**PW SKILLS**

**MONGODB ASSIGNMENT**

Q1. MongoDB is a popular NoSQL (non-relational) database system. Unlike traditional SQL databases, which store data in tables with rows and columns, MongoDB stores data in flexible, JSON-like documents. Non-relational databases are preferred in scenarios where flexibility, scalability, and speed are prioritized over strict data consistency and structured querying. MongoDB is often chosen over SQL databases in applications requiring:

- Flexible schema: MongoDB's document-oriented structure allows for easy modification of data schema without requiring a predefined schema.

- Horizontal scalability: MongoDB can easily scale across multiple servers, making it suitable for large-scale distributed systems.

- Handling unstructured or semi-structured data: MongoDB excels in handling data that doesn't fit well into a tabular format, such as JSON documents or arrays.

Q2. Features of MongoDB include:

- Document-oriented: Stores data in flexible, JSON-like documents.

- High performance: Provides high-speed reads and writes due to its flexible schema and horizontal scalability.

- Scalability: Easily scales out across multiple servers to handle large volumes of data and high traffic.

- Rich query language: Supports powerful query capabilities, including nested queries, range queries, and more.

- Indexing: Allows indexing to improve query performance.

- Aggregation framework: Offers powerful aggregation capabilities for data analysis and reporting.

- Flexible schema: No need to predefine the schema; fields can be added or modified on the fly.

- Replication and high availability: Provides built-in replication for data redundancy and fault tolerance.

- Geospatial indexing: Supports geospatial queries for location-based applications.

- GridFS: Stores large files and binary data efficiently.

Q3. Python code to connect to MongoDB, create a database, and a collection:

```python

from pymongo import MongoClient

# Connect to MongoDB

client = MongoClient('mongodb://localhost:27017/')

# Create a database

mydb = client['mydatabase']

# Create a collection

mycollection = mydb['mycollection']

```

Q4. Python code to insert one record and insert many records into the collection, and print them using `find()` and `find\_one()` methods:

```python

# Insert one record

record\_one = {"name": "John", "age": 30}

result\_one = mycollection.insert\_one(record\_one)

# Insert many records

records\_many = [

{"name": "Alice", "age": 25},

{"name": "Bob", "age": 35},

{"name": "Charlie", "age": 40}

]

result\_many = mycollection.insert\_many(records\_many)

# Print inserted record using find\_one()

print(mycollection.find\_one({"name": "John"}))

# Print all inserted records using find()

for record in mycollection.find():

print(record)

```

Q5. The `find()` method in MongoDB is used to query documents from a collection based on specified criteria. Here's a simple code demonstrating how to use the `find()` method:

```python

# Find documents where age is greater than 30

for record in mycollection.find({"age": {"$gt": 30}}):

print(record)

```

This code retrieves all documents from the collection where the value of the "age" field is greater than 30.

Q6. The `sort()` method in MongoDB is used to sort the documents in a collection based on one or more fields. Here's an example:

```python

# Sort documents by age in ascending order

sorted\_records = mycollection.find().sort("age", 1)

# Print sorted records

for record in sorted\_records:

print(record)

```

This code sorts the documents in the collection based on the "age" field in ascending order (`1` denotes ascending order, `-1` denotes descending order).

Q7. In MongoDB, `delete\_one()` is used to delete a single document that matches the specified criteria, `delete\_many()` is used to delete multiple documents matching the criteria, and `drop()` is used to drop an entire collection.

- `delete\_one()`: Deletes the first document that matches the specified criteria.

- `delete\_many()`: Deletes all documents that match the specified criteria.

- `drop()`: Deletes an entire collection, including all documents and indexes.

These methods are used for removing data from the database when it's no longer needed or to perform cleanup operations.