

Convert MYSQL character set to utf8 and collation to utf8_general_ci

In my current project I had a situation where I had to convert all the database table's character set and collation to utf8_general_ci so to accomplish this task I had written a PHP script which will do it automatically. Here are some basic definitions on few terminologies that I think might help before going towards the script.

A **character set** is a set of symbols and encodings.

A **collation** is a set of rules for comparing characters in a character set.

Collation affects **string** sorting and it doesn't have any effect on numeric data types.

We can specify a collation or a character set only for string types, ie. the types **VARCHAR, CHAR, ENUM, SET, and TEXT types (TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT)**

Steps to change the character set(as well the collation) of database to UTF8(utf8_general_ci)

1. Before running the script for changing the DB collation and character set, please take backup of the current database.

Linux Command to take mysql DB back up:

```
$ mysqldump --host=localhost --user=[DB_USER] --password=[DB_PASSWORD] --database [DB_NAME] --opt --add-drop-table -Q > [/PATH_TO_STORE_THE_DUMP/SQL_File_Name]
```

2. Find list of tables in the DB.

Note: We can either change the character set and collation of each table in the database or we can also change only those tables in which the character set and collation does not match with the target one.

PHP script to find all the tables in the DB.

```
$tabs = array(); //Array that will hold the list of tables in the selected database

//To find the list of tables linked to the database
$query_list_tables = "SHOW TABLES";
$res = mysql_query($query_list_tables);
while (($row = mysql_fetch_row($res)) != null)
{
    $tabs[] = $row[0];
}
```

OR

```
//To find list of the tables for which the collation does not match

$query = "SHOW TABLE STATUS WHERE Collation != TARGET_COLLATION";
$string = mysql_query($query) or die(mysql_error());
while($result = mysql_fetch_array($string))
{
    $tabs[] = $result['Name'];
}
```

3. Drop the indexes except Primary Key

4. Change the non binary string data type (CHAR, VARCHAR, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT) to their equivalent binary datatype as per below.

CHAR → BINARY

TEXT → BLOB

TINYTEXT → TINYBLOB

MEDIUMTEXT → MEDIUMBLOB

LONGTEXT → LONGBLOB

VARCHAR → VARBINARY

Note: If we directly change the character set of column (**with nonbinary data type**) without converting to the equivalent Binary format, then it may be lossy when the column contains characters that are not in both character sets.

The main purpose of converting non binary data type to binary before changing the character set is to remove the existing character set information without performing any character conversion.

When we convert CHAR to Binary, the resulting values will be padded with 0x00 bytes at the end.

5. Convert the column to a nonbinary data type with the proper character set:

Note: As we know when CHAR is converted to Binary, the resulting values will be padded with 0x00 bytes at the end. So, For Binary data type after changing the character and collation, we need to trim the content to get the exact character value.

6. Change Default the collation and character set of the table to the target one.

8. Rebuild table indexes.

7. Change the Default collation and character set of the Database to the target one.

Here is the complete PHP code to convert the database collation that I have written, it seems pretty big as I have added comments to the code for more readability and I also have the code to write the queries executed to a log file for future reference which you can remove if its not needed in your case.

```
<?php
    /* This script will change the collation to "utf8_general_ci" and character
    set to "utf8"
        of the database including all the tables and fields.
        It is HIGHLY suggested to take a MySQL dump prior to running the script
    */

    //This file will store all the queries that has been executed for performing
    the above task
    $filename = "database_collation_query_".date("Y-m-d").".php";
    $f = fopen($filename, "a");

    //function to store the queries executed in the above file and also stores
    the mysql errors if any
    function DB_DETAIL($query, $fileobj)
    {
        //stores the mysql queries
        fwrite($fileobj, "\n");
        fwrite($fileobj, $query);
        //stores the mysql error
        if (mysql_errno())
```

```

        {
            fwrite($fileobj, "\n");
            fwrite($fileobj, mysql_error());
            echo "<b>Mysql Error: " . mysql_error() . "</b>\n";
        }
    }

$username = DB_USERNAME;
$password = DB_PASSWORD;
$db = DB_NAME;
$host = DB_HOST;

// collation you want to change it to:
$target_collate = "utf8_general_ci";
// character set of new collation:
$target_charset = "utf8";
echo "<pre>";

// Open a connection to a MySQL Server
$conn = mysql_connect($host, $username, $password);
//Sets the current active database on the server that's associated with the
specified link identifier
mysql_select_db($db, $conn);

//array that will hold the list of tables in the selected database
$tabs = array();

//step 2 -> to get the list of tables linked to the database
$query_list_tables = "SHOW TABLES";
$res = mysql_query($query_list_tables);
DB_DETAIL($query_list_tables, $f);
while (($row = mysql_fetch_row($res)) != null)
{
    $tabs[] = $row[0];
}

//List of tables where collation is not same as target collation
/* $query = "SHOW TABLE STATUS WHERE Collation != '$target_collate'";
$string = mysql_query($query) or die(mysql_error());
while($result = mysql_fetch_array($string))
{
    $tabs[] = $result['Name'];
}
*/

*/
//Change the collation of each table
foreach ($tabs as $table)
{
    //step 3 -> Drop the indexes except Primary Key
    $query_indexes = "SHOW index FROM {$table}";
    $res = mysql_query($query_indexes);
    DB_DETAIL($query_indexes, $f);
    $indicies = array();
    while (($row = mysql_fetch_array($res)) != null)
    {
        if ($row[2] != "PRIMARY")
        {

```

```

        $indices[] = array("Table" => $row[0],
                           "Non_unique" => $row[1],
                           "Key_name" => $row[2],
                           "Column_name" => $row[4],
                           "Index_type" => $row[10],
                           );
        $query_drop_indexes = "ALTER TABLE {$table} DROP INDEX
{$row[2]}";

        mysql_query($query_drop_indexes);
        DB_DETAIL($query_drop_indexes,$f);
    }
}

//query to get all the fields detail in a table
$query_full_fields = " SHOW FULL FIELDS FROM `{$table}` ";
$rs = mysql_query($query_full_fields);
DB_DETAIL($query_full_fields,$f);

$set = false;
while ($row = mysql_fetch_assoc($rs))
{
    // get the "NULL" property of the field
    if ($row['Null'] == 'YES')
    {
        $nullable = ' NULL ';
    }
    else
    {
        $nullable = ' NOT NULL';
    }

    // get the default value of the field
    if ($row['Default'] == 'NULL') {
        $default = " DEFAULT NULL";
    }
    else if ($row['Default'] != '')
    {
        $default = " DEFAULT
'".mysql_real_escape_string($row['Default'])."'";
    }
    else
    {
        $default = '';
    }

    // Does the field has any comments attribute or not
    if ($row['Comment'] != '') {
        $comment = " COMMENT
'".mysql_real_escape_string($row['Comment'])."'";
    }
    else
    {
        $comment = '';
    }

    //field type

```

```

        $type = $row['Type'];
        $name = $row['Field'];
        $field = mysql_real_escape_string($row['Field']);
        //change the collation of the field in varchar(integer) format
        if (preg_match("/^varchar\((\d+)\)$/i", $type, $mat))
        {
            $size = $mat[1];
            //convert cfield to equivalent binary format
            $query_change_equivalent_binary = "ALTER TABLE {$table}
MODIFY {$name} VARBINARY({$size})";
            mysql_query($query_change_equivalent_binary);
            DB_DETAIL($query_change_equivalent_binary,$f);

            //change the collation of the field and also convert the
            field to actual field type(non-binary format as previous)
            $query_update_collation = "ALTER TABLE `{$table}` CHANGE
`$field` `$field` $row[Type]
CHARACTER SET
$target_charset COLLATE
$target_collate
$nullable $default $comment";
            $string_update_collation =
mysql_query($query_update_collation);
            DB_DETAIL($query_update_collation,$f);

fwrite($f,"\n-----
-----\n");
        }
        //change the collation of the field in char(integer) format
        else if (preg_match("/^CHAR\((\d+)\)$/i", $type, $mat))
        {
            $size = $mat[1];
            //convert cfield to equivalent binary format
            $query_change_equivalent_binary = "ALTER TABLE {$table}
MODIFY {$name} BINARY({$size})";
            mysql_query($query_change_equivalent_binary);
            DB_DETAIL($query_change_equivalent_binary,$f);

            //change the collation of the field and also convert the
            field to actual field type(non-binary format as previous)
            $query_update_collation = "ALTER TABLE `{$table}` CHANGE
`$field` `$field` $row[Type]
CHARACTER SET
$target_charset COLLATE $target_collate $nullable $default $comment";
            $string_update_collation =
mysql_query($query_update_collation);
            DB_DETAIL($query_update_collation,$f);

            //while converting Binary to Char, the resulting values
            will be padded with 0x00 bytes at the end,
            //To remove these bytes
            $query_remove_blankspace = "UPDATE `{$table}` SET `$field` =
TRIM(TRAILING 0x00 FROM `$field`)";
            $string_remove_blankspace =
mysql_query($query_remove_blankspace);
            DB_DETAIL($query_remove_blankspace,$f);

```

```

fwrite($f, "\n-----\n");
-----\n");

}
//change the collation of the field in TINYTEXT format
else if (!strcasecmp($type, "TINYTEXT"))
{
    //convert cfield to equivalent binary format
    $query_change_equivalent_binary = "ALTER TABLE {$table}
MODIFY {$name} TINYBLOB";
    mysql_query($query_change_equivalent_binary);
    DB_DETAIL($query_change_equivalent_binary, $f);

    //change the collation of the field and also convert the
field to actual field type(non-binary format as previous)
    $query_update_collation = "ALTER TABLE `{$table}` CHANGE
`$field` `$field` $row[Type]
CHARACTER SET
$target_charset COLLATE $target_collate $nullable $default $comment";
    $string_update_collation =
mysql_query($query_update_collation);
    DB_DETAIL($query_update_collation, $f);

fwrite($f, "\n-----\n");
-----\n");

}
//change the collation of the field in MEDIUMTEXT format
else if (!strcasecmp($type, "MEDIUMTEXT"))
{
    //convert cfield to equivalent binary format
    $query_change_equivalent_binary = "ALTER TABLE {$table}
MODIFY {$name} MEDIUMBLOB";
    mysql_query($query_change_equivalent_binary);
    DB_DETAIL($query_change_equivalent_binary, $f);

    //change the collation of the field and also convert the
field to actual field type(non-binary format as previous)
    $query_update_collation = "ALTER TABLE `{$table}` CHANGE
`$field` `$field` $row[Type]
CHARACTER SET
$target_charset COLLATE $target_collate $nullable $default $comment";
    $string_update_collation =
mysql_query($query_update_collation);
    DB_DETAIL($query_update_collation, $f);

fwrite($f, "\n-----\n");
-----\n");

}
//change the collation of the field in LONGTEXT format
else if (!strcasecmp($type, "LONGTEXT"))
{
    //convert cfield to equivalent binary format
    $query_change_equivalent_binary = "ALTER TABLE {$table}

```

```

MODIFY {$name} LONGBLOB";
        mysql_query($query_change_equivalent_binary);
        DB_DETAIL($query_change_equivalent_binary,$f);

        //change the collation of the field and also convert the
field to actual field type(non-binary format as previous)
        $query_update_collation = "ALTER TABLE ` $table ` CHANGE
` $field ` ` $field ` $row[Type]
                                CHARACTER SET
$target_charset COLLATE $target_collate $nullable $default $comment";
        $string_update_collation =
mysql_query($query_update_collation);
        DB_DETAIL($query_update_collation,$f);

fwrite($f,"\n-----
-----\n");

    }
    //change the collation of the field in TEXT format
else if (!strcasecmp($type, "TEXT"))
    {
        //convert cfield to equivalent binary format
        $query_change_equivalent_binary = "ALTER TABLE {$table}
MODIFY {$name} BLOB";
        mysql_query($query_change_equivalent_binary);
        DB_DETAIL($query_change_equivalent_binary,$f);

        //change the collation of the field and also convert the
field to actual field type(non-binary format as previous)
        $query_update_collation = "ALTER TABLE ` $table ` CHANGE
` $field ` ` $field ` $row[Type]
                                CHARACTER SET
$target_charset COLLATE $target_collate $nullable $default $comment";
        $string_update_collation =
mysql_query($query_update_collation);
        DB_DETAIL($query_update_collation,$f);

fwrite($f,"\n-----
-----\n");

    }
    //change the collation of the field in ENUM format
else if (stristr($type,"enum"))
    {
        $query_update_collation = "ALTER TABLE ` $table ` CHANGE
` $field ` ` $field ` $row[Type]
                                CHARACTER SET
$target_charset COLLATE $target_collate $nullable $default $comment";
        $string_update_collation =
mysql_query($query_update_collation);
        DB_DETAIL($query_update_collation,$f);

fwrite($f,"\n-----
-----\n");

    }
    //change the collation of the field in ENUM format
else if (stristr($type,"set"))

```

```

        {
            $query_update_collation = "ALTER TABLE ` $table ` CHANGE
` $field ` ` $field ` $row[Type]
                                CHARACTER SET
$target_charset COLLATE $target_collate $nullable $default $comment";
            $string_update_collation =
mysql_query($query_update_collation);
            DB_DETAIL($query_update_collation,$f);

fwrite($f,"\n-----\n");
-----\n");
        }
    }

    // re-build indicies..
    foreach ($indicies as $index)
    {
        //build unique key
        if (!$index["Non_unique"])
        {
            $query_build_index = "CREATE UNIQUE INDEX
{$index["Key_name"]} ON {$table} ({$index["Column_name"]}";
            mysql_query($query_build_index);
            DB_DETAIL($query_build_index,$f);
        }
        //build full text index type
        else if($index["Index_type"] == 'FULLTEXT')
        {
            $query_build_index = "CREATE FULLTEXT INDEX
{$index["Key_name"]} ON {$table} ({$index["Column_name"]}";
            mysql_query($query_build_index);
            DB_DETAIL($query_build_index,$f);
        }
        //build only index type
        else
        {
            $query_build_index = "CREATE INDEX
{$index["Key_name"]} ON {$table} ({$index["Column_name"]}";
            mysql_query($query_build_index);
            DB_DETAIL($query_build_index,$f);
        }
    }
}

// set database charset
$query_database = "ALTER DATABASE {$db} DEFAULT CHARACTER SET
{$target_charset} COLLATE {$target_collate}";
mysql_query($query_database);
DB_DETAIL($query_database,$f);

mysql_close($conn);
?>

```

Please refer the links below which I also referred for more information:

→ <http://stackoverflow.com/questions/105572/a-script-to-change-all-tables-and-fields-to-the-utf-8-bin-collation-in-mysql>

- http://codex.wordpress.org/Converting_Database_Character_Sets
- <http://www.phpwact.org/php/i18n/utf-8/mysql>
- <http://dev.mysql.com/doc/refman/5.1/en/charset-conversion.html>