HDMT Code

File 1:

Hdmthtml.html : HTML file that provides the user interface.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <title>HDMT</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css">
#left {
  margin-right: 15px;
}
#right {
  margin-left: 15px;
}
#left_wrapper {
  width: 10%;
  height:1000px;
  float: left;
  position: relative;
  z-index: 0;
  background-color:#FFFFFF;
}
#middle_wrapper {
  width: 80%;
  height:1000px;
  margin-right: -150px;
  float: left;
  position: relative;
  z-index: 2;
  background-color:#E6E6FA;
```

```
}
#right_wrapper {
  width: 10%;
  height:1000px;
  float: right;
  position: relative;
  z-index: 1;
  background-color:#FFFFFF;
}
</style>
        <script language="javascript" type="text/javascript" src="./libs/js/jquery-1.11.1.js"></script>
        <script language="javascript" type="text/javascript" src="jsfunction.js"></script>
        <script language="javascript" type="text/javascript" src="js_ajaxfunction.js"></script>
        <link rel="stylesheet" href="//code.jquery.com/ui/1.11.2/themes/smoothness/jquery-ui.css">
        <script src="//code.jquery.com/ui/1.11.2/jquery-ui.js"></script>

        <!--for JSON TO HTML TABLE--->
        <script type="text/javascript" src="./libs/js/jquery.jsontotable.min.js"></script>
        <!--for HTML TABLE TO JSON --->
        <script type="text/javascript" src="./libs/js/jquery.tabletojson.min.js"></script>
        <script type="text/javascript" language="javascript">
        function initially()
        {
                document.getElementById('divid_migrate').style.visibility = 'hidden';
        }
        </script>
        <style>
        .jsontotable table, .jsontotable th, .jsontotable td {
```

```
                border: 1px solid black;
                margin: 10px;
                width:50%;
        }
 code {
   white-space: normal;
 }
        </style>
</head>
<body onload="initially();">
 <div id="left_wrapper">
   <div id="left"> <img src="hdmtlogo" height="50" width="100">
   </div>
 </div>
 <div id="middle_wrapper">
   <div id="middle">
        <center><h4>Welcome to Heterogeneous Data Migration Tool (HDMT) !</h4>
        <table border="2" style="border-collapse:collapse; border:solid 3px black;" cellpadding="5">
                <tr>
                        <th COLSPAN="2">SOURCE</th>
                        <th COLSPAN="2">DESTINATION</th>
                </tr>
                <tr>
                        <td>Server: IP Address</td>
                        <td><input type="text" value="172.16.6.23" id="txtid_ip1" name="txtnm_ip1" size="30">
                        <td>Server: IP Address</td>
                        <td><input type="text" value="172.16.6.25" id="txtid_ip2" name="txtnm_ip2" size="30">
                </tr>
                <tr>
                        <td>Server: Port No</td>
```

```
                    <td><input type="text" value="3306" id="txtid_port1"
name="txtnm_port1" size="30">
                    <td>Server: Port No</td>
                    <td><input type="text" value="5432" id="txtid_port2"
name="txtnm_port2" size="30">
                </tr>
                <tr>
                    <td>User Name</td>
                    <td><input type="text" value='root' name="txtusername1"
id="txtusername1" size="30"/></td>
                    <td>User Name</td>
                    <td><input type="text" value='root' name="txtusername2"
id="txtusername2" size="30"/></td>
                </tr>
                <tr>
                    <td>Password</td>
                    <td><input type="password" value='root'
name="txtpassword1" id="txtpassword1" size="30"/></td>
                    <td>Password</td>
                    <td><input type="password" value='root'
name="txtpassword2" id="txtpassword2" size="30"/></td>
                </tr>
                <tr>
                    <td>DBMS</td>
                    <td><select id="sdbtname1" name="sdbtname1" style="width:
220px">
                            <option>SELECT</option>
                            <option>MYSQL</option>
                            <option>PostgreSQL</option>
                            </select><input type="button" id="btnid_ip1"
value="Go">
                    </td>
                    <td>DBMS</td>
```

```
                    <td><select id="sdbtname2" name="sdbtname2" style="width:
220px">
                        <option>SELECT</option>
                        <option>MYSQL</option>
                        <option>PostgreSQL</option>
                        </select><input type="button" id="btnid_ip2"
value="Go">
                    </td>
                </tr>
                <tr>
                    <td>Database Name</td>
                    <td><select id="sdbname1" name="sdbname1" style="width:
220px">
                            <option>SELECT</option>
                        </select><input type="button" id="btnid_db1"
value="Go"></td>
                    <td>Database Name</td>
                    <td><select id="sdbname2" name="sdbname2" style="width:
220px">
                            <option>SELECT</option>
                        </select><input type="button" id="btnid_db2"
value="Go"></td>
                </tr>
                <tr>
                    <td>Table Name</td>
                    <td><select id="stbname1" name="stbname1" style="width:
220px">
                                <option>SELECT</option>

                            </select><input type="button"
id="btnid_table1" value="Go">
                    </td>
                    <td>Table Name</td>
```

```
                    <td><select id="stbname2" name="stbname2" style="width:
220px">
                                        <option>SELECT</option>

                                    </select><input type="button"
id="btnid_table2" value="Go">
                    </td>
                </tr>
            </table>
<br>
            <table>
                <tr>

                    <td valign="top">
                        <table bgcolor="#FFFFFF" border="1" id="tdisp1"
name="tdisp1" cellpadding="5">
                            <tr>
                                <td width = "215" height = "15"
bgcolor="#E6E6FA">Column Names</td>
                            </tr>
                        </table>
                    </td>
                    <td>---->>></td>
                    <td valign="top">
                        <table bgcolor="#FFFFFF" border="1" id="tdisp2"
name="tdisp2" cellpadding="5">
                            <tr>
                                <td width = "215" height = "15"
bgcolor="#E6E6FA">Column Names</td>
                            </tr>
                        </table>
                    </td>
                </tr>
            </table>
```

```
<br>
	<table border="0" width='95%' height="40" bgcolor='#B0C4DE'>
		<tr><td>Field Name : <select name='ddlnm_fldName' id='ddlid_fldName' width="100"></select></td>
			<td>Operator : <select name='ddlnm_opName' id='ddlid_opName'>
				<option>SELECT</option>
				<option>LIKE</option>
				<option>=</option><option>!=</option>
				<option><</option><option><=</option>
				<option>></option><option>>=</option>
			</select></td>
			<td>Value : <input Type="text" name='txtnm_fldVal' id='txtid_fldVal' size="10"></td>
			<td>Limit(Row From) : <input Type="text" name='txtnm_start' id='txtid_start' value="0" size="5">
			(Count) : <input Type="text" name='txtnm_cnt' id='txtid_cnt' size="5" value='10'>
		</td>
		</tr>
	</table>
<br>
<input type="button" value="View Selected Data" id="btnquery" name="btnquery"/>
<br><br>

<div id="divid_migrate" name="divnm_migrate">

<table border="0" width='95%' height="40" bgcolor='#B0C4DE'>
	<tr>
		<td>
		Select Action :
		</td>
		<td>
```

```
                <select id="s_action_id" name="s_action_name" style="width: 300px">
                        <option value='1'>1 - 'Check' Migration Status</option>
                        <option value='2'>2 - 'Add' to Migration Queue</option>
                        <option value ='3'>3 - 'View' Migration Queue</option>
                        <option value ='4'>4 - 'Process' Migration Queue</option>
                        <option value ='5'>5 - 'Clear' Migration Queue</option>
                </select>
                <input type="button" id="btnid_ok" value="OK">
                </td>
                <td>
                        <input type="checkbox" id="chkid_migrate_update" name = "chknm_migrate_update"> ON DUPLICATE KEY UPDATE RECORD
                </td>
        </tr>
</table>

<div id="dialog" title="Basic dialog">
<p>  <a href='migrate_data_to_selected_tables.php' target='_blank'>Migrate Data to Selected Table(s)</a></p>
</div>
        <hr>
</div>
<br><br>
        <div id="jsontotable" class="jsontotable">[Preview Table will appear here]</div>
        </center>
 </div>
</div>
```

```
<div id="right_wrapper">
  <div id="right">
  </div>
</div>
</body>
</html>
```

File 2:

js_ajaxfunction.js : Javascript file that uses jquery and ajax technology for updating part of the user interface screen as per user action. The data sent by the server side file is in JSON format.

```
var norow1, norow2;
//jquery function to  call json to return the array of databases 1
$(document).ready(function(){
   $('#btnid_ip1').click(function() {
   ip=$('#txtid_ip1').val();
         prtno=$('#txtid_port1').val();
         dbtech=$('#sdbtname1').val();
         username=$('#txtusername1').val();
         password=$('#txtpassword1').val();
         $.post("hdmtipaddrclass.php",
{ipaddress:ip,portno:prtno,dbt:dbtech,un:username,pwd:password})
         .done(function( data ){
         if(data.search("ERROR:")==2)
                  alert(data);
         else
                  update_dom(data,"#sdbname1");
         });
   });
});

//jquery function to call json to return the array of databases 2
$(document).ready(function(){
   $('#btnid_ip2').click(function() {
```

```
   ip=$('#txtid_ip2').val();
        prtno=$('#txtid_port2').val();
        dbtech=$('#sdbtname2').val();
        username=$('#txtusername2').val();
        password=$('#txtpassword2').val();
        $.post("hdmtipaddrclass.php",
{ipaddress:ip,portno:prtno,dbt:dbtech,un:username,pwd:password})
        .done(function( data ) {
        if(data.search("ERROR:")==2)
                alert(data);
        else
                update_dom(data,"#sdbname2");
        });
   });
});
//jquery function to assign the list of data in select control
function update_dom(data,ctrlname)
{
        $(ctrlname).empty();
        data=JSON.parse(data);
        for (i=0; i<data.length; i++)
        {
        $(ctrlname).append('<option>' + data[i] + '</option>');
        }
}
//jquery function to call json to return the array of table 1
$(document).ready(function(){
   $('#btnid_db1').click(function() {
        ip=$('#txtid_ip1').val();
        prtno=$('#txtid_port1').val();
   database=$('#sdbname1').val();
        username=$('#txtusername1').val();
        password=$('#txtpassword1').val();
        dbtech=$('#sdbtname1').val();
```

```
        $.post(                                                    "hdmtdbclass.php",
{ipaddress:ip,portno:prtno,dt:dbtech,db:database,un:username,pwd:password})
        .done(function( data ) {
        if(data.search("ERROR:")==2)
                alert(data);
        else
                update_dom(data,"#stbname1");
        });
   });
});
//jquery function to call json to return the array of table 2
$(document).ready(function(){
   $('#btnid_db2').click(function() {
        ip=$('#txtid_ip2').val();
        prtno=$('#txtid_port2').val();
      database=$('#sdbname2').val();
        username=$('#txtusername2').val();
        password=$('#txtpassword2').val();
        dbtech=$('#sdbtname2').val();
        $.post( "hdmtdbclass.php",
{ipaddress:ip,portno:prtno,dt:dbtech,db:database,un:username,pwd:password})
        .done(function( data ) {
        if(data.search("ERROR:")==2)
                alert(data);
        else
                update_dom(data,"#stbname2");
        });
   });
});
//jquery function to call json to return the array of column 1
$(document).ready(function(){
   $('#btnid_table1').click(function() {
        ip=$('#txtid_ip1').val();
        prtno=$('#txtid_port1').val();
```

```
        database=$('#sdbname1').val();
            table=$('#stbname1').val();
            username=$('#txtusername1').val();
            password=$('#txtpassword1').val();
            dbtech=$('#sdbtname1').val();
            $.post(                                                    "hdmttableclass.php",
{ipaddress:ip,portno:prtno,dt:dbtech,un:username,pwd:password,db:database,tab:table
})
            .done(function( data ) {
            if(data.search("ERROR:")==2)
                        alert(data);
            else
            {
                        data_arr = data.split("#");
                        data_col = JSON.parse(data_arr[0]);
                        data_type = JSON.parse(data_arr[1]);
                        cnt = data_col.length;
                        norow1=cnt;
                        genCols('tdisp1',cnt,data_col,data_type);
                        genFieldNames(cnt,data_col);
            }
            });
    });
});

//jquery function to call json to return the array of column 2
$(document).ready(function(){
    $('#btnid_table2').click(function() {
            ip=$('#txtid_ip2').val();
            prtno=$('#txtid_port2').val();
        database=$('#sdbname2').val();
            table=$('#stbname2').val();
            username=$('#txtusername2').val();
            password=$('#txtpassword2').val();
```

```
        dbtech=$('#sdbtname2').val();
        $.post( "hdmttableclass.php",
{ipaddress:ip,portno:prtno,dt:dbtech,un:username,pwd:password,db:database,tab:table
})
        .done(function( data ) {
        if(data.search("ERROR:")==2)
                alert(data);
        else
        {
                data_arr = data.split("#");
                data_col = JSON.parse(data_arr[0]);
                data_type = JSON.parse(data_arr[1]);
                cnt = data_col.length;
                norow2=cnt;
                genCols('tdisp2',cnt,data_col,data_type);
        }
        });
   });
});
//jquery function to call generatequery function on button click
$(document).ready(function(){
   $('#btnquery').click(function() {
        generatequery('tdisp1','tdisp2',norow1, norow2);
   });
});
```

File 3 :

jsfunction.js : Javascript file with jquery and ajax to respond to user action of selecting columns. Output is the data received in table.

```
var col_list2 = "";//declared global to complete select and insert action for destination
var col_list1 = "";//declared global to complete select and insert action for source
//function to generate the columns for a table with the values
function genCols(tablename,tab_noc,columnNames,columnType)
{
```

```
var i, strcol;
strcol=tab_noc;
//identify the table in case of more than once click on the button
var table = document.getElementById(tablename);
//clearing the table in case of more than once click on the button
for(var i = table.rows.length - 1; i > 0; i--)
{
	table.deleteRow(i);
}
// Find a <table> element with id="myTable":
var table = document.getElementById(tablename);
for (i=0;i<strcol;i++)
{
	// Create an empty <tr> element and add it to the 1st position of the table:
	var row = table.insertRow(i+1);
	// Insert new cells (<td> elements) at the 1st and 2nd position of the "new" <tr> element:
	var cell1 = row.insertCell(0);
	// Add some ddl to the new cells:
	var ddA = document.createElement("select");
	//assign the name and id of the dynamically added controls using the table name also
	ddA.name = "name1";
	ddA.id = tablename+"id1"+i;
	ddA.options[ddA.length] = new Option("SELECT", "0");
	ddA.style.width="210px";
	for (j=0;j<strcol;j++)
	{
		//adds the columns names in the specified table and drop down list
		ddA.options[ddA.length] = new Option(columnNames[j], j+1);
	}
	//Add the dropdown to the parent node
```

```
                cell1.appendChild(ddA);
        }
}//end of function genCols()
//function to populate the column names for where condition
function genFieldNames(option_cnt,option_data)
{
        var ddl=document.getElementById('ddlid_fldName');
        var length = ddl.options.length;
        while(ddl.options.length > 0){
   ddl.remove(0);}
        ddl.options[ddl.length] = new Option("SELECT","0");
        for (j=0;j<option_cnt;j++)
                {
                        //adds the columns names in the specified table and drop down
list
                        ddl.options[ddl.length] = new Option(option_data[j], j+1);
                }
}//end of function genFieldNames
//function to send the column values to generate a dynamic query on server side file
function generatequery(tabledisp1, tabledisp2, tab_noc1,tab_noc2)
{
        //declare variables for function scope only
        var selectcol1, selectcol2, strcolumn1, strcolumn2, strtype1;
        var strtable2=document.getElementById("stbname2").value;
        var strtable1=document.getElementById("stbname1").value;
        col_list2 = "";
        //call a function to compose the string for selected columns
        col_list2=get_column_string(tab_noc2, tabledisp2);
        col_list1 = "";
        //call a function to compose the string for selected columns
        col_list1 = get_column_string(tab_noc1, tabledisp1);
        //Reading the db 1 conneciton details
        ip1=$('#txtid_ip1').val();
        prtno1=$('#txtid_port1').val();
```

```
        database1=$('#sdbname1').val();
        username1=$('#txtusername1').val();
        password1=$('#txtpassword1').val();
        dbtech1=$('#sdbtname1').val();
        //Reading the db 2 conneciton details
        ip2=$('#txtid_ip2').val();
        prtno2=$('#txtid_port2').val();
        database2=$('#sdbname2').val();
        username2=$('#txtusername2').val();
        password2=$('#txtpassword2').val();
        dbtech2=$('#sdbtname2').val();
        //added parameter for horizental filteration of rows
        ffName = $('#ddlid_fldName option:selected').text();
        ffOp =  $('#ddlid_opName option:selected').text();
        ffVal =  $('#txtid_fldVal').val();
        recFrom =  $('#txtid_start').val();
        recLimit =  $('#txtid_cnt').val();
        //ajax technology to send data to php file
        $.post("hdmtinsertselect.php",
{queryaction:1,ff_name:ffName,ff_op:ffOp,ff_val:ffVal,rec_from:recFrom,rec_limit:r
ecLimit,fld_list1:col_list1,fld_list2:col_list2,tbl1:strtable1,tbl2:strtable2,ipaddress1:ip
1,
dt1:dbtech1,db1:database1,un1:username1,pwd1:password1,ipaddress2:ip2,dt2:dbtec
h2,db2:database2,un2:username2,pwd2:password2,portno1:prtno1,portno2:prtno2})
        .done(function(response){
        //call function to update dom or user form
        ////alert(response);
        if(response.search("ERROR:")==2)
                alert(response);
        else
                update_dom_table(response);
        //make the migrate button visible for further process only if jsontable is
created
```

```
        if (response) document.getElementById('divid_migrate').style.visibility = 'visible';
        });
}

//function to assign the records to a jsontable pulugin
function update_dom_table(response)
{
        var data = get_jsondataformat(response);
        //remove all from container div
        $("#jsontotable").html("");
        //'tblid_jsontbl' has been set as id in html table created through json plugin
        $.jsontotable(data, { id: '#jsontotable', header: true });
}

//function to convert string in json readable format
function get_jsondataformat(response)
{
        response = response.replace('["','[[');
        response = response.replace('"]', ']]');
        var matchesCount = response.split('","').length - 1;
        for (var i=0; i<matchesCount; i++)
                response=response.replace('","', '],[');
        matchesCount = response.split("'").length - 1;
        for (var i=0; i<matchesCount; i++)
                response=response.replace("'", "");
        return response;
}
//ajax function to capture the click on the dynamically created json table.
$(document).ready(function(){
        $(document.body).on('click', '#tblid_jsontbl tr', function(){
        $(this).css('background', 'white')
                        //use a javascript confirm control
```

```
                        if (confirm('Are you sure you want to remove selected row ?'))
{
                                $(this).remove();
                        } else {
                        $(this).css('background', 'lightgray')
                }
        });
});
$(document).ready(function(){
        $("#dialog").attr('visibility','hidden');
});

$(document).ready(function(){
                $("#btnid_ok").click(function(){
                selected_action = $('#s_action_id').val();
                var update_rec_json = $('#tblid_jsontbl').tableToJSON();
                var strtable2=document.getElementById("stbname2").value;

                var strtable1=document.getElementById("stbname1").value;
                if(selected_action == 4)
                {
                        $("#dialog").attr('display','block');
                        $("#dialog").dialog();
                        alert(selected_action);
                        exit;
                }
                update_rec_json=JSON.stringify(update_rec_json);
                //Reading the db conneciton details
                ip2=$('#txtid_ip2').val();
                database2=$('#sdbname2').val();
                username2=$('#txtusername2').val();
                password2=$('#txtpassword2').val();
                dbtech2=$('#sdbtname2').val();
                prtno2=$('#txtid_port2').val();
```

```
                //Reading the db conneciton details
                dbtech1=$('#sdbtname1').val();
                ip1=$('#txtid_ip1').val();
                database1=$('#sdbname1').val();
                username1=$('#txtusername1').val();
                password1=$('#txtpassword1').val();
                prtno1=$('#txtid_port1').val();
                //check for migration option
                chkmigrationstatus=$('#chkid_migrate').is(":checked");
                chkmigrationupdatestatus=$('#chkid_migrate_update').is(":checked");

                //ajax technology to send data to php file

        $.post("hdmtinsertselect.php",{check_m_u_status:chkmigrationupdatestatus,c
heck_m_status:selected_action,queryaction:2,update_json:update_rec_json,fld_list1:c
ol_list1,fld_list2:col_list2,tbl1:strtable1,tbl2:strtable2,
ipaddress1:ip1,ipaddress2:ip2,dt2:dbtech2,dt1:dbtech1,db1:database1,db2:database2,u
n1:username1,un2:username2,pwd1:password1,pwd2:password2,portno1:prtno1,port
no2:prtno2}).done(function(response){
                alert ("Migration Status - \n" + response);
                });
                //alert(selected_action);
        })
});

//function to migrate the data in html table to json and finally in the destination
database. DEPRICATED
$(document).ready(function(){
                $("#btnid_migrate").click(function(){
                var update_rec_json = $('#tblid_jsontbl').tableToJSON();
                var strtable2=document.getElementById("stbname2").value;

                var strtable1=document.getElementById("stbname1").value;
                update_rec_json=JSON.stringify(update_rec_json);
```

```
                    //Reading the db conneciton details
                    ip2=$('#txtid_ip2').val();
                    database2=$('#sdbname2').val();
                    username2=$('#txtusername2').val();
                    password2=$('#txtpassword2').val();
                    dbtech2=$('#sdbtname2').val();
                    prtno2=$('#txtid_port2').val();
                    //Reading the db conneciton details
                    dbtech1=$('#sdbtname1').val();
                    ip1=$('#txtid_ip1').val();
                    database1=$('#sdbname1').val();
                    username1=$('#txtusername1').val();
                    password1=$('#txtpassword1').val();
                    prtno1=$('#txtid_port1').val();
                    //check for migration option
                    chkmigrationstatus=$('#chkid_migrate').is(":checked");
                    chkmigrationupdatestatus=$('#chkid_migrate_update').is(":checked");

                    //ajax technology to send data to php file

        $.post("hdmtinsertselect.php",{check_m_u_status:chkmigrationupdatestatus,c
heck_m_status:chkmigrationstatus,queryaction:2,update_json:update_rec_json,fld_list
1:col_list1,fld_list2:col_list2,tbl1:strtable1,tbl2:strtable2,
ipaddress1:ip1,ipaddress2:ip2,dt2:dbtech2,dt1:dbtech1,db1:database1,db2:database2,u
n1:username1,un2:username2,pwd1:password1,pwd2:password2,portno1:prtno1,port
no2:prtno2}).done(function(response){
                    alert ("Migration Status - " + response);
                    });
        });
});
//function to compose the column string
function get_column_string(tab_noc, tabledisp)
{
        var strcol=tab_noc;
```

```
        var selectcol, strcolumn, col_list="";
        for (var i=0;i<strcol;i++)
        {
                selectcol = document.getElementById(tabledisp+'id1'+i);

                strcolumn = selectcol.options[selectcol.selectedIndex].innerHTML;
                //consider column names that do not start with the select keyword.
                if (strcolumn != 'SELECT')
                {
                        col_list=col_list+strcolumn;
                        col_list=col_list+",";
                }
        }
        //remove the trailing comma.
        col_list=col_list.substring(0,col_list.length-1);
        return col_list;
}//end of function
```

File 4 :

hdmtipaddressclass.php : PHP file to take server related input information and authenticate it and give the list of databases as json output.

```
<?php
//start of session
session_start();
//create object of class connect_to_server
$obj_ip = new connect_to_server();
//class connect_to_server
class connect_to_server
{
        //data members of connect_to_server for connecting to server
        var $ipaddress;
        var $portno;
        var $dbtech;
        var $username;
```

```
var $password;
//constructor
function connect_to_server()
{
        //to initialize values from session variables
        $this->ipaddress = $_POST['ipaddress'];
        $this->portno = $_POST['portno'];
        $this->dbtech = $_POST['dbt'];
        $this->username = $_POST['un'];
        $this->password = $_POST['pwd'];
        //function to check mysql or mssql server is used
        $this->check_dbtechnology();
}
//function to check mysql or mssql server is used
function check_dbtechnology()
{
        if($this->dbtech == "MYSQL")
        {
                //user defined function to connect to mysql
                $this->fconnect_mysql();
        }
        else if($this->dbtech == "PostgreSQL")
        {
                //user defined function to connect to mysql
                $this->fconnect_pgsql();
        }
        else
        {
                echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                return;
        }
}
//member function to connect to MySQL server
function fconnect_mysql()
```

```
        {
                // Create connection
                $con=mysql_connect($this->ipaddress . ":" . $this->portno,$this->username,$this->password) or die ("..ERROR: Could not connect to MySQL");
                // Check connection and exit in case of exception
                if (!$con)
                {
                        echo "..ERROR: Failed to connect to " . $this->dbtech . ": " . mysql_error();
                        exit;
                }
                else
                {
                        //call mysql_query function to get the list of databases and store in result
                        $result=mysql_query("SHOW DATABASES WHERE `Database` != 'information_schema'");
                        //iterate through the result
                        while($row=mysql_fetch_assoc($result))
                        {
                                //add the result to an array
                                $db_list[]=$row['Database'];
                        }
                        //encode the array using json for data exchange instead of xml
                        $db_list=json_encode($db_list);
                        //close connection
                        mysql_close($con);
                        //send the db list to the client side
                        echo $db_list;
                }
        }//end of function fconnect_mysql()
        //member function to connect to MS SQL server
        function fconnect_pgsql()
        {
```

```
                // Create connection
                $con = pg_connect("host= " . $this->ipaddress . " port = " . $this->portno . " dbname = postgres user= " . $this->username . " password= " . $this->password) or die ("..ERROR: Could not connect to PostgreSQL");
                // Check connection and exit in case of exception
                if (!$con)
                {
                        echo "..ERROR: Failed to connect to Database " . $this->dbtech . " error is : " . pg_last_error();
                        exit;
                }
                else
                {
                        //call pgsql_query function to get the list of databases and store in result
                        $result=pg_query($con, "SELECT datname FROM pg_database WHERE datistemplate = false;");
                        //iterate through the result
                        while($row=pg_fetch_assoc($result))
                        {
                                //add the result to an array
                                $db_list[]=$row['datname']; //datname as defined in pg_database
                        }
                        //encode the array using json for data exchange instead of xml
                        $db_list=json_encode($db_list);
                        //close connection
                        pg_close($con);
                        //send the db list to the client side
                        echo $db_list;
                }
        }//end of function fconnect_pgsql()
}//end of class connect_to_server
?>
```

File 5 :

hdmtdbclass.php : PHP file that takes the server and selected database information and gives the list of tables as json output.

```
<?php
//start of session
session_start();
//create object of class connect_to_server
$obj_db = new connect_to_database();
//class connect_to_server
class connect_to_database
{
	//data members of connect_to_server for connecting to database
	var $ipaddress;
	var $portno;
	var $dbtech;
	var $database;
	var $username;
	var $password;
	//constructor
	function connect_to_database()
	{
		//to initialize values from session variables
		$this->ipaddress = $_POST['ipaddress'];
		$this->dbtech = $_POST['dt'];
		$this->portno = $_POST['portno'];
		$this->database = $_POST['db'];
		$this->username = $_POST['un'];
		$this->password = $_POST['pwd'];
		//function to check mysql or mssql server is used
		$this->check_dbtechnology();
```

```
	}
	//function to check mysql or mssql server is used
	function check_dbtechnology()
	{
		if($this->dbtech == "MYSQL")
		{
			//user defined function to connect to mysql
			$this->fgettablelist_mysql();
		}
		else if($this->dbtech == "PostgreSQL")
		{
			//user defined function to connect to pgsql
			$this->fgettablelist_pgsql();
		}
		else
		{
			echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
			return;
		}
	}
	//member function to connect to MySQL server
	function fgettablelist_mysql()
	{
		// Create connection
		$con=mysql_connect($this->ipaddress . ":" . $this->portno,$this->username,$this->password) or die ("..ERROR: Could not connect to MySQL");
		// Check connection and exit in case of exception
		if (!$con)
		{
			echo "..ERROR: Failed to connect to " . $this->dbtech . ". Error is : " . mysql_error();
			exit;
		}
		else
```

```
                {
                        //call mysql_query function to get the list of tables and store in
result
                        $result = mysql_query("SHOW TABLES FROM " . $this-
>database );
                        //iterate through the result
                        while($row=mysql_fetch_row($result))
                        {
                                //add the result to an array

                                $table_list[]=$row[0];
                        }
                        //encode the array using json for data exchange instead of xml
                        $table_list=json_encode($table_list);
                        //close connection
                        mysql_close($con);
                        //send the db list to the client side
                        echo $table_list;
                }
        }//end of function

        //member function to connect to PostgreSQL server
        function fgettablelist_pgsql()
        {
                // Create connection for PostgreSQL
                $con = pg_connect("host= " . $this->ipaddress . " port = " . $this-
>portno . " dbname = " . $this->database . " user= " . $this->username . " password= "
. $this->password) or die ("..ERROR: Could not connect to PostgreSQL");
                // Check connection and exit in case of exception
                if (!$con)
                {
                        echo "..ERROR: Failed to connect to " . $this->dbtech . " .
Error is : " . pg_last_error();
                        exit;
```

```
                }
                else
                {
                        //call mysql_query function to get the list of tables and store in
result
                        $result = pg_query("SELECT table_name FROM
information_schema.tables WHERE table_schema = 'public';");
                        //iterate through the result
                        while($row=pg_fetch_row($result))
                        {
                                //add the result to an array

                                $table_list[]=$row[0];//echo $table_list;
                        }
                        //encode the array using json for data exchange instead of xml
                        $table_list=json_encode($table_list);
                        //close connection
                        pg_close($con);
                        //send the db list to the client side
                        echo $table_list;
                }
        }//end of function
} //end of class
?>
```

File 6 :

hdmttableclass.php : PHP file to take server, database and table information and to give the list of columns as json output.

```
<?php
session_start();
//create an object of connect_to_table
$obj_table = new connect_to_table();

class connect_to_table
```

```
{
	//data members that are used for connecting and fetching
	var $ipaddress;
	var $portno;
	var $dbtech;
	var $database;
	var $table;
	var $username;
	var $password;
	function connect_to_table()
	{
		//in the constructor to set the data member values using session values

		$this->ipaddress = $_POST['ipaddress'];
		$this->portno = $_POST['portno'];
		$this->dbtech = $_POST['dt'];
		$this->database = $_POST['db'];
		$this->table = $_POST['tab'];
		$this->username = $_POST['un'];
		$this->password = $_POST['pwd'];
		//call function to proceed to mysql or mssql server functions
		$this->check_dbtechnology();
	}
	function check_dbtechnology()
	{
		//check if mysql is selected
		if($this->dbtech == "MYSQL")
		{
			//call function to get the list of columns.
			$this->fgetcolumnlist_mysql();
		}
		//check if ms sql server is selected
		else if($this->dbtech == "PostgreSQL")
		{
```

```
                        //call function to get the list of columns.
                        $this->fgetcolumnlist_pgsql();
                }
                else
                {
                        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                        return;
                }
        }
        //function to display the columns in drop down list from the selected table.
        function fgetcolumnlist_mysql()
        {
                // Create connection
                $con=mysql_connect($this->ipaddress . ":" . $this->portno,$this-
>username,$this->password) or die ("..ERROR: Could not connect to MySQL");

                // Check connection
                if (!$con)
                {
                        echo "..ERROR: Failed to connect to " . $this->dbtech . ". Error
is : " . mysql_error();
                        exit;
                }
                else
                {
                        //connect to database using mysql function
                        mysql_select_db($this->database);
                        //use mysql function to get the result of a query
                        $result = mysql_query("SHOW COLUMNS FROM " . $this-
>table );
                        //iterate the result of query to get the list of column names and
its data type
                        while($row=mysql_fetch_row($result))
                        {
```

```
                        $column_list[]=$row[0];//column names
                        $column_type[]=$row[1];//column data type
                }
                //encode the array using json for data exchange instead of xml
                $column_list=json_encode($column_list);
                $column_type=json_encode($column_type);
                //close connection
                mysql_close($con);
                //send the db list to the client side
                echo $column_list."#".$column_type;
        }//end of database connection activity
}//end of function

function fgetcolumnlist_pgsql()
{
        // Create connection
        $con = pg_connect("host= " . $this->ipaddress . " port = " . $this->portno . " dbname = " . $this->database . " user= " . $this->username . " password= " . $this->password) or die ("..ERROR: Could not connect to PostgreSQL");
        // Check connection and exit in case of exception
        if (!$con)
        {
                echo "..ERROR: Failed to connect to " . $this->dbtech . ". Error is : " . pg_last_error();
                exit;
        }
        else
        {
                //query string to connect to postgres to get the columns of a table
                $sql="SELECT column_name, data_type FROM information_schema.columns WHERE table_name = '" . $this->table . "'";
                //use mysql function to get the result of a query
                $result = pg_query($con,$sql);
```

```
                        //iterate the result of query to get the list of column names and
its data type
                        while($row=pg_fetch_row($result))
                        {
                                $column_list[]=$row[0];//column names
                                $column_type[]=$row[1];//column data type
                        }
                        //encode the array using json for data exchange instead of xml
                        $column_list=json_encode($column_list);
                        $column_type=json_encode($column_type);
                        //close connection
                        pg_close($con);
                        //send the db list to the client side to be updated through ajax
                        echo $column_list."#".$column_type;
                }//end of database connection activity
        }//end of function
}//end of class
?>
```

File 7 :

datatype_compatible_config.php : PHP file that uses the concept of associative array for data type compatibility comparison. This file acts as a data repository which can be added with new meaningful data type associations.

```
<?php
/* This file is used for providing the data type compatibility information. All the
compatible data types will be having the same value
and data type as key ie data type name will be key and its value as integer value.

This file has been included in hdmtinsertselect.php file for the
check_datatype_compatibility() function. */
$arr_equivalent_datatype_set =
array("integer"=>"0","int"=>"0","number"=>"0","int2"=>"0","int4"=>"0",
```

```
"char"=>"1","varchar"=>"1","text"=>"1","string"=>"1","character"=>"1","blob"=>"1
","bpchar"=>"1",
                                        "date"=>"2","datetime"=>"2","year"=>"2",
                                        "real"=>"3"
                                            );
$arr_convertible_datatype_set                                                    =
array("integer"=>array("char","string","varchar","varchar2"),

        "int"=>array("char","string","varchar","varchar2"),

        "real"=>array("text","varchar","varchar2","blob","bpchar"),

        "string"=>array("text","varchar","varchar2","blob","bpchar"),

        "int4"=>array("char","string","varchar","varchar2"),

        "bpchar"=>array("text","varchar","varchar2","blob","string"),

        "date"=>array("text","varchar","varchar2","blob","bpchar")
                                            );
?>
```

File 8 :

hdmtinsertselect.php : PHP file that takes all connection parameter inputs from source and destination database. It selects records from source table to be migrated to destination table. The selected records are displayed in the html table. At this stage few records may be deleted if so desired. The remaining records in the html table is saved in session for migration of data to multiple tables. There are few sub services also to fetch the column data types for compatibility comparison.

```
<?php
//start of session
session_start();
```

```
error_reporting(0);
//create object of class
$obj_dml = new data_insert_select();
//class definition
class data_insert_select
{
	//data members for connecting to source and destination servers
	var $ipaddress1;
	var $portno1;
	var $dbtech1;
	var $database1;
	var $username1;
	var $password1;
	var $col_list1;
	var $tbl1;
	var $ipaddress2;
	var $portno2;
	var $dbtech2;
	var $database2;
	var $username2;
	var $password2;
	var $col_list2;
	var $tbl2;
	//data member for taking the json table data
	var $update_rec_json;
	//data member for identifying the insert or select query
	var $queryaction;
	//data member for getting migration status of check box.
	var $check_m_status;
	//constructor to initialize data members with session values sent from js through jquery
	function data_insert_select()
	{
		$this->ipaddress1 = $_POST['ipaddress1'];
```

```
                $this->portno1 = $_POST['portno1'];
                $this->dbtech1 = $_POST['dt1'];
                $this->database1 = $_POST['db1'];
                $this->username1 = $_POST['un1'];
                $this->password1 = $_POST['pwd1'];
                $this->col_list1 = $_POST['fld_list1'];                 //col      list      for
SELECT query
                $this->tbl1 = $_POST['tbl1'];

                $this->ipaddress2 = $_POST['ipaddress2'];
                $this->portno2 = $_POST['portno2'];
                $this->dbtech2 = $_POST['dt2'];
                $this->database2 = $_POST['db2'];
                $this->username2 = $_POST['un2'];
                $this->password2 = $_POST['pwd2'];
                $this->col_list2 = $_POST['fld_list2'];                 //col      list      for
INSERT query
                $this->tbl2 = $_POST['tbl2'];           //col list for INSET query

                $this->queryaction = $_POST['queryaction'];
                $this->check_m_status = $_POST['check_m_status'];
                $this->check_m_u_status = $_POST['check_m_u_status'];
                //call function to check the query action parameter.
                $this->check_queryaction();
        }//end of constructor
        //function to check the query action parameter for either select or insert action
        function check_queryaction()
        {
                if($this->queryaction == 1)
                        $this->select_records();
                else if ($this->queryaction == 2)
                {
                        $this->update_rec_json  =  $_POST['update_json'];  //get  the
jsonhtml table for insert into destination
```

```
                    if ($this->check_m_status==1)        //check        dataypte
compatibility
                    {
                            $this->check_datatype_compatibility();       //exited
from there, if datatype not compatble
                    }
                    else if ($this->check_m_status==2)   //add to migration queue
                    {
                            $this->compose_query();      //This function composes
the query and save them to session
                    }
                    else if ($this->check_m_status==3)   //add to migration queue
                    {
                            //display the queryies & connection details saved in
session
                            print_r($_SESSION);
                    }
                    else if ($this->check_m_status==4)   //Process    to    migration
queue
                    {
                            //display the queryies & connection details saved in
session

        echo("<script>window.location='http://www.continue.com'</script>");

                    }
                    else if ($this->check_m_status==5)   //clear migration queue
                    {
                            session_destroy();      //To reset/delete all session data
                            echo("Migration Queue 'destroyed' Successfully !!!");
                    }
            }
            else
```

```
                {
                        echo("..ERROR: Please click on [View Selected Data]");
                        return;
                }
        }// end of function check_queryaction()

        //function to select the source records from the available connection information.
        function select_records()
        {
                if ($this->dbtech1 == "MYSQL")
                $rs = $this->fetch_mysql_records();
                else if ($this->dbtech1 == "PostgreSQL")
                        $rs = $this->fetch_pgsql_records();
                else
                {
                        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                        return;
                }
                if(!$rs)
                {
                        echo("..ERROR: No column details selected !!");
                        return;
                }
                $num_rows = $this->get_number_records($rs);
                $colcnt = 0;
                //get col_list1 in array format to validate for duplicate columns
                $arrcol_list1  = explode(',', $this->col_list1);
                $count_col_list1  = count($arrcol_list1);
                $arrcol_list2  = explode(',', $this->col_list2);

                //validate for duplicate columns by calling check_duplicate_column($arrcol_list1)
```

```
            if($this->check_duplicate_column($arrcol_list1))    //exits in case on
duplicate columns
            {
                  echo("..ERROR: Duplicate columns in selected column list 1");
                  return;
            }
            if($this->check_duplicate_column($arrcol_list2))    //exits in case on
duplicate columns
            {
                  echo("..ERROR: Duplicate columns in selected column list 2");
                  return;
            }
            if(count($arrcol_list1)!=count($arrcol_list2))          //exits in case
unequal columns
            {
                  echo("..ERROR: Different number of fields selected in column
list 1 and 2 !");
                  return;
            }
            if ($this->dbtech2 == "MYSQL")
                  $arr_coltype = $this->fetch_mysql_destination();
            else if ($this->dbtech2 == "PostgreSQL")
                  $arr_coltype = $this->fetch_pgsql_destination();
            //json string with the col names of table1 & 2
            $col_list_json = $this->get_cols_json($count_col_list1);
            //returs 1D array/rows containing csv value of fields
            $arr_vals      =      $this->get_compose_string($rs,      $arr_coltype,
$count_col_list1,$col_list_json);
            //function call to return the data in format as per the json_html table

            $html_records = $this->display_records_htmltable($arr_vals);
            echo $html_records;
      }//end of function select_records()
      //function to return the number of rows in a result set
```

```
	function get_number_records($rs)
	{
		if ($this->dbtech1 == "MYSQL")
			$num_rows = mysql_num_rows($rs);
		else if ($this->dbtech1 == "PostgreSQL")
			$num_rows = pg_num_rows($rs);
		else
		{
			echo("..ERROR: Please select MYSQL and PostgreSQL only.");
			return;
		}
		return $num_rows;
	}//end of function get_number_records
	//function to get destination details for mysql
	function fetch_mysql_destination()
	{
		// Create connection
		$con2=$this->db_connect_mysql(2);
		//fetch columns data types
		$strquery = "SELECT " . $this->col_list2 . " FROM " . $this->tbl2 . " WHERE 1=2";
		mysql_select_db($this->database2,$con2) or die("Unable to select db2");
		$rs2 = mysql_query($strquery,$con2);
		if(!$rs2)
		{
			echo("..ERROR: No record ie column list selected (msg1)!!");
			return;
		}
		$arr_coltype = $this->get_column_types ($rs1);
		mysql_close($con2);
		return $arr_coltype;
	}//end of function fetch_mysql_destination()
```

```
	//function to get destination details for postgresql
	function fetch_pgsql_destination()
	{
		// Create connection
		$con2=$this->db_connect_pgsql(2);
		//fetch columns data types
		//$strquery = "SELECT " . $this->col_list2 . " FROM " . $this->tbl2 . "
WHERE 1=2";
		$strquery = "SELECT " . $this->col_list2 . " FROM " . $this->tbl2 . "
WHERE 1=2";
		//echo "..ERROR: " . $strquery;
		$rs = pg_query($con2,$strquery);
		if(!$rs)
		{
			echo("..ERROR: No record ie column list selected (msg2)  !!" .
pg_last_error());
			return;
		}
		$arr_coltype = $this->get_column_types ($rs);
		pg_close($con2);
		return $arr_coltype;
	}//end of function fetch_pgsql_destination()
	//function to fetch source records for mysql
	function fetch_mysql_records()
	{
		// Create connection
		$con1=$this->db_connect_mysql(1);
		mysql_select_db($this->database1,$con1) or die("..ERROR: Could not
connect to MYSQL");
		//Filter criteria
		$ff_name = $_POST['ff_name'];
		$ff_op = $_POST['ff_op'];
		$ff_val = $_POST['ff_val'];
```

```
                $rec_from = $_POST['rec_from'];
                $rec_limit = $_POST['rec_limit'];
                $strquery = "SELECT " . $this->col_list1 . " FROM " . $this->tbl1 ;
                //apply where clause if specified by user
                if(($ff_name !="SELECT") && ($ff_op !="SELECT") && ($ff_val
!==""))
                {
                        $where_clause = " WHERE $ff_name $ff_op '$ff_val'";
                        $strquery .= $where_clause;
                }

                //Apply the row limit.
                if(($rec_from !="") && ($rec_limit !=""))
                {
                        $limit_clause = " LIMIT $rec_from,$rec_limit";
                        $strquery .= $limit_clause;
                }
                $rs = mysql_query($strquery,$con1) or die("..ERROR: Could not
execute MYSQL query");
                mysql_close($con1);
                return $rs;
        }//end of function fetch_mysql_records()
        //function to fetch source records for pgsql
        function fetch_pgsql_records()
        {
                // Create connection
                $con1=$this->db_connect_pgsql(1);
                //selecting the column list from table with columns specified in
col_list1
                $strquery = "SELECT " . $this->col_list1 . " FROM " . $this->tbl1;
                //echo $strquery;
                $rs = pg_query($con1,$strquery);
                pg_close();
                return $rs;
```

```
        }//end of function fetch_pgsql_records()
        //function to check the selection of duplicate column names
        function check_duplicate_column($arrcol_list1)
        {
                $fld_cnt_before = count($arrcol_list1);
                $arrcol_list1  = array_unique($arrcol_list1);
                $fld_cnt_after = count($arrcol_list1);
                if($fld_cnt_before != $fld_cnt_after)
                        return 1;
                else
                        return 0;
        }//end of check_duplicate_column($arrcol_list1)
        //function returns array containg datatype of all the fields in the resultset
        function get_column_types ($result_id)
        {
                $info = array( );                       # create empty array
                if ($this->dbtech1 == "MYSQL" || $this->dbtech2 == "MYSQL")
                {
                        for($i=0;$i<mysql_num_fields($result_id);$i++)
                        $info[] = mysql_field_type($result_id,$i);
                }
                if ($this->dbtech1 == "PostgreSQL" || $this->dbtech2 ==
"PostgreSQL")
                {
                        for($i=0;$i<pg_num_fields($result_id);$i++)

                        $info[] = pg_field_type($result_id,$i);
                }
                //if else
                        //echo "..ERROR: Please select MYSQL or PostgreSQL only.";

                return ($info);
    }//end of get_column_types ($result_id)
```

```
        //function to fetch the column constraints as array
    function get_column_constraints ($result_id)
    {
                $info = array( ); //create empty array
      for($i=0;$i<mysql_num_fields($result_id);$i++)
        $info[] = mysql_field_flags($result_id,$i);
                //print_r ($info);
                return ($info);
    }//end of get_column_constraints
        //function to convert columns string into json compatible type
        function get_cols_json($count_col_list1)
        {
                $col_list1_arr = explode(",",$this->col_list1);
                $col_list2_arr = explode(",",$this->col_list2);
                for($i=0;$i<$count_col_list1;$i++)
                {
                        $col_list_arr[$i] = """ . $col_list1_arr[$i] . " -> " .
$col_list2_arr[$i] . """;
                }
                //convering single quoted array elevem to sq csv.
                $col_list_json = implode(",",$col_list_arr);
                return $col_list_json;
        }//end of function get_cols_json($count_col_list1)
        //function to encode the array values in json compatible format
        function display_records_htmltable($arr_vals)
        {
                $table_list=json_encode($arr_vals);
                return $table_list;
        }//end of function
        //function to compose the string for select query
        function get_compose_string($rs, $arr_coltype,
$count_col_list1,$col_list_json)
        {
                $arr_vals[] = $col_list_json;
```

```
                if ($this->dbtech1 == "MYSQL")
                {
                        while($row = mysql_fetch_array($rs))
                        {
                                $row_val_csv = "";
                                for($colcnt=0;$colcnt<$count_col_list1;$colcnt++)
                                {
                                        $row_val_csv .= "'". $row[$colcnt]."',";
        //json compatible
                                }
                        $row_val_csv = substr("$row_val_csv",0,-1);          //remove
last comman (,)
                        $arr_vals[] = $row_val_csv;
                        }
                }
                else if ($this->dbtech1 == "PostgreSQL")
                {
                        while($row = pg_fetch_array($rs))
                        {
                                $row_val_csv = "";
                                for($colcnt=0;$colcnt<$count_col_list1;$colcnt++)
                                {
                                        $row_val_csv .= "'". $row[$colcnt]."',";
        //json compatible
                                }
                        $row_val_csv = substr("$row_val_csv",0,-1);          //remove
last comman (,)
                        $arr_vals[] = $row_val_csv;
                        }
                }
                else
                {
                        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                        return;
```

```
                }
                return $arr_vals;
        }//end of get_compose_string($rs, $arr_coltype, $count_col_list1)
        //function called to insert records in mysql or postgresql
        function compose_query()
        {
                //echo($this->update_rec_json);
                $json_row = json_decode($this->update_rec_json,true);
                $row_count = count($json_row);
                $row_csv = array();
                $i=0;
                while ($i<$row_count)
                {
                        $row_csv[] = """. implode("','", $json_row[$i]) . """;
                        $i++;
                }
                $arr_vals = $row_csv;
                if ($this->dbtech2 == "MYSQL")
                        $this->compose_save_query_mysql($arr_vals);
        //call function to compose and save query details to session
                else if ($this->dbtech2 == "PostgreSQL")
                        $this->compose_save_query_pgsql($arr_vals);

        } // end of function insert_records()
        //function to insert data in case of mysql
        function compose_save_query_mysql($arr_vals)
        {
                // Create connection
                $con2=$this->db_connect_mysql(2);
                //Composing INSERT query with multiple values
                $i=0;
                $strquery = "INSERT INTO " . $this->tbl2 . " (" . $this->col_list2 . ")
VALUES ";
                $num_rows = count($arr_vals);
```

```
                while($i<$num_rows)
                {
                        $strquery .= "($arr_vals[$i]),";
                        $i++;
                }
                //remove last commas(,) from $strquery;
                $strquery = rtrim($strquery,',');
                if ($this->check_m_u_status=='true')
                {
                        $arrcol_list2  = explode(',', $this->col_list2);
                        $count_col_list2  = count($arrcol_list2);
                        $strqueryupdate = " ON DUPLICATE KEY UPDATE ";
                        $j=0;
                        while($j<$count_col_list2)
                        {
                                $strqueryupdate .= $arrcol_list2[$j] . "= VALUES(" .
$arrcol_list2[$j] . "),";
                                $j++;
                        }
                        $strqueryupdate = rtrim($strqueryupdate,',');
                        $strquery = $strquery . $strqueryupdate;
                }
                //saving to session
                if(isset($_SESSION['qNo']))
                        $_SESSION['qNo'] ++;
                else
                        $_SESSION['qNo'] = 0;
                $str_qNo = "q".$_SESSION['qNo'];
                $_SESSION[$str_qNo]['q'] = "$strquery";                 //query
                $_SESSION[$str_qNo]['db_server'] = $this->ipaddress2;
                $_SESSION[$str_qNo]['db_portno'] = $this->portno2;
                $_SESSION[$str_qNo]['db_usr'] = $this->username2;
                $_SESSION[$str_qNo]['db_pwd'] = $this->password2;
                $_SESSION[$str_qNo]['db_tech'] = $this->dbtech2;
```

```
                $_SESSION[$str_qNo]['db_name'] = $this->database2;

                echo("Query Added to Migration Queue successfully !!!\n Total in
Queue : ". ($_SESSION['qNo'] + 1));
        } // end of function insert_records_mysql()
        //function to insert data in case of pgsql
        function compose_save_query_pgsql($arr_vals)
        {
                // Create connection
                $con2=$this->db_connect_pgsql(2);
                //Composing INSERT query with multiple values
                $i=0;
                $strquery = "INSERT INTO " . $this->tbl2 . " (" . $this->col_list2 . ")
VALUES ";
                $num_rows = count($arr_vals);
                while($i<$num_rows)
                {
                        $strquery .= "($arr_vals[$i]),";
                        $i++;
                }
                //remove last commas(,) from $strquery;
                $strquery = rtrim($strquery,',');
                if ($this->check_m_u_status=='true')
                {
                        echo ("..ERROR: ON DUPLICATE KEY UPDATE facility is
not applicable for PostgreSQL.");
                        exit;
                }
                //saving to session
                if(isset($_SESSION['qNo']))
                        $_SESSION['qNo'] ++;
                else
                        $_SESSION['qNo'] = 0;
                $str_qNo = "q".$_SESSION['qNo'];
```

```
                $_SESSION[$str_qNo]['q'] = "$strquery";
                $_SESSION[$str_qNo]['db_server'] = $this->ipaddress2;
                $_SESSION[$str_qNo]['db_portno'] = $this->portno2;
                $_SESSION[$str_qNo]['db_usr'] = $this->username2;
                $_SESSION[$str_qNo]['db_pwd'] = $this->password2;
                $_SESSION[$str_qNo]['db_tech'] = $this->dbtech2;
                $_SESSION[$str_qNo]['db_name'] = $this->database2;

                echo("Query Added to Migration Queue successfully !!!\n Total in
Queue : ". ($_SESSION['qNo'] + 1));
                pg_close($con2);
        } // end of function insert_records_pgsql()
        //function to check the data type compatibility of source and destination
        function check_datatype_compatibility()
        {
                // Create two connections for source and destination location
                if($this->dbtech1=="MYSQL")
                        $con11=$this->db_connect_mysql(1);
                else if($this->dbtech1=="PostgreSQL")
                        $con1=$this->db_connect_pgsql(1);
                else
                        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                if($this->dbtech2=="MYSQL")
                        $con2=$this->db_connect_mysql(2);
                else if($this->dbtech2=="PostgreSQL")
                        $con2=$this->db_connect_pgsql(2);
                else
                        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                //fetch columns data types
                $strquery = "SELECT " . $this->col_list1 . " FROM " . $this->tbl1 . "
WHERE 1=2";
                if ($this->dbtech1=="MYSQL")
                        $rs1 = mysql_query($strquery,$con11) or die ("..ERROR: No
record ie column list selected (msg3) !!".mysql_error());
```

```
else if($this->dbtech1=="PostgreSQL")
        $rs1 = pg_query($con1,$strquery) or die ("..ERROR: No record ie column list selected (msg4)!!");
else
        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
//fetch columns data types
$strquery = "SELECT " . $this->col_list2 . " FROM " . $this->tbl2 . " WHERE 1=2";
if ($this->dbtech2=="MYSQL")
        $rs2 = mysql_query($strquery,$con2) or die ("..ERROR: No record ie column list selected (msg5)!!");
else if($this->dbtech2=="PostgreSQL")
        $rs2 = pg_query($con2,$strquery) or die ("..ERROR: No record ie column list selected (msg6)!!");
else
        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");

$arr_coltype1 = $this->get_column_types($rs1);
$arr_coltype2 = $this->get_column_types($rs2);
//count the number of elements in coltype array
$n = count($arr_coltype1);

///fetch arr_col_constaints. //
$constraint_arr1 = $this->get_column_constraints($rs1);
require_once("datatype_compatible_config.php");          //file contains the datacompatibility deffination
$flg = "";      //flag to show non-compatable data type
$equvalence_report = "\n Index \t Server1    \t Server2";
for($i=0;$i<$n;$i++)
{
        if($arr_equivalent_datatype_set["$arr_coltype1[$i]"] != $arr_equivalent_datatype_set["$arr_coltype2[$i]"])
                $flg .= $i.",";
```

```
                $equvalence_report .= "\n $i \t\t" . $arr_coltype1[$i] . "\t\t" .
$arr_coltype2[$i] ;                     }
        echo($equvalence_report);
        if($flg !="")   //if datatypes are not equivalent the checking for
compatibility/convertibility
        {
                echo("\n\n..WARNING: Un-equivalent datatype found at
INDEX  $flg  . !!! Checking for convertability of datatype!!!");
                $flg = "";      //flag to show non-compatable data type
                $convertable_report = "\n Index \t Server1    \t Server2";
                for($i=0;$i<$n;$i++)
                {
                        $sub_arr                                                =
$arr_convertible_datatype_set["$arr_coltype1[$i]"];
                        $x=in_array("$arr_coltype2[$i]",$sub_arr);
                        if($x!=1)       //means unconvertable datatype
                                $flg .= $i.",";
                $convertable_report .= "\n $i \t\t" . $arr_coltype1[$i] . "\t\t" .
$arr_coltype2[$i] ;                             }
                echo($convertable_report);
                if($flg !="")
                {
                        echo("\n\n..ERROR: UnCompatable datatype found at
INDEX  $flg  . Migration Process has been aborted !!!");
                        exit;
                }
                else
                {
                        echo("\n \n CONGRATULATIONS ! Data Type
Compatibility process has been successful and is OK for migration !!!\n\n");
                }
        }
        else
        {
```

```
                    echo("\n \n CONGRATULATIONS ! Data Type Verification process has been successful and is OK for migration !!!\n\n");
                }
        }
        //function to connect to mysql database. location specifies the source or destination details. function returns connection.
        function db_connect_mysql($location)
        {
                if ($location == 2)
                {
                        $con = mysql_connect($this->ipaddress2 . ":" . $this->portno2,$this->username2,$this->password2,true) or die ("..ERROR: Could not connect to MYSQL 2" . mysql_error());
                        mysql_select_db($this->database2,$con) or die("Unable to select db2");
                }
                else if ($location == 1)
                {
                        $con = mysql_connect($this->ipaddress1 . ":" . $this->portno1,$this->username1,$this->password1) or die ("..ERROR: Could not connect to MYSQL 1" . mysql_error());
                        mysql_select_db($this->database1,$con) or die("Unable to select db1");
                }
                else
                {
                        echo ("Invalid Server");
                        exit;
                }
                //check if connected
                if (!$con)
                {
                        echo "..ERROR: Failed to connect to server : " . $location . ". Error is : " . mysql_error();
```

```
                exit;
        }
        return $con;
}//end of function db_connect_mysql
//function to connect to postgresql database. location specifies the source or destination details. function returns connection.
function db_connect_pgsql($location)
{
        if ($location == 2)
        {
                $con = pg_connect("host=" . $this->ipaddress2 . " port=" . $this->portno2 . " dbname=" . $this->database2 . " user=" . $this->username2 . " password=" . $this->password2) or die ("..ERROR: Could not connect to PostgreSQL" . pg_last_error());
        }
        else if ($location == 1)
        {
                $con = pg_connect("host= " . $this->ipaddress1 . " port = " . $this->portno1 . " dbname= " . $this->database1 . " user= " . $this->username1 . " password= " . $this->password1) or die ("..ERROR: Could not connect to PostgreSQL" . pg_last_error());
        }
        else
        {
                echo ("Invalid Server");
                exit;
        }
        //check if connected
        if (!$con)
        {
                echo "..ERROR: Failed to connect to server : " . $location . ". Error is : " . pg_last_error();
                exit;
        }
```

```
                return $con;
        }//end of function db_connect_mysql
}       //end of class
?>
```

File 9 :

migrate_data_to_selected_tables.php : PHP file that either commits or rollbacks the insertion of data into the destination table(s). It also displays appropriate reports to DBA / GDBA on completion of the operation.

```
<?php
        session_start();
        if(isset($_SESSION['qNo'])=="")
        {
         echo("<body bgcolor='#E6E6FA'><h2>HDMT Data Migration Status Report</h2><br>There is NO data / table set to migrate !!!<br></body>");
         return;
        }
        $dbtech2 = $_SESSION['q0']['db_tech'];
        $noq=$_SESSION['qNo'];     //number of queries in session. It has been set in htmtinsertselect.php through session
        if ($dbtech2 == "MYSQL")
        {
                echo ("<html><body bgcolor='#E6E6FA'><center><h2>HDMT Data Migration Status Report</h2><table border='1'>");
                $curr_con = db_connect_mysql("q0");                 //fetch current connection.
                mysql_query("BEGIN");
                for($i=0;$i<=$noq;$i++)
                {
                        $queryId = "q".$i;
                        echo("<tr><td>Exeuting Query No : </td><td>" . ($i+1) . "</td></tr>");
                        $curr_query = $_SESSION[$queryId]['q'];   //fetch query connection.
```

```
                $db_name = $_SESSION[$queryId]['db_name'];
                mysql_select_db($db_name,$curr_con) or die("Unable to
select" . $db_name . ":" . mysql_error());
                mysql_query($curr_query,$curr_con);
                echo("<tr><td>Done Query No : </td><td>" . ($i+1) .
mysql_error());
                echo("</td></tr>");
                //check of last execution was successful or NOT.
                if(mysql_errno())
                 {
                        echo("Some error occured while migrating data to one
of the table...<br> Process is being ROLLBACK and aborted.....");
                        mysql_query("ROLLBACK");
                        exit;
                 }
        }//end of for loop
        mysql_query("COMMIT");    //apply the change to database tables;
        echo("</table></center></body>All the data migrated successfully to
destination tables..");
        session_destroy();
    }//end of if
    else if ($dbtech2 == "PostgreSQL")
    {
        echo ("<html><body bgcolor='#E6E6FA'><center><h2>HDMT Data
Migration Status Report</h2><table border='1'>");
        $curr_con = db_connect_pgsql("q0");          //fetch current
connection.
        pg_query("BEGIN");
        for($i=0;$i<=$noq;$i++)
        {
                $queryId = "q".$i;
                echo("<tr><td>Exeuting Query No : </td><td>" . ($i+1) .
"</td></tr>");
```

```
                $curr_query = $_SESSION[$queryId]['q'];   //fetch        query
connection.
                $db_name = $_SESSION[$queryId]['db_name'];
                //mysql_select_db($db_name,$curr_con) or die("Unable to
select db2 :" . mysql_error());
                pg_query($curr_query,$curr_con);
                echo("<tr><td>Done Query No : </td><td>" . ($i+1) .
mysql_error());
                echo("</td></tr>");
                //check of last execution was successful or NOT.
                if(pg_last_error())
                 {
                        echo("Some error occured while migrating data to one
of the table...<br> Process is being ROLLBACK and aborted.....");
                        pg_query("ROLLBACK");
                        exit;
                 }
        }//end of for loop
        pg_query("COMMIT");        //apply the change to database tables;
        echo("</table></center></body>All the data migrated successfully to
destination tables..");
        session_destroy();
    }//end of if
    else
        echo "Process Terminating. Please repeat the migration process.";
    //function to connect to mysql database. location specifies the source or
destination details. function returns connection.
    function db_connect_mysql($str_qNo)
    {
        $db_server = $_SESSION[$str_qNo]['db_server'];
        $db_portno = $_SESSION[$str_qNo]['db_portno'];
        $db_usr = $_SESSION[$str_qNo]['db_usr'];
        $db_pwd = $_SESSION[$str_qNo]['db_pwd'];
        $db_tech = $_SESSION[$str_qNo]['db_tech'];
```

```
		$con = mysql_connect($db_server . ":" . $db_portno,$db_usr,$db_pwd,true) or die ("..ERROR: Could not connect to MYSQL" . mysql_error());
		return $con;
	}//end of function
	//function to connect to postgresql database. location specifies the source or destination details. function returns connection.
	function db_connect_pgsql($str_qNo)
	{
		$db_server = $_SESSION[$str_qNo]['db_server'];
		$db_dbname = $_SESSION[$str_qNo]['db_name'];
		$db_portno = $_SESSION[$str_qNo]['db_portno'];
		$db_usr = $_SESSION[$str_qNo]['db_usr'];
		$db_pwd = $_SESSION[$str_qNo]['db_pwd'];
		$db_tech = $_SESSION[$str_qNo]['db_tech'];

		$con = pg_connect("host= " . $db_server . " port = " . $db_portno . " dbname= " . $db_dbname . " user= " . $db_usr . " password= " . $db_pwd) or die
		("..ERROR: Could not connect to PostgreSQL" . pg_last_error());

		return $con;
	}//end of function
?>
```