SRINIVAS UNIVERSITY

**CITY CAMPUS PANDESHWAR**

**MANGALURU- 575001.**

**INSTITUTE OF COMPUTER AND INFORMATION SCIENCE**

LAMP TECHNOLOGY

**B.C.A V SEMESTER**

# Question Bank (with answers)

****

**Sandhya Bangera**

|  |  |  |  |
| --- | --- | --- | --- |
| **Exam** | **V Semester** | **Paper code** | **20BCASD51/ 20BCAAI51/ 20BCANT51** |
| **Subject** | **Lamp Technology** | **Class** | **BCA III** |
| **Maximum marks** | **50** | **Time** | **2 Hours** |

**Weightage Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Objectives** | **Marks** | **Percentage of Marks** |
| 1. | Knowledge (Remembering) | 05 | 10 |
| 2. | Understanding | 20 | 40 |
| 3. | Application | 15 | 30 |
| 4. | Skill | 10 | 20 |
| **Total** | | **50** | **100** |

**Blueprint**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Unit** | **Remembering** | | **Understand** | | **Application** | | **Skill** | | **Total** |
| **OT** | **SA** | **OT** | **SA** | **OT** | **SA** | **OT** | **SA** |  |
| **1** | 1(1) | 1(4) | 1(1) | 1(4) | - | - | - | - | 10 |
| **2** | - | - | 2(1) | 1(4) | - | 1(4) | - | - | 10 |
| **3** | - | - | - | 1(4) | 2(1) | 1(4) | - | - | 10 |
| **4** | - | - | 1(1) | 1(4) | - | 1(4) | 1(1) | - | 10 |
| **5** | - | - | - | - | 1(1) | - | 1(1) | 2(4) | 10 |
|  | **05** | | **20** | | **15** | | **10** | | **50** |

**UNIT I**

**Mutiple Choice Questions**

1. The programming interface to the kernel is included in which subsystem of operating system.
2. User Applications

# O/S Services

1. Linux Kernel
2. Hardware Controllers
3. Memory hardware is an example of which subsystem of operating system
4. User Applications
5. O/S Services
6. Linux Kernel

# Hardware Controllers

1. Give the full form of VFS

# [Virtual File System](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_3)

1. Visual File System
2. Virus File System
3. Valid File System
4. Full form of IPC

# [Inter-Process Communication](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_5)

1. Intra -Process Communication
2. [Inter-Process Command](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_5)
3. Intra- Process Command
4. Which of the following is responsible for controlling process access to the CPU
5. [Memory Manager](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_2)
6. [Inter-Process Communication](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_5)
7. [Virtual File System](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_3)

# [Process Scheduler](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_1)

1. Give the full form of PID
2. **P**[rocess identification number](http://www.linfo.org/pid.html)
3. Page identification number
4. [Process identical number](http://www.linfo.org/pid.html)
5. Program identical name
6. Special files are in which folder

# /dev

1. /bin
2. /lib
3. /tmp
4. Which of the following value represent ―No restrictions on permissions‖ on **files**

***a) 777***

*b) 755*

*c) 700*

*d) 666*

1. Which of the following value represent ―The file's owner may read, write, and execute the **file**‖.

*a) 777*

*b) 755*

***c) 700***

*d) 666*

1. Which of the following value represent ―The directory owner has full access. Nobody elsehas any rights.‖

*a) 777*

*b) 755*

***c) 700***

*d) 666*

1. Which of the following command is used to ―Prints directory content‖?
2. Cat
3. chmod

# ls

1. mkdir
2. Which of the following command is used Removes directories.
3. Cat
4. chmod

# rmdir

1. rm
2. Which of the following has extra and third party software.

# /opt

1. /root
2. /sbin
3. /tmp
4. Which directory is used for miscellaneous purposes.

# /misc

1. /root
2. /sbin
3. /tmp
4. How is rwx represented in binary form

a) 111 111 111

b) 110 110 110

# c) 111 000 000

d) 000 000 000

1. The main configuration file for configuring Apache is

# httpd.conf

1. srm.conf
2. access.conf
3. apache.conf
4. If a directive must continue onto the next line which of the following must be used as the last character on the previous line

# back-slash '\'

1. colon (:)
2. semicolon (;)
3. underscore (\_)
4. Any line beginning with a character is ignored

# hash (#)

1. colon (:)
2. semicolon (;)
3. underscore (\_)
4. Maximum number of requests to allow during a persistent connection is indicated via

# MaxKeepAliveRequests

1. KeepAliveTimeout
2. MinSpareServers
3. MaxSpareServers
4. Full form of CGI
5. Correct Gateway Interface

# Common Gateway Interface

1. Common Gateway Interconnect
2. Correct Gateway Interconnect
3. IP based Virtual host is also called as
4. Name based Virtual host

# Address-based Virtual host

1. File based virtual host
2. Time based virtual host
3. If your server has 10 IP addresses, how many IP based virtual hosts can be created?

# a) 10

b) 20

c) 30

d) 40

1. Which of the following indicates whether or not to allow persistent connections

# KeepAlive

1. MinSpareServers
2. MaxSpareServers
3. MaxKeepAliveRequests
4. In which of the following file the server should record its process identification number
5. LockFile

# PidFile

1. ScoreBoard File
2. StartServers
3. Which of the following file is used to store internal server process information
4. LockFile
5. PidFile

# ScoreBoard File

1. StartServers

# Long Answer Questions (Application)

1. **What are different file system permissions?**

**Ans.:** Read, Write & Execute Permissions

Permissions are the "rights" to act on a file or directory. The basic rights are read, write, and execute.

Read - A readable permission allows the contents of the file to be viewed. A read permission on a directory allows you to list the contents of a directory.

Write - A write permission on a file allows you to modify the contents of that file. For a directory, the write permission allows you to edit the contents of a directory (e.g. add/delete files).

Execute - For a file the executable permission allows you to run the file and execute a program

or script. For a directory, the execute permission allows you to change to a different directory and make it your current working directory. Users usually have a default group, but they may belong to several additional groups.

Viewing File Permissions

To view the permissions on a file or directory, issue the command ls -l<directory/file>. Remember to replace the information in the **<>** with the actual file or directory name. Below is sample output for the ls command:

-rw-r--r-- 1 root root 1031 Nov 18 09:22 /etc/passwd

The first ten characters show the access permissions. The first dash (-) indicates the type of file (d for directory, s for special file, and - for a regular file). The next three characters (**rw-**) define the owner's permission to the file. In this example, the file owner has read and write permissions only. The next three characters (**r--**) are the permissions for the members of the same group as the file owner (which in this example is read only). The last three characters (**r--**) show the permissions for all other users and in this example it is read only.

1. **What is the different setting for directory permissions?**

**Ans.:** The chmod command is used to control the access permissions for directories. In most ways, the permissions scheme for directories works the same way as they do with files.

However,

the execution permission is used in a different way. It provides control for access to file listing and other things. Here are some useful settings for directories:

*Value Meaning*

*777 (rwxrwxrwx)* No restrictions on permissions. Anybody may list files, create new files in the directory and delete files in the directory. Generally, not a good setting.

*755 (rwxr-xr-x)* The directory owner has full access. All others may list the directory, but cannot create files nor delete them. This setting is common for directories that you wish to share with other users.

*700 (rwx------)*The directory owner has full access. Nobody else has any rights. This setting is useful for directories that only the owner may use and must be kept private from others

1. **List and explain 8 common file system commands.**

**Ans.:**

|  |  |
| --- | --- |
| **Command** | **Meaning** |
| **cat file(s)** | Send content of file(s) to standard output. |
| **chmod *mode* file(s)** | Change access permissions on file(s) |
| **cp sourcefile targetfile** | Copy sourcefile to targetfile. |
| **echo *string*** | Display a line of text |
| **file filename** | Determine file type of filename. |
| **locate *searchstring*** | Print all accessible files matching the search pattern. |
| **ls file(s)** | Prints directory content. |
| **mkdir newdir** | Make a new empty directory. |
| **mv oldfile newfile** | Rename or move oldfile. |
| **Pwd** | Print the present or current working directory. |
| **rm file** | Removes files and directories. |
| **rmdir file** | Removes directories. |
| **wc file** | Counts lines, words and characters in file. |

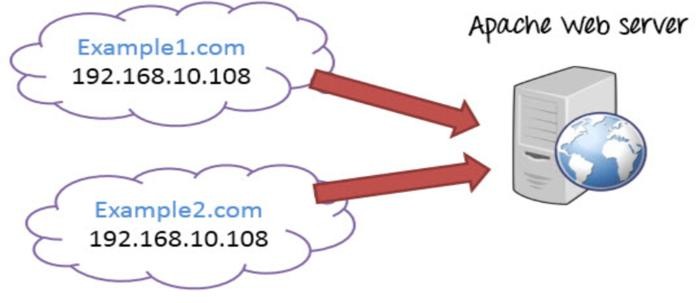
1. **What is virtual host? Differentiate IP based Virtual Hosts and Name based Virtual Hosts.**

**Ans.:** An Apache web server can host multiple websites on the **SAME** server. You do not need separate server machine and apache software for each website. This can achieved using the concept of **Virtual Host** or **VHost.** Any domain that you want to host on your web server will have a separate entry in apache configuration file.

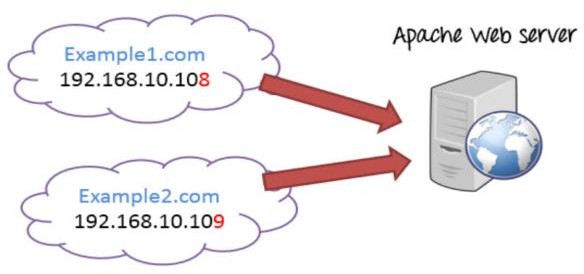
**Types of Apache Virtualhost**

1. Name-based Virtual host
2. Address-based or IP based virtual host and. Name-based Virtual Host

Name based virtual hosting is used to host multiple virtual sites on a single IP address.



In order to configure name based virtual hosting, you have to set the IP address on which you are going to receive the Apache requests for all the desired websites. You can do this by NameVirutalHost directive within the apache configuration i.e. **httpd.conf/apache2.conf file.** IP-based Virtual host In order to setup IP based virtual hosting, you need more than one IP address configured on your server. So, the number of vhost apache will depend on number of IP address configured on your server. If your server has 10 IP addresses, you can create 10 IP based virtual hosts.

****

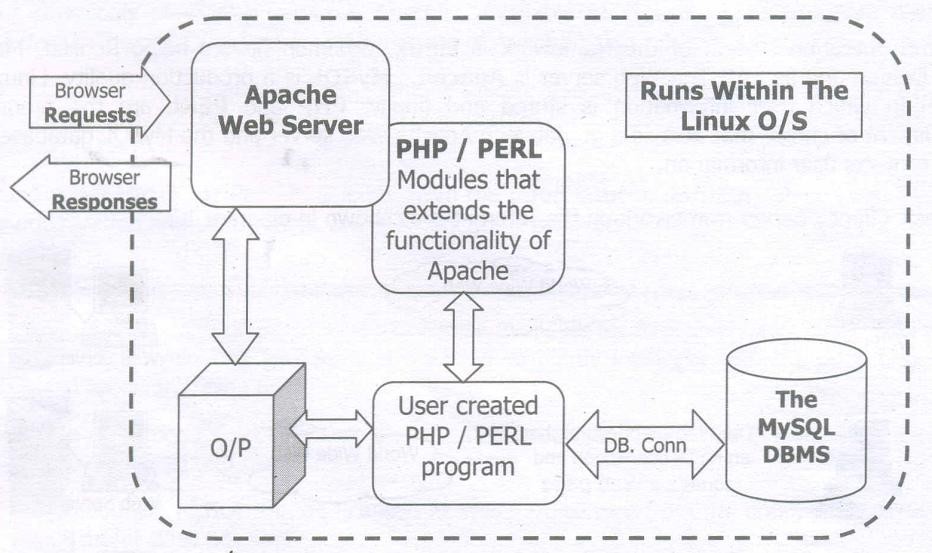
1. **Write a note on PHP and the web server architecture model.**

Ans.: The most commonly used framework on the Internet, for building interactive, database driven websites is L.A.M.P.P. as mentioned earlier this is an acronym for Linux, Apache, MySQL, PHP and PERL. Here the operating system of the framework is Linux. Common flavors being RedHat, Mandrake, SuSE, Debian and so on. The Web server is Apache. MySQL is a production quality, Linux based, RDBMS in which user information is stored and finally, PHP and PERL are the programming

Environment of choice that acts as a go between Apache Web server and the MySQL database engine, which protects user information.

Apache is the Web server responsible for responding to requests received from cIient browsers for information. MySQL is the database in which such information is stored. PHP and PERL are the middleware, programming environment of choice that can:

1. Respond to such information requests being processed by the Web server Apache
2. Access the MySQL database tables where the information requested is stored
3. Convert this to HTML
4. Return this HTML to the client browser via Apache Web Server Decomposing the server side architecture.



Now that the request/response paradigm of the Internet and the framework on which this paradigm can be implemented is known, it is necessary to actually create such a framework on a Linux box to work on.

1. **Describe the different authentication and log files.**

**Ans.:** ErrorLog /logs/error\_log

Custom Log /logs/access\_log common

The various directives that end in Log control indicate whether log files exist at all. Directives also indicate exactly where the log files exist in the Linux file system.

LockFile /var/run/httpd.lock

The LockFile directive sets the path to the lockfile used when Apache is compiled. This directive should normally be left at its default value.

PidFile /var/run/httpd.pid

It is a file in which the server should record its process identification number when it starts. ScoreBoard File /logs/apache\_runtime\_status

It is a file used to store internal server process information. Not air architectures require this But if local architecture does (This will be known because this file will be created when Apache run) then ensure that no two invocations of Apache share the same Scoreboardfile. Timeout 300

Indicates the number of seconds before Apach.ereceives and senqs a time out. KeepAlive On

Indicates whether or not to allow persistent connections (more than one request per connection

Set it to Off to deactivate. MaxKeepAliveRequests 100

Indicates the maximum number of requests to allow during a persistent connection. Set it to allow an unlimited amount. It is recommended that this number is set high, for maximum performance.

KeepAliveTimeout 15

Indicates the number of seconds to wait for the next request from the same client on the same connection.

MinSpareServers 5

MaxSpareServers 20

Indicates the Server-pool size regulation. Rather than making a user guess how many server processes are required, Apache dynamically adapts to the load it sees - that is, it tries to maintain enough server processes to handle the current load, plus a few spare servers to handle transient load spikes (e.g., multiple simultaneous requests from a single Netscape browser).

It doesthis by periodically checking how many servers are waiting for a request If there are fewer than MinSpareServers, it creates a new spare. If there are more than MaxSpareServers, some of the spares die off. The default values are probably OK for most sites.

Number of servers to start initially should be a reasonable ballpark figure. StartServers 8

Port 80

This indicates the port on which the server should run on. User and Group

The Web server's user and group values are denoted by the User and Group settings. This should be set to User ID and Group ID that the server will use to process requests. If the server runs as root, some hacker could exploit the privilege. Normally, people want to run Apache as an underprivileged user for security reasons. On Linux, this can be done, by setting both to nobody.

ServerAdmin root@localhost

Accepts the Email address, where problems with the server should be e-mailed. This address appears on some server-generated pages, such as error documents.

ServerName localhost

This sets the hostname the server will return. Set the name of the server using the ServerName directive. This is especially useful when the computer has multiple names or IP addresses.

DocumentRoot /var/www/html

This indicates the absolute path of the document tree, which is the top directory from which Apache will serve files. The DocumentRoot is the root of the Web tree and it defaults to

/usr/local/apache2/htdocs. Assuming that Apache is installed in/usr/local/apche2/, this can be changed if required.

**Long Answer Questions (Understanding)**

1. **Explain the architecture of Linux operating system.**

**Ans. :** The Linux kernel is composed of five main subsystems that communicate using procedure calls. The architecture of the kernel is one of the reasons that Linux has been successfully adopted by many users. In particular, the Linux kernel architecture was designed to support a large number of volunteer developers. Further, the subsystems that are most likely to need enhancements were architected to easily support extensibility.



**Figure 1.1:** *Decomposition of Linux System into Major Subsystems*

The Linux operating system is composed of four major subsystems:

1. **User Applications** -- the set of applications in use on a particular Linux system will be different depending on what the computer system is used for, but typical examples include a word-processing application and a web-browser.
2. **O/S Services** -- these are services that are typically considered part of the operating system (a windowing system, command shell, etc.); also, the programming interface to the kernel (compiler tool and library) is included in this subsystem.
3. **Linux Kernel** -- this is the main area of interest in this paper; the kernel abstracts and mediates access to the hardware resources, including the CPU.
4. **Hardware Controllers** -- this subsystem is comprised of all the possible physical devices in a Linux installation; for example, the CPU, memory hardware, hard disks, and network hardware are all members of this subsystem
5. **Briefly explain different subsystem of linux kernel.**

**Ans.:** The Linux kernel is composed of five main subsystems:

1. The [Process Scheduler](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_1) (SCHED) is responsible for controlling process access to the CPU. The scheduler enforces a policy that ensures that processes will have fair access to the CPU, while ensuring that necessary hardware actions are performed by the kernel on time.
2. The [Memory Manager](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_2) (MM) permits multiple process to securely share the machine's main memory system. In addition, the memory manager supports virtual memory that allows Linux to support processes that use more memory than is available in the system. Unused memory is swapped out to persistent storage using the file system then swapped back in when it is needed.
3. The [Virtual File System](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_3) (VFS) abstracts the details of the variety of hardware devices by presenting a common file interface to all devices. In addition, the VFS supports several file system formats that are compatible with other operating systems.
4. The [Network Interface](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_4) (NET) provides access to several networking standards and a variety of network hardware.
5. The [Inter-Process Communication](http://oss.org.cn/ossdocs/linux/kernel/a1/index.html#Toc_3_5) (IPC) subsystem supports several mechanisms for process-to-process communication on a single Linux system.
6. **Describe the 2 distinct region of system memory.**

**Ans.:** *System memory* in Linux can be divided into two distinct regions: *kernel space* and *user*

*space*. Kernel space is where the *kernel* (i.e., the core of the operating system) *executes* (i.e., runs) and provides its *services*. User space is that set of memory locations in which *user processes* (i.e., everything other than the kernel) run.

Memory consists of *RAM* (random access memory) cells, whose contents can be *accessed* (i.e., read and written to) at extremely high speeds but are retained only temporarily (i.e., while in use or, at most, while the power supply remains on). Its purpose is to hold programs and data that are currently in use and thereby serve as a high speed intermediary between the CPU (central processing unit) and the much slower *storage*, which most commonly consists of one or more hard disk drives (HDDs).

A *process* is an executing instance of a program. One of the roles of the kernel is to manage individua l user processes within this space and to prevent them from interfering with each other.

Kernel space can be accessed by user processes only through the use of *system calls*. System

calls are requests in a Unix-like operating system by an *active process* for a service performed by the kernel, such as *input/output* (I/O) or process creation. An active process is a process that is currently progressing in the CPU, as contrasted with a process that is waiting for its next turn in the CPU. I/O is any program, operation or device that transfers data to or from a CPU and to or from a peripheral device (such as disk drives, keyboards, mice and printers).

1. **Write a note on Manipulating the Apache2 HTTPD service.**

**Ans.:** Starting the Apache2 HTTPD service: #/usr/local/apache2/bin/a pachectl start

Stopping the Apache2 HTTPD service: # /usr/local/apache2/bin/apachectl stop

The main configuration file for configuring Apache is httpd.conf, which contains directives written in plain text. The location of this file is set at compile-time. When required this may be overridden with the -f command line flag.

The srm.conf and access.conf files can also be used to configure Apache. Other configuration files may be added and their instructions passed to Apache by using the Include directive followed the path that points to where the additional configuration files are.

Any directive may be placed in any of these configuration files. Apache is pretty flexible like that. Changes made to the main configuration files are only recognized by Apache when it is Started or restarted. If any configuration file is actually a directory, Apache will enter that directory and parse any files (and sub-directories) found there as configuration files

Apache works best if there is only one configuration file used and all its directives are placed in that file (i:e.httpd.conf). Apache also reads a file containing document mime types. This filename is set by the Types Config directive, and is mime.types by default.

Apache configuration files contain one directive per line. If a directive must continue onto the next line use back-slash '\' as the last character on the previous line. Directives in configuration files are case-insensitive, but arguments to directives are often case-sensitive. Any line beginning with a hash (#) character is ignored. Blank lines and white spaces before a directive are ignored Configuration files can be checked for syntax errors without starting the server by using

apachectlconfigest or the -t command line option. Some important entries in httpd.conf file.

1. **Explain how the Web server's user and group values are denoted.**

**Ans.:** A *user* is anyone who uses a computer. In this case, we are describing the names which represent those users. It may be Mary or Bill, and they may use the names Dragonlady or Pirate in place oftheir real name. All that matters is that the computer has a name for each account it creates, and it is this name by which a person gains access to use the computer. Some system services also run using restricted or privileged user accounts.

Managing users is done for the purpose of security by limiting access in certain specific ways.

The superuser (root) has complete access to the operating system and its configuration; it is intended for administrative use only. Unprivileged users can use the su and sudo programs for controlled privilege escalation.

Any individual may have more than one account, as long as they use a different name for each account they create. Further, there are some reserved names which may not be used such as "root".

Users may be grouped together into a "group", and users may be added to an existing group to utilize the privileged access it grants.

Linux/Unix operating systems have the ability to multitask in a manner similar to other operating systems. However, Linux's major difference from other operating systems is its ability to have multiple users. Linux was designed to allow more than one user access to the system at the same time. In order for this multiuser design to work properly, there needs to be a method to protect users from each other. This is where permissions come in to play.

1. **Write a note on CGI Model.**

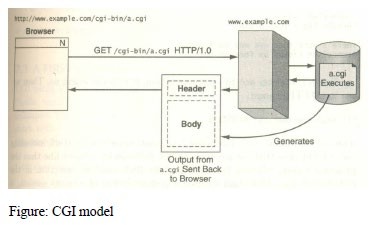
Ans. : Common Gateway Interface (CGI) is a basic way to create dynamic web pages. CGI is a standard for communication between a client and the server. CGI scripts can be written in almost any language. Perl is well suited to the types of text processing common for many tasks, such as search engines and forms interfaces. Other benefits of Perl include portability, ease of programming and overall computational power and performance. And to top it off, the Perl module CGI.pm is a useful way to make Perl CGI script writing quick and easy.

CGI scripts can do simple things that require no input from the client, such as displaying the current time or a random banner when a web page is accessed. Or they can do more complicated tasks involving posted form data from the client, such as entering a credit card number, searching a database and returning the information, and filling out a form.

Figure shown below depicts what happens during the request and execution of a CGI program. The web server recognizes a CGI request by the location of the thing requested (or

by the file name extension). For instance, if we load the URL [*www.example.com/cgi-bin/a.cgi*](http://www.example.com/cgi-bin/a.cgi)into the browser, the web server contacted, [www.example.com.](http://www.example.com/) receives a request such as the following:

GET /cgi-bin/a.cgi HTTP/1.0

The server notices that the directory that contains the thing requested is cgi -bin. It is configured to take the object requested, here a. cgi, which is a program located on the server, and execute it as a stand-alone program. The program generates standard output (in Perl, we would use print()).This output is in an important format: a header, a blank line, and the body. 

The header is a very important piece of information that is sent back to the browser because it tells the browser how to render the data that follows. The primary piece of information that is sent in the header is the Content- l2e. If the header contains content-type: text/plain, the browser displays the data that follows as plain text. If the header contains Content type: text/html, the browser treats the data that follows as HTML and renders it appropriately. And this is what is really important: Programs must output the header, then a blank line, and then the content to be displayed. The blank line is essential-it tells the browser that the header is complete and the body is about to begin.It is easy to see the header, blank line, and body output from a CGI program by using a shell and telnetting to a server. The following code is an example of connecting to a CGI program named test. cgi, which simply prints the content type, the blank line, and some important text:

$ telnet www.not\_a\_real\_web\_server.com 80 Trying 1.299.299.1

Connected to www.not\_a\_real\_web\_server.com (1.299.299.1) Escape character is '-'.

GET /cgi-bin/test.cgi HTTP/l.0

HTTP/I.1 200 OK

Date: Thu, 17 Jan 2002 19:57:05 GMT

Server: Acme Web Server Version 0.001b

Connection: close Content-Type: text/plain

There's more than one way to do it. Connection closed by foreign host

When the server accepts the connection, it tells the client so. Then we see the HTTP request: followed by a blank line. The webserver prints some header stuff, including the content type that the program prints, followed by a blank line that the program prints, followed by an important philosophica l assertion in the next-to-last line. Then, had we used a browser instead of a telnet session ,it would have taken the information in the header and the body and rendered it appropriately based on the content type in the header. Again, that blank line that separates the header and the body is important. If it is not present, a server error will result.

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# UNIT II

1. Which one of the following databases has PHP supported almost since the beginning?
2. Oracle Database
3. SQL
4. SQL+

# MySQL

1. Which one of the following statements is used to create a table?

# CREATE TABLE table\_name (column\_name column\_type);

1. CREATE table\_name (column\_type column\_name);
2. CREATE table\_name (column\_name column\_type);
3. CREATE TABLE table\_name (column\_type column\_name);

# PHP is an example of scripting language.

1. **Server-side**
2. Client-side
3. Browser-side
4. In-side

# Which of the following is not true?

1. PHP can be used to develop web applications.
2. PHP makes a website dynamic
3. PHP applications cannot be compile

# PHP cannot be embedded into html.

1. Which of the following is not used to begin php code
2. <?php
3. <?

# <php

1. “<script language=”php”>

6. PHP is a -------------------

# Loosely typed language

1. Tightly typed language
2. Server typed language
3. Client typed language
4. Full form of PHP

# PHP Hypertext Pre-processor

1. Hypertext Pre-processor
2. Plain Hypertext Pre-processor
3. Parsed Hypertext Pre-Processor
4. Php code ends with

# ;

1. :
2. .

d) ,

1. Which of the following statement is not true about echo?
2. Not written within parenthesis
3. Can output more than 1 string

# Slower than print

1. Echo($arg1[,$arg2……])
2. PHP variables must begin with a sign

# $

1. @
2. &
3. #

11. Constants are defined using which of the following function

1. Include
2. Require
3. **Define**
4. Main

12. Which of the following command gives information about the fields in a table?

1. **Describe**
2. Create
3. Use
4. Select

13. **Which of the following statements prints in PHP?**

1. Out
2. Write
3. **Echo**
4. Display

14. Which of the following is used to add multi line comment in PHP

a) {/ \}

b) //

**c) /\* \*/**

d) {{ }}

15. What will be the output of the following php code?

# a) 0 to 255

b) -128 to 127

c) 10 to 265

d) 8 to 263

1. Which among the following datatypes is not case sensitive
2. VARCHAR
3. TINYBLOB
4. TINYTEXT

# TINYBIT

1. Which of the following command is used to show the database created inside MYSQL.

# SHOW

1. USE
2. CREATE
3. SELECT
4. Which of the following command is used to connect to MYSQL
5. SHOW

# USE

1. CREATE
2. SELECT
3. Each Table within the database is been defined and created by which of the following command
4. SHOW
5. USE

# CREATE

1. SELECT
2. Which of the following command is used to print all the records that match the query?
3. SHOW
4. USE
5. CREATE

# SELECT

1. Which of the following is scalar data?
2. Boolean
3. Integer
4. Float

# Array

1. Which of the following is not true about constant?
2. There is no need to write a dollar sign ($) before a constant

# Constants can be defined by simple assignment

1. Once the Constants have been set, may not be redefined or undefined.
2. Constants may be defined and accessed anywhere without regard to variable scoping rules.

# Long Answer Questions (Application)

1. **Give the syntax of the following**
   1. **Insert**
   2. **Select**
   3. **Update**
   4. **Delete**

## Ans.: INSERT Command

This command is used to insert the records into the table created using CREATE command. The syntax of the same is

Mysql> INSERT INTO name of the table VALUES (the necessary values); mysql> INSERT INTO age\_information

-> (1astname ,firstname, age)

-> VALUES ('Wall', 'Larry', 46);

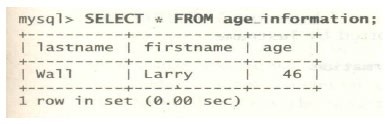
Query OK, 1 row affected (0.00 sec)

The syntax of the command is INSERT INTO, followed by the table in which to insert, a list within parentheses of the fields into which information is to be inserted, and the qualifier VALUES followed by the list of values in parentheses in the same order as the respective fields

## SELECT command

SELECT selects records from the database. When this command is executed from the command line, MySQL prints all the records that match the query.

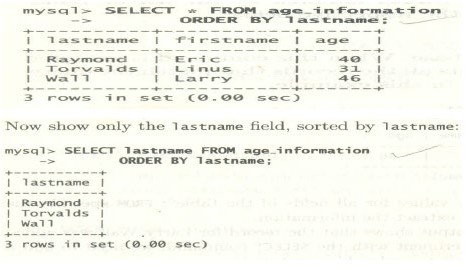
The simplest use of SELECT is shown in this example



The \* means "show values for all fields in the table"; FROM specifies the table from which to extract the information.

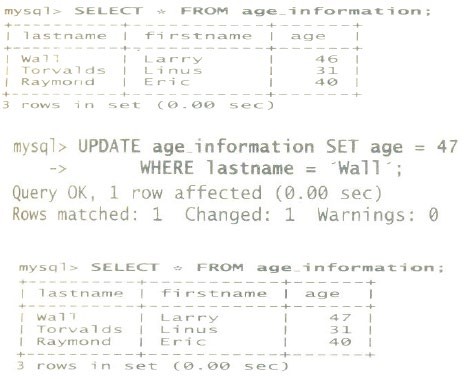
The previous output shows that the record for Larry Wall was added successfully. There are many ways to use the SELECT command-it's very flexible.

First, the table based on 1astname:



## The UPDATE Command

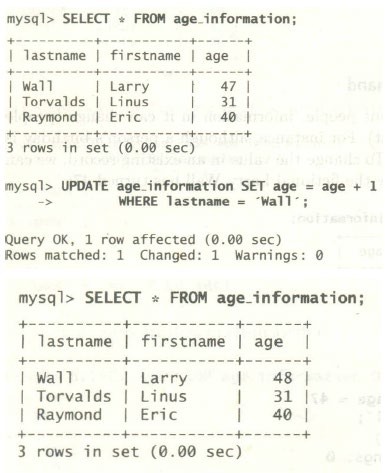
Since the database is about people, information in it can change (people are unpredictable like that). For instance, although a person's birthday is static, their age changes. To change the value in an existing record, we can UPDATE the table. Let's say the fictional Larry Wall has turned 47:



Be sure to use that WHERE clause; otherwise if we had only entered UPDATE age\_information SET age = 47, all the records in the database would have been given the age of 47!

Although this might be good news for some people in these records (how often have the oldtimers said "Oh, to be 47 years old again"-OK, probably not), it might be shocking news to others.This method works, but it requires the database to know that Larry is 46, turning 47.

Instead of keeping track of this, for Larry's next birthday we simply increment his age:



## The DELETE Command

Sometimes we need to delete a record from the table (don't assume the worst-perhaps the person just asked to be removed from a mailing list, which was opt-in in the first place, of course). This is done with the DELETE command:

mysql> DELETE FROM age\_information WHERE lastname = Raymond'; Query OK, 1 row affected (0.00 sec)

1. **Write a note on date and time datatype in MySQL**

**Ans.: Date and Time**

MySQL stores dates and times in the format of YYYY-MM-DD HH:MM:SS, unlike PHP. Your MySQL server dictates where you can store invalid dates or whether all invalid dates should be converted to zeros.

‰ DATETIME contains the date and the time. It has a range from the year 1000 through the year

9999.

‰ DATE contains just the date value.

‰ You can use TIMESTAMP to automatically contain the initial value or automatically update when something changes on the row. It has a range from 1970 through early 2038. It stores all values as of the UTC time zone.

‰ TIME displays the time portion of a date or an elapsed time.

‰ YEAR displays the year. It can be either YEAR(2) or YEAR(4) for two- or four-digit representation of the year. It has a range from the year 1901 through 2155. Two digits between 00 and 69 are converted to 2000 through 2069 and 70 to 99 are converted to 1970 through 1999.

1. **What is the use of auto increment in MySQL?**

**Ans.**  **USING AUTO\_INCREMENT**

The primary key for a table should have the following characteristics:

* + Unique
  + Not Null
  + Not optional
  + Never needs to be changed
  + Does not violate security policies

In addition, a short simple key that can be retrieved quickly helps performance. It can be difficult to find a data field that meets all of these requirements. For that reason, tables are often given artificial keys — arbitrary keys that have no meaning other than to be a primary key. MySQL supports this policy with the AUTO\_INCREMENT attribute. You assign this attribute to a field and MySQL generates a unique sequential number for each new row. You can assign AUTO\_ INCREMENT to either an integer or a floating-point data type, though an integer is the most common. Make sure that the data type you choose is large enough to hold the highest number you

need. The following snippet of code shows the typical specifications for an artificial primary key. The name of the field is id; it is an integer data type that is unsigned, is a required field, will be automatically filled by MySQL, and is assigned as the primary key.

`id` INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

The table keeps track of the next number to be assigned. It starts with 1 unless you tell it differently when you create the table. In Lesson 22, you learn how to add data to the tables. When you add the data, if you do not assign a value to id for new rows, or you assign a NULL or 0, a value is automatically assigned. MySQL has a function, LAST\_INSERT\_ID(), that contains the last AUTO\_INCREMENT value. PHP also has a function that can retrieve this number if you need it.

1. **Explain how the text string is further defined?**

**Ans.: Strings**

There are two types of strings in MySQL. The ﬁrst is text strings, which have character sets and collations. This is the type of string that you use most often. Text strings are further deﬁned as follows:

‰ CHAR: This is the character data type. You deﬁne exactly how many characters are stored. For example, if you want a ﬁeld to be exactly six characters long, you deﬁne it as CHAR(6). If you pass it data that is less than that, it pads with spaces at the end. If you pass it more, the extra characters are truncated. Whether you are truncating blanks or non-blank characters and what error reporting you have set determines what, if any, errors you see. You can go all the way up to 255 characters. Note that some character sets require more than 1 byte to store some characters. The size limits for the text strings are based on the number of characters, not the number of bytes. Trailing spaces are removed when you retrieve the data.

‰ VARCHAR: This data type has a variable number of characters. You specify the maximum number of characters, up to 65,535. If you have a ﬁ eld that could contain up to 50 characters but would likely contain less, you deﬁ ne it as VARCHAR(50). There is a little overhead when using VARCHAR rather than CHAR because 1 or 2 bytes are used to store the length. Trailing spaces are not removed when you retrieve the data.

‰ TEXT: There are four TEXT types — TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT. Like VARCHAR, the TEXT types contain a variable number of characters. The difference between the four types is the maximum number of characters. The type deﬁ nes the maximum number of characters; you do not. See Table 21-1.

TABLE 21-1:

TEXT Type Sizes

TEXT TYPE MAXIMUM CHARACTERS TINYTEXT 255

TEXT 64K

MEDIUMTEXT 16M

LONGTEXT 4G

The second type of string is binary strings, which have no character sets or collations. Character strings contain text, whereas binary strings contain raw data such as images and other media. The binary types are subdivided in the same way that the text strings are, but the size limits are based on the number of bytes, not the number of characters. ‰ BINARY: This is the binary data type. You deﬁne exactly how many bytes are stored. For example, if you want a ﬁ eld to be exactly 6 bytes long, you deﬁne it as BINARY(6). You can go all the way up to 255 bytes. ‰ VARBINARY: This data type has a variable number of bytes. You specify the maximum number of bytes, up to 65,535. If you have a ﬁeld that could contain up to 50 bytes but would likely contain less, you deﬁne it as VARBINARY(50). There is a little overhead when using VARBINARY rather than CHAR because 1 or 2 bytes are used to store the length. ‰ BLOB: There are four BLOB types — TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB. Like VARBINARY, the BLOB types contain a variable number of bytes. The difference between the four types is the maximum number of bytes. The type deﬁnes the maximum number of bytes; you do not. See Table 21-2.

TABLE 21-2: BLOB Type Sizes

BLOB TYPE MAXIMUM BYTES

TINYBLOB 255

BLOB 64K

MEDIUMBLOB 16M

LONGBLOB 4G

1. **Explain the different features of PHP.**

**Ans.:** There are given many features of PHP.

* **Pe rformance** : Script written in PHP executes much faster than those scripts written in other languages such as JSP & ASP.
* **Ope n Source Software** : PHP source code is free available on the web, you can developed all the version of PHP according to your requirement without paying any cost.
* **Platform Inde pe nde nt**: PHP are available for WINDOWS, MAC, LINUX & UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also.
* **Compatibility**: PHP is compatible with almost all local servers used today like Apache, IIS etc.
* **Embe dde d**: PHP code can be easily embedded within HTML tags and script.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **With a simple code explain the basic syntax of PHP.**

**Ans.:** A PHP script starts with <?php and ends with ?>

* + The default file extension for PHP files is ".php"
  + A PHP file normally contains HTML tags, and some PHP scripting code
  + PHP statements are terminated by semicolon (;)

In PHP, all user-defined functions, classes, and keywords (e.g. if, else, while, echo, etc.) are not case-sensitive You can also begin a scripting block with (<?) and end with (?>). This is just a shortened version.

It is always advisable to use the standard form of (<?php) in place of the shortened form (<?) as the former is clearer and generally supported.

“ <?

PHP Here we insert PHP codes

?>

“ <?php

PHP Here we insert PHP codes php

?>

“ <script language=”php”> PHP Here we insert PHP codes

</script>

As in an HTML file, PHP files also have HTML tags in addition to some PHP script code. Some important examples are given as below using the text string “Hello World” and sending it to the browser.

<html>

<body>

<?php echo “Hello World”; ?>

</body></html>

1. **Explain 2 types of comments available in php**

## Ans.: Single Line Comme nt

While there is only one type of comment in HTML, PHP has two types. The first type we will

discuss is the single line comment. The single line comment tells the interpreter to ignore everything that occurs on that line to the right of the comment. To do a single line comment type "//" and all text to the right will be ignored by PHP interpreter. PHP Code:

<?php

echo "Hello World!"; // This will print out Hello World!

echo "<br />Psst...You can't see my PHP comments!"; // echo "nothing";

// echo "My name is Humperdinkle!"; ?>

## Display:

Hello World!

Psst...You can't see my PHP comments

## Multiple Line Comment

Similiar to the HTML comment, the multi-line PHP comment can be used to comment out large blocks of code or writing multiple line comments. The multiple line PHP comment begins with "

/\* " and ends with " \*/ ".

## PHP Code:

<?php

/\* This Echo statement will print out my message to the place in which I reside on. In other words, the World. \*/

echo "Hello World!";

/\* echo "My name is Humperdinkle!";

echo "No way! My name is Uber PHP Programmer!"; \*/

?>

## Display:

Hello World!

**Long Answer Questions (Understanding)**

1. **What is the use of CREATE TABLE and SHOW TABLE Command, Give the syntax?**

## Ans.: CREATE TABLE and SHOW TABLE Command

Each table within the database must be defined and created. This is done with the CREATE TABLE command.

Create a table named age\_information to contain an individua l is first name, last name, and age. MySQL needs to know what kind of data can be stored in these fields. In this case, the first name and the last name are character strings of up to 20 characters each, and the age is an integer:

The Syntax of the same is :

## mysql> CREATE TABLE age\_information (

**->lastname CHAR(20),**

## ->firstname CHAR(20),

**-> age INT**

## -> );

**Query OK, 0 rows affected (0.00 see)**

It appears that the table was created properly (it says OK after all), but this can be checked by executing the SHOWTABLES command. If an error is made, the table can be removed with DROP TABLE.

When a database in MySQL is created, a directory is created with the same name as the database (people, in this example):

SHOW TABLES is used to show all the tables present in the database. The syntax of the same is

## Mysql> SHOW TABLES;

1. **What are steps involved to create table using MySQL**

**Ans.:** Each table within the database must be defined and created. This is done with the CREATE TABLE command.

Create a table named age\_information to contain an individua l is first name, last name, and age. MySQL needs to know what kind of data can be stored in these fields. In this case, the first name and the last name are character strings of up to 20 characters each, and the age is an integer:

The Syntax of the same is :

## mysql> CREATE TABLE age\_information (

**->lastname CHAR(20),**

## ->firstname CHAR(20),

**-> age INT**

## -> );

**Query OK, 0 rows affected (0.00 see)**

1. **Explain php echo statement with example.**

**Ans.:**  **PHP echo state ment**

In PHP ‘echo’ statement is a language construct and not a function, so it can be used without

paranthesis. But we are allowed to use paranthesis with echo statement when we are using more than one arguments with it. The end of echo statement is identified by the semi-colon (‘;’).We can use ‘echo’ to output strings or variables. Below are some of the usage of echo statement in PHP:

**Displaying Strings**: We can simply use the keyword echo followed by the string to be displayed within quotes. Below example shows how to display strings with PHP:

<?php

echo "Hello,This is a display string example!";

?>

Output:

Hello,This is a display string example!

1. **With syntax explain print statement.**

**Ans.: PHP print stateme nt**

The PHP **print** statement is similar to the echo statement and can be used alteranative to echo at many times.It is also language construct and so we may not use parenthesis : print or print(). The main difference between the **print** and **echo** statement is that print statement can have only one agrument at a time and thus can print a single string. Also, print statement always returns a value 1.Like echo, print statement can also be used to print strings and variables. Below are some examples of using print statement in PHP:

□ **Displaying String of Text**: We can display strings with print statement in the same way we did with echo statements. The only difference is we can not display multiple strings

separated by comma(,) with a single print statement. Below example shows how to display strings with the help of PHP print statement:-

<?php

print "Hello, world!";

?>

* Output:
* Hello, world!

1. **Write a note on scope of variable in php.**

## Ans.: Local Variable s

A variable declared in a function is considered local; that is, it can be referenced solely in that

function. Any assignment outside of that function will be considered to be an entirely different variable from the one contained in the function –

Example

<?php

$x = 4; assignx();

echo ("\$x outside of function is $x. <br />"); function assignx ()

{

$x = 0;

echo ( "\$x inside function is $x. <br />");

}

?>

## Function Parame ters

Function parameters are declared after the function name and inside parentheses. They are declared much like a typical variable would be <?php

// multiply a value by 10 and return it to the caller function multiply ($value){

$value = $value \*10; return $value;

}

$retval= multiply (10); Print"Return value is $retval\n";

?>

## Global Variable s

In contrast to local variables, a global variable can be accessed in any part of the program. However, in order to be modified, a global variable must be explicitly declared to be global in the function in which it is to be modified. This is accomplished, conveniently enough, by placing the keyword **GLOBAL** in front of the variable that should be recognized as global. Placing this keyword in front of an already existing variable tells PHP to use the variable having that name. Consider an example –

<?php

$somevar=15; functionaddit(){ GLOBAL $somevar;

$somevar++;

print"Somevar is $somevar";

}

addit();

?>

## Static Variable s

The final type of variable scoping that I discuss is known as static. In contrast to the variables declared as function parameters, which are destroyed on the function's exit, a static variable will not lose its value when the function exits and will still hold that value should the function be called again. You can declare a variable to be static simply by placing the keyword STATIC in front of the variable name

<?php functionkeep\_track(){ STATIC $count =0;

$count++; print $count; print"<br />";

}

keep\_track(); keep\_track(); keep\_track();

?>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Write a note on constants, how is it different from variable.**

Ans.: A constant is a name or an identifier for a simple value. A constant value cannot change during the execution of the script. By default, a constant is case-sensitive. By convention, constant

identifiers are always uppercase. A constant name starts with a letter or underscore, followed by any number of letters, numbers, or underscores. If you have defined a constant, it can never be changed or undefined.

## Differe nce s be tween constants and variable s are

* There is no need to write a dollar sign ($) before a constant, where as in Variable one has to write a dollar sign.
* Constants cannot be defined by simple assignment, they may only be defined using the define() function.
* Constants may be defined and accessed anywhere without regard to variable scoping rules.

□ Once the Constants have been set, may not be redefined or undefined.

# UNIT III

1. **Which of the following is not a valid example of variable declaration?**
   1. $name
   2. $address2
   3. $colour\_30.

# $2name

1. **Give the output for the following code**

$first\_name = 'Chhaya';

**$greeting1 = "Hello, my first name is $first\_name. "; echo $greeting1;**

# Hello, my first name is Chhaya.

* 1. Hello, my last name is $Iast\_name.
  2. $greeting1
  3. Hello, my first name is

1. To concatenate 2 strings which of the following character is used

# .

b. ,

1. $
2. \*
3. are named and indexed collections of other values.
4. Strings

# Arrays

1. Objects
2. Resources
3. hold references to resources external to PHP
   1. Strings
   2. Arrays
   3. Objects

# Resources

1. Which of the following is not a ―special datatype‖?

# Boolean

* 1. Null
  2. Array
  3. Object

1. Which of the following is used to print the next character as a dollar not as a part of variable?
   1. \d
   2. $

c. \\$

# d. \$

1. Which of the following represents conditional operator

# ?:

* 1. ??
  2. ?;
  3. ?.

1. Which keyword is used to halt the current iteration of a loop but it does not terminate the loop

# Continue

* 1. Break
  2. Halt
  3. Stop

1. Which Letter is used to represent ―AM‖ in Date

# A

* 1. a
  2. D
  3. D

1. Which of the following is used to Set the internal pointer of an array to its last element?
   1. count
   2. current
   3. next

# end

1. Which of the following determines if a variable is declared and is different than **NULL**

# Isset

* 1. Empty
  2. Current
  3. Set

1. An array with string index is called as
   1. Numeric array

# Associative array

* 1. Multidimensional array
  2. Indexed Array

1. PHP‘s numerically indexed array begin with position
2. 1
3. 2

# 0

1. -1
2. Which of the following function is used to get the value of the previous element in an array?
   1. last()
   2. before()

# prev()

* 1. previous()

1. Multidimensional arrays are simple arrays that have
   1. One dimensional

# Many arrays stored in them

* 1. No indexes
  2. 1 element

1. For finding nonempty elements in array we use
   1. is\_array ( ) function
   2. sizeof ( ) function
   3. array\_count ( ) function

# count ( ) function

1. When we simply want iteration through looping an array values we can use
   1. current ( )

# foreach ( )

* 1. next ( )
  2. prev()

1. Count ( ) function is identical to
   1. is\_array ( ) function
   2. in\_array ( ) function

# sizeof ( )

* 1. isset ( ) function

1. How does the identity operator === compare two values?

# It converts them to a common compatible data type and then compares the resulting values

* 1. It returns True only if they are both of the same type and value
  2. If the two values are strings, it performs a lexical comparison
  3. It bases its comparison on the C strcmp function exclusively

1. Which of the following is not a global variable?
   1. $\_POST
   2. $\_GLOBALS
   3. $\_REQUEST

# $NUM

1. loop will execute at least once.
   1. While

# Do while

* 1. For
  2. Foreach

1. Which among the following has highest precedence over others?

a. ==

# b. ++

c. &&

d. +

1. Which of the following is unary operator?

# --

* 1. -
  2. \*
  3. &&

1. are used in PHP to performs what is known as variable expansion

# Double quotes

* 1. Single quotes
  2. Comma
  3. Semicolon

# Long Answer Questions (Application)

1. **Explain the following with example**
   1. **Assignment operator**
   2. **Logical operator**

## Ans.: Assignme nt operators

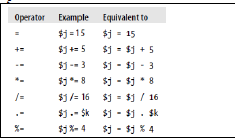
You can use the Assignment Operator to assign a value to a variable. Often a variable is assigned a value of another variable. In this case assignment operators are used. The equal character (=) is used here. Look at the following expression:

Example:

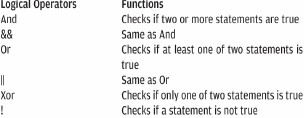
$first\_var = 5;

$second\_var = $first\_var;

Here the values of both ‘$first\_var’ and ‘$second\_var’ variables are assigned the same value i.e.



## Logical operator



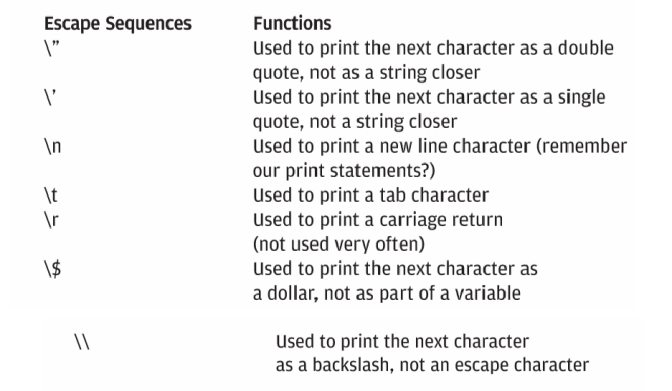
1. **What is Escaping Characters? Write brief notes on Escaping Characters.**

Ans.: **Escape Se que nces:**

You can achieve the same effect in double-quoted strings by using the escape character, which,

in PHP, is a backslash \. Escape sequences, the combination of the escape character \ and a letter, are used to signify that the character after the escape character should be treated specially. For example, if you wanted to have the string "And then he said, "That is amazing!", which was true",you would need escape characters because you have double quotes inside double quotes.

Here is a list of the escape sequences in PHP:



1. **With appropriate example differentiate post increment and pre increment.**

**Ans.: Pre-increment operator**

++$var (Pre-increment : Increments $var by 1, then returns $var)

Post-increment operator

$var++ (Post-increment : Returns $var, then increments $var by one) Example 1: **Pre -incre me nt ope rator**

<?php

$i=2;

//Increments $i by 1, then returns $i echo ++$i."<br>";

echo $i;

?>

## Output

3

3

Example 2: **Post-incre me nt ope rator**

<?php

$i=2;

//Returns $i, then increments $i by one echo $i++."<br>"; echo $i;

?>

Output 2

3

1. **Explain date () function with its syntax. List and explain any eight format which can be used with it.**

## Ans.: Date function()

In PHP the date() function is used to format a timestamp or a date. It arranges a timestamp into a readable date and time. Using the date/ time functions, you can format date and time on the server.

However, these functions entirely depend on the server settings. Here you can use the syntax,

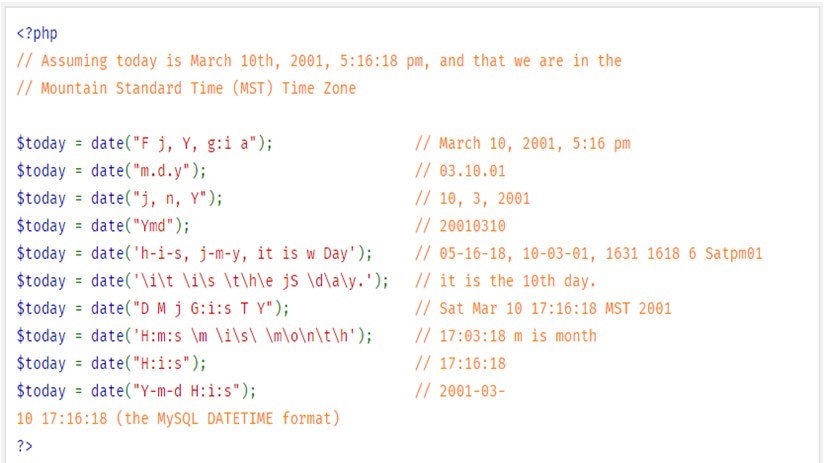
date (format, timestamp). Look at the table below:

Format: This parameter is essential. It assigns the timestamp format.

Timestamp: This Parameter is optional. This takes the Date or/and time that you want to format. If no value is provided then the current time is used for formatting.

In the date function, the first parameter specifies about formatting date and time. Several letters are used to represent date and time formats. Some commonly used letters are given below:

* d - Represents day of a month (01-31)
* D - Represents day in three letter text format
* m - Represents month, as a number (01-12)
* M - Represents month in three letter text format
* Y - Represents year in four digits
* y - Represents year in two digits
* a – “am” or “pm”
* A – “AM” OR “PM”
* F – Full name of the month ( January- December)
* g – Hours in 12- hour format without leading zero (1-12)
* G – Hours in 24- hour format without leading zero (0-23)
* h – Hours in 12- hour format with leading zero (01-12)
* H – Hours in 24- hour format with leading zero (00-23)
* i– minutes with leading zero(00-59)
* j – day of month without leading zero
* l – the full name of the day (Monday- Sunday)
* n – the month as a number, without leading zero(1-12)
* s – the seconds with leading zero(00-59)
* t – the number of days in a given month (28-31)



1. **With example, explain multi-dimensional array.**

Ans.: **Multidime nsional Arrays**

A multi-dimensiona l array each element in the main array can also be an array. And each element

in the sub-array can be an array, and so on. Values in the multi-dime nsional array are accessed using multiple index.

Example

In this example we create a two dimensional array to store marks of three students in three

subjects −

This example is an associative array, you can create numeric array in the same fashion

<html>

<body>

<?php

$marks = array( "mohammad" => array ( "physics" => 35,

"maths" => 30,

"chemistry" => 39

),

"qadir" => array ( "physics" => 30,

"maths" => 32,

"chemistry" => 29

),

"zara" => array ( "physics" => 31,

"maths" => 22,

"chemistry" => 39

)

);

/\* Accessing multi-dimensiona l array values \*/ echo "Marks for mohammad in physics : " ; echo $marks['mohammad']['physics'] . "<br />"; echo "Marks for qadir in maths : ";

echo $marks['qadir']['maths'] . "<br />"; echo "Marks for zara in chemistry : " echo $marks['zara']['chemistry'] . "<br />";

?>

</body>

</html>

Marks for mohammad in physics : 35 Marks for qadir in maths : 32

Marks for zara in chemistry : 39

1. **Explain the following with example, and give respective output.**
   1. **substr\_count()**

# strcasecmp()

Ans.: **substr\_count()**

The substr\_count() function is used to count the number of times a substring appears in a string. This function was introduced in PHP4.

<html>

<body>

<?php print substr\_count("Hello how are you and what are you doing now?", "are"); ?>

</body>

</html> Output 2

## strcase cmp()

The strcasecmp() function is used to compare two case sensitive strings. This function was introduced in PHP3. Look at the example below:

<html>

<body>

<?php

$text1 ="Good morning";

$text2 = "Good morning";

if (strcasecmp($text1, $text2) == 0) {

echo '$text1 is equal to $text2 in a case-insensitive string comparison'; }

?>

</body>

</html> Output

$text1 is equal to $text2 in a case-insensitive string comparison

**Long Answer Questions (Understanding)**

1. **What are the different data types available in PHP**

Ans.: PHP has a total of eight data types which we use to construct our variables:

* + Integers: are whole numbers, without a decimal point, like 4195.
  + Doubles: are floating-point numbers, like 3.14159 or 49.1.
  + Booleans: have only two possible values either true or false.
  + NULL: is a special type that only has one value: NULL.
  + Strings: are sequences of characters, like 'PHP supports string operations.'
  + Arrays: are named and indexed collections of other values.
  + Objects: are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
  + Resources: are special variables that hold references to resources external to PHP (such as database connections).

1. **With syntax and example explain, else if and switch statements in PHP.**

Ans.: **e lseif/e lse if:**

It is a combination of ‘if’ and ‘else’ Control Structure. If the ‘if’ Control Structure’ returns a ‘false’ value, then a different statement is executed by using the ‘else’ Control Structure’.

Example 1:

<?php

$t = date("H");

if ($t < "10") {

echo "Have a good morning!";

} elseif ($t < "20") { echo "Have a good day!";

} else {

echo "Have a good night!";

}

?>

## switch:

This Control Structure is similar to a series of ‘if’ statements.

If you want to select one of many blocks of code to be executed, use the Switch statement. The switch statement is used to avoid long blocks of if..elseif..else code.

Syntax

switch (expression)

{

case label1: code to be executed if expression = label1; break;

case label2: code to be executed if expression = label2; break;

default: code to be executed if expression is different from both label1 and label2;

}

Example

<?php

$favcolor= "red"; switch ($favcolor) { case "red":

echo "Your favoritecolor is red!"; break;

case "blue":

echo "Your favoritecolor is blue!"; break;

case "green":

echo "Your favoritecolor is green!"; break;

default:

echo "Your favoritecolor is neither red, blue, nor green!";

}

?>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Why do we need break and continue statements, explain with example?**

Ans.: **The break statement**

The PHP break keyword is used to terminate the execution of a loop prematurely.

The break statement is situated inside the statement block. If gives you full control and whenever you want to exit from the loop you can come out. After coming out of a loop immediate

statement

to the loop will be executed. Example

In the following example condition test becomes true when the counter value reaches 3 and loop terminates.

<html>

<body>

<?php

$i = 0;

while( $i< 10)

{

$i++;

if( $i == 3 )break;

}

echo ("Loop stopped at i = $i" );

?>

</body>

</html>

This will produce following result: Loop stopped at i = 3

## The continue stateme nt

The PHP continue keyword is used to halt the current iteration of a loop but it does not terminate the loop.

Just like the break statement the continue statement is situated inside the statement block containing the code that the loop executes, preceded by a conditiona l test. For the pass encountering continue statement, rest of the loop code is skipped and next pass starts.

Example

In the following example loop prints the value of array but for which condition becomes true it just skip the code and next value is printed.

<?php

$x=1;

echo 'List of odd numbers between 1 to 10 <br />'; while ($x<=10)

{

if (($x%2)==0)

{

$x++; continue;

}

else

{

echo $x.'<br />';

$x++;

}

}

List of odd numbers between 1 to 10 1

3

5

7

9

1. **What is an array give its syntax? Explain Reset () and Sizeof () function with example.**

Ans. **Arrays**

In PHP, arrays are ordered data maps and are used to store, manage and operate a set of

variables.To put it simply, an array is a data structure that holds multiple data within a single identifier.There are two parts in an Array - Values and Keys. While Values contain information to be stored,Keys are used to identify these values. It is allocated to a single variable. It holds

significant information, popularly termed as Array Elements. This information can be used for a number of times in a program. Either non negative Integers or Strings are used as Keys. The arrays that use non-negative Integers as Keys are termed as Scalar Arrays. These are Associative Arrays that use Strings as keys. An Array may contain different Array(s) popularly known as Multidimensiona l Arrays.

The syntax of an Array is as follows:

$array[key] = value;

## sizeof()

The sizeof() function returns the number of elements in an array. The sizeof() function is an alias of the count() function.

<?php

$cars=array("Volvo","BMW","Toyota "); echo sizeof($cars);

?>

Output:

3

## Reset()

The reset() function moves the internal pointer to the first element of the array.

<?php

$people = array("Peter", "Joe", "Glenn", "Cleveland"); echo next($people) . "<br>";

echo reset($people);

?>

Output Joe

Peter

1. **Define numeric and associative array with example.**

Ans. **Nume ric Array**

These arrays can store numbers, strings and any object but their index will be represented by numbers. By default array index starts from zero.

## Example

Following is the example showing how to create and access numeric arrays.

Here we have used **array()** function to create array. This function is explained in function Reference

<html>

<body>

<?php

$numbers[0] = "one";

$numbers[1] = "two";

$numbers[2] = "three";

$numbers[3] = "four";

$numbers[4] = "five";

foreach( $numbers as $value ) { echo "Value is $value <br />";

}

?>

</body>

</html> **Output** Value is one Value is two Value is three Value is four Value is five

## Associative Arrays

The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

Example

<!DOCTYPE html>

<html>

<body>

<?php

$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

echo "Peter is " . $age['Peter'] . " years old.";

?>

</body>

</html>

## OUTPUT

Peter is 35 years old.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **With syntax and example explain the following:**
   1. **String length**
   2. **Count the Number of Words in a String**
   3. **Reverse a String**
   4. **Search for a Specific Text Within a String**

**Ans.: The PHP strle n() function**

The strlen() function returns the length of a string, in characters. The example below returns the length of the string "Hello world!": strlen() is often used in loops or other functions, when it is important to know when a string ends. (i.e. in a loop, we might want to stop the loop after the last character in a string

<?php

echostrlen("Hello world!"); // outputs 12

?>

Output:

12

## Count The Numbe r of Words in a String

The PHP str\_word\_count() function counts the number of words in a string: Example

<?php

echostr\_word\_count("Hello world!"); // outputs 2

?>

Outputs:

2

## Reve rse a String

The PHP strrev() function reverses a string:

Example

<?php

echostrrev("Hello world!"); // outputs !dlrowolleH

?>

Outputs:

!dlrowolleH

## Se arch For a Spe cific Text Within a String

The PHP strpos() function searches for a specific text within a string.

If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.

The example below searches for the text "world" in the string "Hello world!": Example

<?php

echostrpos("Hello world!", "world"); // outputs 6

?>

Outputs: 6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**UNIT IV**

1. Which predefined variables are used to retrieve information from forms?
2. $GET

# $\_GET

1. $\_SET
2. GET
3. When you use the $\_GET variable to collect data, the data is visible to
4. none
5. only you

# everyone

1. selected few
2. When you use the $\_POST variable to collect data, the data is visible to
3. none

# only you

1. everyone
2. selected few
3. Which variable is used to collect form data sent with both the GET and POST methods?
4. $BOTH
5. $\_BOTH
6. $REQUEST

# $\_REQUEST

1. Which one of the following should not be used while sending passwords or other sensitive information?

# GET

1. POST
2. REQUEST
3. NEXT
4. The is a superglobal variable that stores information passed into the

script through HTTP cookies.

1. $COOKIE
2. **$\_COOKIE**
3. $SET\_COOKIE
4. $ADD\_COOKIE

7. What is the default type of ‗type‘ attribute of <input> element?

1. **Text**
2. Password
3. Numerals
4. Special Characters
5. **In PHP, cookies are set by using the**
   1. **setcookie ( ) function**
   2. set ( ) function
   3. cookie\_set ( ) function
   4. cookie()
6. variable is used to set session
   1. **$\_SESSION**
   2. $\_SETSESSION
   3. SETSESSION
   4. SESSION
7. How do we remove all session variables
   1. **session\_unset();**
   2. session\_destroy();
   3. session\_delete();
   4. session\_stop();
8. Which symbol is used for **Pass by Reference**
   1. **&**
   2. |
   3. %
   4. $
9. How do you get information from a form that is submitted using the "get" method?
   1. **$\_GET[];**
   2. Request.Form;
   3. Request.QueryString;
   4. $\_POST[];
10. Which of the following is the right one to define the function
    1. function {function body}
    2. datatype functionname(parameters){function body}
    3. function {function body}
    4. **function functionname (parameters){function body}**
11. Maximum length of get method

**a. 2048**

b. 2000

c. 2100

d. 3000

1. In PHP default behavior for user defined functions is
   1. **Call-by-value**
   2. Call-by-reference
   3. Call-by-type
   4. Call-by-loc
2. Which of the following methods would you use for sending an email?
3. **mail($to,$subject,$body)**
4. sendmail($to,$subject,$body)
5. mail(to,subject,body)
6. sendmail(to,subject,body)
7. In your PHP application you need to open a file. You want the application to issue a warning and continue execution, in case the file is not found. The ideal function to be used is:

# include()

1. require()
2. nowarn()
3. getFile(false)
4. Sessions doesn‘t allow you to
5. store persistent user preference on a site
6. save user authentication information from page to page
7. create multipage forms

# d) Doesn’t save user authentication information from page to page

1. will produce a fatal error

# Require

* 1. Include
  2. Get
  3. Post

1. Which of the following is a mandatory parameter of setcookie
   1. expiry\_time
   2. cookie\_path

# cookie\_name

* 1. domain

1. can be used to define the cookie access hierarchy
   1. expiry\_time
   2. cookie\_path
   3. cookie\_name

# domain

1. A definition of function starts with .

# Function

* 1. Datatype
  2. Function name
  3. Void

1. things are stored in the $\_FILES array when a file is uploaded?

# 5

1. 8
2. 7
3. 9
4. $\_FILES is an array

# Associative

* 1. Numeric
  2. Indexed
  3. Boolean

# Long Answer Questions (Application)

1. **Explain the use of any 2 form elements in php with appropriate example.**

**Ans.**

First, we ’ll introduce the first HTML element you ’ ll need: form . It delimits the form ’ s area in the page and holds the fields you want your web site users to fill in.

< form action=”formprocess1.php” method=”post” >

< !--form controls go here-- >

< /form >

Notice that the form element has an ending tag and two attributes. The first attribute ( action ) is the recipient page address (the form - processing script). The second attribute ( method ) is the way in which you will send the data to the recipient. You may recall that there are two separate ways of sending a form to its processing script: the POST and the GET methods.

The POST method takes the data from the form fields and sends it through an HTTP header. In this case, the data cannot be seen in the URL. The GET method gets the data from the form fields, encodes it, and appends it to the destination URL.

1. **Explain function declaration in PHP. Give example.**

**Ans.** In all programming and scripting languages, a function is a block of code that is used repetitively in a program. It saves time while developing a web page. In PHP, the concept of function is the same as in other languages. There are some in-built functions in PHP. Besides that, we can define functions as per our requirements. These are called ‘User Defined Functions’.

Look at the elements of a function:

function: all function declarations begin with the word ‘function’.

Name of the function: names to a function are usually assigned in accordance with its utility. Opening and Closing parentheses (()): the opening and closing parentheses are an integral part of a function and you can insert both the opening and closing parentheses together, just after the name of the function. As the dollar sign ($) indicates the existence of a variable, these

parentheses indicate the existence of a function.

Opening and Closing curly braces ({}): the opening curly brace ({) indicates the beginning of the function code and the closing curly brace marks the termination of a function.

Example:

<html>

<body>

<?php

function DisplayTitle()

{

echo “Learning Function”;

}

DisplayTitle();

?>

</body>

</html>

In this example, PHP codes are embedded in HTML. Here, we have used a function ‘DisplayTitle()’. This function starts with the word function and indicates that the character inserted just after this word is a function. It displays the title of the tutorial. Any one who will go through this will understand the purpose of this function from its name.

1. **Write a note on scope of variable.**

**Ans.** The origin from where a function can be accessed is called the function scope. A function, once declared, can be accessed from any section of a program. A variable scope will be local to a function, if defined within a function. Use the global key word while using a variable defined in the body part of the program

1. **Local variable s** A variable declared within a PHP function is local and can only be accessed within that function. (the variable has local scope):

<?php

$a = 5; // global scope function myTest()

{

echo $a; // local scope

}

myTest();

?>

The script above will not produce any output because the echo statement refers to the local scope variable $a, which has not been assigned a value within this scope.

You can have local variables with the same name in different functions, because local variables are only recognized by the function in which they are declared. Local variables are deleted as soon as the function is completed

1. **Global variable s** Global scope refers to any variable that is defined outside of any function. Global variables can be accessed from any part of the script that is not inside a function. To access a global variable from within a function, use the global keyword:

<?php

$a = 5; $b = 10; function myTest()

{

global $a, $b;

$b = $a + $b;

}

myTest(); echo $b;

?>

The script above will output 15.

1. **What are the elements required to build PHP form explain with example?**

**Ans.** GET and POST

When a form is submitted using the GET method, its values are encoded directly in the query string portion of the URL

When a form is submitted using the POST method, its values will not be displayed the query string portion of the URL

An interactive web site requires user input, which is generally gathered through forms. As in the paper - based world, the user fills in a form and submits its content for processing. In a web application, the processing isn ’ t performed by a sentient being; rather, it is performed by a PHP script. Thus, the script requires some sort of coded intelligence.

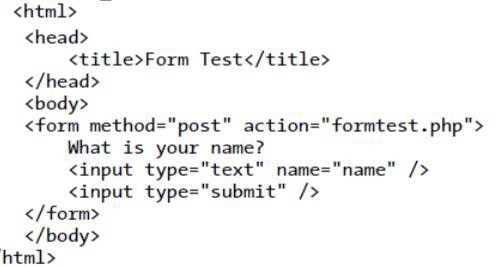
When you fill in a paper form, you generally use a means to deliver its content (the postal service is one example) to a known address (such as a mail - order bookstore). The same logic applies to online forms. The data from an HTML form is sent to a specific location and processed.

The form element is rather simple in HTML. It states where and how it will send the contents of the elements it contains once submitted. It is after that point that PHP comes into play. PHP uses a set of simple yet powerful expressions that, when combined, provide you with the means to do virtually anything you want. The PHP script receives the data from the form and uses it to perform an action such as updating the contents of a database, sending an e - mail, testing the data format, and so on.

Handling forms is a multipart process. First a form is created, into which a user can enter the required details. This data is then sent to the web server, where it is interpreted, often with some error checking. If the PHP code identifies one or more fields that require reentering, the form may be redisplayed with an error message. When the code is satisfied with the accuracy of the input, it takes some action that usually involves the database, such as entering details about a purchase

To build a form, you must have at least the following elements:

* An opening <form> and closing </form> tag
* A submission type specifying either a GET or POST method
* One or more input fields
* The destination URL to which the form data is to be submitted



1. **What is session? How the session is created explain with example.**

**Ans.** When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

Session variables solve this problem by storing user information to be used across multiple

pages (e.g. username, favoritecolor, etc). By default, session variables last until the user closes the browser.

So; Session variables hold information about one single user, and are available to all pages in one application.

## Start a PHP Session

A session is started with the session\_start() function.

Session variables are set with the PHP global variable: $\_SESSION.

Now, let's create a new page called "demo\_session1.php". In this page, we start a new PHP session and set some session variables:

Example

<?php

// Start the session session\_start();

?>

<!DOCTYPE html>

<html>

<body>

<?php

// Set session variables

$\_SESSION["favcolor"] = "green";

$\_SESSION["favanimal"] = "cat"; echo "Session variables are set.";

?>

</body>

</html>

1. **Explain mail() function with its syntax. Give example.**

## Ans. Mail() function

The mail() function is used to mail information to a given recipient. Syntax:

booleanmail(string recipient, string subject, string message [, string addl\_headers])

The recipient is the person to whom the mail has to be sent, the subject is, of course, the subject of the email. The message is the textual body of the email, and the optional input parameter addl\_headers is used to supply any additional header information (such as HTML formatting) that is sent along with the email.

The codespec for a emailinfo.php is

<?php

// Change -$recipient to be the recipient of the form information

$recipient= "admin@ivanbayross.com";

// Email subj ect

$subject= "User Comments from LPOST['name']";

// Extra email headers

$headers= "From: LPOST['email'] ";

// Create a message

$message= "Messagefrom LPOST['name'] \n\n Most PurchasedProduct:

$POST['products'] \n\n Comments: \n\n LPOST['comments']";

// Send the email or produce an error

mail($recipient, $subject, $message, $headers) or die("Could not send email!");

// Send the user an appropriate message

print "Thank you LPOST['name'] for taking a moment to send us your comments!";

?>

Output:

Thank you Sharanam Shah for taking a moment to send us your comments!

# Long Answer Questions (Understanding)

1. **Write a simple php form with at least 4 input fields to display values obtained from input fields.**

**Ans.** <html>

<head>

<title> Form </title>

</head>

<body>

<form action="b.php" method="post">

<table>

<tr>

<td> Name </td>

<td> <input type="text" name="name"/ > </td>

</tr> <tr>

<td colspan="2" style="text-align: center;" >

<input type="submit" name="submit" value="Submit" / > </td>

</tr><tr>

<input type="checkbox" name="check\_list[]" value="C/C++"><labe l>C/C++</labe l><br/>

<input type="checkbox" name="check\_list[]" value="Java"><la bel>Java</label><br/>

<input type="checkbox" name="check\_list[]" value="PHP"><label>PHP</labe l><br/>

</tr>

<tr>

Vegetables <select name="veg" size="1">

<option value="Peas">Peas</option>

<option value="Beans">Beans</option>

<option value="Carrots">Carrots</option>

<option value="Cabbage">Cabbage </option>

<option value="Broccoli">Broccoli</option>

</select>

</tr>

</table>

</form>

</body>

</html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Why do we use the get method, explain the variable used to fetch the values passed by get method, give the drawback of the same?**

## Ans. PHP $\_GET Variable

In PHP, the **$\_GET** variable is used to collect values from HTML forms using method *get*.

Information sent from an HTML form with the GET method is displayed in the browser's address bar, and it has a limit on the amount of information to send.

Example:

<html>

<body>

<form action="registration.php" method="get"> Name: <input type="text" name="name"> Email: <input type="text" name="email">

<input type="submit">

</form>

</body>

</html

* The variable names and values will be visible in URL if HTML forms submitted by the GET method.
* The GET method is restricted to send up to *2048 characters* only.
* When you submit sensitive information like passwords then should not use this method.
* GET method can't be used, to send binary data like images and Word documents.
* GET method data can be accessed using PHP QUERY\_STRING environment variable. PHP $\_GET associative array is used to access all the sent information by GET method

1. **Write a note on POST method and $\_POST variable.**

**Ans.** In PHP, the ***$\_POST*** variable is used to collect values from HTML forms using method *post*. Information sent from a form with the POST method is invisible and has no limits on the amount of information to send.

**Note :** However, there is an 8 MB max size for the POST method, by default (can be changed by setting the post\_max\_size in the php.ini file).

Example

<html>

<body>

<formaction="registration.php"method="post"> Name: <inputtype="text"name="name ">

Email: <inputtype="text"name="ema il">

<inputtype="submit">

</form>

</body>

</html>



The POST method does not have any restriction on data size to be sent. The POST method can be used to send ASCII as well as binary data.

The data sent by POST method goes through HTTP header, so security depends on HTTP

protocol. By using Secure HTTP, you can make sure that your information is secure.

PHP $\_POST associative array is used to access all the sent information by POST method. Variables are not visible in the URL so users can't bookmark your page.

1. **Why do we need include and require statements? Differentiate include and require statements in PHP. With php script.**

**Ans. The include and re quire stateme nts are ide ntical, except upon failure :**

* require will produce a fatal error (E\_COMPILE\_ERROR) and stop the script
* include will only produce a warning (E\_WARNING) and the script will continue

So, if you want the execution to go on and show users the output, even if the include file is missing, use the include statement. Otherwise, in case of FrameWork, CMS, or a complex PHP application coding, always use the require statement to include a key file to the flow of execution.

This will help avoid compromising your application's security and integrity, just in-case one key file is accidentally missing.

Including files saves a lot of work. This means that you can create a standard header, footer, or menu file for all your web pages. Then, when the header needs to be updated, you can only update the header include file.

## Example 1

Assume we have a standard footer file called "footer.php", that looks like this:

<?php

echo "<p>Copyright &copy; 1999-" . date("Y") . " W3Schools.com</p>";

?>

To include the footer file in a page, use the include statement:

<!DOCTYPE html>

<html>

<body>

<h1>Welcome to my home page!</h1>

<p>Some text.</p>

<p>Some more text.</p>

<?php include 'footer.php';?>

</body>

</html>

The output will be

## Welcome to my home page !

1. **Give the syntax of cookie, explain the parameters in it.**

**Ans.** A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

Setting Cookies With PHP

A cookie is created with the setcookie() function. Syntax

<?php

setcookie(cookie\_name, cookie\_value, [expiry\_time], [cookie\_path], [domain], [secure], [httponly]);

?>

HERE,

* Php“setcookie” is the PHP function used to create the cookie.
* “cookie\_name” is the name of the cookie that the server will use when retrieving its value from the $\_COOKIE array variable. It’s mandatory.
* “cookie\_value” is the value of the cookie and its mandatory
* “[expiry\_time]” is optional; it can be used to set the expiry time for the cookie such as 1 hour. The time is set using the PHP time() functions plus or minus a number of seconds greater than 0 i.e. time() + 3600 for 1 hour.
* “[cookie\_path]” is optional; it can be used to set the cookie path on the server. The forward slash “/” means that the cookie will be made available on the entire domain. Sub directories limit the cookie access to the subdomain.
* “[domain]” is optional, it can be used to define the cookie access hierarchy i.e. [www.cookiedomain.com](http://www.cookiedomain.com/) means entire domain while [www.sub.cookiedomain.com](http://www.sub.cookiedomain.com/) limits

the cookie access to [www.sub.cookiedomain.com](http://www.sub.cookiedomain.com/) and its sub domains.

1. **How to create a upload script in PHP, explain the different variables in it.**

**Ans.**

There is one global PHP variable called $\_FILES. This variable is an associate double dimension array and keeps all the information related to uploaded file. Five things are stored in the $\_FILES array when a file is uploaded, So if the value assigned to the input's name attribute in uploading form was file, then PHP would create following variables −

* $\_FILES['file']['tmp\_name'] − the uploaded file in the temporary directory on the web server.
* $\_FILES['file']['name'] − the actual name of the uploaded file.
* $\_FILES['file']['size'] − the size in bytes of the uploaded file.
* $\_FILES['file']['type'] − the MIME type of the uploaded file.
* $\_FILES['file']['error'] – Error code resulting from file upload

## Upload1.html

<html>

<body>

<form action="upload. php" enctype="multipart/form-data" method="post"> Your File Name <input type="file " name="file"/><br/>

<input type="submit" value="Upload" name="upload"/>

</form>

</body>

## upload.php

<?php

if ($\_POST['upload'] )

{

if (

move\_uploaded\_file($\_FILES["file "]["tmp\_name "], "/var/www/" . $\_FILES["file"]["na me"]))

{

echo "uploaded successfully!"; echo"</br>";

echo "Upload: " . $\_FILES["file"]["na me "] . "<br>";

echo "Type: " . $\_FILES["file "]["type "] . "<br>";

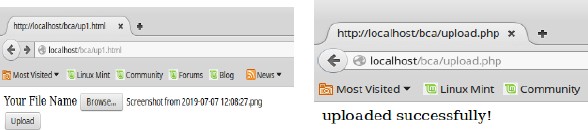
echo "Size: " . ($\_FILES["file "]["size"] / 1024) . " kB<br>"; echo "Stored in: " . $\_FILES["file"]["tmp\_name "];

} else {

print "Upload failed!";

}

}

?>

# ------------------------------------------------------------------------------------------

# UNIT V

1. Which Statement Is Used To Insert A New Data In A Database?
2. INSERT NEW
3. Add New

# INSERT INTO

1. UPDATE
2. Which Of These Commands Will Delete A Table Called XXX If You Have Appropriate Authority:

# DROP TABLE XXX

1. DELETE XXX WHERE Confirm = "YES"
2. DROP XXX
3. DROP XXX WHERE Confirm = "YES"
4. MySQL Runs On Which Operating Systems?
5. Unix And Linux Only

# Unix, Linux, Windows And Others

1. Linux And Mac OS-X Only
2. Any Operating System At All
3. Which Of The Following Is Available In MySQL:
4. CREATE VIEW
5. CREATE SCHEMA
6. CREATE TRIGGER

# CREATE DATABASE

1. Which Of These Is Not A Valid Name For A Column
2. To

# From

1. Far
2. Near
3. Which of the following is NOT available in MySQL:

# FETCH

1. LIKE
2. JOIN
3. SELECT
4. To use MySQL on your computer, you'll need?
5. FTP and Telnet
6. **Some sort of client program to access the databases**
7. A Browser
8. Perl, PHP or Java
9. MySQL is

|  |
| --- |
| a) Protocol |
| b) A Programming language |
| c) A technique for writing reliable programs |
| **d) Relational Database Management System** |

1. Which of the following is used to delete an entire MYSQL database?
2. mysql\_drop\_database
3. mysql\_drop\_entiredb

# mysql\_drop\_db

1. mysql\_drop\_dbase
2. Which of the following commands should be used to create a database named -student‖?
3. CREATE student

# CREATE DATABASE student

1. DATABASE /student
2. DATABSE student
3. A SELECT command without a WHERE clause returns?
4. All the records from a table that match the previous WHERE clause

# All the records from a table, or information about all the records

1. SELECT is invalid without a WHERE clause
2. Nothing
3. What does SQL stand for?
4. Strong question Language

# Structured Query language

1. Structured question language
2. Strong query language
3. With SQL, how do you select a column named "FirstName" from a table named Persons"?
4. Select Persons.Firstname
5. Extract firstname from persons

# Select firstname from persons

1. Select firstname
2. With SQL, how do you select all the columns from a table named "Persons"?
3. Select persons
4. select [all] from persons

# Select \* from persons

1. Select \*.persons
2. The result of select command is
3. File
4. Report

# Table

1. Form
2. Full form of **PDO**

# PHP Data Objects

1. Permanent Data Object
2. Public Data Object
3. Private Data Object
4. Output on executing mysql\_fetch\_row()

# Array

1. String
2. Number
3. Object
4. Using SELECT will return all the fields.

# \*

# .

c) +

d) &

1. Which of the following is not true about MYSQL
2. Fast
3. Easy

# Not Secure

1. Scalable
2. Which function returns an error massage?

# mysql\_error();

1. mysql\_errors();
2. mysql–error();
3. mysql–er();
4. What is function used for close mysql database connection?
5. mysql=close()

# mysql\_close()

1. mysql->die()
2. mysqli->end()
3. Output of **mysql\_query is**

# Boolean

1. Array
2. String
3. Object
4. Offset value starts from

# 0

1. 1
2. 2
3. 3
4. if limit clause has only 1 parameter its considered as
5. offset

# limit

1. same value is considered for both offset and limit
2. error
3. Which of the following takes both numeric and string value as its index
4. mysql\_fetch\_array()

# mysql\_fetch\_assoc()

1. mysql\_fetch\_row()
2. mysql\_fetch()

# Long Answer Questions(Application)

1. **Define MySQL? What are the different advantages of MySQL.**

**Ans.**  MySQL runs on a wide range of operating systems, including UNIX or Linux, Microsoft Windows, Apple Mac OS X, and others.

MySQL is easy to use, yet extremely powerful, fast, secure, and scalable MySQL supports standard SQL (Structured Query Language).

MySQL is ideal database solution for both small and large applications. MySQL is developed, and distributed by Oracle Corporation.

MySQL includes data security layers that protect sensitive data from intruders.

----------------------------------

1. **Explain different ways to connect to database with appropriate syntaxes**

**Ans.:** In PHP you can easily do this using the mysqli\_connect() function. All communication between PHP and the MySQL database server takes place through this connection. Here're the basic syntaxes for connecting to MySQL using MySQLi and PDO extensions:

*Syntax: MySQLi, Procedural way*

$link = mysqli\_connect("hostna me", "username", "password", "database");

*Syntax: MySQLi, Object Oriented way*

$mysqli = new mysqli("hostname ", "username", "password", "database");

*Syntax: PHP Data Objects (PDO) way*

$pdo = new PDO("mysql:host=hostname;dbna me=database", "username", "password"); The *hostname* parameter in the above syntax specify the host name (e.g. localhost), or

IP address of the MySQL server, whereas the *username* and *password* parameters specifies the credentials to access MySQL server, and the *database* parameter, if provided will specify the default MySQL database to be used when performing queries.

1. **Write a note on Select command in php script with where clause.**

**Ans:** The SQL **SELECT** command is used to fetch data from the MySQL database. You can use this command at mysql> prompt as well as in any script like PHP.

Syntax

Here is generic SQL syntax of SELECT command to fetch data from the MySQL table −

SELECT field1, field2,...fieldN FROM table\_name1, table\_name2... [WHERE Clause]

[OFFSET M ][LIMIT N]

* + You can use one or more tables separated by comma to include various conditions using a WHERE clause, but the WHERE clause is an optional part of the SELECT command.
  + You can fetch one or more fields in a single SELECT command.
  + You can specify star (\*) in place of fields. In this case, SELECT will return all the fields.
  + You can specify any condition using the WHERE clause.
  + You can specify an offset using **OFFSET** from where SELECT will start returning records. By default, the offset starts at zero.
  + You can limit the number of returns using the **LIMIT** attribute.

----------------------------

1. **Write a php script to create and delete MySQL Database.**

**Ans.: CREATE**

Optional − if not specified, then the last opened connection by mysql\_connect will be used. Example

<html>

<center>

<h3>Display All</h3>

<?php

$dbhost = 'localhost';

$dbuser = 'root';

$dbpass = 'password';

$conn = mysql\_connect("loca lhost",$dbuser, $dbpass) or die("conn to db failed!"); if(! $conn ) {

die('Could not connect: ' . mysql\_error());

}

echo 'Connected successfully'; mysql\_select\_db( 'TUTORIALS' ); mysql\_close($conn);

?>

</body>

</tr>

</table>

</html>

## DELETE

Try out following example to drop a database.

<?php

$dbhost = 'localhost:3036';

$dbuser = 'root';

$dbpass = 'rootpassword';

$conn = mysql\_connect($dbhost, $dbuser, $dbpass); if(! $conn ) {

die('Could not connect: ' . mysql\_error());

}

$sql = 'DROP DATABASE test\_db';

$retval = mysql\_query( $sql, $conn ); if(! $retval ) {

die('Could not delete database db\_test: ' . mysql\_error());

}

echo "Database deleted successfully\n"; mysql\_close($conn);

?>

1. **With syntax, Explain Update and Delete command.**

**Ans. : Delete**

If you want to delete a record from any MySQL table, then you can use the SQL command DELETE FROM. You can use this command at the mysql> prompt as well as in any script like PHP.

Syntax

The following code block has a generic SQL syntax of the DELETE command to delete data from a MySQL table.

DELETE FROM table\_name [WHERE Clause]

* + If the WHERE clause is not specified, then all the records will be deleted from the given MySQL table.
  + You can specify any condition using the WHERE clause.
  + You can delete records in a single table at a time.

The WHERE clause is very useful when you want to delete selected rows in a table. Eg: $sql = 'DELETE FROM tutorials\_tbl WHERE tid = 1';

## Update

There may be a requirement where the existing data in a MySQL table needs to be modified. You can do so by using the SQL UPDATE command. This will modify any field value of any MySQL table.

Syntax

The following code block has a generic SQL syntax of the UPDATE command to modify the

data in the MySQL table −

UPDATE table\_name SET field1 = new-value1, field2 = new-value2 [WHERE Clause]

You can update one or more field altogether.

You can specify any condition using the WHERE clause. You can update the values in a single table at a time.

The WHERE clause is very useful when you want to update the selected rows in a tab Eg:$result=mysql\_query("update user set fname='$fname',lname='$lna me',age= '$age' where id='$id' ");

1. **Write a note on Limit Clause.**

**Ans.**  In MySQL the LIMIT clause is used with the SELECT statement to restrict the number of rows in the result set. This feature is very helpful for optimizing the page loading time as well as to

enhance the readability of a website. For example you can divide the large number of records in multiple pages using pagination, where limited number of records will be loaded on every page from the database when a user request for that page by clicking on pagination link. The Limit Clause accepts one or two arguments which are offset and count.The value of both the

parameters can be zero or positive integers.

Offset:It is used to specify the offset of the first row to be returned. Count:It is used to specify the maximum number of rows to be returned.

The Limit clause accepts one or two parameters, whenever two parameters are specified, the first is the offset and the second denotes the count whereas whenever only one parameter is specified, it denotes the number of rows to be returned from the beginning of the result set.

Syntax:

SELECT column1, column2, ... FROM table\_name

LIMIT offset, count;

To retrieve the first three rows from the table “Data”, we will use the following query: SELECT \* FROM Data LIMIT 3;

To retrieve the rows 2-3(inclusive) from the table “Data”, we will use the following query: SELECT \* FROM Data LIMIT 1, 2;

----------------------------------------------

# Long Answer Questions(Skill)

1. **What are the different areas where PHP and MySQL are being used together?**
   * **Ans.:** Banner Rotation: Here each banner is a PHP script. Using database random banners are picked up and shown to the visitor. In addition to that a track of the number of times the banner has been viewed can be kept. Banners can be added, changed or edited just by making changes in the database and the script picks the correct banners for all the pages on the site.
   * Forums: Hundreds of forums (message boards) on the internet are run using PHP and MySQL. These are much more efficient than other systems that create a page for each message and offer a wide variety of options. All the pages in the forum can be updated by changing one script.
   * Databases: One quite obvious example is sites, which get all there information from a database. For example Script Avenue is run by a few scripts, which gain all their information from a large database. All the different script categories can be accessed in one script by just changing the URL to access a different part of the database.
   * Web sites: For a large website and changing the design can take a very long time to update and upload all the pages. With PHP and MySQL the whole website could be just one or two PHP scripts. These would access a MySQL database to get the information for the pages. To update the website's design just change one page.
2. **What are the different functions related to database used in PHP?**

## Ans.: Mysql Function in php Connect

Connection to Mysql database can be established by using mysql\_connect function. We can check the success of the function by checking the result. We will get a true result in case connection is established.This function takes three parameters, first one is hostname then user-id and them password.

Here is the function to connect to mysql database mysql\_connect ("$servername","$dbuser","$dbpassword"); **Select Database**

The mysql\_select\_db() function sets the active MySQL database. This function returns TRUE on success, or FALSE on failure.

Syntax mysql\_select\_db(database,connection)

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| database | Required. Specifies the database to select. |
| connection | Optional. Specifies the MySQL connection. If not specified, the last connection opened by mysql\_connect() or mysql\_pconnect() is used. |

Example

<?php

$con = mysql\_connect("localhost", "peter", "abc123"); if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db", $con); if (!$db\_selected)

{

die ("Can\'t use test\_db : " . mysql\_error());

}

mysql\_close($con);

?>

## Close

To close a mysql connection we can use mysql\_close() function. It can takes a optional parameter as link and closes it. If no link identifier is specified then last opened connection is closed. It is not necessary to use mysql\_close() function as all connections are closed at the end of the script execution. Here is the function

mysql\_close()

## Accessing MySQL database from web with PHP : Querying the database:

mysql\_db\_query — Queries a MySQL database.

**mysql\_query()** executes *query* on *database* using the MySQL server connection referenced by *connection* . If no connection handle is specified in the *connection* argument, the last opened connection will be used by default. If no connection is open, **mysql\_query()** attempts to connect to a MySQL database by calling [**mysql\_connect()**](http://www.phpdig.net/ref/rn41re774.html) without arguments.

The value returned depends on the query made. **SELECT**, **DESCRIBE**, **EXPLAIN**, and **SHOW** queries return a MySQL result handle if successful or FALSE on failure. Note that the previous types of queries are considered to have failed only if they're malformed

mysql\_query ("SELECT \* FROM database.table");

## Retriving Query results

*mysql\_fetch\_row()* function returns a record by taking a result identifier. Here it returns the set of data as an array. To get the value from the array we have to use array offset staring from 0. Each call to this mysql\_fetch\_row function returns the next record. Here one record is returned at a time and returns false if there is no more record to return. So we can easily use this function inside a while loop and display all the records. Each time at the starting of the while loop the mysql\_fetch\_row() function is checked and if true then the loop is executed and the records are displayed

-----------------------------------------------------

1. **What are the different items used while creating tables, explain with a simple example.**

**Ans.** PHP uses **mysql\_que ry** function to create or delete a MySQL database. This function takes two parameters and returns TRUE on success or FALSE on failure.

Syntax

bool mysql\_query( sql, connection );

## sql

Required - SQL query to create or delete a MySQL database

## conne ction

Optional - if not specified, then the last opened connection by mysql\_connect will be used.

<html>

<center>

<h3>Display All</h3>

<?php

$dbhost = 'localhost';

$dbuser = 'root';

$dbpass = 'password';

$conn = mysql\_connect("loca lhost",$dbuser, $dbpass) or die("conn to db failed!"); if(! $conn ) {

die('Could not connect: ' . mysql\_error());

}

echo 'Connected successfully<br />';

$sql = 'CREATE DATABASE TUTORIALS';

$retval = mysql\_query( $sql, $conn ); if(! $retval ) {

die('Could not create database: ' . mysql\_error());

}

echo "Database TUTORIALS created successfully\n"; mysql\_close($conn);

?>

</tr>

</table>

</html>

1. **How to Insert values into database using MySQL?**

**Ans.:**

The INSERT INTO statement is used to insert new rows in a database table.

Let's make a SQL query using the INSERT INTO statement with appropriate values, after that we will execute this insert query through passing it to the PHP mysqli\_query() function to insert data in table.

<html>

<center>

<h3>Display All</h3>

<?php

$connect = mysql\_connect("localhost","root","password") or die("conn to db failed!"); mysql\_select\_db ("TUTORIALS") or die ("Db not found" );

$sql = "INSERT INTO tutorials\_tblVALUES('', 'C++', 'Balaguru Swami')";

$result=mysql\_query($sql) or die(mysql\_error()); if($result)

{

?>

<SCRIPT language=JavaScript>

alert("Record Inserted Successfully...");

</script>

<?php

}

?>

</body>

</html>

1. **What are the different methods to fetch the record from database?**

**Ans.:**

Fetch Row

*mysql\_fetch\_row()* function returns a record by taking a result identifier. Here it returns the set of data as an array. To get the value from the array we have to use array offset staring from 0. Each call to this mysql\_fetch\_row function returns the next record. Here one record is returned at a time and returns false if there is no more record to return. So we can easily use this function inside a while loop and display all the records. Each time at the starting of the while loop the

mysql\_fetch\_row() function is checked and if true then the loop is executed and the records are displayed.

<? $query=mysql\_query("se lect \* from student"); echo mysql\_error(); echo "<b>id,na me,class,mark</b><br>"; while($nt=mysql\_fetch\_row($query)){ echo "$nt[0],$nt[1],$nt[2],$nt[3] <br>"; } ?>

Fetch assoc mysql\_fetch\_assoc

The function returns an associative array of strings that corresponds to the fetched row, or FALSE if there are no more rows. The associativity array tells us about the key value pair, whereas the key tells about any column name and the value tells about the row value.

Here we can map the column name as key and value as row. For example. Key is ID and value is corresponding name.

<?php

$dbhost = 'localhost';

$dbuser = 'root';

$dbpass = 'password';

$conn = mysql\_connect("loca lhost",$dbuser, $dbpass,"bca") or die("conn to db failed!"); mysql\_select\_db("bca");

$result = mysql\_query("SELECT \* FROM login"); while ($row = mysql\_fetch\_assoc($result)) {

echo $row["id"]; echo $row["uname"];

}

?>

1. **With appropriate example explain following:**
2. **Function to execute the query**
3. **Function to display the error**

**Ans.:**

1. **Function to execute the query**

## Execute Query

This function mysql\_query() is used to pass a sql query to mysql database. We can write any sql query like insert , select , update , delete etc and pass the query to mysql database. The result of the execution of the query will be known by monitoring the status. We will get true or false based on the success or failure status of the query. We can specify and connecting link as an optional parameter. If no connecting link is specified it assumes default connection.

<?Php

$query="select name,class,mark from student";

$status=mysql\_query($query) or die( "query failed"); if($status){echo "Query is successful";}

else {echo "Qyery failed";}

?>

## Function to display the error

## Error

We can display error message in case of an error generated by mysql query. This meaning full

error message gives idea one the problem or bugs in the script. We can print the error message by using mysql function mysql\_error(). This function returns the error message associated with most recently executed query.

$querry = mysql\_query("SELECT new\_field FROM student"); echo "Error message = ".mysql\_error();

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_