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

MongoDB over SQL databases?

* MongoDB is **an open source NoSQL database management program**. NoSQL is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data.
* 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.
* SQL. NoSQL databases like MongoDB are a good choice **when your data is document-centric and doesn't fit well into the schema of a relational database**, when you need to accommodate massive scale, when you are rapidly prototyping, and a few other use cases.

Q2. State and Explain the features of MongoDB.

* + **Support ad hoc queries** In MongoDB, you can search by field, range query and it also supports regular expression searches.
  + **Indexing** You can index any field in a document.
* **Replication** MongoDB supports Master Slave replicatioA master can perform Reads and Writes and a Slave copies data from the master and can only be used for reads or back up (not writes)
* **Duplication of data** MongoDB can run over multiple servers. The data is duplicated to keep the system up and also keep its running condition in case of hardware failure.
* **Load balancing** It has an automatic load balancing configuration because of data placed in shards.

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

* Code to connect MongoDB to Python

**from** pymongo import MongoClient

import pprint

client = MongoClient()

db = client.javatpoint

employee = {"id": "101",

"name": "Peter",

"profession": "Software Engineer",

}

* Create a database

use sampledb

* Create a collection

>use test

>db.createCollection("SSSIT")

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.

* Insert one record

db.SSSIT.**insert**({"name" : "seomount"})

* Insert many records

db.SSSIT.insertMany([{name:"Ajay",age:20},

{name:"Bina",age:24},

{name:"Ram",age:23}])

* You can read/retrieve stored documents from MongoDB using the ***find()*** method. This method retrieves and displays all the documents in MongoDB in a non-structured way.

db.SSSIT.find()

* The **findOne() function** is used to find one document according to the condition. If multiple documents match the condition, then it returns the first document satisfying the condition.

db.bios.findOne()

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

* The sort() method can be used to sort the metadata values for a calculated metadata field.

The following example used the “food” collection to demonstrate how documents can be sorted using the metadata “textScore.” The field name in the sort() method can be arbitrary as the query system ignores the field name.

db.food.find({$text:{$search: "pizza"}}, {score:{$meta: "textScore"}, \_id: 0}).sort({sort\_example:{$meta: "textScore"}})

Q7. Explain why delete\_one(), delete\_many(), and drop() is used.

* delete\_one(): To delete one document, we use the delete\_one() method.
* Drop():To delete more than one document, use the delete\_many() method.
* delete\_many():To delete more than one document, use the delete\_many() method.