

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

Ans: Python | super() in single inheritance  
Prerequisites: Inheritance, function overriding  
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 another class named Cube which inherits the class Square. Considering this example, we know that each face of a cube is a square and hence, face\_area of Cube represents area of a Square. Now, it makes sense to evaluate face\_area using area() method of the class Square rather than calculating it manually, not only this will save us from rewriting the code but also it will allow to change the area() logic from one place. But as we have overridden the area() method in Cube, we cannot call area() method of Square using self.area().

2. Describe the file-handling system.?

Ans: Software that manages data files. Often called "file managers," and erroneously called "database management systems" or "database managers," file management systems provide the ability to create, enter, change, query and produce reports on one file at a time. A file management system has limited capabilities and is designed to manage individual or group files, such as special office documents and records. It may display report details, like owner, creation date, state of completion and similar features useful in an office environment. However, many other files such as pictures, videos and documents are arranged by the user according to his or her own will. A file management system ultimately is the software used to arrange these files, move them, and work with them. In fact, file management systems take care of how the files are organized rather than just how they are stored. A file management system's tracking component is key to the creation and management of this system, where documents containing various stages of processing are shared and interchanged on an ongoing basis. It consists of a straightforward interface where stored files are displayed. It allows the user to browse, move, and sort them according to different criteria such as date of last modification, date of creation, file type/format, size, etc.

3. In Python, explain multiple inheritance.?

4. Ans: Python Multiple Inheritance  
In this tutorial, you'll learn about multiple inheritance in Python and how to use it in your program. You'll also learn about multi-level inheritance and the method resolution order.  
Python Multiple Inheritance  
A class can be derived from more than one base class in Python, similar to C++. This is called multiple inheritance. In multiple inheritance, the features of all the base classes are inherited into the derived class. The syntax for multiple inheritance is similar to single inheritance.  
Example  
class Base1: pass  
class Base2: pass  
class MultiDerived(Base1, Base2): pass  
Here, the MultiDerived class is derived from Base1 and Base2 classes.  
Multiple Inheritance in Python  
Multiple Inheritance in Python the MultiDerived class inherits from both Base1 and Base2 classes.  
python Multilevel Inheritance  
We can also inherit from a derived class. This is called multilevel inheritance. It can be of any depth in Python. In multilevel inheritance, features of the base class and the derived class are inherited into the new derived class. An example with corresponding visualization is given below. In the multiple inheritance scenario, any specified attribute is searched first in the current class.

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

Ans.: The most commonly used statement of MySQL Select statement. MySQL SELECT statement used to fetch data/records from database table.

In the INSERT INTO statement of MySQL, you can insert single or multiple rows into the database table.

MySQL UPDATE statement, you can update the single row using the UPDATE & WHERE clause statement at a time.

The DELETE statement is used to remove/delete a specific row or multiple rows using the MySQL DELETE & WHERE clause.

Table Of Content

SELECT Statement MySQL

INSERT Statement MySQL

UPDATE Statement MySQL

DELETE Statement MySQL

SELECT Statement MySQL

In MySQL, The SELECT statement is the most commonly used MySQL statement. You can use the SELECT statement to fetch or get data from one or more tables within the database. With SELECT statement of MySQL, you can fetch all fields of table or specified fields.

Syntax

```
SELECT field1, field2 ... field n
```

```
FROM Table
```

```
[WHERE conditions];
```

Params

Fields :- It's a database column name. You can fetch one or more fields using SELECT statement.

Table :- It's a database table name.

WHERE :- It's a option. If You want to filter some data that time specify any condition using the WHERE clause.

SELECT Statement Example

```
SELECT users.id, users.name, users.age
```

```
FROM users
```

```
WHERE status = active
```

5. Describe MongoDB's features.?

Ans.: MongoDB is an open-source document-oriented database that is designed to store a large scale of data and also allows you to work with that data very efficiently. It is categorized under the NoSQL (Not only SQL) database because the storage and retrieval of data in the MongoDB are not in the form of tables.

The MongoDB database is developed and managed by MongoDB, Inc under SSPL (Server Side Public License) and initially released in February 2009. It also provides official driver support for all the popular languages like C, C++, C#, and .Net, Go, Java, Node.js, Perl, PHP, Python, Motor, Ruby, Scala, Swift, Mongoid. So, that you can create an application using any of these languages. Nowadays there are so

many companies that used MongoDB like Facebook, Nokia, eBay, Adobe, Google, etc. to store their large amount of data.

How it works ?

Now, we will see how actually thing happens behind the scene. As we know that MongoDB is a database server and the data is stored in these databases. Or in other words, MongoDB environment gives you a server that you can start and then create multiple databases on it using MongoDB.

Because of its NoSQL database, the data is stored in the collections and documents. Hence the database, collection, and documents are related to each other as shown below: The MongoDB database contains collections just like the MYSQL database contains tables. You are allowed to create multiple databases and multiple collections.

Now inside of the collection we have documents. These documents contain the data we want to store in the MongoDB database and a single collection can contain multiple documents and you are schema-less means it is not necessary that one document is similar to another.

The documents are created using the fields. Fields are key-value pairs in the documents, it is just like columns in the relation database. The value of the fields can be of any BSON data types like double, string, boolean, etc.

The data stored in the MongoDB is in the format of BSON documents. Here, BSON stands for Binary representation of JSON documents. Or in other words, in the backend, the MongoDB server converts the JSON data into a binary form that is known as BSON and this BSON is stored and queried more efficiently.

In MongoDB documents, you are allowed to store nested data. This nesting of data allows you to create complex relations between data and store them in the same document which makes the working and fetching of data extremely efficient as compared to SQL. In SQL, you need to write complex joins to get the data from table 1 and table 2. The maximum size of the BSON document is 16MB.