

QUES-1- What is php? Explain building blocks of php.

PHP (Hypertext Preprocessor) is a popular server-side scripting language used for web development. It is open-source and free to use, making it a popular choice for creating dynamic and interactive web pages. PHP code is executed on the server before the web page is sent to the client's browser, allowing for dynamic content generation and database connectivity.

PHP is widely used for creating a variety of web applications, such as blogs, e-commerce sites, content management systems, and social networking sites. It is also supported by most web servers and operating systems, making it a versatile language for web development.

Some of the key features of PHP include its ability to work with different databases, support for object-oriented programming, and the availability of numerous libraries and frameworks for creating web applications more efficiently. With PHP, developers can create dynamic web pages, handle form data, manipulate and store data, and perform other tasks necessary for creating robust web applications.

The building blocks of PHP are:

1. Variables: PHP variables are used to store data values. They can be of various data types, such as strings, integers, floats, and booleans.
2. Operators: PHP has a range of operators for performing various operations, such as arithmetic, comparison, logical, and assignment operations.

3. Control Structures: PHP has control structures like if-else, switch, for, while, and do-while loops for controlling the flow of code execution.
4. Functions: PHP functions are blocks of code that can be called repeatedly from different parts of a program, making it easier to reuse code.
5. Arrays: PHP arrays are used to store multiple values in a single variable. They can be indexed, associative, or multidimensional.
6. Strings: PHP has powerful string manipulation functions that allow for concatenation, trimming, searching, and replacing.
7. Classes and Objects: PHP supports object-oriented programming, allowing for the creation of classes and objects for modular and reusable code.
8. File Handling: PHP has built-in functions for reading, writing, and manipulating files on the server.
9. Databases: PHP has extensive support for connecting to and interacting with databases, allowing for dynamic content generation and data storage.

Together, these building blocks form the foundation of PHP and enable developers to create robust and dynamic web applications.

## QUES-2-Explain control structure in php.

Control structures in PHP are constructs that allow the programmer to control the flow of program execution based on specific conditions. They are used to make decisions, loop over code blocks, and execute code based on certain conditions. PHP supports various control structures, including if-else statements, switch statements, for loops, while loops, and do-while loops.

1. If-Else Statements: The if-else statement allows a programmer to execute a block of code if a specified condition is true and another block of code if the condition is false. The basic syntax is:

sqlCopy code

```
if (condition) { //code to execute if condition is true } else {  
//code to execute if condition is false }
```

2. Switch Statements: A switch statement allows the programmer to test a variable against a list of values and execute different blocks of code based on the matching value. The basic syntax is:

arduinoCopy code

```
switch (variable) { case value1: //code to execute if variable  
matches value1 break; case value2: //code to execute if  
variable matches value2 break; default: //code to execute if  
variable does not match any values break; }
```

3. For Loops: A for loop allows the programmer to execute a block of code repeatedly based on a specified condition. The basic syntax is:

sqlCopy code

```
for (initial value; condition; increment/decrement) { //code to execute repeatedly }
```

4. While Loops: A while loop allows the programmer to execute a block of code repeatedly based on a specified condition. The basic syntax is:

arduinoCopy code

```
while (condition) { //code to execute repeatedly }
```

5. Do-While Loops: A do-while loop allows the programmer to execute a block of code repeatedly based on a specified condition, but the block of code is executed at least once before checking the condition. The basic syntax is:

arduinoCopy code

```
do { //code to execute repeatedly } while (condition);
```

Using these control structures in PHP, developers can create powerful and flexible programs that can handle a wide range of conditions and situations.

## QUES-3- SHORT NOTES ON:-

### 1.PHP tags

PHP tags are used to enclose PHP code in a web page. There are three types of PHP tags used in PHP programming:

1. **<?php ?>** tags: This is the most common type of PHP tag used to enclose PHP code. The opening tag **<?php** is used to start a PHP code block, and the closing tag **?>** is used to end the PHP code block.
2. **<?= ?>** tags: This is a shorthand version of the **<?php ?>** tags used to print a single value to the web page. The opening tag **<?=** is used to start the PHP code block, and the closing tag **?>** is used to end the PHP code block.
3. **<? ?>** tags: This is an alternative to the **<?php ?>** tags used to enclose PHP code. The opening tag **<?** is used to start a PHP code block, and the closing tag **?>** is optional.

It is important to note that the **<?php ?>** tags are the most commonly used tags and are recommended for use in PHP programming. Additionally, it is important to ensure that there is no white space or HTML code outside of the PHP tags, as it can cause errors in the PHP code.

## 2.Variables

In PHP, variables are used to store data values that can be manipulated and referenced in the program. PHP variables are dynamically typed, which means that they do not have to be declared with a specific data type before use. Instead, PHP automatically assigns a data type to the variable based on the value it contains.

In PHP, variables are defined with a dollar sign (\$) followed by the variable name. Variable names must start with a letter or underscore character and can contain letters, numbers, and underscores. Variable names are case-sensitive, which means that **\$var** and **\$Var** are two different variables.

Assigning a value to a variable is done using the assignment operator (=). For example, to assign the value "Hello World!" to a variable named **\$message**, we can use the following code:

bashCopy code

```
$message = "Hello World!";
```

PHP also supports variable interpolation, which allows variables to be included within double-quoted strings. For example:

perlCopy code

```
$name = "John"; $message = "Hello, $name!"; // Output:  
Hello, John!
```

In addition to simple variables, PHP also supports arrays, which can store multiple values under a single variable name,

and objects, which allow for the creation of complex data structures with properties and methods.

Overall, variables are a crucial part of PHP programming and are used extensively in the creation of dynamic and interactive web applications.

### 3.Data types

In PHP, a data type refers to the type of data that a variable can hold. PHP supports several data types, including:

1. Integer: Integers are whole numbers, either positive or negative, without decimals. In PHP, integers are represented using the **int** or **integer** keyword.
2. Float: Floats (also known as doubles) are numbers with a decimal point or numbers in scientific notation. In PHP, floats are represented using the **float** or **double** keyword.
3. String: Strings are a sequence of characters, enclosed in single or double quotes. In PHP, strings are represented using the **string** keyword.
4. Boolean: Boolean values represent true or false. In PHP, boolean values are represented using the **bool** or **boolean** keyword.
5. Array: Arrays are used to store multiple values under a single variable name. In PHP, arrays can be indexed arrays, associative arrays, or multidimensional arrays.

6. Object: Objects allow for the creation of complex data structures with properties and methods. In PHP, objects are created using classes and can be manipulated using object-oriented programming principles.
7. Null: Null represents a variable with no value. In PHP, null is represented using the **null** keyword.

It is important to note that PHP is a dynamically typed language, which means that the data type of a variable is determined automatically based on the value it contains. This means that a variable can change its data type during the execution of a script.

Understanding data types is crucial for creating effective PHP programs, as it allows developers to properly store, manipulate, and work with different types of data.

## 4.Operators

Operators in PHP are symbols or keywords used to perform operations on variables and values. PHP supports several types of operators, including:

1. Arithmetic operators: Arithmetic operators are used to perform basic mathematical operations such as addition (+), subtraction (-), multiplication (\*), division (/), modulus (%), and exponentiation (\*\*).
2. Assignment operators: Assignment operators are used to assign values to variables. The basic assignment



operator is (=), but PHP also supports shorthand assignment operators such as +=, -=, \*=, /=, and %=.

3. Comparison operators: Comparison operators are used to compare two values and return a boolean (true or false) result. Some common comparison operators in PHP include ==, !=, >, <, >=, and <=.
4. Logical operators: Logical operators are used to combine multiple conditions and return a boolean result. The most common logical operators in PHP are && (AND), || (OR), and ! (NOT).
5. Bitwise operators: Bitwise operators are used to perform operations on the binary representation of values. Some common bitwise operators in PHP include &, |, ^, ~, <<, and >>.
6. String operators: String operators are used to manipulate and concatenate strings. The most common string operator in PHP is the dot (.) operator, which is used to concatenate two strings.
7. Increment/Decrement operators: Increment and decrement operators are used to increase or decrease the value of a variable by one. PHP supports both pre-increment and post-increment operators, as well as pre-decrement and post-decrement operators.

## 5.Expression

In PHP, an expression is a combination of variables, constants, and operators that can be evaluated to produce a value.

Expressions are a fundamental building block of PHP programming, as they are used to perform calculations, manipulate data, and control program flow.

Expressions can take many different forms in PHP, depending on the operators and variables used. For example, a simple arithmetic expression might look like this:

shellCopy code

```
$x = 5; $y = 10; $sum = $x + $y;
```

In this example, the expression **`$x + $y`** is evaluated and the result is stored in the variable **`$sum`**.

PHP supports a wide variety of operators that can be used in expressions, including arithmetic operators, assignment operators, comparison operators, logical operators, bitwise operators, string operators, and increment/decrement operators.

In addition to simple expressions, PHP also supports more complex expressions that include function calls, array references, and object properties. For example:

phpCopy code

```
$names = array("John", "Jane", "Joe"); $first_name =  
strtoupper($names[0]);
```

In this example, the expression **`$names[0]`** is used to access the first element of the **`$names`** array, and the **`strtoupper()`**

function is used to convert it to uppercase. The result of the expression is stored in the variable **\$first\_name**.

QUES-3- What is function in php? Explain its types.

In PHP, a function is a block of code that performs a specific task and can be called from anywhere in a script. Functions are a fundamental building block of PHP programming, as they allow developers to organize code into reusable, modular components.

PHP supports several types of functions, including:

1. Built-in functions: PHP comes with a large number of built-in functions that perform common tasks, such as working with strings and arrays, manipulating dates and times, and sending email.
2. User-defined functions: User-defined functions are functions that are created by the developer and can be called from anywhere in the script. User-defined functions can take parameters and return values, and are used to perform custom operations that are specific to the application being developed.
3. Anonymous functions: Anonymous functions (also known as closures) are functions that do not have a name and can be assigned to a variable. Anonymous functions are often used in PHP to create callbacks and to pass functions as arguments to other functions.

4. Recursive functions: Recursive functions are functions that call themselves, often with different arguments. Recursive functions are useful for performing tasks that require repeated calculations or for working with complex data structures such as trees and graphs.
5. Variadic functions: Variadic functions are functions that can take a variable number of arguments. Variadic functions are useful when the number of arguments is not known in advance, or when the number of arguments may vary from call to call.

QUES-4-Explain array and its types with example.

In PHP, an array is a data structure that stores a collection of values, each identified by a unique key. Arrays are a powerful tool in PHP programming, as they allow developers to store and manipulate large amounts of data in a single variable.

PHP supports three types of arrays:

1. Indexed arrays: Indexed arrays are the simplest type of array in PHP. Each element in an indexed array is identified by a numerical index, starting at 0. For example:

phpCopy code

```
$numbers = array(1, 2, 3, 4, 5); echo $numbers[2]; // Output:  
3
```

In this example, the variable **\$numbers** is an indexed array containing five elements, and the value 3 is retrieved by accessing the element with index 2.

2. Associative arrays: Associative arrays are arrays where each element is identified by a string key instead of a numerical index. For example:

phpCopy code

```
$person = array("name" => "John", "age" => 30, "city" => "New York"); echo $person["age"]; // Output: 30
```

In this example, the variable **\$person** is an associative array containing three elements, each identified by a string key. The value 30 is retrieved by accessing the element with the key "age".

3. Multidimensional arrays: Multidimensional arrays are arrays that contain other arrays as elements. For example:

phpCopy code

```
$students = array( array("name" => "John", "age" => 20), array("name" => "Jane", "age" => 22), array("name" => "Joe", "age" => 21) ); echo $students[1]["name"]; // Output: Jane
```

In this example, the variable **\$students** is a multidimensional array containing three elements, each of which is an associative array with two keys ("name" and "age"). The value "Jane" is retrieved by accessing the element with index 1 and key "name".

## QUES-6-SHORT NOTES:-

### 1.variable scope

In web programming, variable scope refers to the area of a program where a variable can be accessed and manipulated. The scope of a variable is determined by where it is declared within the program.

In PHP, there are four types of variable scope:

1. Global scope: A variable declared outside of any function or class has global scope, which means it can be accessed from anywhere in the script.
2. Local scope: A variable declared inside a function or block of code has local scope, which means it can only be accessed within that function or block of code.
3. Static scope: A static variable is a local variable that retains its value between function calls. Static variables are declared inside a function using the static keyword.
4. Class scope: A variable declared inside a class has class scope, which means it can be accessed from any method within that class. Class variables are also known as properties.

Understanding variable scope is important in web programming, as it helps to prevent naming conflicts and to keep code organized and modular. By using appropriate variable scope, developers can ensure that their code is efficient, maintainable, and easy to debug.

## 2.sorting and reordering array

In web programming, sorting and reordering arrays is a common task when working with large amounts of data. PHP provides a variety of built-in functions for sorting and reordering arrays, which can help developers to quickly and efficiently manipulate their data.

Here are some commonly used functions for sorting and reordering arrays in PHP:

1. `sort()`: This function sorts an indexed array in ascending order based on the value of its elements.
2. `rsort()`: This function sorts an indexed array in descending order based on the value of its elements.
3. `asort()`: This function sorts an associative array in ascending order based on the value of its elements, while maintaining the association between keys and values.
4. `ksort()`: This function sorts an associative array in ascending order based on the keys, while maintaining the association between keys and values.
5. `usort()`: This function sorts an array using a user-defined comparison function.
6. `shuffle()`: This function randomly shuffles the elements of an array.
7. `array_reverse()`: This function reverses the order of the elements in an array.

By using these functions, developers can easily manipulate and reorder arrays to suit their needs. Whether sorting data alphabetically, numerically, or by some other criteria, PHP provides a variety of tools to help make the task fast and efficient.

### 3.php classes

In web programming, PHP classes are an important tool for creating reusable and modular code. A class is a blueprint for creating objects, which are instances of the class. Each object has its own set of properties (variables) and methods (functions), which are defined by the class.

Here are some key concepts related to PHP classes:

1. Class definition: A class definition begins with the **class** keyword, followed by the name of the class and a pair of curly braces. Inside the braces, developers can define properties and methods for the class.
2. Object creation: To create an object from a class, developers use the **new** keyword followed by the class name and parentheses. For example: **\$obj = new MyClass();**
3. Properties: Properties are variables that belong to an object. They can be accessed using the object operator (->) followed by the name of the property. For example: **\$obj->property = "value";**



4. Methods: Methods are functions that belong to an object. They can be called using the object operator (->) followed by the name of the method and parentheses. For example: **\$obj->method();**
5. Constructor: A constructor is a special method that is automatically called when an object is created. It is used to initialize the properties of the object. The constructor method has the same name as the class, and is defined using the **\_\_construct()** function.
6. Inheritance: Inheritance allows developers to create new classes based on existing classes. The new class, called a subclass or derived class, inherits the properties and methods of the parent class, called the superclass or base class.

PHP classes are a powerful tool in web programming, as they allow developers to create reusable and modular code that can be easily maintained and extended. By defining classes and creating objects, developers can create complex web applications that are both efficient and easy to manage.

QUES-7-What are strings in php. Explain various string functions.

In PHP, a string is a sequence of characters, enclosed in quotes (either single or double quotes). Strings can be used to store and manipulate text-based data, such as names, addresses, and messages.

Strings are one of the most important and frequently used data types in PHP. They are used for a wide variety of purposes, including:

1. Displaying text on a webpage: Strings are used to output text to the browser, whether it be a simple greeting message or a large block of HTML markup.
2. Manipulating and formatting text: Strings can be manipulated and formatted using a wide range of built-in functions. For example, the **trim()** function can be used to remove whitespace from the beginning and end of a string, while **substr()** can be used to extract a specific section of a string.
3. Storing and processing data: Strings can be used to store data in a wide range of formats, including numbers, dates, and email addresses. Additionally, strings can be processed using regular expressions, which allow developers to search for and extract specific patterns of text.
4. Performing string searches and replacements: PHP provides a wide range of built-in functions for searching and replacing strings, including **strpos()** for finding the

position of a substring within a string, and **str\_replace()** for replacing a substring with another substring.

Here are some common string functions in PHP:

1. `strlen()`: This function returns the length of a string.
2. `str_replace()`: This function replaces all occurrences of a search string with a replacement string in a given string.
3. `strpos()`: This function finds the position of the first occurrence of a substring in a given string.
4. `substr()`: This function returns a substring of a given string, starting at a specified position and with a specified length.
5. `strtolower()`: This function converts a string to lowercase.
6. `strtoupper()`: This function converts a string to uppercase.
7. `trim()`: This function removes whitespace (or other characters) from the beginning and end of a string.
8. `implode()`: This function joins an array of strings into a single string, separated by a specified delimiter.
9. `explode()`: This function splits a string into an array of strings, using a specified delimiter.
10. `strrev()`: This function reverses a given string.

These functions can be used to perform a wide range of string manipulation tasks, from simple text formatting to

complex search and replace operations. By using these functions, developers can create efficient and flexible PHP programs that can handle a variety of text-based data.

QUES-8- Explain date and time function in php.

PHP provides a comprehensive set of built-in functions for working with date and time values. These functions allow developers to easily manipulate and format dates and times, calculate time differences, and perform other common tasks related to working with date and time data.

Here are some of the most commonly used date and time functions in PHP:

1. **date()** - This function is used to format a date and time value according to a specific format string. The format string can contain various format codes that are replaced with corresponding date and time values.

Example:

bashCopy code

```
$date = date('Y-m-d H:i:s'); echo $date;
```

Output: **2023-05-04 14:30:00**

2. **strtotime()** - This function converts a string representation of a date and time into a Unix timestamp, which is a numeric value that represents the number of seconds since January 1, 1970, 00:00:00 UTC.

Example:

bashCopy code

```
$timestamp = strtotime('2023-05-04 14:30:00'); echo  
$timestamp;
```

Output: **1670111400**

3. **time()** - This function returns the current Unix timestamp.

Example:

bashCopy code

```
$timestamp = time(); echo $timestamp;
```

Output: **1670111400**

4. **strtotime()** and **date()** combined - These two functions can be used together to perform various date calculations, such as adding or subtracting days or hours from a given date.

Example:

bashCopy code

```
$date = '2023-05-04'; $timestamp = strtotime($date . ' +1  
day'); $newDate = date('Y-m-d', $timestamp); echo  
$newDate;
```

Output: **2023-05-05**

5. **gmdate()** - This function is similar to **date()**, but returns the GMT/UTC time instead of the local time.

Example:

bashCopy code

```
$date = gmdate('Y-m-d H:i:s'); echo $date;
```

Output: **2023-05-04 14:30:00**

These are just a few examples of the many date and time functions available in PHP. By using these functions effectively, developers can create powerful and flexible programs that can handle a wide range of date and time-related tasks.

QUES-9- Explain file handling in php (opening, reading, writing, and closing file).

File handling in PHP refers to the ability to read, write, and manipulate files on a server or local computer. PHP provides a set of built-in functions that make it easy to work with files.

Here is a brief overview of how to perform file handling operations in PHP:

1. Opening a file: To open a file in PHP, you can use the **fopen()** function, which takes two arguments: the name of the file to open and the mode in which to open it (e.g. "r" for reading, "w" for writing).

Example:

bashCopy code

```
$file = fopen("example.txt", "r");
```

2. Reading a file: Once a file is open, you can use various functions to read its contents. The most commonly used

function is **fgets()**, which reads a single line of text from the file.

Example:

bashCopy code

```
while(!feof($file)) { $line = fgets($file); echo $line; }
```

3. Writing to a file: To write data to a file, you can use the **fwrite()** function, which takes two arguments: the file handle and the data to write.

Example:

bashCopy code

```
$file = fopen("example.txt", "w"); fwrite($file, "Hello, world!");
```

4. Closing a file: When you are finished working with a file, it is important to close it using the **fclose()** function.

Example:

scssCopy code

```
fclose($file);
```

Here is an example that demonstrates how to open, read, write, and close a file in PHP:

phpCopy code

```
// Open file for reading $file = fopen("example.txt", "r"); //  
Read contents of file and output to screen while(!feof($file)) {  
$line = fgets($file); echo $line; } // Open file for writing and
```

```
write a message $file = fopen("example.txt", "w");  
fwrite($file, "Hello, world!"); // Close file fclose($file);
```

This is just a basic overview of file handling in PHP. There are many other functions available for working with files, such as **fseek()** for setting the file pointer position and **file()** for reading an entire file into an array.

#### QUES-10-SHORT NOTES:-

##### 1.file functions

In web programming, file handling is an important aspect of dealing with files on a server or a local machine. PHP provides a set of built-in file functions that make it easy to work with files.

Here are some of the commonly used file functions in PHP:

1. **fopen()**: This function opens a file and returns a file pointer. It takes two parameters: the name of the file to open and the mode in which to open it.
2. **fclose()**: This function closes an open file pointer.
3. **fread()**: This function reads data from an open file pointer.
4. **fwrite()**: This function writes data to an open file pointer.



5. **feof()**: This function tests whether the end of an open file pointer has been reached.
6. **file\_exists()**: This function checks whether a file or directory exists.
7. **is\_dir()**: This function checks whether a given path is a directory.
8. **is\_file()**: This function checks whether a given path is a file.
9. **unlink()**: This function deletes a file.
10. **rename()**: This function renames a file.
11. **file\_get\_contents()**: This function reads an entire file into a string.
12. **file\_put\_contents()**: This function writes a string to a file.
13. **scandir()**: This function returns an array of files and directories in a given directory.

These file functions make it easy to perform various file handling tasks in PHP, such as opening, reading, writing, renaming, and deleting files, as well as checking whether a file or directory exists or is a file or a directory.

QUES-11-What are cookies, how to set-up cookies in php

Cookies are small pieces of data that are stored on a user's computer by a website. They are commonly used to keep track of user preferences, login information, and other data related to user sessions.

In PHP, cookies can be set up using the **setcookie()** function. This function takes up to six parameters:

1. The name of the cookie
2. The value of the cookie
3. The expiration time of the cookie (in seconds from the current time)
4. The path on the server where the cookie will be available
5. The domain of the cookie
6. Whether the cookie should be sent over secure HTTPS connections only

Here's an example of how to set up a cookie in PHP:

lessCopy code

```
setcookie("username", "john_doe", time() + (86400 * 30),  
"/");
```

This sets a cookie named "username" with the value "john\_doe" that will expire in 30 days. The cookie will be available on the root path of the server ("/").

To retrieve the value of a cookie in PHP, you can use the **\$\_COOKIE** superglobal variable. For example:

bashCopy code

```
$username = $_COOKIE['username'];
```

This retrieves the value of the "username" cookie and assigns it to the **\$username** variable.

It's important to note that cookies can only be set before any output is sent to the browser. This means that you must call the **setcookie()** function before any HTML or other content is printed to the page.

Additionally, cookies can be a security risk if not used properly. It's important to validate and sanitize user input before storing it in a cookie, and to ensure that sensitive information is not stored in a cookie.

QUES-12- Explain session, how to start/stop session in php.

A session in PHP is a way to store data across multiple pages and requests by the same user. When a user visits a website that uses sessions, a unique session ID is generated for that user, which is stored as a cookie on the user's computer or passed as a parameter in the URL. The session ID is then used to identify the user's session and retrieve any stored data.

To start a session in PHP, you need to call the **session\_start()** function at the beginning of each page where you want to use sessions. This function initializes a new session or resumes an existing session if one exists for the current user. Here's an example:

scssCopy code

```
session_start();
```

Once a session has been started, you can store data in it using the **\$\_SESSION** superglobal variable. For example:

bashCopy code

```
$_SESSION['username'] = 'john_doe';
```

This stores the value "john\_doe" in the "username" key of the current user's session.

To end a session in PHP, you can call the **session\_destroy()** function. This destroys all data associated with the current session and removes the session cookie from the user's browser. Here's an example:

scssCopy code

```
session_destroy();
```

It's important to note that the **session\_start()** function must be called before any output is sent to the browser, just like with the **setcookie()** function. Additionally, you should always sanitize and validate user input before storing it in a session to prevent security vulnerabilities.

Overall, sessions are a useful tool for storing user-specific data across multiple pages and requests in a PHP web application.

### QUES-13-How to create database in SQL

To create a database in SQL, you need to use the **CREATE DATABASE** statement. Here's the basic syntax:

sqlCopy code

```
CREATE DATABASE database_name;
```

Replace "database\_name" with the name you want to give your database. Here's an example:

sqlCopy code

```
CREATE DATABASE my_database;
```

This will create a new database named "my\_database".

Once you've created a database, you can create tables to store data within it. Here's an example of creating a table:

sqlCopy code

```
CREATE TABLE users ( id INT(11) NOT NULL  
AUTO_INCREMENT, username VARCHAR(50) NOT NULL,  
password VARCHAR(255) NOT NULL, email VARCHAR(255)  
NOT NULL, PRIMARY KEY (id) );
```

This creates a table named "users" with columns for "id", "username", "password", and "email". The "id" column is set to auto-increment, which means it will automatically assign a unique value to each new row inserted into the table.

You can then insert data into the table using the **INSERT INTO** statement:

sqlCopy code

```
INSERT INTO users (username, password, email) VALUES ('john_doe', 'password123', 'john_doe@example.com');
```

This inserts a new row into the "users" table with the username "john\_doe", password "password123", and email "[john\\_doe@example.com](mailto:john_doe@example.com)".

Overall, creating a database in SQL involves using the **CREATE DATABASE** statement to create the database, creating tables within the database using the **CREATE TABLE** statement, and inserting data into the tables using the **INSERT INTO** statement.

.

#### QUES-14-Data types used in MySQL

MySQL supports various data types that can be used to define columns in tables. The following are the commonly used data types in MySQL:

1. **Numeric Data Types:** Numeric data types are used to store numbers. Some examples of numeric data types in MySQL are **INT**, **FLOAT**, **DOUBLE**, and **DECIMAL**.
2. **Date and Time Data Types:** Date and time data types are used to store date and time values. Some examples of date and time data types in MySQL are **DATE**, **TIME**, **DATETIME**, and **TIMESTAMP**.
3. **String Data Types:** String data types are used to store character data. Some examples of string data types in MySQL are **VARCHAR**, **CHAR**, and **TEXT**.

4. Binary Data Types: Binary data types are used to store binary data, such as images or files. Some examples of binary data types in MySQL are **BLOB** and **LONGBLOB**.
5. Enumeration and Set Data Types: Enumeration and set data types are used to define a list of possible values for a column. **ENUM** data type is used when a column can have only one value from the given list, whereas **SET** data type is used when a column can have multiple values from the given list.
6. Spatial Data Types: Spatial data types are used to store spatial data, such as maps or geographic information. Some examples of spatial data types in MySQL are **GEOMETRY** and **POINT**.

These are some of the commonly used data types in MySQL. Choosing the appropriate data type for a column is important for optimizing storage space and ensuring data integrity.

QUES-15-How to insert/retrieve data from database.

To insert or retrieve data from a MySQL database using PHP, you need to follow these steps:

1. Establish a connection to the database: You can establish a connection to the database using the **mysqli\_connect()** or **PDO** function in PHP.
2. Prepare the SQL query: You need to prepare the SQL query to insert or retrieve data from the database. You can use the **mysqli\_query()** or **PDO::query()** function to execute the SQL query.
3. Insert data into the database: To insert data into the database, you can use the **INSERT INTO** statement in SQL. You can use the **mysqli\_query()** or **PDO::query()** function to execute the SQL query.

For example:

phpCopy code

```
<?php // Establish a connection to the database $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB"; $conn = mysqli_connect($servername, $username, $password, $dbname); // Prepare the SQL query $sql = "INSERT INTO users (name, email) VALUES ('John Doe', 'johndoe@example.com')"; // Insert data into the database if (mysqli_query($conn, $sql)) { echo "New record created successfully"; } else { echo "Error: " . $sql . "<br>" . mysqli_error($conn); } // Close the connection mysqli_close($conn); ?>
```



4. Retrieve data from the database: To retrieve data from the database, you can use the **SELECT** statement in SQL. You can use the **mysqli\_query()** or **PDO::query()** function to execute the SQL query.

For example:

phpCopy code

```
<?php // Establish a connection to the database $servername = "localhost"; $username = "username"; $password = "password"; $dbname = "myDB"; $conn = mysqli_connect($servername, $username, $password, $dbname); // Prepare the SQL query $sql = "SELECT name, email FROM users"; // Retrieve data from the database $result = mysqli_query($conn, $sql); if (mysqli_num_rows($result) > 0) { // Output data of each row while($row = mysqli_fetch_assoc($result)) { echo "Name: " . $row["name"]. " - Email: " . $row["email"]. "<br>"; } } else { echo "0 results"; } // Close the connection mysqli_close($conn); ?>
```

These are the basic steps to insert or retrieve data from a MySQL database using PHP. However, it's important to use prepared statements to prevent SQL injection attacks and ensure the security of your application.

## QUES-16-SHORT NOTES:-

### 1.Using queries and sub-queries

In web programming, using queries and sub-queries in SQL is a powerful technique to retrieve and manipulate data from a database.

A query is a request for data or information from a database, using the **SELECT** statement in SQL. It allows you to retrieve specific data from one or more tables, based on a set of criteria or conditions. You can use various operators, functions, and clauses in SQL to filter, group, sort, and join the data.

A sub-query, also known as a nested query or inner query, is a query inside another query. It allows you to use the results of one query as the input for another query, and perform complex operations on the data. You can use sub-queries in various clauses of SQL, such as **WHERE**, **FROM**, **SELECT**, and **JOIN**.

For example, suppose you have a database with two tables: **employees** and **departments**. The **employees** table has columns such as **id**, **name**, **age**, **salary**, and **department\_id**. The **departments** table has columns such as **id** and **name**.

To retrieve the name and average salary of each department, you can use a sub-query in the **SELECT** clause of the main query, like this:

sqlCopy code

```
SELECT name, (SELECT AVG(salary) FROM employees WHERE
department_id = departments.id) AS avg_salary FROM
departments;
```

This query will retrieve the name of each department and the average salary of the employees in that department, using a sub-query to calculate the average salary based on the **department\_id** column.

Using queries and sub-queries in SQL can help you to perform complex operations on your data, retrieve specific information, and optimize the performance of your database. However, it's important to use them wisely, optimize the queries for efficiency, and avoid security risks such as SQL injection attacks.

## 2.deleting records

In web programming, deleting records from a database is an important operation that allows you to remove unwanted or outdated data from your tables.

To delete records from a table in SQL, you can use the **DELETE** statement with the **WHERE** clause. The **WHERE** clause allows you to specify the conditions that the records must meet to be deleted. For example, the following SQL statement deletes all records from the **employees** table where the **age** is greater than 50:

sqlCopy code

```
DELETE FROM employees WHERE age > 50;
```

It's important to note that the **DELETE** statement removes entire rows from a table, not just individual columns or fields. Therefore, it's important to be careful when using this statement, as it can permanently delete data from your database.

To avoid accidental data loss, you should always make a backup of your database before performing any major operations such as deleting records. You should also use caution when using the **WHERE** clause, and double-check your conditions to make sure you are targeting the correct records.

Overall, deleting records in web programming is an important operation that requires careful consideration and planning to ensure the integrity and security of your data.

### 3. updating records

In web programming, updating records in a database is a common operation that allows you to modify existing data in your tables.

To update records in a table in SQL, you can use the **UPDATE** statement with the **SET** and **WHERE** clauses. The **SET** clause allows you to specify the columns to update and the new values for those columns. The **WHERE** clause allows you to specify the conditions that the records must meet to be updated. For example, the following SQL statement updates the **salary** column of the **employees** table to **50000** for all records where the **department** is **Sales**:

sqlCopy code

```
UPDATE employees SET salary = 50000 WHERE department = 'Sales';
```

It's important to note that the **UPDATE** statement only modifies the specified columns for the selected rows, and does not remove or add any rows to the table.

To avoid accidental data loss or corruption, you should always make a backup of your database before performing any major operations such as updating records. You should also use caution when using the **WHERE** clause, and double-check your conditions to make sure you are targeting the correct records.

Overall, updating records in web programming is an important operation that requires careful consideration and planning to ensure the integrity and security of your data.

#### 4.dropping table and database

In web programming, dropping a table or database is a common operation that allows you to remove unwanted or outdated data from your system.

To drop a table in SQL, you can use the **DROP TABLE** statement, followed by the name of the table you want to delete. For example, the following SQL statement drops the **employees** table:

sqlCopy code

```
DROP TABLE employees;
```

It's important to note that dropping a table permanently deletes all data and metadata associated with that table, so you should use caution when performing this operation.

To drop a database in SQL, you can use the **DROP DATABASE** statement, followed by the name of the database you want to delete. For example, the following SQL statement drops the **mydatabase** database:

sqlCopy code

```
DROP DATABASE mydatabase;
```

Again, it's important to note that dropping a database permanently deletes all data and metadata associated with that database, so you should use caution when performing this operation.

To avoid accidental data loss, you should always make a backup of your database before performing any major operations such as dropping a table or database.

Overall, dropping a table or database in web programming is an important operation that should be performed with caution and after careful consideration of the impact on your system.

## 5.grouping and aggregate data

In web programming, grouping and aggregating data is a common operation in SQL that allows you to group and summarize data based on certain criteria. This can be useful

for generating reports, analyzing data, and making informed decisions.

To group data in SQL, you can use the **GROUP BY** clause in your SQL query. This clause allows you to group the data by one or more columns in the result set. For example, the following SQL query groups the data in the **orders** table by the **customer\_id** column:

vbnetCopy code

```
SELECT customer_id, SUM(total) as total_sales FROM orders  
GROUP BY customer_id;
```

This query will return a result set that shows the total sales for each customer in the **orders** table.

In addition to grouping data, you can also aggregate data in SQL using functions such as **SUM**, **AVG**, **MAX**, **MIN**, and **COUNT**. These functions allow you to perform calculations on the grouped data and return aggregate values. For example, the following SQL query calculates the total number of orders and the average order amount for each customer in the **orders** table:

vbnetCopy code

```
SELECT customer_id, COUNT(*) as total_orders, AVG(total) as  
average_order_amount FROM orders GROUP BY  
customer_id;
```

This query will return a result set that shows the total number of orders and the average order amount for each customer in the **orders** table.

Overall, grouping and aggregating data in web programming can be a powerful tool for analyzing and summarizing data in your application. By using SQL functions and clauses such as **GROUP BY**, **SUM**, and **AVG**, you can gain insights into your data and make informed decisions based on the results.



# UNIT-VI

Accessing MySQL Database From Web With PHP :

Web database architecture, Querying database from the web: checking and filtering

input data, setting up connection, Choosing database to use, querying database,

retrieving the query result, disconnecting from the database.

## NOTES

Accessing a MySQL database from the web with PHP involves several key steps, including establishing a connection, choosing a database, querying the database, retrieving the query result, and disconnecting from the database. Here's an overview of each step:

1. Web database architecture: The web database architecture typically involves a web server running PHP scripts that communicate with a MySQL database server. The PHP scripts can use a variety of methods to connect to the database, including the mysqli and PDO extensions.
2. Querying the database from the web: When querying the database from the web, it's important to check and filter user input data to prevent SQL injection attacks.

This can involve using prepared statements or input validation functions to sanitize user input.

3. Setting up connection: To connect to a MySQL database from PHP, you can use the `mysqli` or `PDO` extension. This involves specifying the host, username, password, and database name to connect to. For example:

bashCopy code

```
$conn = mysqli_connect("localhost", "myuser",  
"mypassword", "mydatabase");
```

4. Choosing database to use: After connecting to the MySQL server, you need to select the database that you want to use. This can be done using the **`mysqli_select_db()`** function or by specifying the database name in the connection string. For example:

bashCopy code

```
mysqli_select_db($conn, "mydatabase");
```

5. Querying database: Once you've established a connection and selected the database, you can query the database using SQL statements. This can involve using functions such as **`mysqli_query()`** or **`PDO::query()`** to execute SQL statements. For example:

bashCopy code

```
$result = mysqli_query($conn, "SELECT * FROM mytable");
```

6. Retrieving query result: After executing a query, you can retrieve the results using functions such as **`mysqli_fetch_array()`**, **`mysqli_fetch_assoc()`**, or

**PDOStatement::fetch()**. These functions allow you to iterate over the rows returned by the query and access the values in each column. For example:

bashCopy code

```
while ($row = mysqli_fetch_assoc($result)) { echo  
$row['column1'] . " " . $row['column2']; }
```

7. Disconnecting from the database: When you're finished querying the database, it's important to close the database connection to free up resources. This can be done using functions such as **mysqli\_close()** or **PDO::close()**. For example:

scssCopy code

```
mysqli_close($conn);
```

Overall, accessing a MySQL database from the web with PHP involves several key steps, including establishing a connection, choosing a database, querying the database, retrieving the query result, and disconnecting from the database. By following best practices for database security and using the appropriate PHP functions and extensions, you can build secure and efficient web applications that interact with MySQL databases.