## Methods/Functions list in our class

Our class has twenty different methods. The lists of all methods of our class with their details are listed as:

1. select\_all($table)

Input: $table is the name of the table

Output: Executes the query select \* from sometable and returns the data as an array.

1. count\_all($table)

Input: $table is the name of the table

Output: Executes the query select \* from sometable and returns the number of rows affecting the query.

1. select\_all\_with\_order\_by($table,$order)

Input: $table is the name of the table,$order is the name of the field or column as a key and the order type like asc,desc as a value in an array e.g. array(‘date’=>’desc’,’id’=>’desc’) or array(‘date’=>’desc’)

Output: Executes the query similar to

select \* from sometable order by id desc,date desc or select \* from sometable order by name asc

and returns the data as an array

1. select\_all\_with\_order\_by\_limit($table,$order,$offset,$limit)

This function number 4 is similar to function number 3.The only additional input to be given is the third and fourth parameter. $offset and $limit .These parameters are the integer e.g. $offset=0 ,$limit=3

Output: Executes the query similar to

select \* from sometable order by date desc limit 0,3

and returns the data as an array

1. select\_all\_with\_condition($table,$condition)

Input:$table is the name of the table.$condition is an array of data with the name of the field as a key and the field value as a value. e.g. $condition=array(“id”=>1,”status”=>’a’);

Output: Executes the query similar to

Select \* from sometable where id=1 and status= ‘a’

And returns the data as an array

1. count\_all\_with\_condition($table,$condition)

This function number 6 is similar to function number 5 .Only difference is the value it returns.

It returns the number of rows affected by the query.while function number 5 returns the data as an array

1. select\_all\_with\_condition\_order\_by($table,$condition,$order)

Input: $table is the name of the table, $condition is the array of fields whose value is to be matched with the name of the field as key and value of that fields as a value.e.g. $condition=array(‘id’=>1,’status’=>’a’)

,$order is the name of the field or column as a key and the order type like asc,desc as a value in an array.e.g. $order=array(‘date’=>’desc’,’id’=>’desc’)

Output:

Select \* from sometable where id=1 and status=’a’ order by date desc,id desc

1. select\_all\_with\_condition\_order\_by\_limit($table,$condition,$order,$offset,$limit)

This function number 8 is similar to function number 7.The only difference is the fourth and fifth parameter.$offset and $limit.they are the integer values. E.g.$offset=0 $limit=5

Output: Executes the query similar to

Select \* from sometable where status=’a’ order by date desc,id desc limit 0,5

1. select\_field($table,$fields)

Function number from 1 to 8 is to select all the fields from the table but functions from 9 to 16 is to select few fields from the table. It is a good practice to execute select query specifying the fields rather than to select all fields for performance of a web application.

Input:$table is the name of the table, $field is the array of data containing the list of fields to select. E.g. $fields=array(‘id’,’title’,’description’)

Note: all arrays are the arrays with the key value pair but only the $field while executing the select query is an array with non-key value pair.

Output: Executes the query similar to

Select title,description from sometable

and returns the data as an array

1. count\_field($table,$fields)

Function number 10 is similar to function number 9.The only difference is the value it returns.It returns the number of rows affected by the query.The query is similar to function number 9.

1. select\_field\_with\_order\_by($table,$fields,$order)

Input: $table is the name of the table. $fields is the array of data containing the list of fields to select. E.g. $fields=array(‘id’,’title’,’description’) ,$order is the name of the field or column as a key and the order type like asc,desc as a value in an array.e.g. $order=array(‘date’=>’desc’,’id’=>’desc’)

Output:

Executes the query similar to

Select id,name,description from sometable order by name asc, id desc

And returns the data as an array

1. select\_field\_with\_order\_by\_limit($table,$fields,$order,$offset,$limit)

This function number 12 is similar to function number 1. The only difference is the fourth and fifth parameters $offset and $limit. These are integer values .$offset=0 $limit=20.the limiting of data while executing the query is effective while doing the pagination of the data.

Output:

Executes the query similar to

Select id,name,description from sometable order by name asc, id desc limit 0,20

And returns the data as an array

1. select\_field\_with\_condition($table,$fields,$condition)

Input:$table is the name of the table . $fields is the array of data containing the list of fields to select. E.g. $fields=array(‘id’,’name’,’price’) and $condition is the array of fields whose value is to be matched with the name of the field as key and value of that fields as a value.e.g. $condition=array(‘cat\_id’=>1,’status’=>’a’)

Output:

Executes the query similar to

Select id,name,price from products where cat\_id=1 and status=’a’

And returns the data or a result set as an array

1. count\_field\_with\_condition($table,$field,$condition)

This function number 14 is similar to function number 13.slightly difference is the field parameter here.In other cases fields(plural) parameter should be an array .Here field(singular) is just a string. $field=’id’ or $field=’date’ mostly singular $field should represent the unique term or a unique field in a table.The return value is the number of rows affected by the query.

Output:Executes the query similar to

Select id from products where cat\_id=1 and status=’a’

Returns the number of rows affected by the above query.

1. select\_field\_with\_condition\_order\_by($table,$fields,$condition,$order)

Input:$table is the name of the table. $fields is the array of data containing the list of fields to select. E.g. $fields=array(‘id’,’name’,’price’) and $condition is the array of fields whose value is to be matched with the name of the field as key and value of that fields as a value.e.g. $condition=array(‘cat\_id’=>1,’status’=>’a’). $order is the name of the field or column as a key and the order type like asc,desc as a value in an array.e.g. $order=array(‘name’=>’asc’,’date’=>’desc’)

Output: Executes the query similar to

Select id,name,price from products where cat\_id=1 and status=’a’ order by name asc,date desc

1. select\_field\_with\_condition\_order\_by\_limit($table,$fields,$condition,$order,$offset,$limit)

This function number 16 is similar to function number 15.The only difference is the fifth and sixth input parameters.$offset and $limit.They are the integer values. e.g. $offset=0 $limit=20

Output:Executes the query similar to

Select id,name,price from products where cat\_id=1 and status=’a’ order by name asc,date desc limit 0,20

And returns the data as an array

1. insert($table,$fields)

This function insert the data into the table. The first input parameter $table is the name of the table. The second input parameter $fields is an array of data containing the name of the field or column where the data to be inserted as a key and the value to be inserted in a respective field or column as a value.$fields=array(‘name’=>$\_POST[‘name’],’price’=>$\_POST[‘price’],’date’=>date(‘Y-m-d’))

Output: Executes the query similar to

Insert into products (name,price,date) values (‘flipflop,’ 10’,’2015-5-3’)

1. latest\_inserted\_id()

Sometimes it is highly required to know the id of the latest inserted item in the database. This function number 18 does this. This function takes no argument. This function works after the insert query is executed. If the insert query is not executed at a time then this function returns the value of 0.If the insert query is executed at the latest time. Then this function returns the id value of that data inserted into the table.

1. update($table, $fields, $condition)

This function fires the update query in a table.

Input: $table is the name of the table

The second input parameter $fields is an array of data containing the name of the field or column as a key and the value that is to be updated in a respective field or column as a value.$fields=array(‘name’=>$\_POST[‘name’],’price’=>$\_POST[‘price’],’date’=>date(‘Y-m-d’))

$condition is the array of fields whose value is to be matched with the name of the field as key and value of that fields as a value.e.g.$condition=array(‘id’=>1)

Output:

Executes the query similar to

Update products set name=’tflipflop’,price=1,date=’2015-8-11’ where id=1

1. delete($table, $condition)

Input:$table is the name of the table $condition is the array of fields whose value is to be matched with the name of the field as key and value of that fields as a value.e.g.$condition=array(‘id’=>1)

Output: Executes the query similar to delete from products where id=1