Homework2

Part 1

A) In the users collection, what is the data type for the field name?

Name has an object data type of a document with 2 nested fields including first and last, and both of them are string.

B) From that same collection, what is the type for the field named phones?

Phones has an array data type of documents with 2 nested fields of kind and number, and both of them are string.

C) In the orders collection, what is the type for the field time_placed? ¶

Time_placed has a string data type.

D) From the same collection, what is the type for the field name selling_price?

Selling_price has a double data type.

E) In the stores collection, what are the min, average and max array length for the categories field?

The categories field has a min of 3, average of 9.68, and max of 16.

Part 2

In [42]:

```
from pymongo import MongoClient
from bson.json_util import dumps

client = MongoClient('mongodb+srv://<>:<>@cluster0.yokf5.mongodb.net/\
test?authSource=admin&replicaSet=atlas-l3ugmu-shard-0&readPreference=primary&appn
ame=MongoDB%20Compass&ssl=true')
shopalot = client.shopalot

# Part A
filter = {'time_placed': '2020-05-26T00:30:28.000Z'}
project = {'_id': 0, 'total_price': 1}

result = shopalot.orders.find(filter = filter, projection = project)
print("Part A:", dumps(result))
print()
```

```
Part A: [{"total_price": 50.7}]
```

In [43]:

```
# Part B
filter = {'category': 'Bread & Bakery', 'list_price': {'$gt': 3}}
sort = list({'list_price': -1}.items())
limit = 10

result = shopalot.products.find(filter = filter, sort = sort, limit = limit)
print("Part B:", dumps(result, indent=2))
print()
```

```
Part B: [
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa76ea"
    },
    "product_id": "MUFUS",
    "category": "Bread & Bakery",
    "name": "Daves Killer Bread Organic White Bread Done Right - 24 Oz",
    "description": "Sandwich Bread",
    "list_price": 6.49
  },
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa77cb"
    },
    "product_id": "G08JV",
    "category": "Bread & Bakery",
    "name": "Fresh Baked Blueberry Muffins - 4 Count",
    "description": "Muffins & Scones",
    "list_price": 5.99
  },
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa77ae"
    },
    "product_id": "UF431",
    "category": "Bread & Bakery",
    "name": "Daves Killer Bread Organic Light Whole Grain - 20.5 Oz",
    "description": "Sandwich Bread",
    "list_price": 5.89
  },
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa76c9"
    },
    "product_id": "6WLH2",
    "category": "Bread & Bakery",
    "name": "Sara Lee Bagels Deluxe Pre-Sliced Plain - 6-22 Oz",
    "description": "Bagels",
    "list_price": 4.99
  },
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa7778"
    },
    "product_id": "CNPR0",
    "category": "Bread & Bakery",
    "name": "Entenmanns Minis Pound Cake 6 Count - 9.25 Oz",
    "description": "Bakery Snacks",
    "list_price": 4.99
  },
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa778f"
    },
    "product_id": "LEEVP",
    "category": "Bread & Bakery",
    "name": "Thomas Bagels Plain Pre Sliced 6 Count - 20 Oz",
    "description": "Bagels",
    "list_price": 4.99
  },
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa7798"
    },
    "product_id": "5KX5O",
    "category": "Bread & Bakery",
    "name": "Thomas Bagels Everything Pre Sliced 6 Count - 20 Oz",
    "description": "Bagels",
    "list_price": 4.99
  },
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa777a"
    },
    "product_id": "ZQX4M",
    "category": "Bread & Bakery",
    "name": "Thomas Bagels Cinnamon Raisin Pre Sliced 6 Count - 20 Oz",
    "description": "Bagels",
    "list_price": 4.99
  },
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa77c2"
    },
    "product_id": "90N22",
    "category": "Bread & Bakery",
    "name": "Entenmanns Minis Pound Cake 6 Count - 9.25 Oz",
    "description": "Bakery Snacks",
    "list_price": 4.99
  },
  {
    "_id": {
      "$oid": "60943ed1037e74ead1aa7751"
    },
    "product_id": "BNGIK",
    "category": "Bread & Bakery",
    "name": "Oroweat Bread Whole Grains 100% Whole Wheat - 24 Oz",
    "description": "Sandwich Bread",
    "list_price": 4.99
  }
]
```

```
In [44]: # Part C
shopalot.products.update_many({'description': {'$exists': False}}, {'$set': {'description': 'Amazing product'}})

print("Part C:")
for x in shopalot.products.find({'description': 'Amazing product'}):
  print(dumps(x, indent=2))
print()
```

```
Part C:
{
  "_id": {
    "$oid": "60943ed1037e74ead1aa76aa"
  },
  "product_id": "0LLE6",
  "category": "Grains, Pasta, & Sides",
  "name": "Rana Fettuccine - 9 Oz",
  "list_price": 2.99,
  "description": "Amazing product"
}
{
  "_id": {
    "$oid": "60943ed1037e74ead1aa76ab"
  },
  "product_id": "TMD3L",
  "category": "Breakfast & Cereal",
  "name": "Nature Valley Granola Bars Sweet & Salty Nut Peanut - 6-1.2 Oz",
  "list_price": 3.99,
  "description": "Amazing product"
}
{
  "_id": {
    "$oid": "60943ed1037e74ead1aa76ac"
  },
  "product_id": "5Z6NV",
  "category": "Pet Care",
  "name": "Friskies Cat Food Classic Pate Variety Pack Box - 24-5.5 Oz",
  "list_price": 19.99,
  "description": "Amazing product"
}
{
  "_id": {
    "$oid": "60943ed1037e74ead1aa76ad"
  },
  "product_id": "X8YHB",
  "category": "Fruits & Vegetables",
  "name": "Cilantro",
  "list_price": 0.99,
  "description": "Amazing product"
}
```

In [45]:

```
# Part D
filter={'year': {'$lte':2008}, 'make': 'Ford', 'color': 'Gray'}
project={'_id': 0, 'license_plate': 1}

result = shopalot.vehicles.find(filter=filter, projection=project)
print("Part D:", dumps(result, indent=2))
print()
```

```
Part D: [
  {
    "license_plate": "7671 HB"
  },
  {
    "license_plate": "12-M590"
  },
  {
    "license_plate": "3UM 870"
  },
  {
    "license_plate": "N 414858"
  },
  {
    "license_plate": "440 RKZ"
  },
  {
    "license_plate": "ERF 840"
  },
  {
    "license_plate": "109 LLU"
  },
  {
    "license_plate": "CIO7338"
  }
]
```

In [46]:

```
# Part E
filter1 = {'category': 'Baby Care', 'list_price': {'$gt': 3, '$lt': 10}}
filter2 = {'category': 'Pet Care', 'list_price': {'$gt': 10}}

result = shopalot.products.count_documents({'$or':[filter1, filter2]})
print("Part E:", result)
print()
```

```
Part E: 29
```

In [47]:

```
# Part F
filter = {'kind': 'SHOPPER', 'phones.kind': 'HOME', 'capacity': {'$exists': True
}}
project = {'_id': 0, 'user_id': 1, 'name': 1}
sort = list({'name.last': 1}.items())
limit = 5

result = shopalot.users.find(filter=filter, projection=project, sort=sort, limit=
limit)
print("Part F:", dumps(result, indent=2))
print()
```

```
Part F: [
  {
    "name": {
      "first": "Ashley",
      "last": "Aguilar"
    },
    "user_id": "YZNCV"
  },
  {
    "name": {
      "first": "Kyle",
      "last": "Arellano"
    },
    "user_id": "Q7W81"
  },
  {
    "name": {
      "first": "Lor",
      "last": "Arnold"
    },
    "user_id": "5MWNO"
  },
  {
    "name": {
      "first": "Joh",
      "last": "Arnold"
    },
    "user_id": "27PTC"
  },
  {
    "name": {
      "first": "Bri",
      "last": "Brown"
    },
    "user_id": "3C98L"
  }
]
```

```
In [48]: # Part G
result = shopalot.orders.aggregate([
{
        '$match': {
            'customer_id': 'GS65P'
        }
    }, {
        '$group': {
            '_id': '$customer_id',
            'total_price': {
                '$sum': '$total_price'
            }
        }
    }, {
        '$addFields': {
            "customer_id": '$_id'
        }

    },{
        '$project': {
            '_id': 0,
            'customer_id': 1,
            'total_price': 1
        }
    }
])

print("Part G:", dumps(result))
print()
```

```
Part G: [{"total_price": 118.12, "customer_id": "GS65P"}]
```

```
In [49]: # Part H
result = shopalot.orders.aggregate([
    {
        '$group': {
            '_id': '$store_id',
            'num_orders': {
                '$sum': 1
            },
            'AVG': {
                '$avg': '$total_price'
            },
            'Max': {
                '$max': '$total_price'
            }
        }
    }, {
        '$sort': {
            'num_orders': -1
        }
    }, {
        '$limit': 10
    }, {
        '$addFields': {
            "store_id": '$_id'
        }

    },{
        '$project': {
            '_id': 0,
            'store_id': 1,
            'num_orders': 1,
            'AVG': 1,
            'Max': 1
        }
    }

])

print("Part H:", dumps(result, indent=2))
print()
```

```
Part H: [
  {
    "num_orders": 121,
    "AVG": 47.96165289256198,
    "Max": 465.82,
    "store_id": "1RMXY"
  },
  {
    "num_orders": 120,
    "AVG": 53.48791666666667,
    "Max": 316.1,
    "store_id": "2TM62"
  },
  {
    "num_orders": 112,
    "AVG": 58.348482142857144,
    "Max": 343.85,
    "store_id": "70GOX"
  },
  {
    "num_orders": 111,
    "AVG": 53.2027027027027,
    "Max": 376.55,
    "store_id": "17KE2"
  },
  {
    "num_orders": 110,
    "AVG": 55.12918181818182,
    "Max": 492.13,
    "store_id": "49TNX"
  },
  {
    "num_orders": 107,
    "AVG": 53.80271028037384,
    "Max": 291.78,
    "store_id": "FW8ZT"
  },
  {
    "num_orders": 107,
    "AVG": 58.51289719626168,
    "Max": 315.34,
    "store_id": "14VAS"
  },
  {
    "num_orders": 107,
    "AVG": 41.19700934579439,
    "Max": 232.8,
    "store_id": "3II5S"
  },
  {
    "num_orders": 107,
    "AVG": 44.42915887850467,
    "Max": 355.36,
    "store_id": "QTB4W"
  },
  {
    "num_orders": 98,
    "AVG": 58.15091836734694,
    "Max": 325.5,
    "store_id": "28JE8"
  }
```

]

In [50]:

```
# Part I
result = shopalot.orders.aggregate([
    {
        '$match': {
            'total_price': {
                '$gt': 100
            }
        }
    }, {
        '$lookup': {
            'from': 'users',
            'localField': 'customer_id',
            'foreignField': 'user_id',
            'as': 'user_info'
        }
    }, {
        '$unwind': {
            'path': '$user_info'
        }
    },{
        '$addFields': {
            'email': '$user_info.email'
        }
    },{
        '$project': {
            '_id': 0,
            'order_id': 1,
            'total_price': 1,
            'email': 1
        }
    }, {
        '$sample': {
            'size': 5
        }
    }
])

print("Part I:", dumps(result, indent=2))
print()
```

```
Part I: [
  {
    "order_id": "Z3VKF",
    "total_price": 239.1,
    "email": "syl789@gmail.com"
  },
  {
    "order_id": "C5LQT",
    "total_price": 278.4,
    "email": "rush88@gmail.com"
  },
  {
    "order_id": "F8299",
    "total_price": 106.91,
    "email": "andreaturner94@gmail.com"
  },
  {
    "order_id": "YMYFF",
    "total_price": 108.09,
    "email": "white73@aol.com"
  },
  {
    "order_id": "UL9R5",
    "total_price": 110.16,
    "email": "mills43@gmail.com"
  }
]
```

```
In [51]: # Part J Extra Credit
         result = shopalot.orders.aggregate([
             {
                 '$match': {
                     'time_placed': {
                         '$gte': '2018-02-16T02:56:00.000Z',
                         '$lte': '2019-01-15T22:06:43.000Z'
                     }
                 }
             }, {
                 '$group': {
                     '_id': '$customer_id',
                     'min_price': {
                         '$min': '$total_price'
                     },
                     'num_orders': {
                         '$sum': 1
                     }
                 }
             }, {
                 '$lookup': {
                     'from': 'users',
                     'localField': '_id',
                     'foreignField': 'user_id',
                     'as': 'user_info'
                 }
             }, {
                 '$unwind': {
                     'path': '$user_info'
                 }
             }, {
                 '$addFields': {
                     'name': '$user_info.name',
                     'user_id': '$user_info.user_id',
                     'email': '$user_info.email'
                 }
             }, {
                 '$project': {
                     '_id': 0,
                     'user_id': 1,
                     'name': 1,
                     'email': 1,
                     'num_orders': 1,
                     'min_price': 1
                 }
             }, {
                 '$match': {
                     'min_price': {
                         '$gte': 20
                     }
                 }
             }, {
                 '$sort': {
                     'num_orders': -1
                 }
             }, {
                 '$limit': 5
             }
         ])
```

```
print("Part J:", dumps(result, indent=2))
print()
```

```
Part J: [
  {
    "min_price": 22.56,
    "num_orders": 4,
    "name": {
      "first": "Doris",
      "last": "Johnson"
    },
    "user_id": "TY6OR",
    "email": "doris.johnson54107@gmail.com"
  },
  {
    "min_price": 35.4,
    "num_orders": 3,
    "name": {
      "first": "Andrew",
      "last": "Cooper"
    },
    "user_id": "HVVCL",
    "email": "cooper53@gmail.com"
  },
  {
    "min_price": 54.68,
    "num_orders": 3,
    "name": {
      "first": "Rebekah",
      "last": "Flores"
    },
    "user_id": "SYPQR",
    "email": "Floresrebekah3481@yahoo.com"
  },
  {
    "min_price": 36.4,
    "num_orders": 3,
    "name": {
      "first": "Kri",
      "last": "Gomez"
    },
    "user_id": "OE4HV",
    "email": "kri96769@yahoo.com"
  },
  {
    "min_price": 22.65,
    "num_orders": 3,
    "name": {
      "first": "Jil",
      "last": "Tanner"
    },
    "user_id": "BK596",
    "email": "tanner66@gmail.com"
  }
]
```