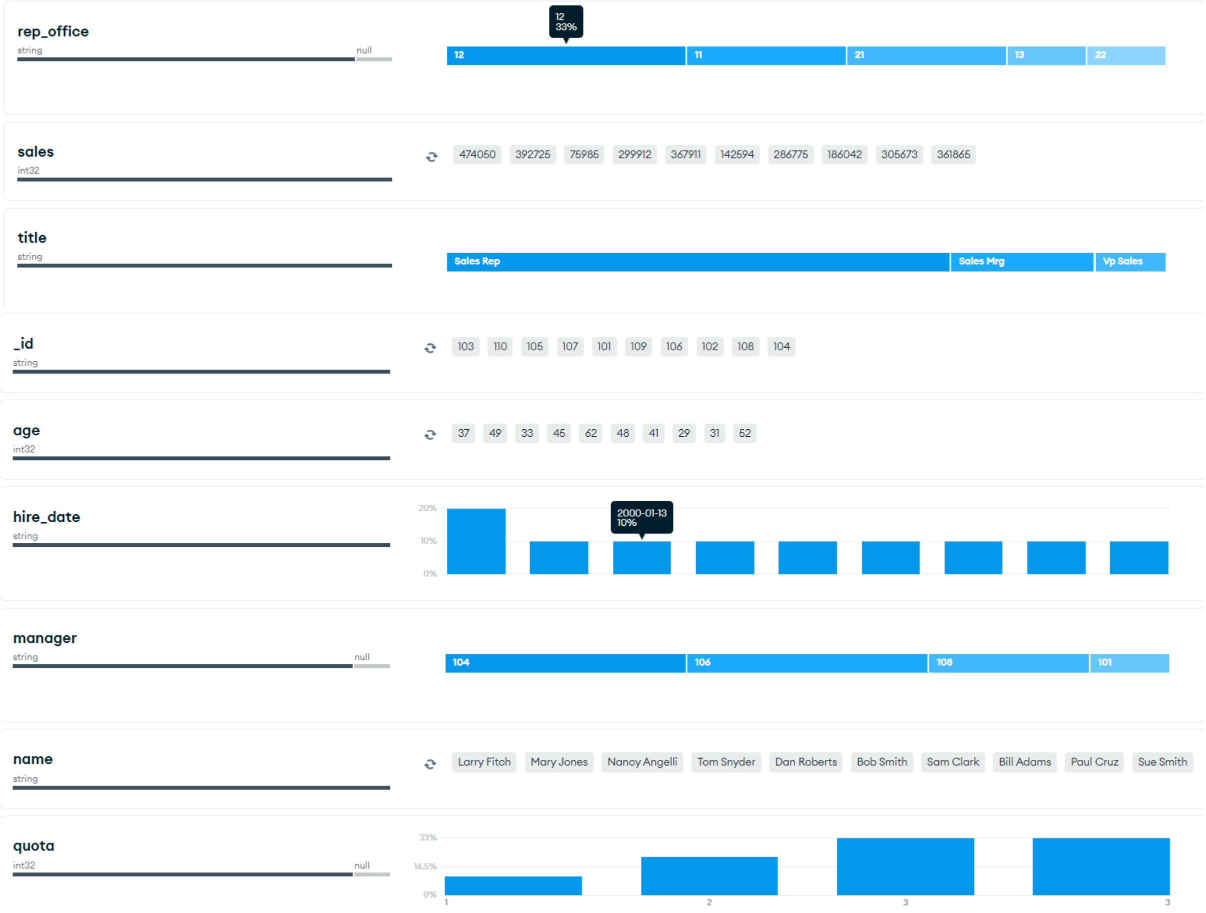
company
string

J.P. Sinclair, Three-Way Lines, Orion Corp., Ace International, First Corp., Smithson Corp., Acme Mfg., Zetacorp, Holm & Landis, Carter & Sons, AAA Investments, Ian & Schmidt, Jones Mfg., Peter Brothers, Chen Associates, Miswest Sytems, Fred Lewis Corp., Solomon Inc., Rico Enterprises, JCP Inc.

credit_limit
int32

cust_rep
string

Salesreps Table Schema



rep_office
string
null

sales
int32

474050 392725 75985 299912 367911 142594 286775 186042 305673 361865

title
string

_id
string

103 110 105 107 101 109 106 102 108 104

age
int32

37 49 33 45 62 48 41 29 31 52

hire_date
string

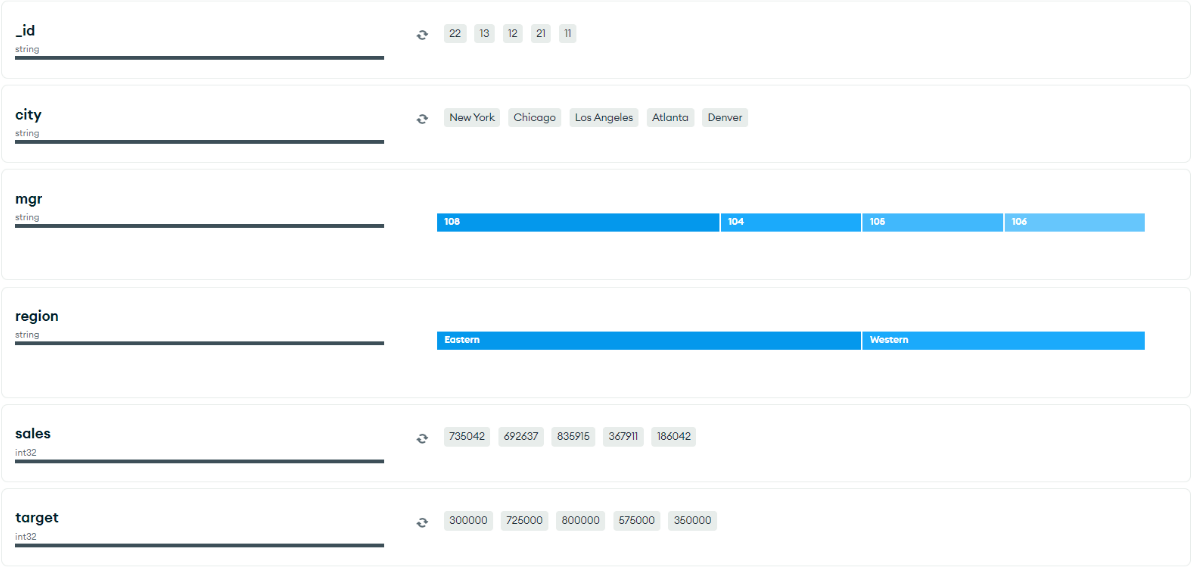
manager
string
null

name
string

Larry Fitch, Mary Jones, Nancy Angelli, Tom Snyder, Dan Roberts, Bob Smith, Sam Clark, Bill Adams, Paul Cruz, Sue Smith

quota
int32
null

Office Table Schema



_id
string
22 13 12 21 11

city
string
New York Chicago Los Angeles Atlanta Denver

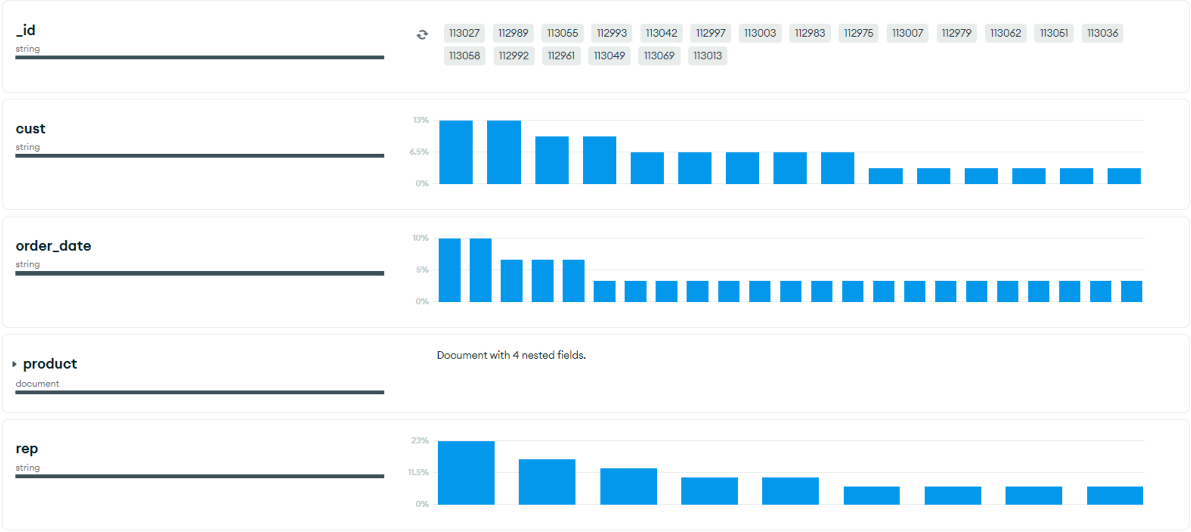
mgr
string

region
string

sales
int32
735042 692637 835915 367911 186042

target
int32
300000 725000 800000 575000 350000

Orders Table Schema



_id
string
113027 112989 113055 112993 113042 112997 113003 112983 112975 113007 112979 113062 113051 113036
113058 112992 112961 113049 113069 113013

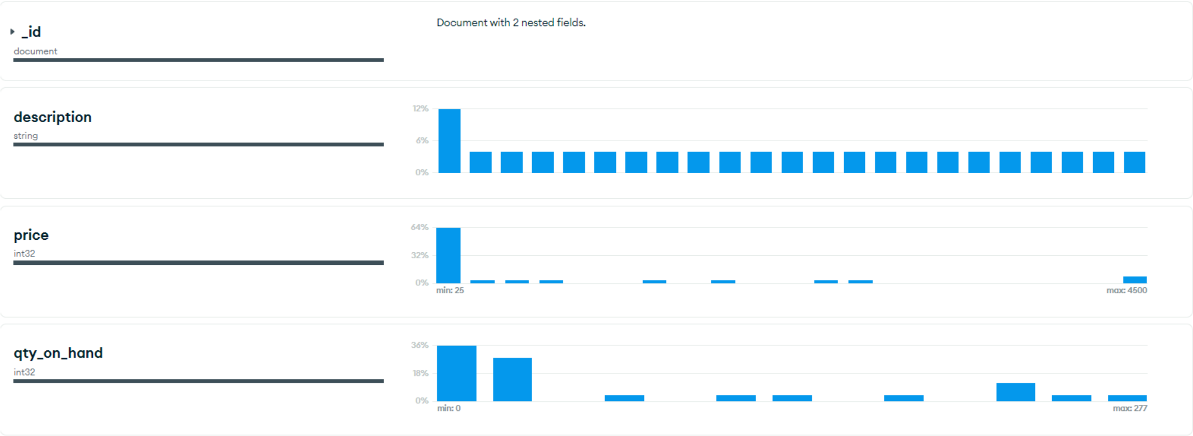
cust
string

order_date
string

product
document
Document with 4 nested fields.

rep
string

Products Table Schema



▸ _id
document

Document with 2 nested fields.

description
string

price
int32

qty_on_hand
int32

1. **Show the name, sales, and quota of Bill Adams:**

```
db.salesreps.find(
 { name: "Bill Adams" },
 { name: 1, sales: 1, quota: 1 }
)
```

2. **List the company names and the product description of all the products each has ordered. Arrange by company:**

```
db.orders.aggregate([
 {
   $lookup: {
     from: "products",
     let: { prod_mfr: "$product.mfr", prod_id: "$product.product_id" },
     pipeline: [
       { $match: { $expr: { $and: [{ $eq: ["$_id.mfr_id", "$$prod_mfr"] }, { $eq: ["$_id.product_id", "$$prod_id"] }] } } },
       { $project: { description: 1 } }
     ],
     as: "product_info"
   }
 },
 {
   $lookup: {
     from: "customers",
     localField: "cust",
     foreignField: "_id",
     as: "customer_info"
   }
 },
 { $unwind: "$product_info" },
 { $unwind: "$customer_info" },
 { $project: { "customer_info.company": 1, "product_info.description": 1 } },
 { $sort: { "customer_info.company": 1 } }
])
```

3. **Show the total value of the inventory on hand for each product. Arrange by total value descending:**

```
db.products.aggregate([
  {
    $project: {
      description: 1,
      total_value: { $multiply: ["$price", "$qty_on_hand"] }
    }
  },
  { $sort: { total_value: -1 } }
])
```

4. **How many customers are there?**

```
db.customers.countDocuments()
```

5. List the offices with a target over $600,000:

```
db.offices.find(
  { target: { $gt: 600000 } },
  { city: 1, target: 1 }
)
```

6. **What is the average sales of all the salespeople?**

```
db.salesreps.aggregate([
 {
   $group: {
     _id: null,
     avg_sales: { $avg: "$sales" }
   }
 }
])
```

7. **List orders over $25,000, including the name of the salesperson who took the order and the name of the customer who placed it:**

```
db.orders.aggregate([
 { $match: { "product.amount": { $gt: 25000 } } },
 {
   $lookup: {
     from: "salesreps",
     localField: "rep",
     foreignField: "_id",
     as: "salesrep_info"
   }
 },
 {
   $lookup: {
     from: "customers",
     localField: "cust",
     foreignField: "_id",
     as: "customer_info"
   }
 },
 { $unwind: "$salesrep_info" },
 { $unwind: "$customer_info" },
 { $project: { "salesrep_info.name": 1, "customer_info.company": 1, "product.amount": 1 } }
])
```

8. **How many sales offices have salespeople who are over quota?**

```
db.salesreps.aggregate([
  { $match: { $expr: { $gt: ["$sales", "$quota"] } } },
  { $group: { _id: "$rep_office" } },
  { $count: "offices_with_over_quota" }
])
```

9. **Show the name, sales, and office for each salesperson. Order by increasing sales:**

```
db.salesreps.find({}, { name: 1, sales: 1, rep_office: 1 }).sort({ sales: 1 })
```

10. **List all the companies who have ordered any size widget, and the widget they ordered:**

```
db.orders.aggregate([
  {
    $lookup: {
      from: "products",
      let: { prod_mfr: "$product.mfr", prod_id: "$product.product_id" },
      pipeline: [
        { $match: { $expr: { $and: [{ $eq: ["$_id.mfr_id", "$$prod_mfr"] }, { $eq: ["$_id.product_id", "$$prod_id"] }, { $regexMatch: { input: "$description", regex: "Widget" } }] } } },
        { $project: { description: 1 } }
      ],
      as: "widget_info"
    }
  },
  { $unwind: "$widget_info" },
  {
    $lookup: {
      from: "customers",
      localField: "cust",
      foreignField: "_id",
      as: "customer_info"
    }
  },
  { $unwind: "$customer_info" },
  { $project: { "customer_info.company": 1, "widget_info.description": 1 } }
])
```

11. **List the city, region, and amount that sales are over/under target for each office:**

```
db.offices.aggregate([
  {
    $project: {
      city: 1,
      region: 1,
      difference_from_target: { $subtract: ["$sales", "$target"] }
    }
  }
])
```

12. **What is the total number of each part that has been ordered?**

```
db.orders.aggregate([
  {
    $group: {
      _id: "$product.product_id",
      total_qty_ordered: { $sum: "$product.qty" }
    }
  }
])
```

13. **List the salespeople, the city they work in, and the manager of the office in which they work:**

```
db.salesreps.aggregate([
  {
    $lookup: {
      from: "offices",
      localField: "rep_office",
      foreignField: "_id",
      as: "office_info"
    }
  },
  { $unwind: "$office_info" },
  { $project: { name: 1, "office_info.city": 1, "office_info.mgr": 1 } }
])
```

14. **List all orders showing order number, amount, customer name, and the customer’s credit limit where the order was greater than $20,000:**

```
db.orders.aggregate([
  { $match: { "product.amount": { $gt: 20000 } } },
  {
    $lookup: {
      from: "customers",
      localField: "cust",
      foreignField: "_id",
      as: "customer_info"
    }
  },
  { $unwind: "$customer_info" },
  {
    $project: {
      _id: 1,
      "product.amount": 1,
      "customer_info.company": 1,
      "customer_info.credit_limit": 1
    }
  }
])
```

15. **Are there any customers who are over their credit limit? If so, list the customer, the total amount the customer has on order, and the credit limit:**

```
db.orders.aggregate([
  {
    $group: {
      _id: "$cust",
      total_amount: { $sum: "$product.amount" }
    }
  },
  {
    $lookup: {
      from: "customers",
      localField: "_id",
      foreignField: "_id",
      as: "customer_info"
    }
  },
  { $unwind: "$customer_info" },
  { $match: { $expr: { $gt: ["$total_amount", "$customer_info.credit_limit"] } } },
  {
    $project: {
      "customer_info.company": 1,
      total_amount: 1,
      "customer_info.credit_limit": 1
    }
  }
])
```

16. **List the salespeople with a higher quota than their manager:**

```
db.salesreps.aggregate([
  {
    $lookup: {
      from: "salesreps",
      localField: "manager",
      foreignField: "_id",
      as: "manager_info"
    }
  },
  { $unwind: "$manager_info" },
  { $match: { $expr: { $gt: ["$quota", "$manager_info.quota"] } } },
  {
    $project: {
      name: 1,
      quota: 1,
      "manager_info.name": 1,
      "manager_info.quota": 1
    }
  }
])
```

17. **List salespeople who work in different offices than their managers, show the name and office where each works:**

```
db.salesreps.aggregate([
  {
    $lookup: {
      from: "salesreps",
      localField: "manager",
      foreignField: "_id",
      as: "manager_info"
    }
  },
  { $unwind: "$manager_info" },
  { $match: { $expr: { $ne: ["$rep_office", "$manager_info.rep_office"] } } },
  {
    $project: {
      name: 1,
      rep_office: 1,
      "manager_info.name": 1,
      "manager_info.rep_office": 1
    }
  }
])
```

18. **What is the total order size for each salesperson? Order by increasing sales:**

```
db.orders.aggregate([
  {
    $group: {
      _id: "$rep",
      total_sales: { $sum: "$product.amount" }
    }
  },
  {
    $lookup: {
      from: "salesreps",
      localField: "_id",
      foreignField: "_id",
      as: "salesrep_info"
    }
  },
  { $unwind: "$salesrep_info" },
  {
    $project: {
      "salesrep_info.name": 1,
      total_sales: 1
    }
  },
  { $sort: { total_sales: 1 } }
])
```

19. **List all the customers whose sales representative is a manager. Arrange by increasing company name:**

```
db.customers.aggregate([
 {
  $lookup: {
   from: "salesreps",
   localField: "cust_rep",
   foreignField: "_id",
   as: "salesrep_info"
  }
 },
 { $unwind: "$salesrep_info" },
 { $match: { "salesrep_info.title": { $regex: "Mrg" } } },
 {
  $project: {
   company: 1,
   "salesrep_info.name": 1
  }
 },
 { $sort: { company: 1 } }
])
```

20. **What is the total order size for each salesperson whose orders total more than $30,000?**

```
db.orders.aggregate([
 {
  $group: {
   _id: "$rep",
   total_sales: { $sum: "$product.amount" }
  }
 },
 { $match: { total_sales: { $gt: 30000 } } },
 {
    $lookup: {
      from: "salesreps",
      localField: "_id",
      foreignField: "_id",
      as: "salesrep_info"
    }
  },
  { $unwind: "$salesrep_info" },
  {
    $project: {
      "salesrep_info.name": 1,
      total_sales: 1
    }
  }
])
```

21. **List the offices where the sales target for the office exceeds the sum of the individual salespeople’s quotas:**

```
db.salesreps.aggregate([
  {
    $group: {
      _id: "$rep_office",
      total_quota: { $sum: "$quota" }
    }
  },
  {
    $lookup: {
      from: "offices",
      localField: "_id",
      foreignField: "_id",
      as: "office_info"
    }
  },
  { $unwind: "$office_info" },
  { $match: { $expr: { $gt: ["$office_info.target", "$total_quota"] } } },
  {
    $project: {
      "office_info.city": 1,
      "office_info.target": 1,
      total_quota: 1
    }
  }
])
```

22. **List the salespeople whose quotas are equal to or higher than the target of the Atlanta sales office:**

```
db.offices.aggregate([
  { $match: { city: "Atlanta" } },
  {
    $lookup: {
      from: "salesreps",
      localField: "_id",
      foreignField: "rep_office",
      as: "salesrep_info"
    }
  },
  { $unwind: "$salesrep_info" },
  { $match: { $expr: { $gte: ["$salesrep_info.quota", "$target"] } } },
  {
    $project: {
      "salesrep_info.name": 1,
      "salesrep_info.quota": 1
    }
  }
])
```

23. **List the salespeople who do not work in offices managed by Larry Fitch**

```
db.salesreps.aggregate([
 {
   $lookup: {
     from: "offices",
     localField: "rep_office",
     foreignField: "_id",
     as: "office_info"
   }
 },
 { $unwind: "$office_info" },
 { $match: { "office_info.mgr": { $ne: "108" } } },
 {
   $project: {
     name: 1,
     rep_office: 1
   }
 }
])
```

24. **List the products for which an order of $25,000 or more has been received:**

```
db.orders.aggregate([
 { $match: { "product.amount": { $gte: 25000 } } },
 {
   $lookup: {
     from: "products",
     let: { prod_mfr: "$product.mfr", prod_id: "$product.product_id" },
     pipeline: [
       { $match: { $expr: { $and: [{ $eq: ["$_id.mfr_id", "$$prod_mfr"] }, { $eq: ["$_id.product_id",
"$$prod_id"] }] } } },
       { $project: { description: 1 } }
   ],
   as: "product_info"
  }
 },
 { $unwind: "$product_info" },
 {
  $project: {
   "product_info.description": 1
  }
 }
])
```

25. **List the companies who placed an order with a sales rep that is not the sales rep that usually calls on them. Include the names of the sales reps, indicating by attribute name who took the order:**

```
db.orders.aggregate([
 {
  $lookup: {
   from: "customers",
   localField: "cust",
   foreignField: "_id",
   as: "customer_info"
  }
 },
 { $unwind: "$customer_info" },
 { $match: { $expr: { $ne: ["$customer_info.cust_rep", "$rep"] } } },
 {
  $lookup: {
   from: "salesreps",
   localField: "customer_info.cust_rep",
   foreignField: "_id",
   as: "usual_salesrep_info"
  }
  },
  { $unwind: "$usual_salesrep_info" },
  {
    $lookup: {
      from: "salesreps",
      localField: "rep",
      foreignField: "_id",
      as: "order_taking_rep_info"
    }
  },
  { $unwind: "$order_taking_rep_info" },
  {
    $project: {
      "customer_info.company": 1,
      "usual_salesrep_info.name": "usual_sales_rep",
      "order_taking_rep_info.name": "order_taking_rep"
    }
  }
])
```