```
# Install pymongo if applicable
!pip3 install pymongo
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pymongo in /usr/local/lib/python3.10/dist-packages (4.3.3)
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from pymongo) (2.3.0)
```

```
# setup the client. Make sure to remove your password while submitting

from pymongo import MongoClient
from bson.json_util import dumps

client = MongoClient("mongodb+srv://bixingjian19:Bxj20020405@hw3.seazer3.mongodb.net/?retryWrites=true&w=majority")
client.sample
db = client['interchange']
```

```
# Use Python comments to answer Q1 below
# 1.A Document with 2 nested fields.
# 1.B
#    i. _id
#    ii. objectid
#    iii. The ObjectId is a built-in data type used as the default identifier
#         for documents in a collection. It is a 12-byte identifier.
#         They are primarily designed to be globally unique and to provide
#.        a reasonable ordering when sorting documents by their _id field.
#         Timestamp is based on the number of seconds elapsed since the Unix epoch.
#         Machine identifier represent the identifier of the machine or process
#.        that generated the identifier. This helps ensure that each machine or
#.        process produces unique ObjectId values.
#         Process identifier represent a unique identifier for the process generating the identifier.
#.        In a multi-threaded environment, this helps differentiate ObjectId values generated by different threads.
#         Counter (3 bytes)represent an incrementing value that starts with a random number.
#.        This counter is used to ensure uniqueness in cases where multiple ObjectId
#.        values are generated within the same second, by the same process, and on the same machine.
#    iv. yes
# 1.C buyer.user_id, seller.user_id
# 1.D string, png, jpeg, mp4
# 1.E array, 1.09
```

```
# 2A

query = {"item_id": "6N0EN", "is_service": True}
services = db.items.find(query)
print(dumps(services, indent=2))
```


```
[
  {
    "_id": {
      "$oid": "645c187b0d00376d80974bcb"
    },
    "_oid": "5d146dcc-922a-8262-0d50-1c5fb266c238",
    "category": "Others",
    "description": null,
    "frequency": "yearly",
    "item_id": "6N0EN",
    "name": "Nail Paint",
    "price": 1669.16,
    "seller": {
      "user_id": "67EYU",
      "list_date": "2022-09-13"
    },
    "is_good": false,
    "is_service": true
  }
]
```

```
# 2B

query = {"is_good": True}
sort = [("seller.list_date", -1)]
limit = 3

goods = db.items.find(query).sort(sort).limit(limit)
print(dumps(goods, indent=2))

```


```
[
  {
    "_id": {
      "$oid": "645c18710d00376d80972b9b"
    },
    "_oid": "ac5be233-7e83-7b39-b0ce-65a139d9d089",
    "category": "Clothing, Shoes & Jewelry",
    "description": null,
    "item_id": "52X3Q",
    "name": "Necklace",
    "price": 441.53,
    "seller": {
      "user_id": "EXO6D",
      "list_date": "2022-10-29"
    },
    "is_good": true,
    "is_service": false
  },
  {
    "_id": {
      "$oid": "645c18710d00376d809729f0"
    },
    "_oid": "ac5be233-7e83-7b39-b0ce-65a139d9cd28",
    "category": "Electronics",
    "description": null,
    "item_id": "KG6E5",
    "name": "Charger",
    "price": 788.81,
    "seller": {
      "user_id": "KYTZ7",
      "list_date": "2022-10-29"
    },
    "is_good": true,
    "is_service": false
  },
  {
    "_id": {
      "$oid": "645c18710d00376d809729d4"
    },
    "_oid": "ac5be233-7e83-7b39-b0ce-65a139d9ccef",
    "category": "Sports & Outdoors",
    "description": null,
    "item_id": "JSRBX",
    "name": "Basketball",
    "price": 351.15,
    "seller": {
      "user_id": "2LVZE",
      "list_date": "2022-10-29"
    },
    "is_good": true,
    "is_service": false
  }
]
```


```
# 2C

query = {"$and": [{"is_buyer": True}, {"is_seller": True}]}
count = db.users.count_documents(query)
print(dumps(count, indent=2))
```

```
27
```

```
# 2D

service_outlier_query = {"is_service": True, "price": {"$lt": 5}}
good_outlier_query = {"is_good": True, "price": {"$gt": 1999}}
query = {"$or": [service_outlier_query, good_outlier_query]}
outlier_count = db.items.count_documents(query)
print(dumps(outlier_count, indent=2))

```

```
13
```

```
# 2E

buyers_filter = {"is_buyer": True}
add_fields_stage = {"$addFields": {"category_count": {"$size": "$categories"}}}
match_stage = {"$match": {"category_count": {"$gte": 13}}}
project_stage = {"$project": {
    "_id": 0,
    "email": 1,
    "full_name":
     {
        "$cond": [
            {"$eq": [{"$ifNull": ["$name.first", ""]}, ""]},
            "$name.last",
            {"$concat": ["$name.last", ", ", "$name.first"]}
             ]
      }
      }
      }
sort_stage = {"$sort": {"email": 1}}
results = db.users.aggregate([
    {"$match": buyers_filter},
    add_fields_stage,
    match_stage,
    project_stage,
    sort_stage
])
for document in results:
    print(dumps(document, indent=2))

```

```
{
  "email": "03Jordanpatricia91782@gmail.com",
  "full_name": "Jordan"
}
{
  "email": "carrollchristine@hotmail.com",
  "full_name": "Carroll, Christine"
}
{
  "email": "dia70@gmail.com",
  "full_name": "Dixon"
}
{
  "email": "donovandestiny36@gmail.com",
  "full_name": "Donovan, Destiny"
}
{
  "email": "smith.Nicholas@gmail.com",
  "full_name": "Smith, Nicholas"
}
```

```
# 2F
match_stage = {"$match": {"buyer.user_id": {"$exists": True}}}
group_stage = {"$group": {"_id": "$seller.user_id", "total_sold_items": {"$sum": 1}}}
sort_stage = {"$sort": {"total_sold_items": -1}}
denserank_stage = {
        "$setWindowFields": {
            "sortBy": {"total_sold_items": -1},
            "output": {
                "denseRank": {"$denseRank": {}}
            }
        }
    }
limit_stage = {'$match': {'denseRank': {'$lte': 3}}}
project_stage = {"$project": {
            "seller_id": "$_id",
            "_id": 0,
            "seller.user_id": 1,
            "total_sold_items": 1,
            "denseRank": 1
        }
    }

pipeline = [match_stage, group_stage, sort_stage, denserank_stage, limit_stage, project_stage]

cursors = db.items.aggregate(pipeline)
```

```
cursors = db.items.aggregate(pipeline)
print(dumps(list(cursors), indent=2))
```


```
[
  {
    "total_sold_items": 8,
    "denseRank": 1,
    "seller_id": "3FLK5"
  },
  {
    "total_sold_items": 7,
    "denseRank": 2,
    "seller_id": "SZXQH"
  },
  {
    "total_sold_items": 6,
    "denseRank": 3,
    "seller_id": "HVJUT"
  },
  {
    "total_sold_items": 6,
    "denseRank": 3,
    "seller_id": "3KLP3"
  },
  {
    "total_sold_items": 6,
    "denseRank": 3,
    "seller_id": "IJ61L"
  },
  {
    "total_sold_items": 6,
    "denseRank": 3,
    "seller_id": "X9W2Z"
  }
]
```


```
# 2G

lookup_stage = {
        '$lookup': {
            'from': 'ads',
            'localField': 'item_id',
            'foreignField': 'item_id',
            'as': 'ads'
        }
    }
match_stage = {
        '$match': {
            '$and': [
                {"seller.user_id": "XZJXD"},
                {"seller.list_date": {"$lt": "2022-05-30"}},
                {"$or": [{"ads": {"$size": 0}}, {"ads": {"$exists": False}}]},
                {"buyer": {"$exists": False}}
            ]
        }
    }
project_stage = {
        '$project': {
            '_id': 0,
            'item_id': 1
        }
    }
pipeline = [lookup_stage, match_stage, project_stage]

cursors = db.items.aggregate(pipeline)
print(dumps(cursors, indent=2))
```


```
[
  {
    "item_id": "FZ9GO"
  },
  {
    "item_id": "HV9TT"
  }
]
```


```
# 2H

```

```
# Update

match_stage = {"$match": {'$or': [{'content': {'$exists': False}}, {'content': ''}]}}
join_stage = {
            '$lookup': {
                'from': 'items',
                'localField': 'item_id',
                'foreignField': 'item_id',
                'as': 'items'
            }
        }
unwind_stage = {'$unwind': '$items'}
project_stage = {'$project': {'content': {'$cond': [{'$eq': ['$items.is_good', True]},
                                          {'$concat': ['Treat yourself to the best ', '$items.name']},
                                          {'$concat': ['Transform your life with ', '$items.name']}]}
                                          , '_oid': 1, 'ad_id': 1, 'item_id': 1, 'pic_num': 1,
                                          'plan': 1, 'seller': 1}}
set_stage = {'$set': {'content': '$content'}}
merge_stage = {'$merge': {
      'into': 'ads',
      'on': '_id',
      'whenMatched': 'replace',
      'whenNotMatched': 'discard'
    }}
pipeline = [match_stage, join_stage, unwind_stage, project_stage, set_stage, merge_stage]
cursors = db.ads.aggregate(pipeline)
print(dumps(cursors, indent = 2))



# Find queries

print("Treat yourself to the best: ",
      db.ads.count_documents({'content': {'$regex': '^Treat yourself to the best'}}));
print("Transform your life with: ",
      db.ads.count_documents({'content': {'$regex': '^Transform your life with'}}));


```

```
[]
Treat yourself to the best:  1118
Transform your life with:  1128
```

```
# 2I

match_stage = {"$match": {"rating_date": {'$gte': '2022-01-01', '$lt': '2023-01-01'}}}
set_stage = {"$addFields": {
    "weighted_rating": {
        "$add": [
            {"$multiply": [0.2, {"$ifNull": ["$pricing", 0]}]},
            {"$multiply": [0.5, {"$ifNull": ["$quality", 0]}]},
            {"$multiply": [0.3, {"$ifNull": ["$delivery", 0]}]}
        ]
    }
}}
group_stage = {"$group": {
    "_id": "$seller_id",
    "avg_weighted_rating": {"$avg": "$weighted_rating"}
}}
match_stage2 = {"$match": {"avg_weighted_rating": {"$gte": 3.2}}}
project_stage = {"$project": {"_id": 0, "user_id": "$_id", "avg_weighted_rating": 1}}

pipeline = [match_stage, set_stage, group_stage, match_stage2, project_stage]

res = db.ratings.aggregate(pipeline)
print(dumps(res, indent=2))

```


```
[
  {
    "avg_weighted_rating": 3.6,
    "user_id": "VWQZC"
  },
  {
    "avg_weighted_rating": 3.3,
    "user_id": "G7ZYX"
  }
]
```

```
# 2J
matchid_stage = {
        "$match": {
            "item_id": {"$regex": "^A"}
        }
    }
lookupitem_stage = {
        "$lookup": {
            "from": "items",
            "localField": "item_id",
            "foreignField": "item_id",
            "as": "items"
        }
    }
unwinditem_stage = {
        "$unwind": "$items"
    }
lookupuser_stage = {
        "$lookup": {
            "from": "users",
            "localField": "items.seller.user_id",
            "foreignField": "user_id",
            "as": "users"
        }
    }
unwinduser_stage = {
        "$unwind": "$users"
    }
group_stage = {
        "$group": {
            "_id": {
                "item_id": "$items.item_id",
                "email": "$users.email",
                "name": "$items.name"
            },
            "count": {"$sum": 1}
        }
    }
match_count_stage = {
        "$match": {
            "count": {"$eq": 2}
        }
    }
sort_stage = {
        "$sort": {
            "_id.email": 1
        }
    }
limit_stage = {
        "$limit": 2
    }
project_stage = {
        "$project": {
            "name": "$_id.name",
            "email": "$_id.email",
            "_id": 0
        }
    }

pipeline = [matchid_stage, lookupitem_stage, unwinditem_stage, lookupuser_stage,
            unwinduser_stage, group_stage, match_count_stage, sort_stage, limit_stage,
           project_stage ]

res = db.pictures.aggregate(pipeline)
print(dumps(res, indent=2))
```

```
[
  {
    "name": "Chair",
    "email": "039dan8786@gmail.com"
  },
  {
    "name": "Marker",
    "email": "08jes96676@aol.com"
  }
]
```

✓ 0 秒 完成时间：01:22