

1. Explain super() in the context of inheritance.

Ans:- At a fairly abstract level, `super()` provides the access to those methods of the super-class (parent class) which have been overridden in a sub-class (child class) that inherits from it. Consider the code example given below, here we have a class named `Square` and an another class named `Cube` which inherits the class `Square`.

```
class Square:
```

```
    def __init__(self, side):
```

```
        self.side = side
```

```
    def area(self):
```

```
        return self.side * self.side
```

```
class Cube(Square):
```

```
    def area(self):
```

```
        face_area = self.side * self.side
```

```
        return face_area * 6
```

```
    def volume(self):
```

```
        face_area = self.side * self.side
```

```
        return face_area * self.side
```

```
class Cube(Square):
```

```
    def area(self):
```

```
        return super().area() * 6
```

```
    def volume(self):
```

```
        return super().area() * self.side()
```

2. Describe the file-handling system.

Ans:- Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files. The concept of file handling has stretched over various other languages, but the implementation is either complicated or lengthy, but like other concepts of Python, this concept here is also easy and short. Python treats file differently as text or binary and this is important. Each line of code includes a sequence of characters and they form text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun. Let's start with Reading and Writing files.

Working of open() function

F=open("filename","mode")

Where the following mode is supported:

1. **r**: open an existing file for a read operation.
2. **w**: open an existing file for a write operation. If the file already contains some data then it will be overridden.
3. **a**: open an existing file for append operation. It won't override existing data.
4. **r+**: To read and write data into the file. The previous data in the file will not be deleted.
5. **w+**: To write and read data. It will override existing data.
6. **a+**: To append and read data from the file. It won't override existing data.

3. In Python, explain multiple inheritance.

Ans:- Multiple Inheritance is a type of **inheritance** in which **one class** can **inherit properties(attributes and methods)** of **more than one parent classes**.

```
class hulk:
    def smash(self):
        return "I smash"

class banner:
    def speak(self):
        return "I've got the brains!"

class smarthulk(hulk, banner):
    pass

s1 = smarthulk()
print(s1.smash(), "and", s1.speak())
```

output:-

```
I smash and I've got the brain!
```

4. Write the MySQL query syntax for INSERT, UPDATE, and DROP.

Ans:- In MySQL, You can use the INSERT INTO statement to insert data to your database table. With INSERT INTO statement of MySQL, you insert single row or multiple rows into database table.

Syntax

In MySQL INSERT INTO statement syntax is :-

```
INSERT INTO table_name ( field1, field2,...fieldN )
VALUES ( value1, value2,...valueN );
```

Params

- **Fields :-** It's a database column name. You can insert value into in it.
- **table_name :-** It's a database table name.

- Value :- It's your column values, that you want to insert into database

In MySQL, You can use the UPDATE statement to update or change data to your database table. With UPDATE statement of MySQL, you can change or modify table data by using MySQL WHERE Clause.

Syntax

In MySQL UPDATE statement syntax is :-

```
UPDATE table_name SET field1=new-value1, field2=new-value2  
[WHERE Clause]
```

Params

- table_name: – This is the name of a database table.
- Fields: – This is a database column name. Where you can update the value in it
- Value: – These are your column values, which you want to update in the database.
- WHERE: – In MySQL, where the clause is used to update specific rows in a table.

The **DROP DATABASE** statement is used to drop an existing SQL database.

Syntax

DROP DATABASE databasename;

5. Describe MongoDB's features

Ans:- MongoDB Features

Let us now define and understand each feature :

1. Ad Hoc Queries Support

Basically, when we are in the Database Design phase, we have no idea of what queries might be executed. So, when we say, MongoDB supports Ad Hoc Queries, it means that the MongoDB supports queries that were not known while establishing a structure for the database. This is an amazing feature that makes MongoDB stand out. Ad hoc queries function in a way to better the performance and are real-time.

2. Indexing

One of the most important features for a Database is Indexing, which results in improvements. We can index any field in MongoDB. Creating indexes helps in faster search results. Indexing is possible with any field or key, in a document.

Code:

```
db.test.getIndexes()
```

3. Replication

Simply speaking, replication is a process of, distributing the data on multiple servers and keeping everything synchronized. Primary Node and Secondary Nodes are introduced here.

Working: Whenever a primary node, with the data, is down or experiences failure, the secondary node will be the working primary node, making it possible for the data to be available.

This feature is important regarding Data Storage and Backup, as it allows us to recover and restore, in case of failure in hardware or services. Thanks to this replication, that data is made available with multiple copies on different locations. Time is saved, and no operation is halted due to this amazing feature.

4. Schema-less Database

One of the most primarily discussed features of MongoDB is, it being Schemaless. Meaning that there can be multiple documents in a collection, with different keys, and these keys might not be found in other documents. This is the major reason behind MongoDB's flexibility, with data.

We can have multiple documents or records in a single, same collection, but with different key values pairs. Check the below-attached screenshot for reference.

Code:

```
db.test.find().pretty()
```

5. Document Oriented

Being Document Oriented is a great feature that helps MongoDB stand out among others. We have tables and rows columns structure for SQL, we have fields in key values pair, with MongoDB.

Code:

```
db.educba.find().pretty().limit(1)
```

6. Aggregation Pipeline

Aggregation Framework is one of the most efficient features offered by MongoDB. Simply speaking it is a process of creating a pipeline of multiple operations and getting a final result, filtered.

Code:

```
db.fun.aggregate([{$match: {"City":"Pune"}},{$sort:
{_id:-1}}])
```

The above query will return documents that have Pune as their City, and it will print the sorted result. Refer the below-attached screenshot.

.

7. GridFS

When a file exceeds the BSON document limit of 16 MB, it is divided into multiple small chunks and stored separately in various documents. These chunks have a size of 255KB, excluding the last chunk. GridFS, which stands for Grid File System, use two separate collections. One collection is used to store the larger file's chunks, while the second collection is used to store the

metadata. When we execute a query for this file, the GridFS will collect and return all the chunks together. GridFS also implements the Indexing, which allows the query execution for returning the file easier.

8. Sharding

When we are encountered with large datasets, we can implement the Sharding of Data. Meaning, the large datasets are split and shared across multiple machines. The massive data can cause unexpected problems, but the implementation of sharding can be useful. Sharding is the process of database partitioning and spreading across multiple machines, while the replication is the process of making multiple copies of the database. The data is distributed over multiple collections, and these collections are known as “Shards”.

9. High Performance

MongoDB is open-source, is one the highest performing database. With the implementation of replication and indexing, query execution and data fetching are faster. With developing multiple applications, the need to check the performance is mandatory. Database Profiling, which collects the

complete data for every operation that is executed against the MongoDB instance. The output provided by the DB Profiler can help us understand the queries and operations that are being inefficient.