PHP – Funzioni per la gestione di database MySQL

Funzione	Sintassi	Note
mysqli_connect — Apre una connessione ad un server MySQL	$link = **mysqli_connect**("127.0.0.1", "my_user", "my_password", "my_db"); $mysqli = new **mysqli**('localhost', 'my_user', 'my_password', 'my_db');	Restituisce un identificativo di connessione MySQL in caso di successo oppure **FALSE** in caso di fallimento.
mysqli_query — Invia una query MySQL	mixed **mysqli_query** (mysqli $link , string $query [, int $resultmode = MYSQLI_STORE_RESULT]) $result = $mysqli->**query**($sql)	Invia una query al database. Solo per le istruzioni SELECT, SHOW, EXPLAIN o DESCRIBE restituisce un identificativo di risorsa o **FALSE** se la query non è stata eseguita correttamente. Per altri tipi di istruzioni SQL, restituisce **TRUE** in caso di successo e **FALSE** in caso di errore.
mysqli_fetch_array — Carica una riga del risultato come un array associativo, un array numerico o entrambi.	mixed **mysqli_fetch_array** (mysqli_result $result [, int $resulttype = MYSQLI_BOTH]) $row = $result->**fetch_array**(MYSQLI_NUM); MYSQLI_ASSOC o MYSQLI_BOTH	Restituisce un array che corrisponde alla riga caricata o **NULL** se non ci sono più righe.
mysqli_num_rows — Ottiene il numero di righe in un risultato	int **mysqli_num_rows** (mysqli_result $result) $row_cnt = $result->**num_rows**;	Restituisce il numero di righe in un risultato. Questo comando è valido solo per le istruzioni **SELECT**.
mysqli_affected_rows — Ottiene il numero di righe coinvolte nelle precedenti operazioni MySQL	int **mysqli_affected_rows** (mysqli $link) $mysqli->**affected_rows**	Restituisce il numero di righe coinvolte nell'ultima query **INSERT, UPDATE o DELETE.** Se l'ultima query fallisce, questa funzione restituisce -1.
mysqli_insert_id — Ottiene l'identificativo generato dalla precedente operazione INSERT	mixed **mysqli_insert_id** (mysqli $link) $mysqli->**insert_id**	restituisce l'identificativo generato per una colonna AUTO_INCREMENT dalla precedente query INSERT. Restituisce 0 se la precedente query non ha generato un valore AUTO_INCREMENT.
mysqli_error — Restituisce il testo del messagio di errore della precedente operazione MySQL	string **mysqli_error** (mysqli $link) $mysqli->**error**	Restituisce il testo dell'errore dall'ultima funzione MySQL, oppure '' (la stringa vuota) se nessun errore intercorre
mysqli_connect_error — Restituisce un errore di connessione a MySQL	string **mysqli_connect_error** (void) $mysqli->**connect_error**	Restituisce l'ultima stringa di errore dopo l'ultima chiamata a mysqli_connect() o NULL se non c'è errore
mysqli_real_escape_string-Crea una stringa SQL "legale"	string **mysqli_real_escape_string** (mysqli $link , string $escapestr)	I caratteri codificati sono *NUL (ASCII 0), \n, \r, \, ', ", e Control-Z*.
mysql_close — Chiude una connessione MySQL	bool **mysql_close** ([resource `$ identificativo_connessione`])	Restituisce **TRUE** in caso di successo, **FALSE** in caso di fallimento.

PHP – Funzioni per la gestione di database MySQL

Indice dei contenuti

PHP – Funzioni per la gestione di database MySQL

- mysqli::select_db — Selects the default database for database queries
- mysqli::send_query — Send the query and return
- mysqli::set_charset — Sets the default client character set
- mysqli::set_local_infile_default — Unsets user defined handler for load local infile command
- mysqli::set_local_infile_handler — Set callback function for LOAD DATA LOCAL INFILE command
- mysqli::$sqlstate — Returns the SQLSTATE error from previous MySQL operation
- mysqli::ssl_set — Used for establishing secure connections using SSL
- mysqli::stat — Gets the current system status
- mysqli::stmt_init — Initializes a statement and returns an object for use with mysqli_stmt_prepare
- mysqli::store_result — Transfers a result set from the last query
- mysqli::$thread_id — Returns the thread ID for the current connection
- mysqli::thread_safe — Returns whether thread safety is given or not
- mysqli::use_result — Initiate a result set retrieval
- mysqli::$warning_count — Returns the number of warnings from the last query for the given link

Indice dei contenuti

PHP – Funzioni per la gestione di database MySQL

```
link: http://php.net/manual/en/mysqli.examples-basic.php
<?php
// Let's pass in a $_GET variable to our example, in this case
// it's aid for actor_id in our Sakila database. Let's make it
// default to 1, and cast it to an integer as to avoid SQL injection
// and/or related security problems. Handling all of this goes beyond
// the scope of this simple example. Example:
//   http://example.org/script.php?aid=42
if (isset($_GET['aid']) && is_numeric($_GET['aid'])) {
    $aid = (int) $_GET['aid'];
} else {
    $aid = 1;
}

// Connecting to and selecting a MySQL database named sakila
// Hostname: 127.0.0.1, username: your_user, password: your_pass, db: sakila
$mysqli = new mysqli('127.0.0.1', 'your_user', 'your_pass', 'sakila');

// Oh no! A connect_errno exists so the connection attempt failed!
if ($mysqli->connect_errno) {
    // The connection failed. What do you want to do?
    // You could contact yourself (email?), log the error, show a nice page, etc.
    // You do not want to reveal sensitive information

    // Let's try this:
    echo "Sorry, this website is experiencing problems.";

    // Something you should not do on a public site, but this example will show
you
    // anyways, is print out MySQL error related information --
 you might log this
    echo "Error: Failed to make a MySQL connection, here is why: \n";
    echo "Errno: " . $mysqli->connect_errno . "\n";
    echo "Error: " . $mysqli->connect_error . "\n";

    // You might want to show them something nice, but we will simply exit
    exit;
}

// Perform an SQL query
$sql = "SELECT actor_id, first_name, last_name FROM actor WHERE actor_id = $aid";
if (!$result = $mysqli->query($sql)) {
    // Oh no! The query failed.
    echo "Sorry, the website is experiencing problems.";

    // Again, do not do this on a public site, but we'll show you how
    // to get the error information
    echo "Error: Our query failed to execute and here is why: \n";
    echo "Query: " . $sql . "\n";
    echo "Errno: " . $mysqli->errno . "\n";
    echo "Error: " . $mysqli->error . "\n";
    exit;
}

// Phew, we made it. We know our MySQL connection and query
// succeeded, but do we have a result?
if ($result->num_rows === 0) {
    // Oh, no rows! Sometimes that's expected and okay, sometimes
    // it is not. You decide. In this case, maybe actor_id was too
    // large?
    echo "We could not find a match for ID $aid, sorry about that. Please try ag
ain.";
    exit;
```

```
}

// Now, we know only one result will exist in this example so let's
// fetch it into an associated array where the array's keys are the
// table's column names
$actor = $result->fetch_assoc();
echo "Sometimes I see " . $actor['first_name'] . " " . $actor['last_name'] . " o
n TV.";

// Now, let's fetch five random actors and output their names to a list.
// We'll add less error handling here as you can do that on your own now
$sql = "SELECT actor_id, first_name, last_name FROM actor ORDER BY rand() LIMIT
5";
if (!$result = $mysqli->query($sql)) {
    echo "Sorry, the website is experiencing problems.";
    exit;
}

// Print our 5 random actors in a list, and link to each actor
echo "<ul>\n";
while ($actor = $result->fetch_assoc()) {
    echo "<li><a href='" . $_SERVER['SCRIPT_FILENAME'] . "?aid=" . $actor['actor
_id'] . "'>\n";
    echo $actor['first_name'] . ' ' . $actor['last_name'];
    echo "</a></li>\n";
}
echo "</ul>\n";

// The script will automatically free the result and close the MySQL
// connection when it exits, but let's just do it anyways
$result->free();
$mysqli->close();
?>
```

Altri Link:

http://php.net/manual/en/mysqli.quickstart.statements.php