Q1.

""" MongoDB is a popular, open-source NoSQL database management system that is designed to store and manage large volumes of data in a flexible, schema-less format.

Non-relational databases, often referred to as NoSQL databases, are a category of database management systems that are not based on the traditional relational database model.

When dealing with data that doesn't fit neatly into a fixed table structure, such as JSON-like documents or variable attributes, MongoDB's flexible schema is advantageous. MongoDB is well-suited for storing log files and event data, where data records can have varying attributes and formats.

"""

Q2.

"""

- Document-Oriented: MongoDB is a document-oriented database, where data is stored in flexible, JSON-like documents.
- Schema Flexibility: MongoDB provides schema flexibility, allowing you to store data without a predefined schema
- Automatic Sharding: MongoDB supports automatic data sharding, allowing it to partition large data sets across multiple servers.
- High Availability: MongoDB provides features like replica sets, which offer data redundancy and failover in case of server failures.
- Text Search: MongoDB provides full-text search capabilities, making it suitable for applications that require searching and indexing of text data.

"""

Q3.

"""

- import pymango

"""

In [ ]:
```
import pymongo

connection_string = "mongodb+srv://prajwalm:123@cluster.mongodb.net/your_databas
```

```
client = pymongo.MongoClient(connection_string)

database_name = "database"
db = client[database_name]

collection_name = "your_collection"
collection = db[collection_name]

data_to_insert = {"name": "John Doe", "email": "johndoe@example.com"}
inserted_document = collection.insert_one(data_to_insert)

print("Inserted document ID:", inserted_document.inserted_id)
```

Q4.

In [1]:

```
%pip install pymango
import pymongo

connection_string = "mongodb+srv://prajwalm:123@cluster.mongodb.net/your_databas

client = pymongo.MongoClient(connection_string)


database_name = "your_database"
db = client[database_name]
collection_name = "your_collection"
collection = db[collection_name]

data_to_insert_one = {"name": "Alice", "email": "alice@example.com"}
inserted_document_one = collection.insert_one(data_to_insert_one)
print("Inserted document ID (One Record):", inserted_document_one.inserted_id)

data_to_insert_many = [
    {"name": "Bob", "email": "bob@example.com"},
    {"name": "Charlie", "email": "charlie@example.com"},
    {"name": "David", "email": "david@example.com"},
]
inserted_documents_many = collection.insert_many(data_to_insert_many)
print("Inserted document IDs (Many Records):", inserted_documents_many.inserted_

found_document_one = collection.find_one({"name": "Alice"})
print("Found One Record:", found_document_one)

found_documents_many = collection.find()
print("Found Many Records:")
for document in found_documents_many:
    print(document)
```

```
Requirement already satisfied: pymango in /opt/conda/lib/python3.10/site-packag
es (0.1.1)
Requirement already satisfied: requests>=2.4.3 in /opt/conda/lib/python3.10/sit
e-packages (from pymango) (2.28.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/lib/python3.
10/site-packages (from requests>=2.4.3->pymango) (1.26.13)
Requirement already satisfied: charset-normalizer<3,>=2 in /opt/conda/lib/pytho
n3.10/site-packages (from requests>=2.4.3->pymango) (2.1.1)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/
site-packages (from requests>=2.4.3->pymango) (2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-p
ackages (from requests>=2.4.3->pymango) (3.4)
Note: you may need to restart the kernel to use updated packages.
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[1], line 2
      1 get_ipython().run_line_magic('pip', 'install pymango')
----> 2 import pymongo
      4 connection_string = "mongodb+srv://prajwalm:123@cluster.mongodb.net/you
r_database?retryWrites=true&w=majority"
      6 client = pymongo.MongoClient(connection_string)

ModuleNotFoundError: No module named 'pymongo'
```

Q5.

""" The find() method takes an optional query object as its argument, allowing you to specify criteria for filtering documents.

"""

In [3]:
```
"""
import pymongo

connection_string = "mongodb+srv://prajwalm:1123@cluster.mongodb.net/your_databa

client = pymongo.MongoClient(connection_string)

database_name = "your_database"
db = client[database_name]
collection_name = "students"
collection = db[collection_name]

query = {"age": 20}

cursor = collection.find(query)

for document in cursor:
    print(document)
"""
```

Out[3]:
```
'\nimport pymongo\n\nconnection_string = "mongodb+srv://prajwalm:1123@cluster.m
ongodb.net/your_database?retryWrites=true&w=majority"\n\nclient = pymongo.Mongo
Client(connection_string)\n\ndatabase_name = "your_database"\ndb = client[datab
ase_name]\ncollection_name = "students"\ncollection = db[collection_name]\n\nqu
ery = {"age": 20}\n\ncursor = collection.find(query)\n\nfor document in curso
r:\n    print(document)\n'
```

Q6.

"""

- the sort() method is used to specify the sorting order of the documents in the result set of a query.

"""

collection.find(query).sort(sort_key, sort_order)

Q7.

"""

- The delete_one() method is used to delete a single document (record) that matches a specific filter or criteria within a MongoDB collection.
- The delete_many() method is used to delete multiple documents that match a given filter or criteria within a MongoDB collection.
- The drop() method is used to remove an entire collection (equivalent to a table in relational databases) from a MongoDB database.

"""