SQL

SQL stands for Structured Query Language. It is a language that enables us to create and operate on relational databases, which are sets of related information stored in tables. Initially it was called ***Sequel (Structured English Query Language).*** The SQL language is defined by ANSI (American National Standards Institute) and is currently accepted as well by ISO (International Standards Organization). There exist many programs that can manage data and use SQL very efficiently like Oracle, Microsoft SQL, Microsoft Access, Visual FoxPro etc.

SQL Data Types

Data type	**Syntax**	**Description**
Character	char / character (size)	Strings of text(fixed length)
Varchar	varchar / varchar2(size)	Strings of text(variable length)
Integer	int / integer	Number with no decimal part
Number	number(size [,scale])	Numeric with/without decimal part
Decimal	dec (size,scale)	Decimal value
Date	date	Date value
Time	time	Time value

SQL Commands

- **DDL** stands for Data Definition Language, is also called schema definition language as it provides commands for creation, modification and deletion of tables. Thus DDL commands are concerned with the overall structure of the database.

Ex**:** CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE VIEW, DROP VIEW

- **DML** stands for Data Manipulation Language, which provides commands for the manipulation of the contents of the table. The manipulation of data includes the insertion of new rows into the tables; replacement or changing of existing values in the rows with the new ones; and the deletion of existing rows from the tables.

Ex**:** INSERT, UPDATE, DELETE, SELECT

- **DCL** stands for Data Control Language, which consists of features that determine whether a user is permitted to perform a particular action. This is considered part of DDL in ANSI.

Ex**:** Grant, Revoke

<u>Grant</u>:- It is used to allow privileges.
<u>Revoke</u>:- To revoke already granted privileges.

1. CREATE TABLE

This command is used to create a table having a set of columns with constraints. Each table must have least one column.

Syntax:

```
CREATE TABLE TableName(Columnname Datatype Size [ColumnConstraint , …………….);
```

Ex: CREATE TABLE student (Rollno number (3), studname char(10),mark integer);

Column Constraints

These are some keywords by which some restrictions can be implemented in the desired columns and these are written immediately after the data type of the column. These constraints ensure database integrity and hence they are often called **Database integrity constraints**. The following are the constraints:

- ***NOT NULL***

This constraint specifies that a column can never have null (empty, but not zero or blank) values.

- ***UNIQUE***

It ensures that no two rows have the same value in the specified column. It is used to avoid duplication. The values must not be repeated across the columns.

- ***PRIMARY KEY***

It declares a column as the primary key of the table. This constraint is very similar to unique constraint except that it can be applied only to one column or a combination of columns. Also since primary key cannot contain a null value. This constraint must be defined only on the columns defined as a not null.

- ***DEFAULT***

It is used to set a default value for a column, when the user does not enter a value for the column.

- ***CHECK***

This constraint limits values that can be inserted into a column of a table. It is used to check with a specified condition.

2. INSERT INTO

This command is used for inserting a row or tuple in the table. That is new records can be added to the table.

Syntax:

```
INSERT INTO  Tablename VALUES (value1, value2,……..);
```

Ex: INSERT INTO student VALUES (100,’Pooja’, 85);

3. SELECT

This command is used to retrieve or select rows or columns from a table.

Syntax**:**

```
SELECT Columnname1, ……..FROM  Tablename;
```

- To display the roll number of all students in the student table

SELECT rollno FROM student;

- To display all the details of the table student

SELECT * FROM student;

WHERE

It is used to select rows that satisfy some conditions with select/delete/update command.

Syntax**:**

```
SELECT  ColumnName1,……FROM  Tablename [WHERE condition];
```

- To display the name and course of the students whose percent greater than 75.

SELECT name, course FROM student WHERE percent>75;

AND

- To display all male students in science course

SELECT * FROM student WHERE sex='m' AND course='science';

OR

- To display admission no and name of students whose course is science or commerce and percent>65

SELECT admno, name FROM student WHERE course='science' OR course='commerce' AND percent >65;

BETWEEN…. AND

- To display the details of those students with percentage between 70 and 80

SELECT * FROM student WHERE percent BETWEEN 70 AND 80;

IS NULL

- To display the name of students whose course is not specified.

SELECT name from student where course IS NULL;

NOT

- Display student details whose course is specified.

Select * from student where course IS NOT NULL;

DISTINCT

The keyword distinct is used with select command to avoid duplication of rows in the selection. It eliminates the redundant data. Another keyword **ALL** used to display all values with duplicates.

- A query to know the courses of the student table

SELECT DISTINCT course FROM student;

- To display the departments of employee table

SELECT Distinct department from employee;

LIKE

This operator is used for setting a condition with pattern matching. The string matching operator % (percent) substitutes a group of characters and the symbol _ (underscore) substitutes only one character of the value.

- To display the names which start with letter   s from student table

SELECT name FROM student WHERE name LIKE 's%';

- To display the names start with letter s and end with letter a

SELECT name FROM student WHERE name LIKE 's%a';

- To display the names whose second letter is a

SELECT name FROM student WHERE name LIKE '_a%';

- To display all 4 letter names

SELECT name FROM student WHERE name LIKE '_ _ _ _';

IN

This operator defines a set in which a given value may or may not be included. It defines a set by explicitly naming the members of the set in the parenthesis, separated by commas.

- To display the details of those students studying in either commerce or humanities

SELECT * FROM student WHERE course IN ('commerce', 'humanities');

- To display the details of those students studying in neither commerce nor humanities

SELECT * FROM student WHERE course NOT IN ('commerce', 'humanities');

ORDER BY

The order by clause is used to sort either in ascending or descending order. The order is to be specified using the keywords ASC (for ascending) or DESC (for descending) along with the column name. If the order is not specified, the arrangement will be in the ascending order.

Syntax**:**

SELECT ColumnName1,………FROM Tablename [WHERE condition] ORDER BY columnname ASC/DESC;

- To display the details of students in the alphabetic order of their name

SELECT * FROM student ORDER BY name;

- To display female students, toppers first in the list.

SELECT * FROM student WHERE sex='F' ORDER BY percent DESC;

Aggregate Functions

The SQL includes a number of built in functions to answer some more specific queries. It is also known as **Summary Functions**.

SUM( )	Total of the values
AVG( )	Average of the values
MIN( )	Smallest of the values
MAX( )	Largest of the values
COUNT( )	Number of values or rows

- To find the highest, lowest and average percent in the table student

SELECT MAX (percent), MIN (percent), AVG (percent) FROM student;

- To find the average percent of those students studying in science group

SELECT AVG (percent) FROM student WHERE course='science';

- To display the total number of students

SELECT COUNT (*) from student;

- To display the total number of students in science batch

SELECT COUNT (*) from Student where course='science';

GROUP BY

The rows of a table can be grouped together based on a common value using the group by clause.

- To display each course and number of students in each

SELECT course, COUNT (*) FROM student GROUP BY course;

- To display total mark of students in each course

SELECT course, SUM (total) from student GROUP BY course;

HAVING

The having clause is used to specify a condition with group by clause.

- To display the number of students in science group

SELECT course, COUNT (*) FROM student GROUP BY course HAVING course='science';

- To display total salary paid to managers

SELECT designation, SUM (salary) from employee GROUP BY designation having designation='manager';

4. ALTER TABLE

This command is used to change the structure of the table by

- Adding new columns
- Modifying the existing column in terms of type or size
- Remove the existing column

Adding new columns

Syntax:

```
ALTER TABLE tablename ADD (columnname datatype size [columnconstraint]);
```

Modifying the existing column

Syntax:

```
ALTER TABLE tablename MODIFY (columnname newdatatype newsize);
```

Delete the existing column

Syntax:

```
ALTER TABLE tablename DROP column columnname;
```

- To change the size of name field from 10 to 15

ALTER TABLE student MODIFY (name char(15));

- To add a new column to set the grade, which is two characters.

ALTER TABLE student ADD (grade char (2));

- Delete the column course.

Alter table student drop column course;

5. UPDATE

It is used to change the values in a column of specified rows. The new values are added by using the SET keyword.

Syntax:

```
UPDATE tablename SET  columnname=value,………[where condition];
```

- To change the percent of Martin Thomas from 65 to 70

UPDATE student SET percent=70 WHERE name='Martin Thomas';

- To increment 1 percentage to all science students

UPDATE student SET percent=percent+1 WHERE course='science';

- Change the name Rajiv to Rajeev

Update student set name='Rajeev' where name='Rajiv';

- Increase the salary of the employee 'Mahesh' by 1000 and change his designation to Clerk.

Update employee set salary = salary + 1000, designation = 'clerk' where name = 'Mahesh';

6. DELETE

It is used to delete rows in a table. If where clause is not used, all the rows in the table will be deleted.

Syntax:

```
DELETE from  tablename[where  condition];
```

- To delete all science students

DELETE FROM student WHERE course='science';

- To delete all employees

DELETE FROM employee;

- To delete all female students in commerce batch

DELETE FROM student WHERE course='commerce' AND sex='F';

7. DROP TABLE

This command is used to delete a table.

Syntax:

```
DROP TABLE tablename;
```

- To delete the table student

DROP TABLE student;

8. CREATE VIEW

This command is used to create a virtual table from an existing table.

Syntax:

```
CREATE VIEW  ViewName AS SELECT   Columnname1,……….FROM
tablename[WHERE  condition]:
```

- To create a view named studview with the details of science students

Create view studview as select * from student where course=’science’;

- To create an employee view with empid and empname

Create view empview as select empid,empname from employee;

9. DROP VIEW

This command is used to delete a view table.

Syntax:

```
DROP VIEW viewname;
```

- To delete the view named studview.

Ex: DROP VIEW studview;

SQL Commands	
DDL	**DML**
Create Table	Insert
Alter Table	Select
Drop Table	Update
Create View	Delete
Drop View	

Keys

A key is a field or an attribute of a table.There are five types of keys:

1)Candidate Key

It is an attribute or a set of attributes that uniquely identifies a table.

2)Primary Key

It is one of the candidate key that uniquely identifies a table.

3)Alternate Key

It is a candidate key that is not the primary key.

4)Super Key

It consists of a primary key and one or more attributes of a table.

5)Foreign Key

It is a primary key that is a candidate key of another table.

MySQL

MySQL is a relational database management system.It was originally developed in Sweden by David Axmark,Allan Larsson and Michael Widenius. The most popular open-source database in the world is MySQL. **MySQL** is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. **MySQL** was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL. MySQL can be used for a variety of applications, but is most commonly found on Web servers.

Characteristics of MySQL

1)MySQL is released under an open source license, so it is customizable.

2)It is very powerful and simple to setup and easy to use.

3)MySQL uses a standard form of the well known ANSI SQL standards.

4)MySQL is a platform independent application which works on many operating systems like Windows,UNIX,LINUX etc.

5)MySQL has compatability with many languages including Java,C++,PHP,PERL etc.

6)MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.

7)MySQL follows the working of a client/server architecture.

8)MySQL is free to use so that we can download it from MySQL official website without any cost.

9)MySQL supports multi-threading that makes it easily scalable. It can handle almost any amount of data, up to as much as 50 million rows or more.

10)MySQL is considered one of the very fast database languages, backed by a large number of the benchmark test.

11)MySQL allows transactions to be rolled back, commit, and crash recovery.

12)MySQL provides a unified visual database graphical user interface tool named "**MySQL Workbench**" to work with database architects, developers, and Database Administrators.
13)MySQL version 8.0 provides support for dual passwords: one is the current password, and another is a secondary password, which allows us to transition to the new password.

Disadvantages/Drawbacks of MySQL

- MySQL version less than 5.0 doesn't support ROLE, COMMIT, and stored procedure.
- MySQL does not support a very large database size as efficiently.
- MySQL doesn't handle transactions very efficiently, and it is prone to data corruption.
- MySQL is accused that it doesn't have a good developing and debugging tool compared to paid databases.
- MySQL doesn't support SQL check constraints.

PHP

PHP is a general-purpose scripting language especially suited to web development.It is used for creating dynamic webpages.PHP stands for **PHP: Hypertext Preprocessor**.The initial name of PHP was Personnel Homepage Tools. It is also an interpreted language.It is an opensource software and was created by Rasmus Lerdorf in 1994.It is more robust and error free than many other server side scripting language and can access more than 20 different databases including MySQL.PHP can also be used to output images ,pdf files and flash movies.Today many large scale websites are developed in PHP.

Features

- As PHP can be easily embedded into html,which is easy to convert and already existing static website into dynamic one.
- It is compatable with a wide variety of databases.
- It is an independent platform and can run across Windows,Linux or Mac servers.
- It supports most of the webservers including IIS(Internet Information Server) , but most often used is freely available Apache server.
- It supports most of the database softwares including Oracle,but most often used is freely available MySQL databse.
- It is simple and easy to learn.
- It operates much faster than other scripting languages.
- PHP offers many levels of security to prevent malicious attacks.

- It has a huge support from open source community.
- It is completely free and can legally be used for personnel and commercial website development.
- The syntax of PHP is almost same as C.
- It can handle for accessing cookies and encrypt data.
- PHP is a case sensitive language.

Common uses of PHP

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.
- It can be used to **create Web applications** like Social Networks, Blogs, eCommerce websites etc.
- ## PHP Programming steps

1) Setting up the development environment

To setup a web development environment for a php program to run successfully.For this,we need to install separately or from a set of free open source software package like WAMP,LAMP,XAMPP which includes Apache web server,MySQL and PHP.
WAMP-Windows Apache MySQL PHP
LAMP- Linux Apache MySQL PHP
XAMPP-X-OS Apache MySQL PHP PERL

2) Writing and running the script

A simple text editor is used for writing php scripts.The php documents are to be saved with .php extensions. The scripts should be written as,

```
<?php
Scripts
?>
```

-OR-

```
<?
Scripts
?>
```

-OR-

```
<script language="php">
Scripts
</script>
```

3)Combining html and php

We can embed the php code into standard html documents using the special tags <?php and ?>

```
<html>
<head>
<title>A program</title>
</head>
<body>
<?php
echo  "Hello";
?>
</body>
</html>
```

Comments in PHP

In php code, a comment is a line that is not read or execute as part of the program.The two types of comments are:

1)Single line comment

```
//comment text
```

2)Multiline comment

```
/* comment

Text

*/
```

Output statements

The two basic statements are used to display output on a webpage.They are echo and print.

Ex: echo(“Welcome”);

Print(“Welcome”);

Echo	Print
Can take more than one parameter when used without parenthesis	Takes only one parameter
Does not return any value	It returns true on one on successful output and false if it was unable to print out the string
Little faster than print	Little slower than echo

Variables

Variables are storage locations for holding data.In php, a variable name starts with the $ sign followed by the name of the variable.Unlike other programming languages, a variable does not need to be declared before using it.Php automatically determines the datatype of the variable,depending on the type of value assigned to it.

Syntax:

$ variablename=value;

Variable naming rules

- A variable name must start with the $ sign followed by the name of the variable.
- The name of the variable must start with a letter or the underscore character.
- The name of the variable can only contain alpha numeric characters and underscore.
- Variable names are case sensitive.
- Keywords are not allowed.
- Spaces and special symbols are not allowed.

Variable Scope

Scope can be defined as the range of availability a variable has to the program in which it is declared. PHP variables can be one of three scope types −

- Local variables:A variable declared in a function is called Local variables.
- Global variables:A global variable can be accessed in any path of the program.
- Static variables:A static variable will not loose its value when the function exit and will still hold that value should the function be called again.

Constant

A constant is a name or an identifier for a simple value. A constant value cannot change during the execution of the script. 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.

To define a constant you have to use define() function and to retrieve the value of a constant, you have to simply specifying its name. You can also use the function constant() to read a constant's value if you wish to obtain the constant's name dynamically.

constant() function: As indicated by the name, this function will return the value of the constant.This is useful when you want to retrieve value of a constant, but you do not know its name, i.e. It is stored in a variable or returned by a function.

Data types of PHP

The data types in php can be divided into two:

1) Core data types

It contains integer,float,Boolean and string

2) Special data types

It contains null,arrays,objects and resources

- **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.Null data type can also be use it for emptying variables.
- **Strings** − are sequences of characters, like 'PHP supports string operations.'
- **Arrays** − are named and indexed collections of other values.
- **Objects** – An object can contain variables and functions that process a task.
- **Resources** − are special variables that hold references to resources external to PHP (such as database connections).

PHP Operators

Operators are used to perform operations on variables and values.PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators

- String operators

Arithmetic Operators

The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
**	Exponentiation

Assignment Operators

The PHP assignment operators are used with numeric values to write a value to a variable.The basic assignment operator in PHP is "=".

Operator	Name
=	Assignment
+=	Addition assignment
-=	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%=	Modulus assignment

Comparison Operators(Relational Operators)

The PHP comparison operators are used to compare two values (number or string):

Operator	Name
==	Equal to
!= or <>	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than

<=	Less than or equal to

Increment / Decrement Operators

The PHP increment operators are used to increment a variable's value.The PHP decrement operators are used to decrement a variable's value.

Operator	**Name**
++$x	Pre increment
$x++	Post increment
--$x	Pre decrement
$x--	Post decrement

Logical Operators

The PHP logical operators are used to combine conditional statements.

Operator	**Name**
And or &&	Logical AND
Or or ||	Logical OR
Not	Logical NOT

String Operators

PHP has two operators that are specially designed for strings.

Operator	**Name**
.	Concatenation
.=	Concatenation assignment

Concatenation Operator (.)

It joints the right and left string into a single string.

```
Ex: $string1='Good';
$string2='Morning';
$new=$string1.string2;
echo $new;
            o/p-Good Morning
```

Concatenation assignment operator (.=)

It adds the value placed on the rightside on the equal operator with the value placed on the left side of the equal operator.

```
Ex: $string1='Good';
$string2='Morning';
$string1.=$string2;
echo $string1;
        o/p-Good Morning
```

Escape Sequences in PHP

It is used for giving special meaning to represent line breaks, tabs, alert and more. Escape sequences are started with the escaping character backslash (\) followed by the character which may be an alphanumeric or a special character.

widely used Escape Sequences in PHP

1. \' – To **escape** ' within single quoted **string**.
2. \" – To **escape** " within double quoted **string**.
3. \\ – To **escape** the backslash.
4. \$ – To **escape** $.
5. \n – To add line breaks
6. \t – To add tab space.
7. \r – For carriage return.

PHP Programs

1) Display Hello

```
<?php
echo "Hello";
?>
```

2) Find the sum of two numbers

```
<?php
$a=10;
$b=20;
$c=$a+$b;
echo "Sum: ",$c;
?>
```

3)Find the area of a triangle

```
<?php
$b = 2;
$h = 5;
$area=0.5*$b*$h;
echo "area is",$area;
?>
```

Control statements(Flow control statements)

Control statements are conditional statements that execute a block of statements if the condition is correct. The statement inside the conditional block will not execute until the condition is satisfied. PHP supports a number of different control structures:

1) if

The if statement checks the truthfulness of an expression and, if the expression is true, evaluates a statement.

Syntax:

```
if (expression)
statement
```

Qn: Find the largest of two numbers

```
<?php
$a=10;
$b=5;
if($a>$b)
echo “Big is”,$a;
?>
```

2)If else

Syntax:

```
if (expression)
statement
else
statement
```

Qn: Find the largest of two numbers

```
<?php
$a=10;
$b=5;
if($a>$b)
echo “Big is”,$a;
else
echo “Big is”,$b;
?>
```

Qn: To check a number is odd or even

```
<?php
$n=5;
if($n % 2==0)
echo “It is an even number”;
else
echo “It is an odd number”;
?>
```

3) Else if ladder

Syntax:

```
if (expression)
statement
else if(expression)
statement
:
:
```

```
:
else
statement
```

Qn: Find the largest of two numbers

```
<?php
$a=10;
$b=5;
if($a>$b)
echo “Big is”,$a;
else if($a<$b)
echo “Big is”,$b;
else
echo “Both are equal”;
?>
```

Qn: To check a number is +ve,-ve or zero

```
<?php
$n=5;
if($n>0)
echo “Positive”;
else if($n<0)
echo “Negative”;
else
echo “Zero”;
?>
```

4) switch

The switch statement is somewhat similar to the `else if ladder` statement.

Syntax:

```
switch(expression)
{
case value1:
statement;
break;
case value2:
statement;
break;
:
:
:
default:
statement;
}
```

Qn: To display the dayname of a week

```
<?php
$day=5;
switch($day)
{
case 1:
```

```
echo “Sunday”;
break;
case 2:
echo “Monday”;
break;
:
:
default:
echo “Invalid”;
}
?>
```

Looping statements in PHP

Loops in PHP are used to execute the same block of code a specified number of times. PHP supports following four loop types.

- **for** − loops through a block of code a specified number of times.
- **while** − loops through a block of code if and as long as a specified condition is true.
- **do...while** − loops through a block of code once, and then repeats the loop as long as a special condition is true.
- **foreach** − loops through a block of code for each element in an array.

for

Syntax:

```
for (initialization; test condition; update expression)
{
body of the loop;
}
```

Ex: Display first 10 natural numbers

```
<?php
for($i=1;$i<=10;$i++)
{
echo “$i”;
}
?>
```

while

Syntax:

```
initialization;
while(test condition)
{
body of the loop;
update expression;
}
```

Ex: Display first 10 natural numbers

```
<?php
$i=1;
while($i<=10)
```

```
{
echo “$i”;
$i++;
}
?>
```

do while

Syntax:

```
initialization;
do
{
body of the loop;
update expression;
}while(test condition);
```

Ex: Display first 10 natural numbers

```
<?php
$i=1;
do
{
echo “$i”;
$i++;
}while($i<=10);
?>
```

foreach

Syntax:

```
for each (array as value)
{
  code to be executed;
}
```

Ex:

```
<?php
 $new= array (1, 2, 3, 4, 5);
 foreach($new as $value)
 {
 echo "Value is $value";
 }
?>
```

O/P

```
Value is 1
Value is 2
Value is 3
Value is 4
Value is 5
```

Array in PHP

An array is a special variable, which can hold more than one value at a time.It is a data structure that stores one or more similar type of values in a single value. The arrays are helpful to create a list of elements of similar types, which can be accessed using their index or key.The main advantages of PHP arrays are less code,easy to traverse and sorting.

In PHP, the `array()` function is used to create an array.It creates and returns an array. It allows to create indexed, associative and multidimensional arrays.

In PHP, there are three types of arrays:

- **Indexed/Numeric arrays** - An array with a numeric index. Values are stored and accessed in linear fashion. By default,array index starts with '0'.

There are two ways to define indexed array:

Syntax1:
$arrayname=array(value1,value2,value3,……);
Ex: $mark=array(65,73,56);

Syntax2:
$arrayname[key]=value;
Ex: $mark[0]=65;
$mark[1]=73;
$mark[2]=56;

- **Associative arrays** – Arrays with named keys are called associative arrays. This stores element values in association with key values rather than in a strict linear index order. The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index.

There are two ways to define associative array:

Syntax1:
$arrayname=array(key1=>value1,key2=value2,……);
Ex: $salary=array("Anoop"=>"35000","John"=>"45000","Kartik"=>"20000");
Syntax2:
$arrayname[key]=value;
Ex: salary["Anoop"]="35000";
$salary["John"]="45000";
$salary["Kartik"]="20000";

- **Multidimensional arrays** - An array containing one or more arrays and values are accessed using multiple indices. A multi-dimensional 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-dimensional array are accessed using multiple index.

PHP array functions

PHP provides various array functions to access and manipulate the elements of array. The important PHP array functions are given below.

array( )

PHP array() function creates and returns an array. It allows you to create indexed, associative and multidimensional arrays.

Syntax:

```
$arrayname= array (values);
```

array_change_key_case()

Changes all keys in an array to lowercase or uppercase

Syntax:

```
array_change_key_case ( array $array [, int $case = CASE_LOWER ] )
```

array_chunk()

PHP array_chunk() function splits array into chunks. By using array_chunk() method, you can divide array into many parts.

Syntax:

```
array_chunk (array $array , int $size )
```

count()

PHP count() function counts all elements in an array.

Syntax:

```
count ( mixed $array_or_countable [, int $mode = COUNT_NORMAL ] )
```

sort()

PHP sort() function sorts all the elements in an array.

Syntax:

```
sort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

array_reverse()

PHP array_reverse() function returns an array containing elements in reversed order.

Syntax:

```
array_reverse ( array $array)
```

array_search()

PHP array_search() function searches the specified value in an array. It returns key if search is successful.

Syntax:

```
array_search ($needle , array $arrayname)
```

array_intersect

PHP array_intersect() function returns the intersection of two array. In other words, it returns the matching elements of two arrays.

Syntax:

```
array array_intersect ( array $array1 , array $array2 [, array $... ] )
```

PHP Functions

PHP functions are similar to other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value. PHP function is a piece of code that can be reused many times. It can take input as argument list and return value.

Advantages of PHP Functions

Code Reusability: PHP functions are defined only once and can be invoked many times, like in other programming languages.

Less Code: It saves a lot of code because you don't need to write the logic many times. By the use of function, you can write the logic only once and reuse it.

Easy to understand: PHP functions separate the programming logic. So it is easier to understand the flow of the application because every logic is divided in the form of functions.

Types of PHP Functions

1)Built-in Functions

Built in functions are predefined **functions** in **PHP** that exist in the installation package. These PHP inbuilt functions are what make PHP a very efficient and productive scripting language. The built in functions of PHP can be classified into many categories.PHP has over 1000 built-in functions that can be called directly, from within a script, to perform a specific task.

These functions can be subdivided into multiple categories as stated below.

String functions
Numeric functions
Date and time functions
Array functions
Directory functions

String Function

strtolower(); -> converts all characters of the string to lower case

strtoupper(); -> converts all characters of the string to upper case

ucfirst(); -> converts first letter to upper case

ucwords(); -> converts first letter of each word to upper case

strlen(); -> counts number of characters in a string and returns integer value

trim(); -> trims unnecessary spaces

ltrim(); -> trims unnecessary spaces from left

rtrim(); -> trims unnecessary spaces from right

substr(int,int) -> prints a string from defined initial character number to defined last number

strpos() -> finds position of the string

Numeric Function

abs() -> returns positive value of a number

sqrt() -> returns square root of a number

round() -> rounds a floating number

rand() -> generates a random integer

pow() -> returns x raised to the power of y

min() -> returns the lowest value from an array

max() -> returns the highest value from an array

bindec() -> converts a binary number to a decimal number

decbin() -> converts a decimal number to a binary number

Date/Time Functions

date_diff()	Returns the difference between two dates
date()	Formats a local date and time
time()	Returns the current time

Array Functions

array() -> creates an array

sizeof() -> counts the number of values in an array

sort() -> sorts an indexed array in ascending order

arsort() -> sorts an associative array in descending order according to the value

Directory Functions

getcwd() -> get current working directory

mkdir() -> make directory

rmdir() -> remove directory

dirname() -> directory name

2) User Defined Functions

Besides the built-in PHP functions, it is possible to create your own functions.

- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute automatically when a page loads.
- A function will be executed by a call to the function.

Function Declaration/Definition

A user-defined function declaration/Definition starts with the word `function`:

Syntax

```
function functionName()
{
code to be executed;
}
```

Function Calling

Syntax:

Functionname();

Example

```
<?php
function writeMsg()
{
 echo "Hello world!";
}
writeMsg(); // call the function
?>
```

Function Arguments

Information can be passed to functions through arguments. An argument is just like a variable.Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

Parameter passing to Functions

PHP allows us two ways in which an argument can be passed into a function:

- **Pass/Call by Value:** On passing arguments using pass by value, the value of the argument gets changed within a function, but the original value outside the function remains unchanged. That means a duplicate of the original value is passed as an argument.
- **Pass/Call by Reference:** On passing arguments as pass by reference, the original value is passed. Therefore, the original value gets altered. In pass by reference we actually pass the address of the value, where it is stored using ampersand sign(&).

Function with arguments

Informations can be passed to functions through arguments. Arguments are specified after the function name inside the parenthesis.

Ex: Sum of two numbers

```
<?php
function add($a,$b)
```

```
{
$c=$a+$b;
return $c;
}
$c=add(5,10);
echo $c;
?>
```

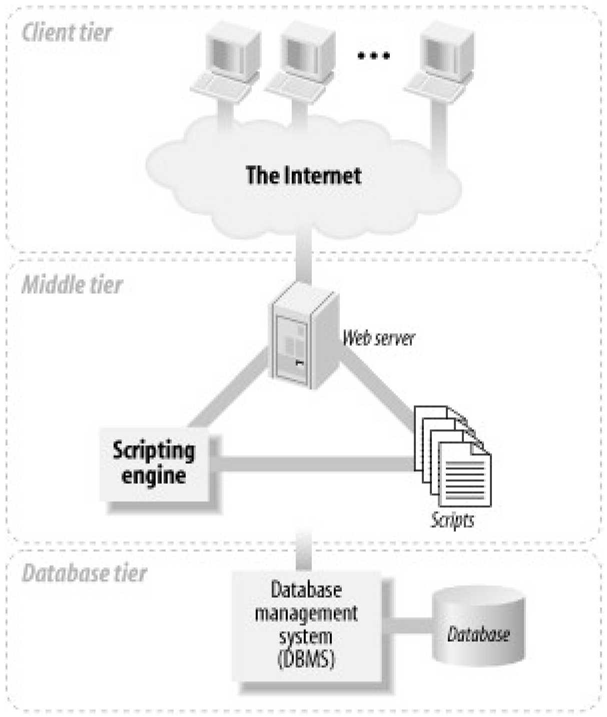
3-Tier Architecture

3-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

Three-tier architecture has the following three tiers:

- **Presentation Tier:** This is the topmost level of the application. It is the frontend in which the content is entered by the browser. The content may be html,php,javascript etc. The Presentation Tier shows information related to such services as Order Entry and Accounting applications.
- **Application or Logical Tier:** It is the middle portion of the architecture in which the content processing, accept the request, produce the result etc by using a webserver like Apache. The Application Tier (middle tier) controls the business logic and processes based on Service calls from the Presentation Tier (User Interface).
- **Database Access Tier:It is the back end in which the dbms software manages and provide access to the data. It executes query by using SQL command.** This tier consists of database/data storage system and data access layer. A **Data Layer** which is a database management system that provides access to application data.

The three-tier architecture model of a web database application



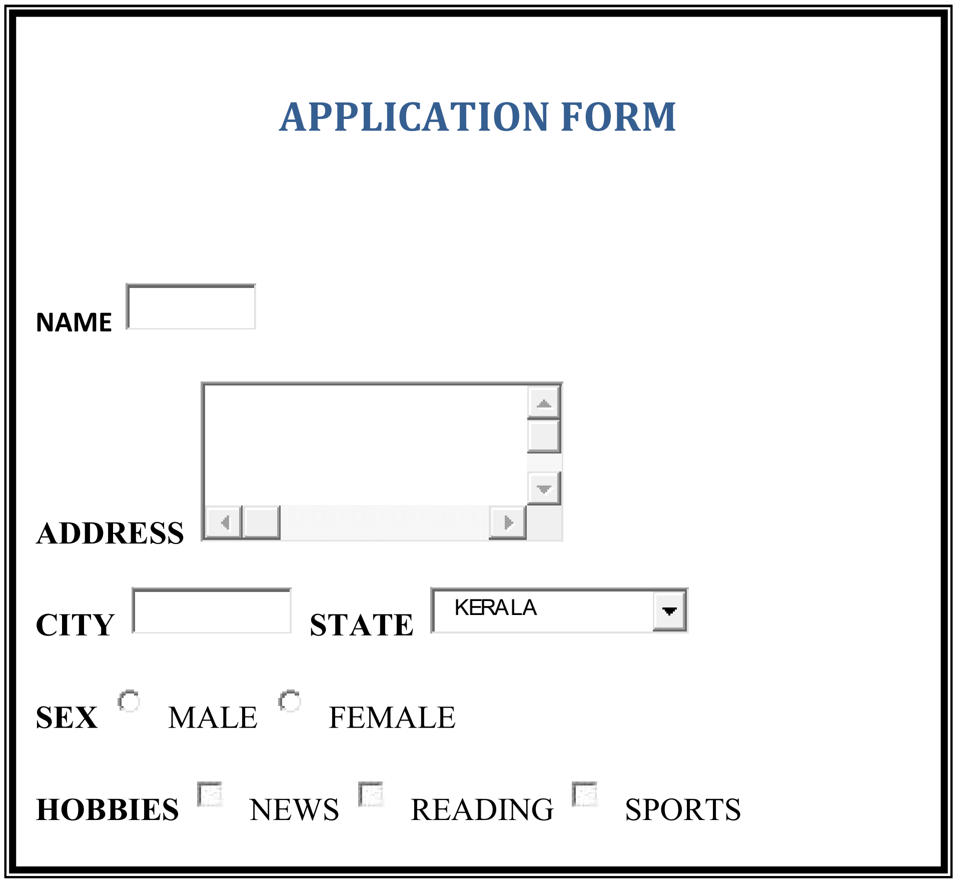
PHP Forms

In php,forms are used to collect information from the client and pass it to the server.Php is capable of accepting and manipulating input from all the form elements like textbox,combobox button etc.

There are two ways the browser or client can send information to the web server.They are Get and POST

Get	**Post**
Data is appended to the program	Data is stored separately and is forwarded
Used to pass less volumes of data	Used to pass large volumes of data
It is a faster method	It is a slower method
It is no secure as the data is visible during submission	It is secure as the data is not visible during submission
Page link can be bookmarked	Page link can never be bookmarked
It limits 2000 characters	There is no restriction
May be used for sending non sensitive data	Used for sending sensitive data

Script for creating a form (file upload form)


APPLICATION FORM

NAME

ADDRESS

CITY STATE KERALA

SEX MALE FEMALE

HOBBIES NEWS READING SPORTS

```
<HTML>
<BODY>
<H1 ALIGN=CENTER>APPLICATION FORM</H1>
<FORM action="upload.php" method="post">
<LABEL>NAME</LABEL>&NBSP;
<INPUT TYPE=TEXT><P>
<LABEL>ADDRESS</LABEL>&NBSP;
<TEXTAREA>
</TEXTAREA><P>
<LABEL>CITY</LABEL>&NBSP;
<INPUT TYPE=TEXT>&NBSP;
<LABEL>STATE</LABEL>&NBSP;
<SELECT>
<OPTION>KERALA
<OPTION>TAMIL NADU
```

```
<OPTION>KARNATAKA
<OPTION>ANDHRA PRADESH
</SELECT><P>
<LABEL>SEX</LABEL>&NBSP;
<INPUT TYPE=RADIO NAME=SEX>MALE&NBSP;
<INPUT TYPE=RADIO NAME=SEX>FEMALE<P>
<LABEL>HOBBIES</LABEL>&NBSP;
<INPUT TYPE=CHECKBOX>NEWS&NBSP;
<INPUT TYPE=CHECKBOX >READING&NBSP;
<INPUT TYPE=CHECKBOX>SPORTS<P>
<INPUT TYPE=SUBMIT VALUE=SUBMIT>&NBSP;
<INPUT TYPE=RESET VALUE=RESET>
</FORM>
</BODY>
</HTML>
```

Send an Email on Form Submission Using PHP

The mail() function allows you to send emails directly from a script. PHP mail is the built in PHP function that is used to send emails from PHP scripts.

Syntax

mail(*to,subject,message,headers,parameters*);

1. $to email address is the email address of the mail recipient.
2. $subject is the email subject.
3. $message is the message to be sent.
4. $headers contain other **email** parameters like BCc, Cc etc.
    - CC is the acronym for carbon copy. It's used when you want to send a copy to an interested person.
    - BCC is the acronym for blind carbon copy.The email addresses included in the BCC section will not be shown to the other recipients.

Ex:

```
<?php
$to                =                "somebody@example.com";
$subject               =              "My              subject";
$txt              =               "Hello              world!";
```

```
$headers = "From: webmaster@example.com" . "\r\n" .
"CC: somebodyelse@example.com";
mail($to,$subject,$txt,$headers);
?>
```

PHP Errors

PHP Error occurs when something is wrong in the PHP code. The error can be as simple as a missing semicolon, or as complex as calling an incorrect variable.

The four types of PHP errors are:

- Warning **Error**.
- Notice **Error**.
- Parse **Error**.
- Fatal **Error**.

1)Warning Error

The main reason of warning errors are including a missing file. This means that the PHP function call the missing file.A **warning error in PHP** does not stop the script from running. It only warns you that there is a problem, one that is likely to cause bigger issues in the future.

The most common causes of warning errors are:

- Calling on an external file that does not exist in the directory
- Wrong parameters in a function

2)Notice Error

It is similar to warning error. It means that the program contains something wrong but it allows the execution of script.**Notice errors** are minor errors. Often, the system is uncertain whether it's an actual error or regular code. Notice errors usually occur if the script needs access to an undefined variable.

3)Parse Error (Syntax)

It is the type of error done by the programmer in the source code of the program. The syntax error is caught by the compiler. After fixing the syntax error the compiler compile the code and execute it. Parse errors can be caused dues to unclosed quotes, missing or Extra parentheses, Unclosed braces, Missing semicolon etc. The compiler catches the error and terminates the script.

Parse errors are caused by:

- Unclosed brackets or quotes
- Missing or extra semicolons or parentheses
- Misspellings

4)Fatal Error

It is the type of error where PHP compiler understands the PHP code but it recognizes an undefined function. This means that function is called without the definition of function. **Fatal errors** are ones that crash your program and are classified as critical errors. There are three (3) types of fatal errors:

1. **Startup fatal error** (when the system can't run the code at installation)
2. **Compile time fatal error** (when a programmer tries to use nonexistent data)
3. **Runtime fatal error** (happens while the program is running, causing the code to stop working completely)

PHP error constants

- **E_ERROR :** A fatal error that causes script termination
- **E_WARNING :** Run-time warning that does not cause script termination
- **E_PARSE :** Compile time parse error.
- **E_NOTICE :** Run time notice caused due to error in code
- **E_CORE_ERROR :** Fatal errors that occur during PHP's initial startup (installation)
- **E_CORE_WARNING :** Warnings that occur during PHP's initial startup
- **E_COMPILE_ERROR :** Fatal compile-time errors indication problem with script.
- **E_USER_ERROR :** User-generated error message.
- **E_USER_WARNING :** User-generated warning message.
- **E_USER_NOTICE :** User-generated notice message.
- **E_STRICT :** Run-time notices.
- **E_RECOVERABLE_ERROR :** Catchable fatal error indicating a dangerous error
- **E_DEPRECATED :** Run-time notices.

PHP Global Variables – Superglobal Arrays

Global variables are variables that are always accessible from anywhere in the code. We can access it from any function, class or file. Usually global variables are declared in the initial part of a script and outside a function.Some predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

The PHP superglobal variables are:

Superglobal	**Description**
$GLOBALS	Variables available in the global scope.PHP stores all global variables in an array called $GLOBALS[index]
$_SERVER	Information about the server such as headers,paths and script locations.
$_GET	Data passed using the HTTP GET Method
$_POST	Data passed using the HTTP POST Method
$_REQUEST	Data passed via an HTTP request. It contains the contents of the $_GET, $_POST and

	$_COOKIE superglobals. It is used to collect data after submitting an HTML form.
$_FILES	Data passed by an HTML file input
$_SESSION	Current session data specific to the user
$_COOKIE	Data stored on the user's browser as a cookie

Connecting PHP to MySQL Database

For a PHP program to access data from the database, the program should first establish a connection to the database server such as MySQL and then specify the database to use. After this, PHP program be able to issue SQL queries to the database.

Connection to a MySQL database is done in four steps:

1) **Open a connection to MySQL**

The function mysql_connect( ) is used to establish a connection between PHP and MySQL Database server. It takes three string arguments, the host/server name, the username and the password. The function returns a link identifier when it successfully connects to the specified MySQL server or NULL up on error.

$link_id=mysql_connect("local host","userid","password");

2) **Specify the database you want to open**

The MySQL database server may contain a lot of databases. mysql_select_db( ) function is used to select a particular MySQL database from them. It takes a string which is the database name and a link identifier as argument. If it finds the database, it returns true, or else false is returned.

$link_id=mysql_select_db($database);

3) **Reading data from database**

Reading data from the database is done in two steps:

a. Execute the SQL query on the database using the function mysql_query( ).It sends the query to the database and returns a resource type query handle on successful execution.

   $result_set=mysql_query(query,connection);

b. Populate the rows in to an array using the function mysql_fetch_array. It takes a result handle returned from mysql_query( ),and returns an array corresponding to the fetched row.

   $fetched_row=mysql_fetch_array($result_set);

4) **Close the Connection**

   The function mysql_close( ) is used to close a connection to the database server. The link to the MySQL server is automatically closed when the PHP script is terminated.

   mysql_close($db_handle);

mysql_connect( ) and mysql_pconnect( )

Mysql_connect() opens a new connection to the database while mysql_pconnect() opens a persistent connection to the database. These are used to include a file to php page.

mysql_connect

This function will create a new connection to the database once your script starts executing and closes the connection to the database once script execution ends. This will make a connection to the database every time our scripts start the execution.

mysql_pconnect

When you try to connect your database using this function, then it will search for existing connection made to the database using same username and password, if the connection found then it will return the resource ID, otherwise, it will make the connection and return the resource ID.

require( ) and include( )

The include (or require) statement takes all the text/code/markup that exists in the specified file and copies it into the file that uses the include statement.

The **include** and **require** statements are identical, 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.The only **difference** is — the **include()** statement will only generate a **PHP** warning but allow script execution to continue if the file to be **included** can't be found, whereas the **require()** statement will generate a fatal error and stops the script execution.

Syntax of the include() and require()

include("path/to/filename"); -Or- include "path/to/filename";
require("path/to/filename"); -Or- require "path/to/filename";

Interfaces in PHP

An *interface* is a structure which defines a list of methods that must exist in a class. *Interfaces* are a good way to allow many different classes to be used in the same way. The implements keyword can be used to make a class use an *interface*

Sessions in PHP

A session creates a file in a temporary directory on the server where registered session variables and their values are stored. This data will be available to all pages on the site during that visit..Unlike a cookie, the information is not stored on the user's computer. **PHP session** creates unique user id for each browser to recognize the user and avoid conflict between multiple browsers.

Start a PHP Session & set session variables

A session is started with the **session_start()** function. Session variables are set with the PHP global variable: **$_SESSION**.

Ex: Create a new page called "demo_session1.php". In this page, we start a new PHP session and set some session variables

```
<html>
<body>
<?php
// Start the session
session_start();
?>
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
</body>
</html>
```

Destroy a PHP Session

To remove all global session variables by using **session_unset()** and destroy the session by using **`session_destroy()`**

Ex:

```
<html>
<body>
<?php
session_start();
?>
<?php
// remove all session variables
session_unset();
// destroy the session
session_destroy();
?>
</body>
</html>
```

Turning on Auto Session

To work with a session, we need to explicitly start or resume that session. To start a session, you must find the following line in your php.ini file and change the value from 0 to 1.

session.auto_start=0;

You don't need to call start_session() function to start a session when a user visits your site if you can set **session.auto_start** variable to 1 in **php.ini** file.

PHP Cookies

A cookie is a small file with the maximum size of 4KB that the web server stores on the client computer.They are typically used to keeping track of information such as a username that the site can retrieve to personalize the page when the user visits the website next time.A cookie is often used to identify a user. PHP transparently supports HTTP cookies.With PHP, you can both create and retrieve cookie values.

Cookies are often used to perform following tasks:

- **Session management**: Cookies are widely used to manage user sessions. For example, when you use an online shopping cart, you keep adding items in the cart and finally when you checkout, all of those items are added to the list of items you have purchased. This can be achieved using cookies.
- **User identification**: Once a user visits a webpage, using cookies, that user can be remembered. And later on, depending upon the search/visit pattern of the user, content which the user likely to be visited are served.
- **Tracking / Analytics**: Cookies are used to track the user. Which, in turn, is used to analyze and serve various kind of data of great value, like location, technologies (e.g. browser, OS) form where the user visited, how long (s)he stayed on various pages etc.

Create/Set Cookies

PHP provided **setcookie()** function to set a cookie. This function requires upto six arguments and should be called before <html> tag. For each cookie this function has to be called separately.

setcookie(name, value, expire, path, domain, security);

1. **Name:** It is used to set the name of the cookie.
2. **Value:** It is used to set the value of the cookie.
3. **Expire:** It is used to set the expiry timestamp of the cookie after which the cookie can't be accessed.

4. **Path:** It is used to specify the path on the server for which the cookie will be available.
5. **Domain:** It is used to specify the domain for which the cookie is available.
6. **Security:** It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists.

Ex: Create two cookies **name** and **age** these cookies will be expired after one hour.

```
<?php
  setcookie("name", "John Watkin", time()+3600, "/","", 0);
  setcookie("age", "36", time()+3600, "/", "",  0);
?>
<html>
     <head>
     <title>Setting Cookies with PHP</title>
  </head>
     <body>
     <?php echo "Set Cookies"?>
  </body>

</html>
```

Accessing Cookies

PHP provides many ways to access cookies. Simplest way is to use either **$_COOKIE** or **$HTTP_COOKIE_VARS** variables. You can use **isset()** function to check if a cookie is set or not.

Delete Cookies

The **setcookie()** function can be used to delete a cookie. If you set the expiration date in past, cookie will be deleted.

Different ways to embed a script in html file

- Placing scripts inside the <Body> tag
- Include scripts in the head section of the web page
- Place the scripts into an external file and then link to that file from within the html document

settype()

The settype() function converts a variable to a specific type.

Syntax

settype(*variable*, *type*);

Ex:

```
<?php
$a = "32"; // string
settype($a, "integer"); // $a is now integer
?>
```

O/P- 32

gettype()

The gettype() function returns the type of a variable.

Syntax

gettype(*variable*);

Ex:

```
<?php
$a = 3;
echo gettype($a) ;
?>
```

O/P-integer

Privileges/permissions for users in Mysql

- ALL - Allow complete access to a specific database. If a database is not specified, then allow complete access to the entirety of MySQL.
- CREATE - Allow a user to create databases and tables.
- DELETE - Allow a user to delete rows from a table.
- DROP - Allow a user to drop databases and tables.
- EXECUTE - Allow a user to execute stored routines.
- GRANT OPTION - Allow a user to grant or remove another user's privileges.
- INSERT - Allow a user to insert rows from a table.
- SELECT - Allow a user to select data from a database.
- SHOW DATABASES- Allow a user to view a list of all databases.
- UPDATE - Allow a user to update rows in a table.

MySQL String Functions

Function	Description
ASCII	Returns the ASCII value for the specific character
CHAR_LENGTH	Returns the length of a string (in characters)
CONCAT	Adds two or more expressions together
FORMAT	Formats a number to a format like "#,###,###.##", rounded to a specified number of decimal places
INSTR	Returns the position of the first occurrence of a string in another string
LCASE	Converts a string to lower-case
LEFT	Extracts a number of characters from a string (starting from left)
LENGTH	Returns the length of a string (in bytes)
LOCATE	Returns the position of the first occurrence of a substring in a string
LOWER	Converts a string to lower-case
LTRIM	Removes leading spaces from a string
MID	Extracts a substring from a string (starting at any position)
REPLACE	Replaces all occurrences of a substring within a string, with a new substring

REVERSE	Reverses a string and returns the result
RIGHT	Extracts a number of characters from a string (starting from right)
RTRIM	Removes trailing spaces from a string
STRCMP	Compares two strings
SUBSTR	Extracts a substring from a string (starting at any position)
TRIM	Removes leading and trailing spaces from a string
UCASE	Converts a string to upper-case
UPPER	Converts a string to upper-case

MySQL Numeric Functions

Function	**Description**
ABS	Returns the absolute value of a number
AVG	Returns the average value of an expression
COUNT	Returns the number of records returned by a select query
EXP	Returns raised to the power of a specified number
LEAST	Returns the smallest value of the list of arguments
MAX	Returns the maximum value in a set of values
MIN	Returns the minimum value in a set of values
MOD	Returns the remainder of a number divided by another number
POW	Returns the value of a number raised to the power of another number
RAND	Returns a random number
ROUND	Rounds a number to a specified number of decimal places
SIGN	Returns the sign of a number
SQRT	Returns the square root of a number
SUM	Calculates the sum of a set of values
TRUNCATE	Truncates a number to the specified number of decimal places

MySQL Date Functions

Function	**Description**
ADDTIME	Adds a time interval to a time/datetime and then returns the time
CURDATE	Returns the current date
CURTIME	Returns the current time
DATEDIFF	Returns the number of days between two date values
DATE_ADD	Adds a time/date interval to a date and then returns the date
DAY	Returns the day of the month for a given date

DAYNAME	Returns the weekday name for a given date
DAYOFMONTH	Returns the day of the month for a given date
DAYOFWEEK	Returns the weekday index for a given date
DAYOFYEAR	Returns the day of the year for a given date
HOUR	Returns the hour part for a given date
LOCALTIME	Returns the current date and time
MINUTE	Returns the minute part of a time/datetime
MONTH	Returns the month part for a given date
MONTHNAME	Returns the name of the month for a given date
NOW	Returns the current date and time
SECOND	Returns the seconds part of a time/datetime
SYSDATE	Returns the current date and time
TIME	Extracts the time part from a given time/datetime
WEEK	Returns the week number for a given date
WEEKDAY	Returns the weekday number for a given date
WEEKOFYEAR	Returns the week number for a given date
YEAR	Returns the year part for a given date
YEARWEEK	Returns the year and week number for a given date

MySQL Advanced Functions

Function	Description
BIN	Returns a binary representation of a number
CONV	Converts a number from one numeric base system to another
DATABASE	Returns the name of the current database
IF	Returns a value if a condition is TRUE, or another value if a condition is FALSE
USER	Returns the current MySQL user name and host name
VERSION	Returns the current version of the MySQL database