Project #2

November 10, 2022

0.0.1 Mark Attwood IS 257 650

[1]:
```
from pymongo import MongoClient
import numpy as np
import pandas as pd
from datetime import datetime
```

[2]:
```
client = MongoClient()
```

[3]:
```
type(client)
```

[3]:
```
pymongo.mongo_client.MongoClient
```

[4]:
```
for daba in client.list_databases():
    print(daba, type(daba))
```

```
{'name': 'Attwood', 'sizeOnDisk': 724992, 'empty': False} <class 'dict'>
{'name': 'DBA', 'sizeOnDisk': 380928, 'empty': False} <class 'dict'>
{'name': 'admin', 'sizeOnDisk': 40960, 'empty': False} <class 'dict'>
{'name': 'config', 'sizeOnDisk': 98304, 'empty': False} <class 'dict'>
{'name': 'local', 'sizeOnDisk': 81920, 'empty': False} <class 'dict'>
{'name': 'nobel', 'sizeOnDisk': 376832, 'empty': False} <class 'dict'>
```

[5]:
```
db = client["DBA"]
```

[6]:
```
db.list_collection_names(filter = {})
```

[6]:
```
['Products', 'Customers']
```

[7]:
```
customer = db.Customers.find_one()
product = db.Products.find_one()
```

[8]:
```
print(list(customer.keys()))
print(list(product.keys()))
```

```
['_id', 'first_name', 'last_name', 'phone', 'acct_created', 'cust_id',
'shipping_addresses', 'primary_address', 'payment_method', 'order']
['_id', 'product_id', 'weight', 'count', 'cost', 'center_type',
```

```
'chocolate_type', 'dietary', 'bar_type', 'filling_type', 'qoh',
'reorder_indicator', 'reviews', 'product_name']
```

[9]:
```
for key in product.keys():
    print(f"{key}: {product[key]}\n")
```

```
_id: 63699d696532e28b6fd8df26

product_id: 1FE8EC

weight: 1 oz

count: 1

cost: 22.5

center_type: fruit

chocolate_type: None

dietary: None

bar_type: None

filling_type: marshmallow

qoh: 260

reorder_indicator: 0

reviews: [{'customer_id': 'trtrec1615', 'rating': 0.5, 'review': 'Sint sunt anim
labore aliqua non ipsum culpa officia esse voluptate magna in eu ex. Sed ipsum
sint in dolore officia culpa consectetur magna tempor ea irure aliqua aliquip
laborum sunt. Dolor aute excepteur nostrud est consectetur dolor ex velit irure
laborum amet lorem commodo ut consectetur irure.'}, {'customer_id':
'comume7611', 'rating': 4, 'review': 'Excepteur ut dolore mollit excepteur nisi
enim occaecat cupidatat minim in nulla reprehenderit. Irure ea dolor ea veniam
fugiat dolore ut enim occaecat cupidatat ut in aute.'}, {'customer_id':
'lashor4447', 'rating': 2, 'review': 'In duis sint reprehenderit consequat qui
in ut aliqua enim. Ipsum tempor ad in cillum nisi culpa velit ullamco ut.'},
{'customer_id': 'amwine6355', 'rating': 4.5, 'review': 'Eu fugiat culpa veniam
amet sit in et duis dolor laboris voluptate. Officia id qui velit eu sunt culpa
sit nulla commodo tempor occaecat mollit dolor consequat pariatur.'},
{'customer_id': 'debenl5286', 'rating': 3, 'review': 'Nisi ad reprehenderit
reprehenderit aliqua incididunt adipiscing sunt cillum duis in nostrud occaecat
nostrud. Amet amet aliquip in cupidatat aute qui sunt ad ipsum occaecat ut
consectetur sunt officia dolor ut irure ad.'}, {'customer_id': 'ligrot1266',
'rating': 3, 'review': 'Excepteur consectetur dolor enim non esse elit ex
```

```
pariatur ipsum do proident. In sed mollit commodo deserunt esse consectetur ea
eu aliquip duis aliqua dolore fugiat fugiat excepteur.'}, {'customer_id':
'cltomi3359', 'rating': 2.5, 'review': 'Aliqua minim qui dolore laboris amet qui
enim in proident veniam consequat.'}]

product_name: Lollypop Fruit Marshmallow Jackolanterns
```

[10]:
```
for key in customer.keys():
    print(f"{key}: {customer[key]}\n")
```

```
_id: 636975cf92e8018b85258ad7

first_name: Johnny

last_name: Trahan

phone: 370-977-9493

acct_created: 2021-08-04 00:00:00

cust_id: jotrah9493

shipping_addresses: [{'street_address': '8045 Alberta St', 'city':
'Jonesborough', 'state': 'TN', 'zip': '37659', 'address_id': '7C28'}]

primary_address: {'street_address': '8045 Alberta St', 'city': 'Jonesborough',
'state': 'TN', 'zip': '37659', 'address_id': '7C28'}

payment_method: {'card_number': '3676-9161-2729-7373', 'card_type': 'American
Express', 'exp_date': datetime.datetime(2023, 4, 27, 0, 0), 'cvv': 659,
'billing_address': {'street_address': '8045 Alberta St', 'city': 'Jonesborough',
'state': 'TN', 'zip': '37659', 'address_id': '7C28'}}

order: [{'order_id': '6C1CCE9803', 'order_date': datetime.datetime(2022, 3, 23,
0, 0), 'order_items': [{'product_name': 'Sour Chocolate And Walnuts Assorted
Chocolate Twists', 'price': 43.75, 'count': 1}, {'product_name': 'Double Dark
Chocolate Brittle', 'price': 3.5, 'count': 1}], 'address_id': '7C28',
'shipping_cost': 15.94, 'promotion': 'None'}, {'order_id': '30733C9FFC',
'order_date': datetime.datetime(2022, 6, 24, 0, 0), 'order_items':
[{'product_name': 'Mini Dark Chocolate Marshmallow Krispys', 'price': 17.5,
'count': 2}], 'address_id': '7C28', 'shipping_cost': 23.54, 'promotion':
'None'}, {'order_id': '765213BB98', 'order_date': datetime.datetime(2022, 8, 21,
0, 0), 'order_items': [{'product_name': 'Assorted Bats', 'price': 11.25,
'count': 2}], 'address_id': '7C28', 'shipping_cost': 36.12, 'promotion':
{'code': 'E2E7CCCF2022', 'discount': 0.3, 'start_date': datetime.datetime(2020,
10, 23, 0, 0), 'end_date': datetime.datetime(2022, 11, 15, 0, 0)}}]
```

1 Part A: Create queries to provide the following key metrics:

1.1 a. Generate a query to show all your customers sorted by customer last name

[11]:
```
last_names = np.asarray(db.Customers.distinct("last_name"))
```

[12]:
```
print(last_names[:5])
print(len(last_names))
```

```
['Aaland' 'Abarr' 'Abdulkarim' 'Abebe' 'Acquilano']
499
```

1.2 b. Generate a query to show the number customers by state and sorted by state in ascending order

[13]:
```
# From original documents
list(db.Customers.aggregate([
    {"$project": {"first_address": {"$first": "$shipping_addresses"}}},
    {"$project": {"state": "$first_address.state"}},
    {"$group": {"_id": "$state", "Count": {"$sum": 1}}},
    {"$sort": {"_id": 1}},
    {"$limit": 10}
]))
```

[13]:
```
[{'_id': 'AK', 'Count': 3},
 {'_id': 'AL', 'Count': 14},
 {'_id': 'AR', 'Count': 10},
 {'_id': 'AZ', 'Count': 7},
 {'_id': 'CA', 'Count': 24},
 {'_id': 'CO', 'Count': 10},
 {'_id': 'CT', 'Count': 4},
 {'_id': 'DC', 'Count': 1},
 {'_id': 'DE', 'Count': 1},
 {'_id': 'FL', 'Count': 22}]
```

[14]:
```
# From appended documents (added a non-array address field)
list(db.Customers.aggregate([
    {"$group": {"_id": "$primary_address.state", "Count": {"$sum": 1}}},
    {"$sort": {"_id": 1}},
    {"$limit": 10}
]))
```

[14]:
```
[{'_id': 'AK', 'Count': 3},
 {'_id': 'AL', 'Count': 14},
 {'_id': 'AR', 'Count': 10},
 {'_id': 'AZ', 'Count': 7},
 {'_id': 'CA', 'Count': 24},
```

```
 {'_id': 'CO', 'Count': 10},
 {'_id': 'CT', 'Count': 4},
 {'_id': 'DC', 'Count': 1},
 {'_id': 'DE', 'Count': 1},
 {'_id': 'FL', 'Count': 22}]
```

1.3 c. Generate a query to show all your products

[15]:
```
product_names = db.Products.distinct("product_name")
```

[16]:
```
product_names[1:20]
```

[16]:
```
['Assorted Bats',
 'Assorted Chocolate Bar Caramel Square',
 'Assorted Fruit Dark Chocolate Ganache Bats',
 'Assorted Nut And Chews Dark Chocolate Bats',
 'Assorted Soft Dark Chocolate Balls',
 'Assorted White Chocolate Dark Chocolate Ganache Bar',
 'Birthday Assorted Chocolate Nut Free Brittle',
 'Birthday Assorted Chocolate Sugar Free Krispys',
 'Birthday Chocolate Bar Bar',
 'Birthday Creme Brittle',
 'Birthday Milk Chocolate Jackolanterns',
 'Birthday Nut And Chews Krispys',
 'Birthday Soft Milk Chocolate Brittle',
 'Birthday Twists',
 'Butterscotch Assorted Chocolate Candy Bar Twists',
 'Butterscotch Chocolate And Walnuts Milk Chocolate Jackolanterns',
 'Butterscotch Chocolate Bar Dark Chocolate Ganache Krispys',
 'Butterscotch Dark Chocolate Balls',
 'Butterscotch Dark Chocolate Nut Free Twists']
```

1.4 d) Generate a query to show all products by names and count sold for a specific date range

[17]:
```
# Count of each products sold in 2021
list(db.Customers.aggregate([
    {"$match": {"order.order_date": {"$gte": datetime(2021, 1, 1)},
                "order.order_date": {"$lt": datetime(2022, 1, 1)}}},
    {"$unwind": "$order"},
    {"$unwind": "$order.order_items"},

    {"$group": {"_id": "$order.order_items.product_name", "total": {"$sum":␣
 ↪"$order.order_items.count"}}},
    {"$sort": {"total": -1}},
    {"$limit": 10}
```

```
]))
```

[17]:
```
[{'_id': 'Gold Twists', 'total': 71},
 {'_id': 'Mini Balls', 'total': 68},
 {'_id': 'Gold Chocolate Bar Caramel Krispys', 'total': 65},
 {'_id': 'Mini Chocolate And Walnuts Milk Chocolate Ganache Bar', 'total': 63},
 {'_id': 'Triple Chocolate Bar Milk Chocolate Ganache Leaves', 'total': 59},
 {'_id': 'Cinnamon Fruit Balls', 'total': 58},
 {'_id': 'Mini Dark Chocolate Milk Chocolate Ganache Square', 'total': 57},
 {'_id': 'Butterscotch Square', 'total': 52},
 {'_id': 'Maple Jackolanterns', 'total': 50},
 {'_id': 'Triple Nut And Chews Dark Chocolate Ganache Bundle', 'total': 46}]
```

2 Part B: More Advanced Aggregation Piplelines

2.1 e. Generate a report to show the bestselling product by month, quarter and year

10 top selling products among all years

[18]:
```
list(db.Customers.aggregate([
    {"$unwind": "$order"},
    {"$unwind": "$order.order_items"},
    {"$group": {"_id": "$order.order_items.product_name",
                "revenue": {"$sum": {"$multiply":["$order.order_items.count",␣
 ↪"$order.order_items.price"]}}}},
    {"$sort": {"revenue": -1}},
    {"$limit": 10}
]))
```

[18]:
```
[{'_id': 'Lollypop White Chocolate Creme Bar', 'revenue': 6064.25},
 {'_id': 'Maple Jackolanterns', 'revenue': 4510.5},
 {'_id': 'Birthday Milk Chocolate Jackolanterns', 'revenue': 3965.0},
 {'_id': 'Assorted Fruit Dark Chocolate Ganache Bats', 'revenue': 3952.5},
 {'_id': 'Sour Chocolate And Walnuts Assorted Chocolate Twists',
  'revenue': 3937.5},
 {'_id': 'Cinnamon Fruit Balls', 'revenue': 3744},
 {'_id': 'Butterscotch Square', 'revenue': 3538.75},
 {'_id': 'Assorted Nut And Chews Dark Chocolate Bats', 'revenue': 3500.0},
 {'_id': 'Double Chocolate And Walnuts Egg Free Leaves', 'revenue': 3448.75},
 {'_id': 'Triple Nut And Chews White Chocolate Bundle', 'revenue': 3423.75}]
```

Total sales for each year

[19]:
```
list(db.Customers.aggregate([
    {"$unwind": "$order"},
    {"$unwind": "$order.order_items"},
    {"$project": {"product_name": "$order.order_items.product_name",
```

```
                    "price": "$order.order_items.price",
                    "quantity": "$order.order_items.count",
                    "revenue": {"$multiply":["$order.order_items.count", "$order.
 ↪order_items.price"]},
                    "order_date": "$order.order_date"}},
    {"$bucket": {
        "groupBy": "$order_date",
        "boundaries": [datetime(2020, 1, 1), datetime(2021, 1, 1),␣
 ↪datetime(2022, 1, 1), datetime(2023, 1, 1)],
        "output": {
                "revenue": {"$sum": {"$multiply":["$quantity", "$price"]}}
        }
    }},
]))
```

[19]:
```
[{'_id': datetime.datetime(2020, 1, 1, 0, 0), 'revenue': 356.5},
 {'_id': datetime.datetime(2021, 1, 1, 0, 0), 'revenue': 36758.5},
 {'_id': datetime.datetime(2022, 1, 1, 0, 0), 'revenue': 136176.25}]
```

[20]:
```
docs = list(db.Customers.aggregate([
    {"$unwind": "$order"},
    {"$unwind": "$order.order_items"},
    {"$project": {"_id": 0,
                    "product_name": "$order.order_items.product_name",
                    "price": "$order.order_items.price",
                    "quantity": "$order.order_items.count",
                    "revenue": {"$multiply":["$order.order_items.count", "$order.
 ↪order_items.price"]},
                    "order_date": "$order.order_date",
                    "year": {"$year": "$acct_created" },
                    "month": {"$month": "$order.order_date"},
                    "quarter": {"$cond": [{ "$lte": [{ "$month": "$order.
 ↪order_date" }, 3] }, "1st Quarter",
                                {"$cond": [{ "$lte": [{ "$month": "$order.
 ↪order_date" }, 6] }, "2nd Quarter",
                                {"$cond": [{ "$lte": [{ "$month": "$order.
 ↪order_date" }, 9] }, "3rd Quarter",
                                        "4th Quarter"],
                },],},],},},
    },
    {"$group": {"_id":  ["$product_name", "$quarter", "$year"],
                "revenue": {"$sum": {"$multiply":["$quantity", "$price"]}}}}},
    #{"$sort": { "revenue": -1,"year": 1}},
    #{"$limit": 10}
]))
```

[21]:
```
index = -1
best_items = []
years = [2020, 2021, 2022]
quarters = ["1st Quarter", "2nd Quarter", "3rd Quarter", "4th Quarter"]

for year in years:
    for quarter in quarters:
        best_revenue = 0
        index += 1
        best_items.append("temp")
        for doc in docs:
            if doc['_id'][1] == quarter and doc['_id'][2] == year:
                if doc["revenue"] > best_revenue:
                    best_revenue = doc["revenue"]
                    best_items[index] = doc

print("Best Revenue Generator by Quarter \n")
x = enumerate(best_items)
for i in range(len(best_items)):
    item = next(x)
    print(f"{item[1]['_id'][1]}, {item[1]['_id'][2]} \n{item[1]['_id'][0]}:␣
 ↪${item[1]['revenue']}\n")
```

```
Best Revenue Generator by Quarter

1st Quarter, 2020
Lollypop White Chocolate Creme Bar: $525.25

2nd Quarter, 2020
Mini Nut And Chews Leaves: $397.5

3rd Quarter, 2020
Lollypop White Chocolate Creme Bar: $477.5

4th Quarter, 2020
Sour Chocolate And Walnuts Assorted Chocolate Twists: $306.25

1st Quarter, 2021
Assorted Fruit Dark Chocolate Ganache Bats: $1069.5

2nd Quarter, 2021
Lollypop White Chocolate Creme Bar: $1337.0

3rd Quarter, 2021
Maple Jackolanterns: $1255.5

4th Quarter, 2021
```

```
Maple Jackolanterns: $1023.0

1st Quarter, 2022
Butterscotch Sugar Free Dark Chocolate Ganache Twists: $530.75

2nd Quarter, 2022
Mini Marshmallow Brittle: $830.0

3rd Quarter, 2022
Sour Chocolate And Walnuts Assorted Chocolate Twists: $1356.25

4th Quarter, 2022
Assorted Fruit Dark Chocolate Ganache Bats: $1255.5
```

[22]:
```
index = -1
worst_items = []
years = [2020, 2021, 2022]
quarters = ["1st Quarter", "2nd Quarter", "3rd Quarter", "4th Quarter"]

for year in years:
    for quarter in quarters:
        worst_revenue = 100000
        index += 1
        worst_items.append("temp")
        for doc in docs:
            if doc['_id'][1] == quarter and doc['_id'][2] == year:
                if doc["revenue"] < worst_revenue:
                    worst_revenue = doc["revenue"]
                    worst_items[index] = doc

print("Worst Revenue Generator by Quarter \n")
x = enumerate(worst_items)
for i in range(len(worst_items)):
    item = next(x)
    print(f"{item[1]['_id'][1]}, {item[1]['_id'][2]} \n{item[1]['_id'][0]}:␣
 ↪${item[1]['revenue']}\n")
```

```
Worst Revenue Generator by Quarter

1st Quarter, 2020
Double Chocolate And Walnuts Kosher Jackolanterns: $2.25

2nd Quarter, 2020
Cinnamon Nut And Chews Bats: $1.5

3rd Quarter, 2020
Sour Nut And Chews Assorted Chocolate Twists: $3
```

```
4th Quarter, 2020
Double Chocolate And Walnuts Kosher Jackolanterns: $2.25

1st Quarter, 2021
Double Chocolate And Walnuts Kosher Jackolanterns: $4.5

2nd Quarter, 2021
Cinnamon Nut And Chews Bats: $10.5

3rd Quarter, 2021
Assorted White Chocolate Dark Chocolate Ganache Bar: $13

4th Quarter, 2021
Maple Chocolate And Caramel Chocolate Bar Square: $3.25

1st Quarter, 2022
Triple Chocolate Bar Milk Chocolate Ganache Leaves: $1.25

2nd Quarter, 2022
Assorted White Chocolate Dark Chocolate Ganache Bar: $1

3rd Quarter, 2022
Double Chocolate And Walnuts Kosher Jackolanterns: $4.5

4th Quarter, 2022
Assorted White Chocolate Dark Chocolate Ganache Bar: $4
```

2.2 Generate a report to show the total revenue for each product sold between January 2022 and June 2022. Extra credit: 5 points

[23]:
```
list(db.Customers.aggregate([
    {"$match": {"order.order_date": {"$gte": datetime(2022, 1, 1)},
                "order.order_date": {"$lt": datetime(2022, 7, 1)}}},
    {"$unwind": "$order"},
    {"$unwind": "$order.order_items"},
    {"$project": {"_id": 0,
                  "product_name": "$order.order_items.product_name",
                  "price": "$order.order_items.price",
                  "quantity": "$order.order_items.count",
                  "revenue": {"$multiply":["$order.order_items.count", "$order.
 ↪order_items.price"]}
                 }
    },

    {"$group": {"_id": "$product_name", "total_sales": {"$sum": "$revenue"}}},
```

```
    {"$sort": {"total": -1}},
    {"$limit": 10}
]))
```

[23]:
```
[{'_id': 'Maple Chocolate And Walnuts Dark Chocolate Ganache Bats',
  'total_sales': 1729.0},
 {'_id': 'Double Chocolate And Walnuts Kosher Jackolanterns',
  'total_sales': 101.25},
 {'_id': 'Cinnamon Nut And Chews Bats', 'total_sales': 66.0},
 {'_id': 'Double White Chocolate Candy Bar Bats', 'total_sales': 2242.5},
 {'_id': 'Mini Marshmallow Brittle', 'total_sales': 2656.0},
 {'_id': 'Sour Bar', 'total_sales': 949.0},
 {'_id': 'Assorted Bats', 'total_sales': 630.0},
 {'_id': 'Sour Fruit White Chocolate Krispys', 'total_sales': 1433.25},
 {'_id': 'Gold Nut And Chews Krispys', 'total_sales': 832.5},
 {'_id': 'Gold Nut And Chews Bats', 'total_sales': 1551}]
```

3 Extra Queries

3.1 Which 3 customers spent the most on shipping in 2021?

[24]:
```
list(db.Customers.aggregate([

    {"$unwind": "$order"},
    {"$match": {"order.order_date": {"$gte": datetime(2021, 1, 1)},
                "order.order_date": {"$lt": datetime(2022, 1, 1)}}},
    {"$group": {"_id": {"$concat": ["$first_name", " ", "$last_name"]},
                "shipping_costs": {"$sum": "$order.shipping_cost"}}},
    {"$sort": {"shipping_costs": -1}},
    {"$limit": 3}
]))
```

[24]:
```
[{'_id': 'Camryn Staal', 'shipping_costs': 171.75},
 {'_id': 'Hessie Casem', 'shipping_costs': 169.76},
 {'_id': 'Hugo Rogotzke', 'shipping_costs': 157.49}]
```

3.2 How many new accounts created each year?

[25]:
```
list(db.Customers.aggregate([
    {"$project": {
        "name": {"$concat": ["$first_name", " ", "$last_name"]},
        "year": {"$year": "$acct_created"}}},
    {"$group": {"_id": "$year", "accounts created": {"$sum": 1}}},
]))
```

[25]:
```
[{'_id': 2021, 'accounts created': 265},
 {'_id': 2022, 'accounts created': 196},
 {'_id': 2020, 'accounts created': 39}]
```

3.3 Which 3 items in 2021 had the greatest count sold?

[26]:
```
list(db.Customers.aggregate([
    {"$match": {"order.order_date": {"$gte": datetime(2021, 1, 1)},
                "order.order_date": {"$lt": datetime(2022, 1, 1)}}},
    {"$unwind": "$order"},
    {"$unwind": "$order.order_items"},

    {"$group": {"_id": "$order.order_items.product_name", "total": {"$sum":␣
 ↪"$order.order_items.count"}}},
    {"$sort": {"total": -1}},
    {"$limit": 3}
]))
```

[26]:
```
[{'_id': 'Gold Twists', 'total': 71},
 {'_id': 'Mini Balls', 'total': 68},
 {'_id': 'Gold Chocolate Bar Caramel Krispys', 'total': 65}]
```

3.4 How many total items has the company sold?

[27]:
```
# How many items sold?

docs = db.Customers.find({}, ["order"])

n_items_sold = 0

for doc in docs:
    for order in doc["order"]:
        for item in order["order_items"]:
            n_items_sold += item["count"]

print(n_items_sold)
```

```
7116
```