```
!pip install pymongo
 → Collecting pymongo
              Downloading pymongo-4.8.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (22 kB)
          Collecting dnspython<3.0.0,>=1.16.0 (from pymongo)
             Downloading dnspython-2.6.1-py3-none-any.whl.metadata (5.8 kB)
          Downloading pymongo-4.8.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
                                                                                                1.2/1.2 MB 36.7 MB/s eta 0:00:00
          Downloading dnspython-2.6.1-py3-none-any.whl (307 kB)
                                                                                                 307.7/307.7 kB 17.2 MB/s eta 0:00:00
          Installing collected packages: dnspython, pymongo
          Successfully installed dnspython-2.6.1 pymongo-4.8.0
import pandas as pd
from pymongo import MongoClient
import time
import json
from bson import ObjectId
from datetime import datetime
import os
import time
import json
import sys
\verb|client = MongoClient("mongodb+srv://ayushlokhande0:NufT140wTqeUcfFa@cluster0.803ckl6.mongodb.net/?retryWrites=true\&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.netryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.netryWrites=true&w=majority&appName=Cluster0.803ckl6.mongodb.netryWrites=true&w=majority&appName=Cluster0.
db = client['amenda_010']
collection = db['alex']
documents = collection.find()
docs_list = list(documents)
df = pd.DataFrame(docs_list)
print(df.head())
                                                                                                   product_id
                                                                                                                          customer id
 \rightarrow
                                                          _id transaction_id
              66a95ee3dd4f7h5h25502a6f
                                                                                             1
                                                                                                                    2
                                                                                                                                         2950
               66a95ee3dd4f7b5b25502a70
                                                                                             2
                                                                                                                    3
                                                                                                                                         3120
               66a95ee3dd4f7b5b25502a71
                                                                                             3
                                                                                                                   37
                                                                                                                                          402
               66a95ee3dd4f7b5b25502a72
                                                                                             4
                                                                                                                   88
                                                                                                                                         3135
          4
               66a95ee3dd4f7b5b25502a73
                                                                                             5
                                                                                                                   78
                                                                                                                                          787
              transaction_date online_order order_status
                                                                                                                         brand product_line
                                                                                                                                            Standard
          0
                          2017-02-25
                                                                   0.0
                                                                                   Approved
                                                                                                                        Solex
                          2017-05-21
                                                                   1.0
                                                                                   Approved
                                                                                                         Trek Bicycles
                                                                                                                                            Standard
          1
                                                                                                            OHM Cycles
                         2017-10-16
                                                                                                                                            Standard
          2
                                                                   0.0
                                                                                   Approved
                                                                                  Approved
                         2017-08-31
                                                                                                                                            Standard
          3
                                                                   9.9
                                                                                                      Norco Bicycles
          4
                         2017-10-01
                                                                   1.0
                                                                                  Approved Giant Bicycles
                                                                                                                                            Standard
             product_class product_size list_price standard_cost
          0
                           medium
                                                     medium
                                                                               71.49
                                                                                                             53.62
                           medium
                                                      large
                                                                           2091.47
                                                                                                           388.92
                                                     medium
                                                                           1793.43
                                                                                                           248.82
                                 low
          3
                           medium
                                                     medium
                                                                           1198.46
                                                                                                           381.10
          4
                           medium
                                                                                                           709.48
                                                       large
                                                                           1765.30
               product_first_sold_date
          0
                                                41245.0
                                                41701.0
          1
          2
                                                36361.0
          3
                                                36145.0
          4
                                                42226.0
Start coding or generate with AI.
start_time = time.time()
document = collection.find()
docs_list = list(document)
end_time = time.time()
query_time = end_time - start_time
```

```
df = pd.DataFrame(docs_list)
print(df.head())
                              _id transaction_id product_id customer_id \
\overline{2}
       66a95ee3dd4f7b5b25502a6f
                                                           2
                                                                      2950
                                               1
       66a95ee3dd4f7b5b25502a70
     1
                                               2
                                                           3
                                                                      3120
       66a95ee3dd4f7b5b25502a71
     2
                                               3
                                                          37
                                                                       402
     3 66a95ee3dd4f7b5b25502a72
                                               4
                                                          88
                                                                      3135
     4 66a95ee3dd4f7b5b25502a73
                                                          78
                                                                       787
       transaction_date online_order order_status
                                                              brand product_line \
     0
             2017-02-25
                                 0.0
                                         Approved
                                                              Solex
                                                                        Standard
             2017-05-21
                                                     Trek Bicycles
     1
                                  1.0
                                          Approved
                                                                        Standard
             2017-10-16
                                                       OHM Cycles
                                                                        Standard
     2
                                  0.0
                                          Approved
     3
             2017-08-31
                                          Approved Norco Bicycles
                                                                        Standard
                                  0.0
     4
                                                                        Standard
             2017-10-01
                                  1.0
                                         Approved Giant Bicycles
       \verb|product_class| product_size | list\_price | standard_cost | \\
     0
              medium
                           medium
                                        71.49
                                                       53.62
                           large
     1
              medium
                                      2091.47
                                                      388.92
     2
                low
                           medium
                                      1793.43
                                                       248.82
     3
              medium
                           medium
                                      1198.46
                                                       381.10
     4
              medium
                                      1765.30
                                                      709.48
                           large
        product_first_sold_date
     0
                        41245.0
     1
                        41701.0
                        36361.0
     2
     3
                        36145.0
     4
                        42226.0
Start coding or generate with AI.
total_size = sum(len(json.dumps(doc)) for doc in docs_list)
\rightarrow
     TypeError
                                               Traceback (most recent call last)
     <ipython-input-20-f15d34560b73> in <cell line: 1>()
     ----> 1 total_size = sum(len(json.dumps(doc)) for doc in docs_list)

    4 frames

     /usr/lib/python3.10/json/encoder.py in default(self, o)
         177
        178
     --> 179
                     raise TypeError(f'Object of type {o.__class__.__name__}} '
                                     f'is not JSON serializable')
         180
        181
     TypeError: Object of type ObjectId is not JSON serializable
 Next steps:
              Explain error
total_size = sum(len(json.dumps(doc)) for doc in docs_list)
                                               Traceback (most recent call last)
     <ipython-input-21-f15d34560b73> in <cell line: 1>()
     ----> 1 total_size = sum(len(json.dumps(doc)) for doc in docs_list)
                                     4 frames –
     /usr/lib/python3.10/json/encoder.py in default(self, o)
        177
         178
                     raise TypeError(f'Object of type {o.__class__.__name__}} '
     --> 179
         180
                                     f'is not JSON serializable')
     TypeError: Object of type ObjectId is not JSON serializable
 Next steps: Explain error
def serilaize_doc(doc):
 if 'id' in doc:
   doc['_id'] = str(doc['_id'])
 return doc
```

```
import json
from datetime import datetime
def serilaize_doc(doc):
 if 'id' in doc:
   doc['_id'] = str(doc['_id'])
 # Handle datetime objects
 for key, value in doc.items():
      if isinstance(value, datetime):
         doc[key] = value.isoformat() # Convert datetime to ISO format string
 return doc
total_size = sum(len(json.dumps(serilaize_doc(doc))) for doc in docs_list)
print(total_size)
₹ 7505698
query_latency = end_time - start_time
throughput = total_size / query_latency if query_latency > 0 else float('inf')
metrics_report = {
    "total_documents": len(docs_list),
    "data size": f"{total size} bytes",
    "query_latency": f"{query_latency:.6f} seconds",
    "throughput": f"{throughput:.2f} bytes/second"
}
print("\nData Transfer Metrics Report:")
for metric, value in metrics_report.items():
   print(f"{metric}: {value}")
\overline{\mathcal{F}}
     Data Transfer Metrics Report:
     total_documents: 20000
     data_size: 7505698 bytes
     query_latency: 103.771113 seconds
     throughput: 72329.36 bytes/second
Start coding or generate with AI.
start_time = time.time()
document = collection.find()
docs_list = list(document)
metrics_df = pd.DataFrame([metrics_report])
CSV_FILE_PATH = "metrics_report.csv"
if os.path.exists(CSV_FILE_PATH):
 existing_metrics_df = pd.read_csv(CSV_FILE_PATH)
if os.path.exists(CSV_FILE_PATH):
 metrics_df.to_csv(CSV_FILE_PATH, mode='a', header=False, index=False)
 metrics_df.to_csv(CSV_FILE_PATH, index=False)
Start coding or generate with AI.
print("\nMerics saved to performance_metrics.csv")
     Merics saved to performance metrics.csv
CSV_FILE_PATH = "metrics_report.csv"
metrics_df = pd.read_csv(CSV_FILE_PATH)
print(metrics_df.head(10))
```

```
total_documents
                            data_size
                                           query_latency
                                                                     throughput
                 20000 7505698 bytes 103.771113 seconds 72329.36 bytes/second
current_timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
metrics_report = {
    "timestamp": current_timestamp,
    "total_documents": len(docs_list),
    "data_size": total_size,
    "query_latency": query_latency,
    "throughput": throughput
metrics_df = pd.DataFrame([metrics_report])
csv_file_path = "performance_metrics.csv"
if os.path.exists(csv_file_path):
   # Append to the existing CSV file
   metrics_df.to_csv(csv_file_path, mode='a', header=False, index=False)
   # Create a new CSV file with headers
   metrics_df.to_csv(csv_file_path, mode='w', header=True, index=False)
print("\nMetrics saved to performance_metrics.csv")
    Metrics saved to performance_metrics.csv
CSV_FILE_PATH = "performance_metrics.csv"
metrics_df = pd.read_csv(CSV_FILE_PATH)
print(metrics_df.head(10))
                 timestamp total_documents data_size query_latency \
    0 2024-07-31 10:39:04
                                                          103.771113
                                    20000 7505698
         throughput
    0 72329.358062
from google.colab import files
files.download('/content/performance_metrics.csv')
```

Start coding or $\underline{\text{generate}}$ with AI.