Assignment

Q1. What is MongoDB? Explain non-relational databases in short. In which scenarios it is preferred to use MongoDB over SQL databases?

Answer.

MongoDB is a non-relational document database that provides support for JSON-like storage. The MongoDB database has a flexible data model that enables you to store unstructured data, and it provides full indexing support, and replication with rich and intuitive AP

A non-relational database is a database that does not use the tabular schema of rows and columns found in most traditional database systems. Instead, non-relational databases use a storage model that is optimized for the specific requirements of the type of data being stored.

While MongoDB supports JSON querying, a SQL Database uses SQL query processing. The basic characteristics make MongoDB a more dynamic and complex option that is fit for hierarchical data while a SQL Database remains more predefined and suited for other kinds of data storage.

SQL databases are typically used in applications that require complex queries and transaction management, whereas NoSQL databases are used in applications that require high performance and scalability, such as web applications and mobile apps.

Q2. State and Explain the features of MongoDB.

Answer: Features of MongoDB include the following:

*__Replication:__ A replica set is two or more MongoDB instances used to provide high availability. Replica sets are made of primary and secondary servers. The primary MongoDB server performs all the read and write operations, while the secondary replica keeps a copy of the data. If a primary replica fails, the secondary replica is then used.

*__Scalability:__ MongoDB supports vertical and horizontal scaling. Vertical scaling works by adding more power to an existing machine, while horizontal scaling works by adding more machines to a user's resources.

*__Load balancing__: MongoDB handles load balancing without the need for a separate, dedicated load balancer, through either vertical or horizontal scaling.

*__Schema-less:__ MongoDB is a schema-less database, which means the database can manage data without the need for a blueprint.

*__Document:__ Data in MongoDB is stored in documents with key-value pairs instead of rows and columns, which makes the data more flexible when compared to SQL databases.

Q3. Write a code to connect MongoDB to Python. Also, create a database and a collection in MongoDB.

Answer:

```
import pymongo
from pymongo.mongo_client import MongoClient
uri = "mongodb+srv://ashish:password@cluster0.2j0tmuk.mongodb.net/?retryWrites=true&w=majority"
# Create a new client and connect to the server
client = MongoClient(uri)
# Send a ping to confirm a successful connection
try:
    client.admin.command('ping')
    print("Pinged your deployment. You successfully connected to MongoDB!")
except Exception as e:
  print(e)
db= client['pwskills']
col_create= db["myrecord"]
data={"Name":"Ashish",
     "Class":"Data science",
     "Type":"Regular"}
col_create.insert_one(data1)
```

Q4. Using the database and the collection created in question number 3, write a code to insert one record, and insert many records. Use the find() and find_one() methods to print the inserted record.

Answer:

```
data={"Name":"Ashish",
     "Class":"Data science",
     "Type":"Regular"}
col_create.insert_one(data1)
all_records= [{'Name': 'Ashish',
     'Course': 'Data Science',
     'duration': '2 Months'},

    {'Name': 'XYZ',
     'Course': 'C++',
     'duration': '8 Months'},

    {'Name': 'abc',
     'Course': 'dev',
     'duration': '7 Months'},
]
col_create.insert_many(all_records)
col_create.find_one()
for i in col_create.find():
    print(i)
```

Q5. Explain how you can use the find() method to query the MongoDB database. Write a simple code to demonstrate this.

Answer: The **find()** method in MongoDB selects documents in a collection or view and returns a cursor to the selected documents.

```
data={"Name":"Ashish",
     "Class":"Data science",
     "Type":"Regular"}
col_create.insert_one(data1)
all_records= [{'Name': 'Ashish',
     'Course': 'Data Science',
     'duration': '2 Months'},

    {'Name': 'XYZ',
     'Course': 'C++',
     'duration': '8 Months'},

    {'Name': 'abc',
     'Course': 'dev',
     'duration': '7 Months'},
]
col_create.insert_many(all_records)
col_create.find_one()
for i in col_create.find():
    print(i)
```

Q6. Explain the sort() method. Give an example to demonstrate sorting in MongoDB.

Answer: To sort documents in MongoDB, you need to use sort() method. The method accepts a document containing a list of fields along with their sorting order. To specify sorting order 1 and -1 are used. 1 is used for ascending order while -1 is used for descending order.

SYNTAX:

```
db.collection_name.find().sort({field_name: sort order})
```

Q7. Explain why delete_one(), delete_many(), and drop() is used.

Answer:

1: deleteOne() deletes the first document that matches the filter.

2: deleteMany() method allows you to remove all documents that match a condition from a collection

3: **drop()** method is used to drop a collection from a database