```
<?php

# PHP MySQL to MySQLi migration
#       Redefines deprecated or missing mysql_ functions and
calls mysqli_ functions for PHP5.5+.

# notes
#       mysql constants are directly translated to mysqli, so the
actual value may differ
#       mysql_escape_string now takes the last used connection

# to check if an item is a (mysql) resource or a mysqli object
# note that all types of resources must get through as this is a
# generic replacement for is_resource()
function is_mysqli_or_resource($r) {
# get the type of the variable
switch(gettype($r)) {
# if it is a resource - could be mysql, file handle etc...
case 'resource':
return true;
# if it is an object - must be a mysqli object then
case 'object':
# is this an instance of mysqli?
if ($r instanceof mysqli) {
# make sure there is no connection error
return !($r->connect_error);
}
# or is this an instance of a mysqli result?
if ($r instanceof mysqli_result) {
return true;
}
return false;
# negative on all other variable types
default:
return false;
}
}

# alias for is_mysqli_or_resource()
function is_mysql_resource($r) {
return is_mysqli_or_resource($r);
}

# alias for is_mysqli_or_resource()
function is_generic_resource($r) {
return is_mysqli_or_resource($r);
}

# to check if an item is a resource/object - replace is_resource
with this
# old version, this will break if testing file handles too
function is_mysql_resource_old($result) {

# first try to treat as resource if original mysql is loaded
if (extension_loaded('mysql')) return is_resource($result);

# or if mysqli is loaded, try to check object
if (extension_loaded('mysqli')) return is_object($result);

die('Fatal error, mysqli extension not loaded.');
}

# only do this is mysql extension is not there
if (!extension_loaded('mysql')) {

# check if mysqli extension is loaded - its required as we rely
on it
if (!extension_loaded('mysqli')) die('Fatal error, mysqli
extension not loaded.');

# --- helper variables and constants ----------------------------
------------------------------------------------------------------
------

# a list of connections, used to get the last one
$mysql_links = array();

# our own constants to reach default connection values in INI
file
define('MYSQL_DEFAULT_HOST', ini_get("mysql.default_host"));
define('MYSQL_DEFAULT_USER', ini_get("mysql.default_user"));
define('MYSQL_DEFAULT_PASSWORD',
ini_get("mysql.default_password"));

# --- MySQL constants (from PHP.net) ----------------------------
------------------------------------------------------------------
------

# MySQL client constants
define('MYSQL_CLIENT_COMPRESS', MYSQLI_CLIENT_COMPRESS);
# Use compression protocol
define('MYSQL_CLIENT_IGNORE_SPACE', MYSQLI_CLIENT_IGNORE_SPACE);
# Allow space after function names
define('MYSQL_CLIENT_INTERACTIVE', MYSQLI_CLIENT_INTERACTIVE);
```

```
# Allow interactive_timeout seconds
# (instead of wait_timeout ) of
# inactivity before closing the connection.
define('MYSQL_CLIENT_SSL', MYSQLI_CLIENT_SSL);
# Use SSL encryption. This flag is only
# available with version 4.x of the MySQL
# client library or newer. Version 3.23.x is
# bundled both with PHP 4 and Windows binaries
# of PHP 5.

# mysql_fetch_array() uses a constant for the different types of
result arrays. The following constants are defined:

# MySQL fetch constants
define('MYSQL_ASSOC', MYSQLI_ASSOC);    # Columns are returned
into the array having
# the fieldname as the array index.

define('MYSQL_BOTH', MYSQLI_BOTH);      # Columns are returned
into the array having
# both a numerical index and the fieldname as
# the array index.

define('MYSQL_NUM', MYSQLI_NUM);        # Columns are returned
into the array having a
# numerical index to the fields. This index
# starts with 0, the first field in the result.

# --- helper functions ----------------------------------------
------------------------------------------------------------

# internal function to convert bitflags of mysqli to flags in
text of mysql
# thanks to andre at koethur dot de at
http://www.php.net/manual/en/mysqli-result.fetch-
fields.php#101828
function mysql_field_bitflags_to_flags($flags_num) {

$flags = array();
$constants = get_defined_constants(true);
foreach ($constants['mysqli'] as $c => $n) {
if (preg_match('/MYSQLI_(.*)_FLAG$/', $c, $m)) {
if (!array_key_exists($n, $flags)) {
$flags[$n] = $m[1];
}
}
}

$result = array();
foreach ($flags as $n => $t) {
if ($flags_num & $n) {
$result[] = $t;
}
}
return implode(' ', $result);
}

# function to convert bit-types of mysqli to types in text of
mysql
# thanks to andre at koethur dot de at
http://www.php.net/manual/en/mysqli-result.fetch-
fields.php#101828
function mysql_field_bittypes_to_types($type_id) {

$types = array();
$constants = get_defined_constants(true);
foreach ($constants['mysqli'] as $c => $n) {
if (preg_match('/^MYSQLI_TYPE_(.*)/', $c, $m)) {
$types[$n] = $m[1];
}
}

return array_key_exists($type_id, $types)? $types[$type_id] :
NULL;
}

# lib helper function - to ensure mysql link as mysqli always
needs one
# but mysql takes last one
function mysql_ensure_link($link_identifier) {
# no link specified
if ($link_identifier === NULL) {
global $mysql_links;

# no connection at all - then go null
if (!count($mysql_links)) return NULL;

# get the last item of the array
$last = end($mysql_links);

# return the last stored link
return $last['link'];
}

return $link_identifier;
```

```
}

# --- MySQL functions (from PHP.net) ----------------------------
------------------------------------------------------------------
------

# mysql_affected_rows - Get number of affected rows in previous
MySQL operation
# int mysql_affected_rows ([ resource $link_identifier = NULL ] )
# int mysqli_affected_rows ( mysqli $link )
function mysql_affected_rows($link_identifier = NULL) {
# mysql_affected_rows = -1 if the last query failed
# mysqli_affected_rows = -1 indicates that the query returned an
error
$temp =
mysqli_affected_rows(mysql_ensure_link($link_identifier));
if ($temp === NULL || $temp === false) {
return -1;
}
return $temp;
}

# mysql_client_encoding - Returns the name of the character set
# string mysql_client_encoding ([ resource $link_identifier =
NULL ] )
# mysqli_character_set_name ( mysqli $link )
function mysql_client_encoding($link_identifier = NULL) {
# note that mysqlI_client_encoding ALSO is deprecated, so we
cannot use that
# mysql_client_encoding/mysqli_character_set_name = Returns the
default character set name for the current connection.
$temp =
mysqli_character_set_name(mysql_ensure_link($link_identifier));
if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_close - Close MySQL connection
# bool mysql_close ([ resource $link_identifier = NULL ] )
# bool mysqli_close ( mysqli $link )
function mysql_close($link = NULL) {

# mysql_close/mysqli_close = returns TRUE on success or FALSE on
failure.

global $mysql_links;
$link = mysql_ensure_link($link);

$thread_id = isset($link->thread_id) && is_numeric($link-
>thread_id) ? $link->thread_id : false;

$result = mysqli_close($link);

# did the removal suceed and and we have thread id
if ($result && $thread_id) {
# walk the links
foreach ($mysql_links as $k => $v) {

# does this thread-id match the one we just removed?
if ($v['thread_id'] === $thread_id) {
# then remove it from connection array
array_splice($mysql_links, $k, 1);
break;
}
}

# when connection already has been closed this error appears:
# Couldn't fetch mysqli in mysql.php on line xxx
# and this gives null instead of false
} else if ($result === null) {
return false;
}

return $result;
}

# mysql_connect - Open a connection to a MySQL Server
# resource mysql_connect ([ string $server =
ini_get("mysql.default_host") [, string $username =
ini_get("mysql.default_user") [, string $password =
ini_get("mysql.default_password") [, bool $new_link = false [,
int $client_flags = 0 ]]]]] )
# mysqli mysqli_connect ([ string $host =
ini_get("mysqli.default_host") [, string $username =
ini_get("mysqli.default_user") [, string $passwd =
ini_get("mysqli.default_pw") [, string $dbname = "" [, int $port
= ini_get("mysqli.default_port") [, string $socket =
ini_get("mysqli.default_socket") ]]]]]] )
function mysql_connect($server = MYSQL_DEFAULT_HOST, $username =
MYSQL_DEFAULT_USER, $password = MYSQL_DEFAULT_PASSWORD, $new_link
= false, $client_flags = 0) {
global $mysql_links;
```

```
# no newlink but s/u/p matches prev ones-take last link
if (!$new_link) {
global $mysql_links;

# are there prev links?
if (count($mysql_links)) {

# get the last one made
$last = end($mysql_links);

# does the s/u/p match last one?
if ($server === $last['server'] && $username ===
$last['username'] && $password === $last['password'] &&
is_resource($last['link'])) {
# then take that
return mysql_ensure_link(NULL);
}
}
}

# try to connect using current credentials
$link = mysqli_connect($server,$username,$password,"");

if (mysqli_connect_errno()) {
# printf("Connect failed: %s\n", mysqli_connect_error());
# die();
return false;
}

# store this
$mysql_links[] = array(
'thread_id' => $link->thread_id,
'server' => $server,
'username' => $username,
'password' => $password,
'link' => $link
);

return $link;
}

# mysql_create_db - Create a MySQL database
# bool mysql_create_db ( string $database_name [, resource
$link_identifier = NULL ] )
# CREATE DATABASE
function mysql_create_db($database_name, $link_identifier = NULL)
{
# mysql_create_db/mysql_query+CREATE DATABASE = false on error
return mysql_query('CREATE DATABASE
'.mysql_real_escape_string($database_name), $link_identifier);
}

# mysql_data_seek - Move internal result pointer
# bool mysql_data_seek ( resource $result , int $row_number )
# bool mysqli_data_seek ( mysqli_result $result , int $offset )
function mysql_data_seek($result , $row_number) {
# mysql_data_seek/mysqli_data_seek = false on error
$temp = mysqli_data_seek($result, $row_number);
if ($temp === NULL) {
return false;
}
return true;
}

# mysql_db_name - Retrieves database name from the call to #
mysql_list_dbs
# string mysql_db_name ( resource $result , int $row [, mixed
$field = NULL ] )
# SELECT DATABASE()
function mysql_db_name($result , $row, $field = NULL) {
# return mysql_query('SELECT DATABASE()',
mysql_ensure_link($link_identifier));

# null does not fit mysql_result
$field = $field === null ? 0 : $field;

return mysql_result($result, $row, $field);
}

# mysql_db_query - Selects a database and executes a query on it
# resource mysql_db_query ( string $database , string $query [,
resource $link_identifier = NULL ] )
# mysqli_select_db() then the query
function mysql_db_query($database, $query, $link_identifier =
NULL) {
# mysql_db_query = false on error, mysql_query+sql = false on
error
if (mysql_select_db($database, $link_identifier) !== true) {
return false;
}
return mysql_query($query, $link_identifier);
}
```

```
# mysql_drop_db - Drop (delete) a MySQL database
# bool mysql_drop_db ( string $database_name [, resource
$link_identifier = NULL ] )
# DROP DATABASE
function mysql_drop_db($database_name, $link_identifier = NULL) {
# mysql_drop_db = false on error, mysql_query+DROP DATABASE =
false on error
return mysql_query('DROP DATABASE
'.mysql_real_escape_string($database_name), $link_identifier);
}

# mysql_errno -Returns the numerical value of the error message
from previous MySQL operation
# int mysql_errno ([ resource $link_identifier = NULL ] )
# int mysqli_errno ( mysqli $link )
function mysql_errno($link_identifier = NULL) {
# mysql_errno/mysqli_errno = returns a number, 0 if no error
$temp = mysqli_errno (mysql_ensure_link($link_identifier));
if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_error - Returns the text of the error message from
previous MySQL operation
# string mysql_error ([ resource $link_identifier = NULL ] )
# string mysqli_error ( mysqli $link )
function mysql_error($link_identifier = NULL) {
# mysql_error/mysqli_error = returns empty string on no error
$temp = mysqli_error(mysql_ensure_link($link_identifier));
if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_escape_string - Escapes a string for use in a #
mysql_query
# string mysql_escape_string ( string $unescaped_string )
# string mysqli::real_escape_string ( string $escapestr )
function mysql_escape_string($unescaped_string) {
# mysql_escape_string = returns the escaped string
# mysql_real_escape_string = returns FALSE on error
return mysql_real_escape_string($unescaped_string);
}

# mysql_fetch_array - Fetch a result row as an associative array,
a numeric array, or both
# array mysql_fetch_array ( resource $result [, int $result_type
= MYSQL_BOTH ] )
# mixed mysqli_fetch_array ( mysqli_result $result [, int
$resulttype = MYSQLI_BOTH ] )
function mysql_fetch_array($result, $result_type = MYSQL_BOTH) {

# mysql_fetch_array = Returns an array of strings that
corresponds to the fetched row, or FALSE if there are no more
rows
# mysqli_fetch_array = Returns an array of strings that
corresponds to the fetched row or NULL if there are no more rows
in resultset

# store the result in a temporarily array
$temp = mysqli_fetch_array($result, $result_type);

# is the result null?
if ($temp === NULL) {
# then return false as the old function did
return false;
}
return $temp;
}

# mysql_fetch_assoc - Fetch a result row as an associative array
# array mysql_fetch_assoc ( resource $result )
# array mysqli_fetch_assoc ( mysqli_result $result )
function mysql_fetch_assoc ($result) {
# mysql_fetch_assoc = returns FALSE if there are no more rows
# mysqli_fetch_assoc = returns NULL if there are no more rows in
resultset
$temp = mysqli_fetch_assoc($result);

# is the result null?
if ($temp === NULL) {
# then return false as the old function did
return false;
}
return $temp;
}

# mysql_fetch_field - Get column information from a result and
return as an object
# object mysql_fetch_field ( resource $result [, int
$field_offset = 0 ] )
```

```
# object mysqli_fetch_field ( mysqli_result $result ) - but
field_offset is missing
function mysql_fetch_field($result, $field_offset = NULL) {
# if field offset is specified
if (is_numeric($field_offset)) {
# then seek to that
mysqli_field_seek($result, $field_offset);
}
$temp = mysqli_fetch_field($result);

if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_fetch_lengths - Get the length of each output in a result
# array mysql_fetch_lengths ( resource $result )
# array mysqli_fetch_lengths ( mysqli_result $result )
function mysql_fetch_lengths($result) {
# mysql_fetch_lengths/mysqli_fetch_lengths = FALSE on error
$temp = mysqli_fetch_lengths($result);
if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_fetch_object - Fetch a result row as an object
# object mysql_fetch_object ( resource $result [, string
$class_name [, array $params ]] )
# object mysqli_fetch_object ( mysqli_result $result [, string
$class_name [, array $params ]] )
function mysql_fetch_object ($result, $class_name=NULL,
$params=NULL) {

# mysql_fetch_object = FALSE if there are no more rows
# mysqli_fetch_object = NULL if there are no more rows in
resultset

if ($class_name !== NULL && $params !== NULL) {
$temp = mysqli_fetch_object($result, $class_name, $params);
} else if ($class_name !== NULL) {
$temp = mysqli_fetch_object($result, $class_name);
} else {
$temp = mysqli_fetch_object($result);
}

# is the result null?
if ($temp === NULL) {
# then return false as the old function did
return false;
}
return $temp;
}

# mysql_fetch_row - Get a result row as an enumerated array
# array mysql_fetch_row ( resource $result )
# mixed mysqli_fetch_row ( mysqli_result $result )
function mysql_fetch_row ($result) {

# mysql_fetch_row = FALSE if there are no more rows
# mysqli_fetch_row = NULL if there are no more rows in result set

$temp = mysqli_fetch_row($result);

# is the result null?
if ($temp === NULL) {
# then return false as the old function did
return false;
}
return $temp;
}

# mysql_field_flags - Get the flags associated with the specified
field in a result
# string mysql_field_flags ( resource $result , int $field_offset
)
# mysqli_fetch_field_direct() [flags]
# -> object mysqli_fetch_field_direct ( mysqli_result $result ,
int $fieldnr )
function mysql_field_flags($result, $field_offset) {
# mysql_field_flags = FALSE on failure
# mysqli_fetch_field_direct = FALSE if no field information for
specified fieldnr is available
$tmp = mysqli_fetch_field_direct($result, $field_offset);
if (!is_object($tmp)) return false;
$tmp = (array)$tmp;
return isset($tmp['flags']) ?
mysql_field_bitflags_to_flags($tmp['flags']) : false;
}

# mysql_field_len - Returns the length of the specified field
# int mysql_field_len ( resource $result , int $field_offset )
```

```
# mysqli_fetch_field_direct() [length]
# -> object mysqli_fetch_field_direct ( mysqli_result $result ,
int $fieldnr )
function mysql_field_len($result, $field_offset) {
# mysql_field_len = FALSE on failure
# mysqli_fetch_field_direct = FALSE if no field information for
specified fieldnr is available
$tmp = mysqli_fetch_field_direct($result, $field_offset);
if (!is_object($tmp)) return false;
$tmp = (array)$tmp;
return isset($tmp['length']) ? $tmp['length'] : false;
}

# mysql_field_name - Get the name of the specified field in a
result
# string mysql_field_name ( resource $result , int $field_offset
)
# mysqli_fetch_field_direct() [name] or [orgname]
# -> object mysqli_fetch_field_direct ( mysqli_result $result ,
int $fieldnr )
function mysql_field_name($result, $field_offset) {
# mysql_field_name = FALSE on failure
# mysqli_fetch_field_direct = FALSE if no field information for
specified fieldnr is available
$tmp = mysqli_fetch_field_direct($result, $field_offset);
if (!is_object($tmp)) return false;
$tmp = (array)$tmp;
return isset($tmp['name']) ? $tmp['name'] : false;
}

# mysql_field_seek - Set result pointer to a specified field
offset
# bool mysql_field_seek ( resource $result , int $field_offset )
# bool mysqli_field_seek ( mysqli_result $result , int $fieldnr )
function mysql_field_seek($result, $field_offset) {
# mysql_field_seek/mysqli_field_seek = FALSE on failure
$temp = mysqli_field_seek($result, $field_offset);
if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_field_table - Get name of the table the specified field
is in
# string mysql_field_table ( resource $result , int $field_offset
)
# mysqli_fetch_field_direct() [table] or [orgtable]
# -> object mysqli_fetch_field_direct ( mysqli_result $result ,
int $fieldnr )
function mysql_field_table($result, $field_offset) {
# mysql_field_table = error return value not defined
# mysqli_fetch_field_direct = FALSE if no field information for
specified fieldnr is available
$tmp = mysqli_fetch_field_direct($result, $field_offset);
if (!is_object($tmp)) return false;
$tmp = (array)$tmp;
return isset($tmp['table']) ? $tmp['table'] : false;
}

# mysql_field_type - Get the type of the specified field in a
result
# string mysql_field_type ( resource $result , int $field_offset
)
# mysqli_fetch_field_direct() [type]
# -> object mysqli_fetch_field_direct ( mysqli_result $result ,
int $fieldnr )
function mysql_field_type($result, $field_offset) {
# mysql_field_type = error return value not defined
# mysqli_fetch_field_direct = FALSE if no field information for
specified fieldnr is available
$tmp = mysqli_fetch_field_direct($result, $field_offset);
if (!is_object($tmp)) return false;
$tmp = (array)$tmp;
return isset($tmp['type']) ?
mysql_field_bittypes_to_types($tmp['type']) : false;
}

# mysql_free_result - Free result memory
# bool mysql_free_result ( resource $result )
# void mysqli_free_result ( mysqli_result $result )
function mysql_free_result($result) {
# mysql_free_result = FALSE on failure
# mysqli_free_result = No value is returned.
mysqli_free_result($result);
# note that mysqli does not return any boolean, so we do it
return true;
}

# mysql_get_client_info - Get MySQL client info
# string mysql_get_client_info ( void )
# string mysqli_get_client_info ( mysqli $link )
function mysql_get_client_info($link_identifier = null) {
# mysql_get_client_info/mysqli_get_client_info = not defined what
```

```
is returned on error
# note that mysql does not have a link argument while mysqli does
return
mysqli_get_client_info(mysql_ensure_link($link_identifier));
}

# mysql_get_host_info - Get MySQL host info
# string mysql_get_host_info ([ resource $link_identifier = NULL
] )
# string mysqli_get_host_info ( mysqli $link )
function mysql_get_host_info ($link_identifier = NULL) {
# mysql_get_host_info = FALSE on failure
# mysqli_get_host_info = error return value not defined
$temp =
mysqli_get_host_info(mysql_ensure_link($link_identifier));
if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_get_proto_info - Get MySQL protocol info
# int mysql_get_proto_info ([ resource $link_identifier = NULL ]
)
# int mysqli_get_proto_info ( mysqli $link )
function mysql_get_proto_info($link_identifier = NULL) {
# mysql_get_proto_info = FALSE on failure
# mysqli_get_proto_info = error return value not defined
$temp =
mysqli_get_proto_info(mysql_ensure_link($link_identifier));
if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_get_server_info - Get MySQL server info
# string mysql_get_server_info ([ resource $link_identifier =
NULL ] )
# string mysqli_get_server_info ( mysqli $link )
function mysql_get_server_info($link_identifier = NULL) {
# mysql_get_server_info = FALSE on failure
# mysqli_get_server_info = error return value not defined
$temp =
mysqli_get_server_info(mysql_ensure_link($link_identifier));
if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_info - Get information about the most recent query
# string mysql_info ([ resource $link_identifier = NULL ] )
# string mysqli_info ( mysqli $link )
function mysql_info($link_identifier = NULL) {
# mysql_info = FALSE on failure
# mysqli_info = returns empty string on failure
$temp = mysqli_info(mysql_ensure_link($link_identifier));
if ($temp === NULL) {
return false;
}
return $temp;
}

# mysql_insert_id - Get the ID generated in the last query
# int mysql_insert_id ([ resource $link_identifier = NULL ] )
# mixed mysqli_insert_id ( mysqli $link )
function mysql_insert_id($link_identifier = NULL) {
# mysql_insert_id = FALSE if no MySQL connection was established
# mysqli_insert_id = error value not defined
$temp = mysqli_insert_id(mysql_ensure_link($link_identifier));
if ($temp === null) {
return false;
}
return $temp;
}

# mysql_list_dbs - List databases available on a MySQL server
# resource mysql_list_dbs ([ resource $link_identifier = NULL ] )
# SQL Query: SHOW DATABASES
function mysql_list_dbs ($link_identifier = NULL) {
global $mysql_list_dbs_cache;

# mysql_list_dbs/mysql_query = FALSE on failure
$temp = mysql_query('SHOW DATABASES',
mysql_ensure_link($link_identifier));

$mysql_list_dbs_cache = $temp;

# when no working link is passed we get null
if ($temp === NULL) {
return false;
}
```

```
return $temp;
}

# mysql_list_fields - List MySQL table fields
# resource mysql_list_fields ( string $database_name , string
$table_name [, resource $link_identifier = NULL ] )
# SQL Query: SHOW COLUMNS FROM sometable
function mysql_list_fields ($database_name, $table_name,
$link_identifier = NULL) {
# mysql_list_fields/mysql_query = FALSE on failure
return mysql_query('SHOW COLUMNS FROM
'.mysql_real_escape_string($database_name).'.`'.mysql_real_escape
_string($table_name).'`', mysql_ensure_link($link_identifier));
}

# mysql_list_processes - List MySQL processes
# resource mysql_list_processes ([ resource $link_identifier =
NULL ] )
# mysqli_thread_id()
function mysql_list_processes($link_identifier = NULL) {
# mysql_list_processes = FALSE on failure
$temp = mysql_query("SHOW PROCESSLIST",
mysql_ensure_link($link_identifier));
if ($temp === null) {
return false;
}
return $temp;
}

# mysql_list_tables - List tables in a MySQL database
# resource mysql_list_tables ( string $database [, resource
$link_identifier = NULL ] )
# SQL Query: SHOW TABLES FROM sometable
function mysql_list_tables ($database_name, $table_name,
$link_identifier = NULL) {
# mysql_list_tables/mysql_query = FALSE on failure
return mysql_query('SHOW TABLES FROM
'.mysql_real_escape_string($database_name),
mysql_ensure_link($link_identifier));
}

# mysql_num_fields - Get number of fields in result
# int mysql_num_fields ( resource $result )
# int mysqli_field_count ( mysqli $link )
function mysql_num_fields ($result) {

# mysql_num_fields/mysqli_fetch_fields = FALSE on failure
# mysql takes a result, where mysqli takes link and takes the
most recent query
# so instead we fetch all the fields and then count that
$tmp = mysqli_fetch_fields($result);
if ($tmp === null) {
return false;
}
return count($tmp);
}

# mysql_num_rows - Get number of rows in result
# int mysql_num_rows ( resource $result )
# int mysqli_num_rows ( mysqli_result $result )
function mysql_num_rows($result) {
# mysql_num_rows = FALSE on failure
# mysqli_num_rows = error return value not defined
return mysqli_num_rows($result);
}

# mysql_pconnect - Open a persistent connection to a MySQL server
# resource mysql_pconnect ([ string $server =
ini_get("mysql.default_host") [, string $username =
ini_get("mysql.default_user") [, string $password =
ini_get("mysql.default_password") [, int $client_flags = 0 ]]]] )
# mysqli_connect() with p: host prefix
function mysql_pconnect($server = MYSQL_DEFAULT_HOST, $username =
MYSQL_DEFAULT_USER, $password = MYSQL_DEFAULT_PASSWORD,
$client_flags = 0) {
# mysql_pconnect/mysql_connect = FALSE on error
return mysql_connect('p:'.$server, $username, $password, true,
$client_flags);
}

# mysql_ping - Ping a server connection or reconnect if there is
no connection
# bool mysql_ping ([ resource $link_identifier = NULL ] )
# bool mysqli_ping ( mysqli $link )
function mysql_ping($link_identifier = NULL) {
# mysql_ping/mysqli_ping = FALSE on error
$temp = mysqli_ping(mysql_ensure_link($link_identifier));
if ($temp === NULL) {
return false;
}
return $temp;
}
```

```
# mysql_query - Send a MySQL query
# resource mysql_query ( string $query [, resource
$link_identifier = NULL ] )
# mixed mysqli_query ( mysqli $link , string $query [, int
$resultmode = MYSQLI_STORE_RESULT ] )
function mysql_query ($query, $link_identifier = NULL) {
# mysql_query/mysqli_query = FALSE on error
return mysqli_query(mysql_ensure_link($link_identifier), $query);
}

# mysql_real_escape_string - Escapes special characters in a
string for use in an SQL statement
# string mysql_real_escape_string ( string $unescaped_string [,
resource $link_identifier = NULL ] )
# string mysqli_real_escape_string ( mysqli $link , string
$escapestr )
function mysql_real_escape_string($unescaped_string,
$link_identifier = NULL) {
# mysql_real_escape_string = FALSE on error
# mysqli_real_escape_string = error return value not defined
return
mysqli_real_escape_string(mysql_ensure_link($link_identifier),
$unescaped_string);
}

# mysql_result - Get result data
# string mysql_result ( resource $result , int $row [, mixed
$field = 0 ] )
# no equivalent function exists in mysqli - mysqli_data_seek() in
conjunction with mysqli_field_seek() and mysqli_fetch_field()
function mysql_result($result , $row , $field = 0) {
# mysql_result = FALSE on failure
# try to seek position, returns false on failure
if (mysqli_data_seek($result, $row) === false) return false;
$row = mysqli_fetch_array($result);
if ($row === NULL || !isset($row[$field])) return false;
return $row[$field];
}


# mysql_select_db - Select a MySQL database
# bool mysql_select_db ( string $database_name [, resource
$link_identifier = NULL ] )
function mysql_select_db ($database_name, $link_identifier =
NULL) {
# mysql_select_db/mysqli_select_db = FALSE on failure
return mysqli_select_db(mysql_ensure_link($link_identifier),
$database_name);
}

# mysql_set_charset - Sets the client character set
# bool mysql_set_charset ( string $charset [, resource
$link_identifier = NULL ] )
# bool mysqli_set_charset ( mysqli $link , string $charset )
function mysql_set_charset($charset, $link_identifier = NULL) {
# mysql_set_charset/mysqli_set_charset = FALSE on failure
return mysqli_set_charset(mysql_ensure_link($link_identifier),
$charset);
}

# mysql_stat - Get current system status
# string mysql_stat ([ resource $link_identifier = NULL ] )
# string mysqli_stat ( mysqli $link )
function mysql_stat($link_identifier = NULL) {
# mysql_stat = NULL on error
# mysqli_stat = FALSE on error
$temp = mysqli_stat(mysql_ensure_link($link_identifier));
if ($temp === FALSE) {
return NULL;
}
return $temp;
}

# mysql_tablename - Get table name of field
# string mysql_tablename ( resource $result , int $i )
# no mysqli equivalent exists - SHOW TABLES [FROM db_name] [LIKE
'pattern']
function mysql_tablename ($result, $i) {
#       return mysql_query('SHOW COLUMNS FROM
"'.mysql_real_escape_string($database_name).'.'.mysql_real_escape
_string($table_name).'"', mysql_ensure_link($link_identifier));
return mysql_result($result, $i);
}

# mysql_thread_id - Return the current thread ID
# int mysql_thread_id ([ resource $link_identifier = NULL ] )
# int mysqli_thread_id ( mysqli $link )
function mysql_thread_id($link_identifier = NULL) {
# mysql_thread_id = FALSE on failure
# mysqli_thread_id = no error return value defined
$temp = mysqli_thread_id(mysql_ensure_link($link_identifier));
if ($temp === NULL) {
return false;
}
```

```
return $temp;
}
}

?>
```