

Knowledge flow

Learning PHP and MySQL





“LEARNING STARTS WITH VIEWING THE WORLD DIFFERENTLY.”

Knowledge flow- A mobile learning platform provides apps, eBooks and video tutorials.

Knowledge flow brings you a learning eBook of ***Learning PHP and MySQL***. This eBook is for all information technology, computer science and web development students and professionals across the world.

Follow us on

[Facebook](#)

[Google plus](#)

[Twitter](#)

For more information visit us at

[Knowledgeflow.in](#)

[knowledgeflowapps.blogspot.in](#)

Thank you for using Knowledge flow eBooks



LEARNING PHP AND MYSQL

[1. Introduction of PHP](#)

[2. Features of PHP](#)

[3. Variables and Arrays](#)

[4. Operators](#)

[5. Functions](#)

[6. Introduction of SQL](#)

[7. Operators and Conditions of SQL](#)

[8. Queries, Sub-queries and 3VL](#)

[9. Data Control and Drawbacks](#)

[10. PHP with MySQL](#)

[11. Create and Insert in MySQL using PHP](#)

[12. Update and Retrieve in MySQL using PHP](#)

[13. Delete in MySQL using PHP](#)

[14. More eBooks and Apps](#)



Disclaimer

This eBook contents is for informational and study purposes only. The **Knowledge flow** makes no claims, promises, or guarantees about the accuracy, completeness, or adequacy of the contents of this eBook and no legal liability or other responsibility is accepted by **Knowledge flow** for any errors, omissions, or statements on this eBook.



Introduction

PHP is originally a personalized home page *i.e.* Now used as a server side scripting language. It is used and executed on server. It was introduced mainly for web designing purpose. *PHP* was actually created **Rasmus Lerdorf** in 1994. At present it is hypertext preprocessor. It is used for server side web development. It can be used as a command line scripter. It can be used as a GUI for the client. It helped many software frameworks to provide RAD *i.e.* rapid application development. It provides dynamic contents to client.



Logo of PHP

History of PHP

- *PHP* development was started by **Rasmus Lerdorf** in 1994.
- He started keeping his personal home page to write a series of CGI *i.e.* common gateway interface in C.
- In 1997 the parser was rewritten by **Zeev Suraski** and **Andy Gutmans** which led to the existence of basic *PHP*.
- There had been only two stable releases with all security and bug fixes.
- *PHP* has a mascot which is blue elephant which was designed by Vincent Pointer.

Future of PHP

- The version 5.0 and 5.5 which were released are respectively supported till date June 2016 and August 2017.
- Version 7.0 will be released in mid October of 2015 and will be supported till 2018.
- Version 7.0 is released with the feature of uniform variable syntax, bitwise shift consistency across platform and return type declaration.
- Version 7.0 will complement its already existing parameter type declaration.
- Version 7.0 will also rectify all the already existing long time flaws of *PHP*.

Licensing of PHP

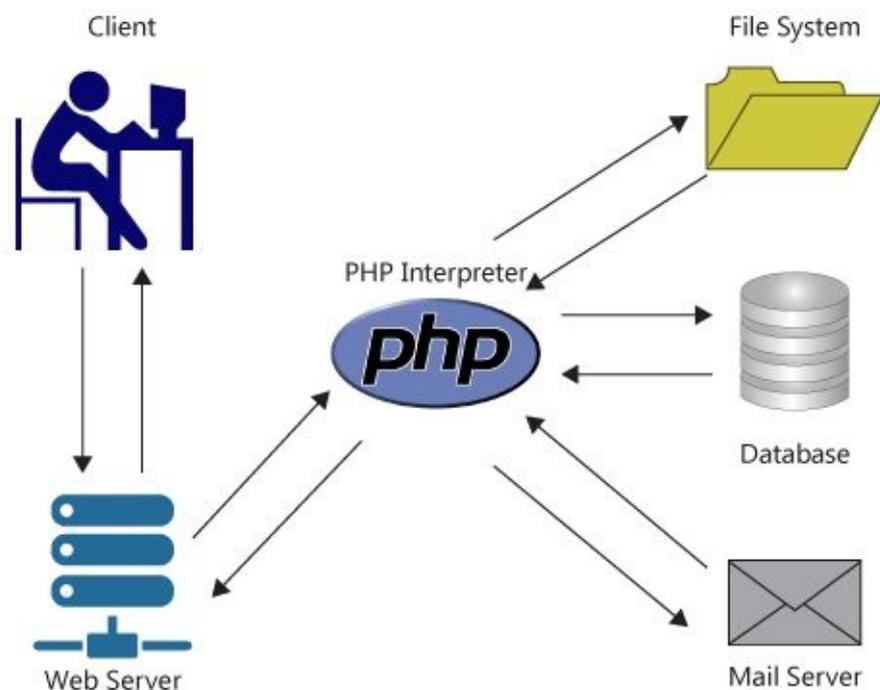
- *PHP* is a free software released free of cost software.
- Its license states that- product derived from *PHP* may not be called *PHP*, nor may *PHP* name should be appeared without prior written permission from group@php.net.
- You may indicate that your software had been working with *PHP* by saying FOO instead of *PHP*.
- Should not state it as *PHPFOO* or *phpfoo*.



Features of PHP

They are as follows.

- **Server side-** *PHP* won't execute within the local machines web browser, instead it will be executed in the web server.
- **Cross plat formed-** This feature allows the *PHP* script to run and executed on any operating system and web server.
- **HTML embedded language-** This means the commands and statement of *PHP* is already embedded in the HTML document.



Working of PHP

PHP supports many databases and it is open source software which is free to download and use.

Uses of PHP

- *PHP* can be used for creation of dynamic website.
- For building template that would make the site maintenance easy.
- For creation of graphics on the fly.
- For maintaining the differentiation of the type of content to be served on the basis of the users browser, IP address, date and time and various other details.
- For conducting a survey online.
- To connect databases with the web. For e.g. oracle and My SQL.
- Helps in providing a link of communication with external web.

- It performs read and process XML.

Simple scripting in PHP

```
<HTML>
<HEAD>
<TITLE>
</HEAD>
<BODY>
<?
//HELLO IN PHP
PRINT ("HELLO");
?>
</BODY>
</HTML>
```

The *PHP* statement is follow-

```
<? PRINT ("HELLO") ;?>
```

This would display the message “hello” in the window browser.

The *PHP* print function displays the value with the parenthesis as output in new window.

Important things about *PHP* scripts-

- *PHP* is case sensitive.
- *PHP* statements are stated between <? ?>.
- It always ends with semi-colon.
- Comments can be added using // in the beginning of the line.
- *PHP* file should not have .html extension.
- File should have *.PHP* extension.
- If the file don't have *.PHP* extension nor has .html extension the file won't found, parsed and executed.

Advantage of *PHP* with high security i.e. safe mode-

- The ownership should be same for the script being executed and file from which it is read or written.

- The directory should be owned by the owner of the *PHP* script.
- *PHP* cannot execute external programs which are not from the system.



Variables

Variables are used to store values like strings numbers etc. When a variable is declared, it can be used again and again in your script. Symbol used to represent *PHP* variables is “\$”.

Syntax

```
$ var _ name = value;
```

There is no need to specify in *PHP* that the variable use belong to which data type. *PHP* converts the variable into correct data type. A variable name can also contain alpha numerical characters but it should not contain space.

Example

```
<?php
$name = 'Raj';
$year born = 1977;
$current year = 2015;
$age = $current year - $year born;
Print (“$name is $age years old”)
?>
```

Predefined variables

The work environment that is been used by you is been accessed by the *PHP* script. Such variables are stored in a special hash data like `$_ENV` and `$_SERVER`. These variables can be used only one two in count at a time.

Example

```
<HTML>
<HEAD><TITLE>Example 2</TITLE> </HEAD>
<BODY>
<?
print(“hello>”);
print(“you are using $_SERVER[HTTP_SERVER_AGENT]<br>”);
```

```
print("your internet address is $_SERVER[REMOTE_ADDR]<br>");
```

```
?>
```

```
</BODY>
```

```
</HTML>
```

Variable from external source

Html forms (GET and POST)

Information from the form when submitted to *PHP* is automatically made available to the particular script.

Example

```
<form action = "foo.php" method = "post">
```

```
Name- <input type = "text" name = "username" /><br />
```

```
Email- <input type = "text" name= "email" /><br/>
```

```
<input type = "submit" name = "submit" value = "submit me!" />
```

```
</form>
```

Arrays

Array is an ordered map actually in *PHP* .This type of optimization is so that it can have several different uses. Arrays value can be arrays, trees and even multidimensional arrays.

Example

```
Array (  
Key => value,  
Key a => value2,  
Key b => value3,  
....  
)
```

Array can also be made using a key/value pair combination. These are also considered as associative arrays and hashes. An easy way to compress hash work is to think it as a table.

Example

Country (Key)	City (Value)
Cambodia	Phnom Penh
Lebanon	Beirut
Cuba	Habana
Indonesia	Jakarta
Finland	Helsinki

Table

```
<? php // create the array (  
'Cambodia' => 'Phom Penh',  
'Lebanon' => 'Beirut',  
'Cuba' => 'Habana',  
'Indonesia =>' Jakarta',
```

```
'Finland' => 'Helsinki');
```

```
Print ($capital ['Cambodia']);
```

```
?>
```

Predefined constant of array

- CASE_LOWER
- CASE_UPPER
- SORT_ASC
- SORT_DESC
- COUNT_NORMAL
- COUNT_RECURSIVE
- EXTR_SKIP
- EXTR_OVERWRITE
- EXTR_IF_EXISTS.



Operators

Arithmetic operators

<i>Example</i>	<i>Name</i>	<i>Result</i>
$-\$a$	Negation	Opposite of $\$a$
$\$a + \b	Addition	Sum of $\$a$ and $\$b$
$\$a - \b	Subtraction	Difference of $\$a$ and $\$b$
$\$a * \b	Multiplication	Product of $\$a$ and $\$b$
$\$a / \b	Division	Quotient of $\$a$ and $\$b$
$\$a \% \b	Modulus	Remainder of $\$a$ and $\$b$

The division operator would return float value unless the two operands are integers.

Example

```
<? php
```

```
Echo (5 % 3)."\n";
```

```
Echo (5 % -3)."\n";
```

```
Echo (-5 %3)."\n";
```

```
Echo (-5%-3)."\n";
```

```
?>
```

Assignment operator

There are two types of assignment operators which are commonly used in *PHP*. These are represented as “=” and “=>”.

Example

```
<? php
```

```
$a= ($b = 4) + 5;
```

```
?>
```

Bitwise operator

It allows evaluation and manipulation of a specific bit.

<i>Example</i>	<i>Name</i>	<i>Result</i>
$a \& b$	And	Are set in both a and b
$a b$	Or	Are set in either a or b
$a \wedge b$	Xor	Not in both
$\sim a$	Not	Are set in a are not set
$a \ll b$	Shift left	Shifts bits of a b to left
$a \gg b$	Shift right	Shift bits of a b to right

Comparison operator

It allows the comparison to be done by the use.

<i>Example</i>	<i>Name</i>	<i>Result</i>
$a == b$	Equal	True if $a = b$
$a === b$	identical	True if $a = b$
$a != b$	Not equal	True if $a \neq b$
$a <> b$	Not equal	True if $a \neq b$
$a !== b$	Not identical	True if $a \neq b$
$a < b$	Less than	True if a strictly less than

		\$b
\$a>\$b	Greater than	True if \$a greater than \$b

Error control operator

It is represented by '@'. Using this it will ignore any error that is been intended by the expression.

Example

```
<? php
```

```
$my_file = @file ('non_existent_file')
```

```
$value = @$cache [$key];
```

```
?>
```

Execution operator

Supports execution operator: back ticks (`). Use of back tick is identical to the shell_exec ().

Example

```
<? php
```

```
$output = `ls -al`;
```

```
echo "<pre> $output</pre>";
```

```
?>
```

Increment decrement operator

They are pre and post increment decrement operator in c style.

<i>Example</i>	<i>Name</i>	<i>Effect</i>
++\$a	Pre-increment	Increases by one
\$a++	Post-increment	Returns ,then increases
—\$a	Pre-decrement	Decreases then returns
\$a—	Post-decrement	Returns then decreases

Logical operator

The use of two different variables is to get different precedence as result.

<i>Example</i>	<i>Name</i>	<i>Result</i>
\$a and \$b	And	True if both are true
\$a or \$b	Or	True if either is true
\$a xor \$b	Xor	True if either is true
!\$a	Not	True if \$a is not true
\$a&&\$b	And	True if both are true
\$a \$b	Or	True if either is true



Functions

- Functions are groups of PHP statements grouped together that are or can be used over and over again.
- It has more than 1000 built in and user defined function.
- Function don't execute immediately.
- Functions only execute when u call them.
- Statement may contain multiple functions.

User-defined function

It can be defined as shown in the example.

Example

```
<?php
Function foo ($arg_1, $arg_2, ....., $args_n )
Echo " example.\n" ;
Return $retval ;
}
?>
```

Function argument

Through a argument list the information can be passed to the function. This is a comma delimited list of expression.

Example

```
<? php
Function takes_array ($input)
{
Echo "$input[0] +$input[1] = ",$input[0]+ $input[1];
}
?>
```

Returning values

Values are returned using the optional return statement. Any type can be returned, including arrays and objects.

Example

```
<? php  
Function square ($num)  
{  
Return $num * $num;  
}  
Echo square (4);  
?>
```

Internal (built-in) function

These are already set in the library. There also function that requires certain specified *PHP* extension compiled.

Example

To create image function we use *imagecreatetruecolor ()*.

Anonymous function

This function is also known as closure, this has no specific name.

Example

```
<? php  
Echo preg_replace_callback ('~-([a-z]) ~', function ($match [1]) ;  
{, 'hello-world');  
?>
```



SQL

SQL stands for standard query language which is used to access databases. It also used to manipulate database and it is an ANSI standard. MySQL is the widely used open source Relational database management system. MySQL is generally RDBMS being used for developing web-based software applications and you can add SQL queries to the MySQL database to insert, retrieve, update or delete data.



Logo of MySQL

History

- It was developed in 1970 at *IBM* with Donald D. Chamberlin and Raymond F. Boyce.
- It was first named as *SEQUEL*.
- This was sold to US NAVY, C.I.A and U.S. Government agencies after its potential was recognized.

Uses of SQL

- Used to execute query against database.
- To retrieve a data from database.
- To insert record.
- To update record.
- To delete record.
- To create a new database.
- To create new tables in database.
- To create various storage procedures in database.
- To create various kind of views in a database.
- To create permissions for the functioning of various procedures on tables and views.

How to use SQL in website

- One of the RDMS PROGRAM.
- Use of *PHP* or *ASP*.
- Use of *SQL* for the data to be retrieved.
- Use *HTML/CSS*.

Most important SQL command

- *Select* – To extract any needed or desired record or data from database.
- *Update* – To update or modify any data in the database.
- *Delete* – To delete or remove any data from database.
- *Insert into* – To add or insert into database.
- *Create database* – To create or start a new database.
- *Alter database* – To modify or edit database.
- *Drop table* – To remove or delete a table in database.
- *Creates index* – To create a new index.
- *Drop index*- To remove or delete an index.

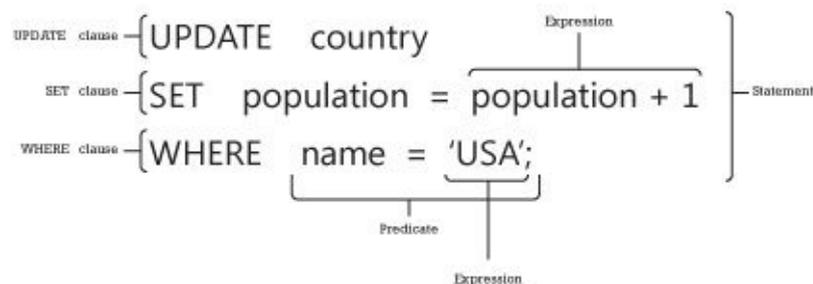
Syntax of SQL

SELECT *FROM Customers;

SQL is not case sensitive. This means both select and SELECT have the same meaning.

Various language elements of SQL languages

- Clauses
- Expression
- Predicate
- Query
- Statement
- Insignificant white space



Language elements of SQL



Operators in SQL

SQL operators are as follows.

Operator	Description	Example
=	Equal to	John = 'author'
<>	Not equal to	Dept<>height
>	Greater than	100>6
<	Less than	6<100
≥	Greater than or equal	Rate≥2.6
≤	Less than or equal	Rate ≤2.6
BETWEEN	In an inclusive range	Range BETWEEN 10 AND 20
LIKE	Match the character	NAME LIKE 'WILLY%'
IS OR IS NOT	Compare to null	AGE IS OR IS NOT RAJ
IN	Equal to one of the multiple possible value	ID IN (100,120,134)
AS	Used to change name when viewing the result or the particular value	Select student AS 'section a'
IS NOT DISTINCT FROM	Equal to value given or both are missing data	Loan IS NOT DISTINCT FORM-APPLICABLE

Conditions

SQL expression-

Case- It is one of the searching techniques of finding the matching case. It was used in SQL-92.

Example

CASE WHEN N > 0

THEN 'POSITIVE VALUE'

WHEN < 0

THEN 'NEGATIVE VALUE'

ELSE

'ZERO'

When- This condition is the source through which the data is compared with the output which is set is similar to its condition. That time the action which has to be take is defined using when the condition with the when statement is stated.

Example

CASE a

WHEN 1

THEN 'ONE ORANGE'

WHEN 2

THEN 'TWO ORANGES'

ELSE

'CANNOT COUNT THAT VALUE'

Then- This statement is used to execute particular value or display particular value when the condition in *WHEN* statement is satisfied.

Example

CASE b

WHEN 3

THEN 'three ORANGE'

WHEN 4

THEN 'four ORANGES'

ELSE

'CANNOT COUNT THAT VALUE'.

Else- When the when condition is not satisfied the process leads to another option, that is defined as *else* statement.

Example

WHEN 5

THEN 'FIVE ORANGES'

ELSE

'CANNOT COUNT THAT VALUE'.

End- It is used to terminate the program which leads the system to stop the execution.

Example

CASE x

WHEN 1

THEN 'COUNT 1'

WHEN 2

THEN 'COUNT 2'

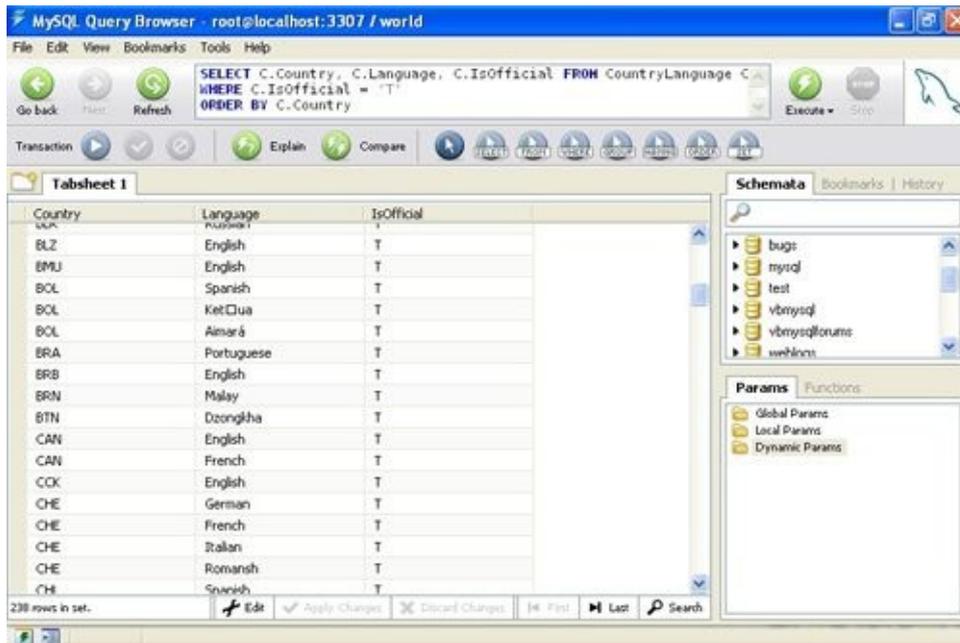
ELSE 'ERROR'

END



Queries

- Query is the most commonly used and one of the key operations in *SQL*.
- It is declared using *SELECT* statement.
- It provides user with various methods to describe and show his/her desired data.
- An '*' is used to show that the query should return all columns of the particular that is queried.
- It is a request information method used in the *SQL* therefore it is a key factor and mostly the back bone of the *SQL*.



MySQL query window

Various clauses and keywords used in *SQL* are as follows.

- *FROM clause*- This shows from where or which table to retrieve data from.
- *Where clause*- This includes a predicate for comparison this helps in restricting of the row that is been returned by the query.
- *Group by clause*- Used to project rows that are identical or have common value.
- *Having clause*- It is used in filtration of row where it can be combined with *Group by clause*.
- *Order by clause*- It sorts or arranges the data in a form it's needed to be arranged.

Example

SELECT Book title AS TIM,

COUNT (*) AS Authors

FROM Book

JOIN Book_TIM

ON Book . isbn = Book_author . isbn

GROUP BY Book . title

Title	Authors
SQL Examples and Guide	4
The Joy of SQL	1
An Introduction to SQL	2
Pitfalls of SQL	1

Output

Sub-Queries

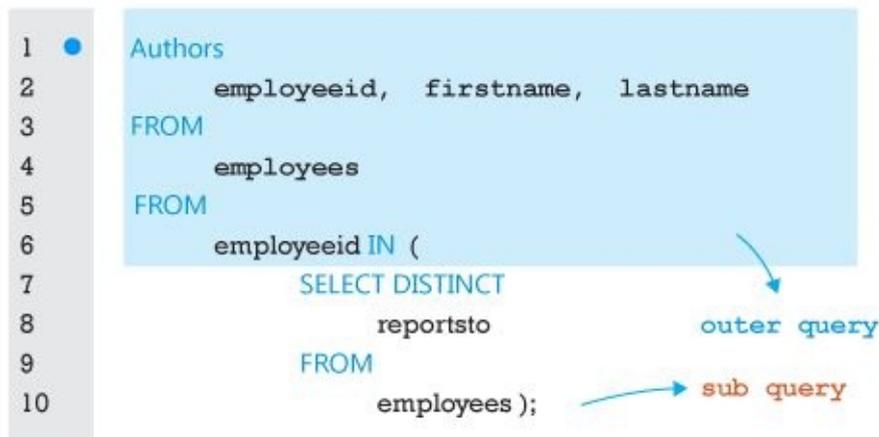
These are nested queries. A query that is in another query and this is embedded within the where clause. Sub-Queries are usually used to return data which will be further used in the main query.

Rules of Sub-queries

- It has to be enclosed within parenthesis.
- It can have only one column in SELECT clause.
- It is not allowed to use ORDER BY query whereas instead of that GROUP BY is used.
- It cannot have immediate enclosure in a set function.

Use of sub-queries

- It is used to join one table with other tables operation.
- This also provides faster remedies. This provides a hierarchical execution.
- Sub query can use values from query outside, these are correlated sub query.



Sub-Query

Example

SELECT isbn,

Name,

Price,

FROM Book

WHERE price < (SELECT AVG (PRICE) FROM Book)

ORDER BY title;

3VL

3VL stands for three-valued logic to SQL. There are 3 fragments of 3VL which are AND, OR and NOT. There tables which are followed in 3VL.

P AND q		p		
		True	False	Unknown
q	True	True	False	Unknown
	False	False	False	False
	Unknown	Unknown	False	Unknown

AND table

p or q		p		
		True	False	Unknown
q	True	True	True	True
	False	True	False	Unknown
	Unknown	True	Unknown	Unknown

OR table

q	NOT q
True	False
False	True
Unknown	Unknown

Not table

NULL is one of the most focused element of the 3VL and also had been very controversial. It is just used to indicate the missing or absence of value and it does not

belong to any data domain.



Null representation

Therefore because of its existence issue *NULL* can never ever result in either true or false so third new logical value is introduced that is “*UNKNOWN*”.



Data controls

The *Data Control Language (DCL)* allows users to access and manipulated data. Its main statements are as follow.

- *GRANT* allows one or more than one user to perform an operation on an object.
- *REVOKE* to eliminate a default grant.

Example

```
GRANT SELECT, UPDATE
```

```
ON example
```

```
TO some_user, another_user;
```

```
REVOKE SELECT, UPDATE
```

```
ON example
```

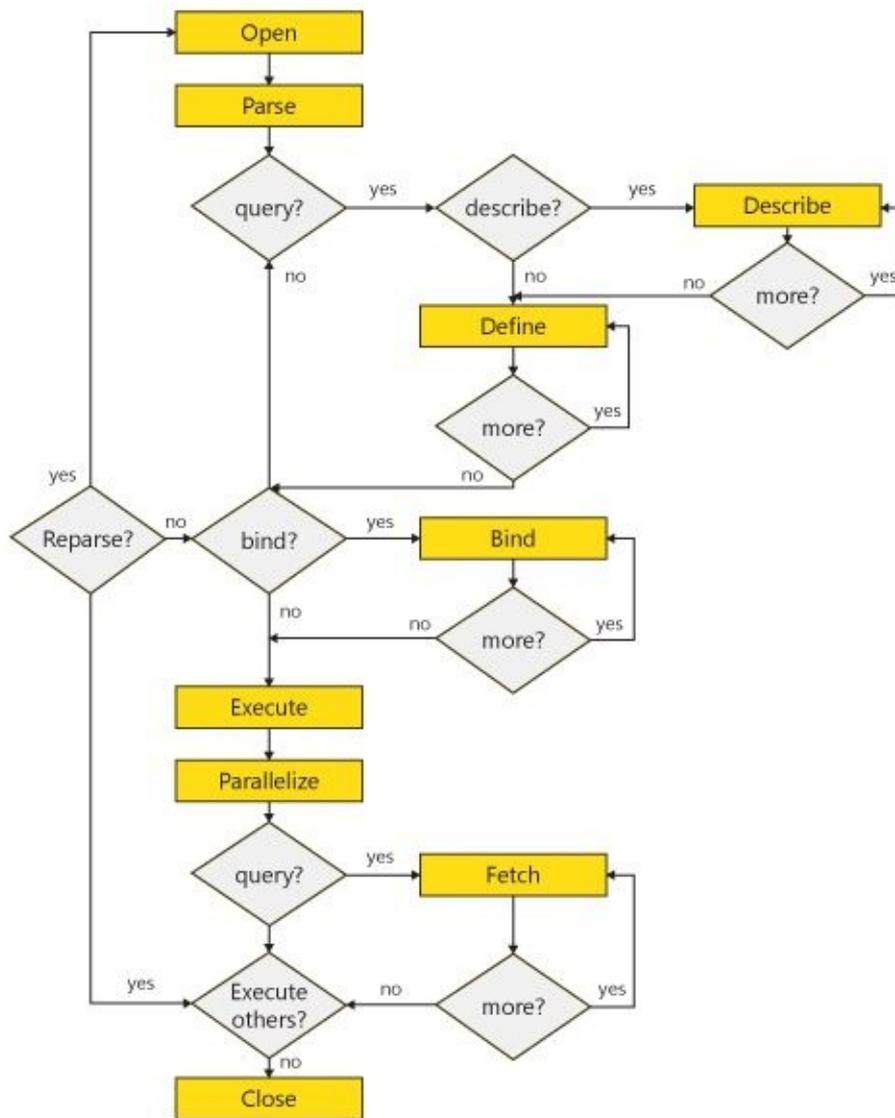
```
FROM some_user, another_user;
```

It carries out functions such as connect insert, select, update, execute and delete.

Procedural extensions

The designing of *SQL* is done in a form of query contained relation database. The procedural extension diagram of *SQL* is shown below. This allows programming contained in form of routines which. Some elements known in procedural extension of *SQL* are as follows.

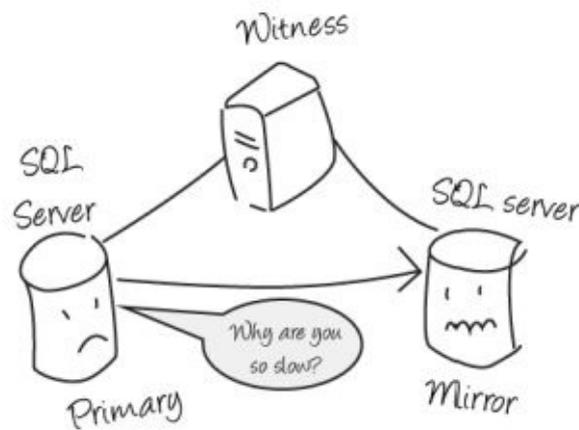
- Polymorphism with the parameters in *SQL* of *IN*, *OUT* and *INOUT* modes.
- Various procedures calling statement.
- Atomic BEGIN and END call.
- Locally declared *SQL* statement.
- Error signaling and cursor processing.
- Declared data types.
- Various conditional control and looping statement.



Procedural extension

Drawbacks

- Portability do lack between database systems.
- Implementation is not supported with entire freedom because of the size and complexity of the *SQL*.
- The behavior of the implementation at several regions of database is confusing some times.
- There is little commercial incentive to vendors to enforce it as an easier task for the user to bring any changes in database suppliers.
- User's evaluation of database software avoids factors such as higher performances for the standard priority.



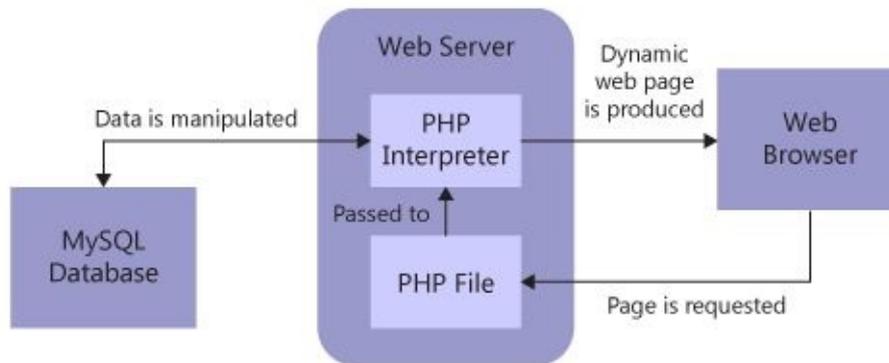
Performance issue

- *SQL* has also been found to have a conflict in compatibility with the consumer's prior database.



PHP with MySQL

PHP connected to MySQL for data manipulation. It also provides cross platform functionality.



PHP with MySQL

In a web server it there is a *PHP* interpreter which get an input from the *PHP* file space. This input is nothing

PHP creates a *MySQL*. It can be created by using–

- *MySQLi*- My SQL improvement.
- *PDO*- *PHP* data objects.

Example

MySQLi

```
<? Php
```

```
$servername = "local host";
```

```
$username = "username";
```

```
$password = "password";
```

```
// Create connection
```

```
$conn = new mysqli($servername, $username, $password);
```

```
// Check connection
```

```
if ($conn->connect_error)
```

```
{
```

```
die ("Connection failed: " . $conn->connect_error);
```

```
}
```

```
// Create database

$sql = "CREATE DATABASE myDB";

if ($conn->query($sql) === TRUE)
{
    echo "Database created successfully";
}

else

{
    echo "Error creating database: " . $conn->error;
}

$conn->close();

?>
```

PDO

```
<? php

$servername = "local host";

$username = "username";

$password = "password";

try

{
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "CREATE DATABASE myDBPDO";
    // use exec() because no result returned
    $conn->exec($sql);
    echo "Database created <br>";
}

}
```

```
catch(PDOException $e)
{
echo $sql . "<br>" . $e->getMessage ();
}

$conn = null;

?>
```

Syntax rules

- Query has to be quoted in PHP.
- Numeric value must be avoided.
- NULL value should not be quoted, it should be avoided.

High efficient statements

- Preparation statement- Is an SQL statement use to create and send a template to the particular database.
- Execute statement- Is used to bind all the retrieved values and to execute them.



Creating MySQL database using PHP

Admin should be privilege to create database. *PHP* uses `mysql_query` for creation of the *SQL* database. This has two parameter functioning which is true when creation is successful and false when a failure occurs.

Syntax

```
Bool mysql_query (sql, connection);
```

Example

```
<? php
```

```
$dbhost = 'localhost :30306' ;
```

```
$dbuser = 'root';
```

```
$dbpass = 'password'
```

```
$conn = mysql_connection ($dbhost, $dbuser, $dbpass);
```

```
If (!conn)
```

```
{
```

```
Die ('could not connect'. mysql_error ());
```

```
}
```

```
Echo 'connect success';
```

```
$sql = 'create database test-tb';
```

```
$retval = mysql_query ($sql, $conn);
```

```
If ( ! $retval)
```

```
{
```

```
Die ('could not create database'. mysql_error ());
```

```
}
```

```
Echo "database test_db created database\n";
```

```
Mysql_close ($conn);
```

```
?>
```

Insert of data in MySQL database using PHP

The insert statement carries out the function using *PHP* by using query statement `mysql_query` for the SQL INSERT statement.

Example

```
<? php
$dbhost = 'localhost: 3036';
$dbuser = 'root';
$dbpass = 'password';
$conn = mysql_connect ($dbhost, $dbuser, $dbpass);
if (! $conn)
{
    die ('Could not connect'. mysql_error ());
}
$sql = 'INSERT INTO employee '
'(emp_name, emp_address, emp_salary, join_date) '
'VALUES ("guest", "XYZ", 2000, NOW ())';

mysql_select_db ('test_db');
$retval = mysql_query ($sql, $conn);
if (! $retval)
{
    die ('could not enter data: '. mysql_error ());
}
echo "Entered data successfully\n";
mysql_close ($conn);
?>
```



Update in MySQL Database using PHP

For updating in MySQL there is a need of the command to be carried out that is SQL UPDATE statement through *PHP* function `mysql_query`.

Example

```
<html>
<head>
<title>Update a Record in MySQL Database</title>
</head>
<body>
<?php
if(isset($_POST['update']))
{
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'password';
$conn = mysql_connect ($dbhost, $dbuser, $dbpass);
If (! $conn)
{
die ('Could not connect: ' . mysql_error());
}
$emp_id = $_POST['emp_id'];
$emp_salary = $_POST['emp_salary'];
$sql = "UPDATE employee "
"SET emp_salary = $emp_salary "
"WHERE emp_id = $emp_id";
mysql_select_db ('test_db');
```

```

$retval = mysql_query ($sql, $conn);
if (! $retval)
{
die('Could not update data: ' . mysql_error());
}
echo "Updated data successfully\n";
mysql_close($conn);
}
else
{
?>
<form method="post" action="<?php $_PHP_SELF ?>">
<table width="400" border="0" cellspacing="1" cellpadding="2">
<tr>
<td width="100">Employee ID</td>
<td><input name="emp_id" type="text" id="emp_id"></td>
</tr>
<tr>
<td width="100">Employee Salary</td>
<td><input name="emp_salary" type="text" id="emp_salary"></td>
</tr>
<tr>
<td width="100"> </td>
<td> </td>
</tr>
<tr>
<td width="100"> </td>

```

```
<td>
<input name="update" type="submit" id="update" value="Update">
</td>
</tr>
</table>
</form>
<?php
}
?>
</body>
</html>
```

The updating performed using *PHP* in MySQL provides more efficiency and exception.

Retrieval in MySQL Database using PHP

Retrieval in *SQL* is termed as fetch therefore to perform retrieval in *SQL* using *PHP* a code is been in use which is as follow

Syntax

```
Mysql_fetch_arrays ( )
```

Example

```
<?php
```

```
$dbhost = 'localhost:3036';
```

```
$dbuser = 'root';
```

```
$dbpass = 'password';
```

```
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
```

```
if(! $conn )
```

```
{
```

```
die('Could not connect: ' . mysql_error());
```

```
}
```

```
$sql = 'SELECT emp_id, emp_name, emp_salary FROM employee';
```

```
mysql_select_db('test_db');
```

```
$retval = mysql_query( $sql, $conn );
```

```
if(! $retval )
```

```
{
```

```
die('Could not get data: ' . mysql_error());
```

```
}
```

```
while($row = mysql_fetch_array($retval, MYSQL_ASSOC))
```

```
{
```

```
echo "EMP ID :{$row['emp_id']} <br> “.
```

```
“EMP NAME : {$row['emp_name']} <br> “.
```

```
“EMP SALARY : {$row[‘emp_salary’]} <br> “.
```

```
“_____<br>”;
```

```
}
```

```
echo “Fetched data successfully\n”;
```

```
mysql_close($conn);
```

```
?>
```



Deleting of MySQL Database using PHP

The database once is not needed and the user don't want to have it any more or want to free space then the user can pass a statement that would state to delete the database through the `mysql_query` for deletion of data. The term used in *SQL* for delete is *drop*.

Example

```
<? php
$dbhost = 'localhost: 3036';
$dbuser = 'root';
$dbpass = 'password';
$conn = mysql_connect ($dbBhost, $dBUser, $dBBpass);
if (! $conn)
{
Die ('could not connect: '. mysql_error ());
}
$sql = 'DROP DATABASE test_d';
$retval = mysql_query ($sql, $conn);
If (! $retval)
{
Die ('Could not delete database db_test 'mysql_error ());
}
Echo "Database deleted successfully\n";
mysql_close ($conn);
?>
```

Deleting of data from MySQL using PHP

When intended or the user wants to delete any data from the *MySQL* database using *PHP* the user can pass a statement stating that through `mysql_query`.

Example

```
<html>
<head>
<title>Delete from MySQL Database</title>
</head>
<body>
<?php
if(isset($_POST['delete']))
{
$dBBhost = 'localhost:3036';
$dBBuser = 'root';
$dBBpass = 'password';
$conn = mysql_connect($dBBhost, $dBBuser, $dBBpass);
if(! $conn )
{
Die ('could not connect: '. mysql_error ());
}
$emp_id = $_POST ['emp_id'];
$sql = "DELETE employee ".
"WHERE emp_id = $emp_id";
mysql_select_db ('test_d');
$retval = mysql_query ($sql, $conn);
if (! $retval)
```

```

{
Die ('could not delete data: '. mysql_error ());
}
Echo "Deletion successful\n";
mysql_close ($conn);
}
else
{
?>
<form method="post" action="<? php $_PHP_SELF?>">
<table width="400" border="0" cellspacing="1" cellpadding="2">
<tr>
<td width="100">Employee ID</td>
<td><input EPRORNAME="ePROR_NAMED" type="text" id="ePROR_idD"></td>
</tr>
<tr>
<td width="100"> </td>
<td> </td>
</tr>
<tr>
<td width="100"> </td>
<td>
<input EPROR="delete" type="submit" id="delete" value="Delete">
</td>
</tr>
</table>
</form>

```

```
<?php  
}  
?>  
</body>  
</html>
```

You can take back up of the *SQL* database using following methods.

- Using command of *SQL* through *PHP*.
- Using *mysqldump* with the help of *PHP*.
- By using a *phpMyAdmin* user interface.



Knowledge flow: more eBooks and Apps

- [Get more Google Play Books](#)
- [Get more eBooks](#)
- [Get more apps from Google Play store](#)
- [Get more apps from Amazon app store](#)

