

```
<?php
```

```
/*
 * OWNER: Amit Yadav
 * Dated: 13th April 2010
 * This script matches the addresses from two different database tables
 * This script uses the "levenshtein" algorithm to calculate the difference
 * between the two strings, the lesser the distance the more exact the match is
 * if the distance calculated b/w the two strings is zero then there is an exact match
 *
 * The script needs the DB connection parameter to be passed through the command lines
 *
 * HOW TO CALL THIS SCRIPT FROM CLI
 * Use the command below to call this script
 * php fuzzy-match.php [host-name] [user-name] [password] [DB-port] [Database-Name]
 *
 */
set_time_limit(0);

$username = $argv[2];

$pass = (strtolower($argv[3]) == 'nil') ? '' : $argv[3];

$port = (strtolower($argv[4]) == 'nil') ? '' : $argv[4];

$host = ($port != 'nil') ? $argv[1] . ':' . $port : $argv[1];

$dbname = $argv[5];

$db = mysql_connect($host, $username, $pass);
```

```
$srmSiteRecords = array();

if($db){
    if(mysql_select_db($database)){
        //CHECK IF THE DUMMY SITE IS THERE IN THE DB
        $resultSet = mysql_query('SELECT id FROM sites WHERE id = 0');

        // IF DUMMY SITE NOT FOUND THEN INSERT A DUMMY SITE IN THE DATABASE
        if(!$resultSet){
            mysql_query("INSERT INTO sites SET id=0, address1='Unknown'");
        }

        // CHECK IF THE DUMMY COMPANY EXISTS
        $resultSet = mysql_query('SELECT id FROM companies WHERE id = 0');

        // IF THE DUMMY SITE DOES NOT EXISTS THEN INSERT THE DUMMY SITE
        if(!$resultSet){
            mysql_query("INSERT INTO companies SET id=0,
customer_name='Unknown'");
        }

        $resultSet = mysql_query('SELECT id, address1, address2, city, state, country,
zipcode FROM sites');

        // ADD ALL THE SITE ADDRESS TO AN ARRAY SO THAT WE NEED NOT HIT THE
DATABASE EVERY TIME
```

```
while ($rowSite = mysql_fetch_assoc($resultSet)) {  
    $siteStr = "";  
    $siteStr .= (trim($rowSite['address1']) != "")?  
trim(addslashes($rowSite['address1'])) . ' ': "  
    $siteStr .= (trim($rowSite['address2']) != "")?  
trim(addslashes($rowSite['address2'])) . ' ': "  
    $siteStr .= (trim($rowSite['city']) != "")? trim(addslashes($rowSite['city']))  
. ' ': "  
    $siteStr .= (trim($rowSite['state']) != "")?  
trim(addslashes($rowSite['state'])) . ' ': "  
    $siteStr .= (trim($rowSite['country']) != "")?  
trim(addslashes($rowSite['country'])) . ' ': "  
    $siteStr .= (trim($rowSite['zipcode']) != "")?  
trim(addslashes($rowSite['zipcode'])) : "  
    $srmSiteRecords[$rowSite['id']] = $siteStr;  
}
```

```
//GET ALL THE STUDENTS RECORDS THAT DO NOT HAVE A MATCHING SITE
```

```
$resultSet = mysql_query('SELECT id, division, department, address1, address2,  
city, state, country, zipcode FROM students' /* WHERE site_id IS NULL */);
```

```
while ($rowStudent = mysql_fetch_assoc($resultSet)) {  
  
    // INPUT STUDENT ADDRESS TO MATCH  
    $srmStudentRec = "";  
    $srmStudentRec .= (trim($rowStudent['address1']) != "")?  
trim(addslashes($rowStudent['address1'])) . ' ': "  
    $srmStudentRec .= (trim($rowStudent['address2']) != "")?  
trim(addslashes($rowStudent['address2'])) . ' ': "
```

```

                                $srmStudentRec .= (trim($rowStudent['city']) != '')?
trim(addslashes($rowStudent['city'])) . ' ' : '';

                                $srmStudentRec .= (trim($rowStudent['state']) != '')?
trim(addslashes($rowStudent['state'])) . ' ' : '';

                                $srmStudentRec .= (trim($rowStudent['country']) != '')?
trim(addslashes($rowStudent['country'])) . ' ' : '';

                                $srmStudentRec .= (trim($rowStudent['zipcode']) != '')?
trim(addslashes($rowStudent['zipcode'])) : '';

//MATCH THE STUDENT ADDRESS TO THE SITES ADDRESS
ARRAY CREATED AT THE START OF THE FILE

                                $keyMatched = get_matching_address($srmStudentRec,
                                $srmSiteRecords, $rowStudent['id']);

                                if(is_numeric($keyMatched)){

                                        mysql_query('UPDATE students SET site_id = ' .
                                $keyMatched . ' WHERE id = ' . $rowStudent['id'], $db);

                                }elseif($keyMatched == true){

                                        mysql_query("INSERT INTO sites SET company_id=0,
                                division=" . trim(addslashes($rowStudent['division'])) . ", department=" .
                                trim(addslashes($rowStudent['department'])) . ",address1=" .
                                trim(addslashes($rowStudent['address1'])) . ", address2=" . trim(addslashes($rowStudent['address2'])) .
                                ", city=" . trim(addslashes($rowStudent['city'])) . ", state=" . trim(addslashes($rowStudent['state'])) . ",
                                country=" . trim(addslashes($rowStudent['country'])) . ", zipcode=" .
                                trim(addslashes($rowStudent['zipcode'])));

                                        mysql_query('UPDATE students SET site_id = ' .
                                @mysql_insert_id() . ' WHERE id = ' . $rowStudent['id'] . ' AND site_id IS NULL', $db);

                                }else{
```

```
mysql_query('UPDATE students SET site_id = 0 WHERE
id = ' . $rowStudent['id'] . ' AND site_id IS NULL',$db);
    }
}

}else{
    echo 'No database with the name specified found';
}
}else{
    echo "Cannot establish connection to localhost";
}
```

```
function get_matching_address($srmStudentRec, $srmSiteRecords, $srmStudentRecId){
    //IF THE STUDENT ADDRESS IS NULL OR N/A THEN RETURN,
    //IN THIS CASE A DUMMY SITE IS ASSIGNED TO THE STUDENT
    if(trim($srmStudentRec) == " || $srmStudentRec == 'N/A'){
        return false;
    }

    // NO SHORTEST DISTANCE FOUND, YET
    $shortest = -1;
    $keyMatched = 0;
```

```
// loop through sites addresses to find the matching address
foreach ($srmSiteRecords as $srmSiteId => $srmSiteRecord) {

    // CALCULATE THE DISTANCE BETWEEN THE STUDENT ADDRESS,
    // AND THE SITES ADDRESS
    $lev = levenshtein($srmStudentRec, $srmSiteRecord);

    // CHECK FOR AN EXACT MATCH
    if ($lev == 0) {

        // CLOSEST ADDRESS IS THIS ONE (EXACT MATCH)
        $closest = $srmSiteRecord;
        $shortest = 0;
        $keyMatched = $srmSiteId;

        // BREAK OUT OF THE LOOP; WE'VE FOUND AN EXACT MATCH
        break;
    }

    // IF THIS DISTANCE IS LESS THAN THE NEXT FOUND SHORTEST
    // DISTANCE, OR IF A NEXT SHORTEST WORD HAS NOT YET BEEN FOUND
    if ($lev <= $shortest || $shortest < 0) {

        // SET THE CLOSEST MATCH, SHORTEST DISTANCE, AND THE MATCHED

SITE ID

        $closest = $srmSiteRecord;
        $shortest = $lev;
    }
}
```

```
        $keyMatched = $srmsiteld;
    }
}

if($keyMatched){ // A MATCH HAS BEEN FOUND FOR THE STUDENT ADDRESS

    $distance = $shortest;

    echo "Input word: $srmsStudentRec ==> ";

    if ($shortest == 0) {

        echo " $closest, $keyMatched\n\n";

        return $keyMatched;

    } else {

        echo " $closest?, $keyMatched\n\n";

        return $keyMatched;

    }

}

}else

    // NO MATCH FOUND FOR THE STUDENT ADDRESS IN THIS CASE A NEW SITE
ENTRY WILL BE CREATED

    // WITH THE ADDRESS AS OF THE STUDENT AND ASSIGNED TO THE STUDENT,
THE COMPANY WILL BE DUMMY

    // IN THIS CASE

    return true;

}

?>
```