**Решение проблем с кодировкой**

Кодировка таблицы — UTF-8, кодировка нужных столбцов — UTF-8, кодировка в заголовках страницы — UTF-8, кодировка в браузере — UTF-8. Ну и конечно подвело промежуточное звено, кодировка соединения — cp1251. А все лишь потому, что не проверили идентичность настроек тестового и рабочего сервера БД. Глупая ошибка, но данных в БД уже полно и надо как-то их вернуть. Прогуглив все, что можно, было найдено несколько интересных sql-запросов, способы с шаманством кодировок в дампах, с помощью блокнота, и один php-скрипт, который отработав безошибочно, просто выдал те же иероглифы. Что то не помогло, что то просто не подходит.

Моё решение банальное, но я не нашел подобного на просторах сети.
Пишем маленький php скрипт перегона нужных таблиц в нужный формат.

<?php

define('BD\_HOST','localhost');

define('BD\_PASS','password');

define('BD\_LOGIN','login'); //доступ к БД

define('BD\_FROM','bd1'); //исходная БД

define('BD\_IN','bd2'); //конечная БД

$tables = array ( 'table1',

 'table2',

 'table3',

 'table4',

 'table5'); //список нужных таблиц для конвертирования, эти таблицы должны быть как в исходной, так и в конечной БД(в конечной, должны быть пустыми)

foreach ($tables as $table) {

 mysql\_connect(BD\_HOST,BD\_LOGIN,BD\_PASS);

 mysql\_select\_db(BD\_FROM);

 $query\_select\_from = "SELECT \* FROM $table;";

 $result = mysql\_query($query\_select\_from);

 mysql\_close(); //запросили все косячные данные и закрыли соединение

 mysql\_connect(BD\_HOST,BD\_LOGIN,BD\_PASS);

 mysql\_select\_db(BD\_IN);

 mysql\_query("set names utf8"); //открыли новое соединение, но уже указав все недостающие настройки, из за которых произошло такое недопущение

 while ($t = mysql\_fetch\_assoc($result)) {

 $values = '';

 $f = false;

 foreach($t as $val) {

 $p = ($f)?',':'';

 $f = true;

 $add = (is\_int($val))?"$val":"'$val'";

 $values.= "$p $add";

 };

 $insert\_in = "INSERT $table VALUES($values)";

 mysql\_query($insert\_in); //запись прочтенной строки, уже в нормальном соединении

 };

 mysql\_close(); //закрываем соединение, что бы открыть его уже для другой таблицы.

};

?>

Вот и все. Скрипт сделал именно то, что было нужно и писался за пару минут, поэтому могут быть удивления, что это же можно сделать более гибким и оптимизированным, а еще и ООП неплохо прикрутить бы.

2-й вариант решения проблемы

й

Источник <https://habr.com/ru/post/201802/>