ADVANCE INTERNET INFORMATION PROCESSING
ASSIGNMENT – 2

SUBMITTED BY

Kommuru Jaya Naga Bhavana

A04711981

SUBMITTED TO

Prof. WUXU PENG

DEPARTMENT OF COMPUTER SCIENCE

TEXAS STATE UNIVERSITY

QUESTION 1

1. (10 + 15 + 15 = 40 pts) Extend the "Table" class discussed in PHP lecture as follows:

(1) Add a new method "rmRow" that allows a user to remove a row from the given table. The method "rmRow" takes a row of values and verifies if there exists a row that has all the given values. If yes, it will remove it. If that row doesn't exist, then you program should provide proper error message/warning information.

(2) Similar to the "addRowAssocArray" method, add a new method "rmRowAssocArray" that will remove a row using associative array as parameters. If that row doesn't exist, then you program should provide proper error message/warning information.

(3) Add a new method "addCol" that adds a new column to a given table. Besides the new column name to be added the method can take an optional value parameter that will be used as value of all existing rows for the added column. If this value parameter is not given then all existing rows will have value "null" for the new column.

(4) Add a new method "rmCol" that removes a column. The only parameter of the method is the name of the column.

(5) Add a new method "renameCol" that renames one or more columns. A complete implementation must provide a web page that provides all of the four options above.

The Table.php file is stored in:

/home/Students/j_k201/public_html/ASSIGNMENT2/A2Q1/Table.php

For executing all the operations on a table with values is as follows:

Table.php

```
<!DOCTYPE html>
<html>
<head>
<title>AIIP ASSIGNMENT</title>
</head>

<body>
<?php

class Table
{
    var $table_array = array();
    var $headers = array();
    var $cols;

    function Table ( $headers )
    {
        $this->headers = $headers;
        $this->cols = count ($headers);
    }

    function addRow ( $row )
    {
```

```
    if ( count ($row) != $this->cols )
        return false;
    array_push ( $this->table_array, $row );
    return true;
}


function addRowAssocArray ( $row_assoc )
{
    $row = array();
    foreach ( $this->headers as $header )
    {
        if ( ! isset ($row_assoc[$header] ) )
            $row_assoc[$header] = "";
        $row[] = $row_assoc[$header];
    }
    array_push($this->table_array, $row);
    return true;
}


function output ( )
{
    print "<pre>";
    foreach ( $this->headers as $header )
    print "<b>$header</b> ";
    print "\n";
    foreach ( $this->table_array as $y )
```

```
        {
              foreach ( $y as $xcell )
              print "$xcell ";
              print "\n";
        }
        print "</pre>";
  }

  function rmRow ( $row )
  {
$key = array_keys($this->table_array, $row);
  foreach($key as $searchKey)
  {
    unset($this->table_array[$searchKey]);
  }
return true;
  }

  function rmRowAssocArray ( $row_assoc )
  {
    $row = array();
    foreach ( $this->headers as $header )
    {
      if ( ! isset ($row_assoc [$header] ) )
        $row_assoc[$header] = "";
      $row[] = $row_assoc [$header];
    }
    $key = array_keys($this->table_array, $row);
```

```
   foreach($key as $newKey)
   {
         unset($this->table_array[$newKey]);
}
return true;
}

function addCol($header, $value=NULL )
{
      if(isset($header))
         {
   $this->headers[] = $header;
   $this->cols +=1;
}
if(!isset($value))
   $value = "null";
$index = count($this->table_array);
         for($i = 1; $i <= $index; $i++)
   $this->table_array[$i][] = $value;
}

function rmCol($rm_col)
{
$position = array_search($rm_col, $this->headers);
if($position != "")
{
            unset($this->headers[$position]);
   foreach($this->table_array as &$x)
```

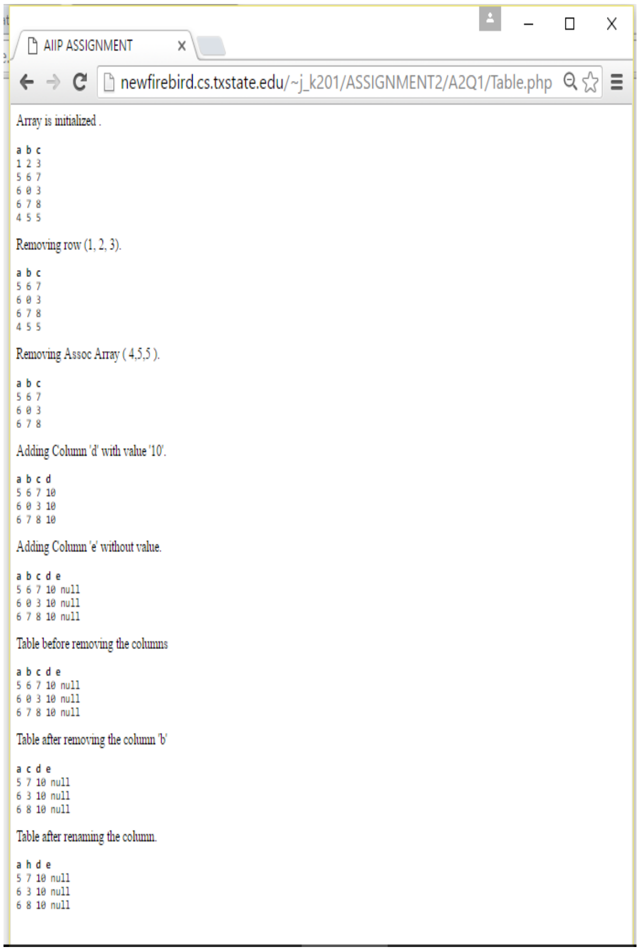
```
        {
                    unset($x[$position]);
          }
      }
      else
        return false;
      }

      function renameCol($old_col,$new_col)
      {
            $position = array_search($old_col, $this->headers);
            if($position != "")
          {
                  $this->headers[$position]=$new_col;
            }
          else
                  return false;
      }
}

$test = new table ( array ("a", "b", "c") );
$test->addRow ( array (1, 2, 3 ) );
$test->addRow ( array (5, 6, 7 ) );
$test->addRowAssocArray ( array ("b"=>0, "a"=>6, "c"=>3 ) );
$test->addRow ( array (6, 7, 8 ) );
$test->addRow( array(4,5,5));

print "Array is initialized .\n ";
```

```
$test->output();


$test->rmRow (array (1, 2, 3 ) );
print"Removing  row (1, 2, 3). \n";
$test->output();



$test->rmRowAssocArray ( array ("b"=>5, "a"=>4, "c"=>5 ) );
print"Removing Assoc Array ( 4,5,5 ). \n";
$test->output();

print "Adding Column 'd' with value '10'.\n";
$test->addCol('d',10);
$test->output();

print "Adding Column 'e' without value.\n";
$test->addCol('e');
$test->output();

print "Table before removing the columns";
print "\n";
$test->output();

print "Table after removing the column 'b'";
$test->rmCol('b');
$test->output();

print "Table after renaming the column.";
```

```
$test->renameCol('c','h');

$test->output();

?>


</body>

</html>
```

Now after running the program by typing the url in the web browser the output we get is as below:

URL : http://newfirebird.cs.txstate.edu/~j_k201/ASSIGNMENT2/A2Q1/Table.php


AIIP ASSIGNMENT
newfirebird.cs.txstate.edu/~j_k201/ASSIGNMENT2/A2Q1/Table.php

```
Array is initialized .

a b c
1 2 3
5 6 7
6 0 3
6 7 8
4 5 5

Removing row (1, 2, 3).

a b c
5 6 7
6 0 3
6 7 8
4 5 5

Removing Assoc Array ( 4,5,5 ).

a b c
5 6 7
6 0 3
6 7 8

Adding Column 'd' with value '10'.

a b c d
5 6 7 10
6 0 3 10
6 7 8 10

Adding Column 'e' without value.

a b c d e
5 6 7 10 null
6 0 3 10 null
6 7 8 10 null

Table before removing the columns

a b c d e
5 6 7 10 null
6 0 3 10 null
6 7 8 10 null

Table after removing the column 'b'

a c d e
5 7 10 null
6 3 10 null
6 8 10 null

Table after renaming the column.

a h d e
5 7 10 null
6 3 10 null
6 8 10 null
```

QUESTION 2

2. (60 pts) Consider the sample program sample3.pc given in Oracle 10g's document (We use it in Oracle 11g here). Here is the comment section on top of the program file:

This program connects to ORACLE, declares and opens a cursor, fetches in batches using arrays, and prints the results using the function print_rows()

You can compile and run sample3.pc to verify its output.

Convert and revise that program to a complete Web application as follows. Instead of just fetching the department number, names and salary of all employees in every department, your application will present a Web interface to a user. The interface has two options:

- an option for a user to enter a department number OR name. Then your application will return and display the employee name, salary, and commission of all employees in that department.
- an option for a user to enter two salary values. Then your application will return and display the department name, employee name, and salary value for all employees with salary value between the two given salary values.
- an option for a user to enter two commission values. Then your application will return and display the department name, employee name, and salary value for all employees with salary value between the two given commission values.

You can use either Perl or PHP as the CGI scripting language. The two tables emp and dept involved in this problem are available in the sample use scott's schema.

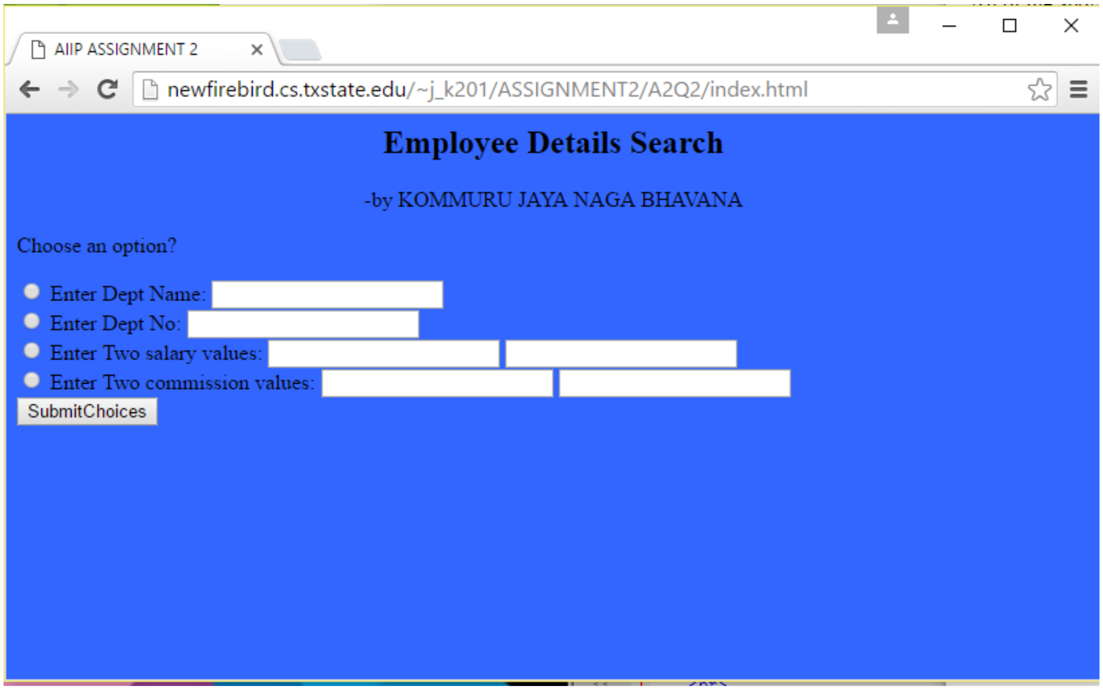
STEP 1: Creating HTML page.

<u>Index.html</u>

```
<html>
<head>
<title>AIIP ASSIGNMENT 2</title>
</head>

<body bgcolor="#3366ff" link="#000000">

  <center><h2>Employee Details Search</h2></center>
  <center>-by KOMMURU JAYA NAGA BHAVANA</center>

  <body>
    <form action="/~j_k201/cgi-bin/EmployeeSearch.pl" method="get">

    <p>Choose an option?</p>

    <input type="radio" name="empsearch" value="DeptSearch1" />
                        Enter Dept Name:
            <input name="dept_name" type="text" size="20"/>
            <br>
            <input type="radio" name="empsearch" value="DeptSearch2" />
                        Enter Dept No:
            <input name="dept_no" type="number" size="20"/>
    <br>
            <input type="radio" name="empsearch" value="sal" />
        Enter Two salary values:
            <input name="sal1" type="number" size="20"/>
        <input name="sal2" type="number" size="20"/>
    <br>
        <input type="radio" name="empsearch" value="comm" />
      Enter Two commission values:
        <input name="comm1" type="number" size="20"/>
        <input name="comm2" type="number" size="20"/>
        <br>
        <input name="SubmitChoices" type="submit" align = "center" value =
"SubmitChoices"/>


    </form>
  </body>
</html>
```

This index.html page can be accessed through web browser as:

URL: http://newfirebird.cs.txstate.edu/~j_k201/ASSIGNMENT2/A2Q2/index.html


AIIP ASSIGNMENT 2
newfirebird.cs.txstate.edu/~j_k201/ASSIGNMENT2/A2Q2/index.html

Employee Details Search

-by KOMMURU JAYA NAGA BHAVANA

Choose an option?

○ Enter Dept Name:

○ Enter Dept No:

○ Enter Two salary values:

○ Enter Two commission values:

SubmitChoices

STEP 2: Now for retrieving the data from the database we need to mention the following commands in the C++ code as :

- an option for a user to enter a department number OR name. Then your application will return and display the employee name, salary, and commission of all employees in that department.

  ```
  EXEC SQL DECLARE c1 CURSOR FOR
        SELECT  ename, sal , comm FROM emp e,dept d WHERE e.deptno=d.deptno;
  EXEC SQL OPEN c1;
  EXEC SQL DECLARE c1 CURSOR FOR
        SELECT e. ename, e.sal , e.comm FROM emp e,dept d WHERE e.deptno=d.deptno
  and d.dname;
  EXEC SQL OPEN c1;
  ```

- an option for a user to enter two salary values. Then your application will return and display the department name, employee name, and salary value for all employees with salary value between the two given salary values.

  ```
  EXEC SQL DECLARE c1 CURSOR FOR
        SELECT  e.ename, e.sal , d.dname FROM emp e,dept d WHERE e.sal>sal_1 and
  e.sal<sal_2;
  EXEC SQL OPEN c1;
  ```

- an option for a user to enter two commission values. Then your application will return and display the department name, employee name, and salary value for all employees with salary value between the two given commission values.

  ```
  EXEC SQL DECLARE c1 CURSOR FOR
        SELECT  e.ename, e.sal , d.dname FROM emp e,dept d WHERE e.sal>comm_1
  and e.sal<comm_2;
  EXEC SQL OPEN c1;
  ```

STEP 3: The CGI scripting language as php or perl is used to code for the Web application:

EmployeeSearch.php

```
<html>
<head>
<title>ASSIGNMENT 2</title>
</head>

<body>
<h1>Employee Details Search</h1>

<?php

//create short variable names
$dept_name = $_POST['dept_name'];
$dept_no = $_POST['dept_no'];
$sal1 = $_POST['sal1'];
$sal2 = $_POST['sal2'];
$comm1 = $_POST['comm1'];
$comm2 = $_POST['comm2'];

if(!$dept_name || $dept_no || $sal1 || $sal2 || $comm1 || $comm2)
{
echo 'You have not entered search details please try again';
exit;
}

if(!get_magic_quotes_gpc())
```

```
{
$dept_name = addslashes($dept_name);
$dept_no = addslashes($dept_no);
$sal1 = addslashes($sal1);
$sal2 = addslashes($sal2);
$comm1 = addslashes($comm1);
$comm2 = addslashes($comm2);
}

@ $db = new mysqli('localhost','J_K201@CSDBORA','****','J_K201');

if(mysqli_connect_errno())
{
echo 'Error could not connect to database';
exit;
}

if($dept_no)
$query = "select ename,sal,comm from emp e,dept d where d.deptno = ".$dept_no"";

if($dept_name)
$query = "select ename,sal,comm from emp e,dept d where d.dname = ".$dept_name" and
d.deptno = e.deptno;

if($sal1 && $sal2)
$query = "select e.ename,e.sal,e.comm from emp e,dept d where e.sal>sal1 and e.sal<sal2;

if($comm1 && $comm2)
```

```
$query = "select e.ename,e.sal,d.dname from emp e,dept d where e.sal>e.comm1 and e.sal<e.comm2;


$result = $db->query($query);


$num_results = $result_rows;


echo "<p>Results found</p>";


for($i=0;$i<num_results;$i++)
{


$row = $result->fetch_assoc();
echo "<center>";
echo "<table border = 1>";
echo stripslashes($row);
echo "<br />";
}


$result->free();
$db->close();


?>
</body>
</html>
```

EmployeeSearch.pl

```
#!/usr/bin/perl
```

```
require "common.cgi";


sub  query_job

{

  local( $dept_name,$dept_no,$sal1,$sal2,$comm1,$comm2, $comfile);

  $dept_name = "$contents{dept_name}";

  $dept_no = "$contents{dept_no}";

  $sal1 = "$contents{sal1}";

  $sal2 = "$contents{sal2}";

  $comm1 = "$contents{comm1}";

  $comm2 = "$contents{comm2}";

  $comfile = "/home/Students/j_k201/public_html/demo/proc/unix-version/c++ \"$option\"
\"$value\"";

            $proc_returns = `$comfile`;

            return $proc_returns;

}


sub print_job

{

            print "<table border=1>\n";

            print "$query_return";

            print "</table>";

}


&get_parameters( *contents );

$query_return = &query_job;


print "Content-Type: text/html \n\n";

&print_header;
```

```
print "<center><h3>Employee Details based on search</h3></center>\n";
&print_job($query_return);
```