69/B Monwara Plaza (Level-5)
East Panthapath
Dhaka-1205
Cell: 01713 432003

Lecture Sheet on "PHP+MySQL"
Day-9: (PHP+MySQL)

Objectives

- Describe how arguments and PHP functions relate to the database
- Differentiate between simple and complex functions depending on execution results
- Understand the importance of associative and indexed arrays.
- Modify specific scripts to allow for secure data transmission
- Update database records using PHP scripts

Topics

- Connecting to MySQL and selecting the database
- Executing simple queries
- Retrieving query results
- Ensuring secure SQL
- Counting returned records
- Updating records with PHP

mysql_connect()

mysql_connect — Open a connection to a MySQL Server

Description:

mysql_connect ([string $server [, string $username [, string $password [, bool $new_link [, int $client_flags]]]]])

Opens or reuses a connection to a MySQL server.

Parameters

server

The MySQL server. It can also include a port number. e.g. "hostname:port" or a path to a local socket e.g. ":/path/to/socket" for the localhost. If the PHP directive mysql.default_host is undefined (default), then the default value is 'localhost:3306'. In SQL safe mode, this parameter is ignored and value 'localhost:3306' is always used.

username

The username. Default value is defined by mysql.default_user. In SQL safe mode, this parameter is ignored and the name of the user that owns the server process is used.

password

The password. Default value is defined by mysql.default_password. In SQL safe mode, this parameter is ignored and empty password is used.

new_link

If a second call is made to **mysql_connect()** with the same arguments, no new link will be established, but instead, the link identifier of the already opened link will be returned. The *new_link* parameter modifies this behavior and makes **mysql_connect()** always open a new link, even if **mysql_connect()** was called before with the same parameters. In SQL safe mode, this parameter is ignored.

client_flags

The *client_flags* parameter can be a combination of the following constants: 128 (enable *LOAD DATA LOCAL* handling), **MYSQL_CLIENT_SSL**, **MYSQL_CLIENT_COMPRESS**, **MYSQL_CLIENT_IGNORE_SPACE** or **MYSQL_CLIENT_INTERACTIVE**.

Return Values

Returns a MySQL link identifier on success, or **FALSE** on failure.

Examples

Example: mysql_connect() example

```
<?php
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
if (!$link) {
   die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
mysql_close($link);
?>
```

mysql_query

mysql_query — Send a MySQL query

Description

resource **mysql_query** (string $query [, resource $link_identifier])

mysql_query() sends an unique query (multiple queries are not supported) to the currently active database on the server that's associated with the specified *link_identifier*.

Parameters

query **:** A SQL query. The query string should not end with a semicolon.

link_identifier **:** The MySQL connection. If the link identifier is not specified, the last link opened by mysql_connect() is assumed. If no such link is found, it will try to create one as if mysql_connect() was called with no arguments. If by chance no connection is found or established, an **E_WARNING** level warning is generated.

Return Values

For SELECT, SHOW, DESCRIBE, EXPLAIN and other statements returning resultset, **mysql_query()** returns a **resource** on success, or **FALSE** on error. For other type of SQL statements, UPDATE, DELETE, DROP, etc, **mysql_query()** returns **TRUE** on success or **FALSE** on error. The returned result resource should be passed to mysql_fetch_array(), and other functions for dealing with result tables, to access the returned data. Use mysql_num_rows() to find out how many rows were returned for a SELECT statement or mysql_affected_rows() to find out how many rows were affected by a DELETE, INSERT, REPLACE, or UPDATE statement.

69/B Monwara Plaza (Level-5)
East Panthapath
Dhaka-1205
Cell: 01713 432003

mysql_query() will also fail and return **FALSE** if the user does not have permission to access the table(s) referenced by the query.

Examples

Example : Invalid Query

The following query is syntactically invalid, so **mysql_query()** fails and returns **FALSE.**

```
<?php
$result = mysql_query('SELECT * WHERE 1=1');
if (!$result) {
    die('Invalid query: ' . mysql_error());
}

?>
```

mysql_affected_rows

mysql_affected_rows — Get number of affected rows in previous MySQL operation

Description

int **mysql_affected_rows** ([resource $link_identifier])

Get the number of affected rows by the last INSERT, UPDATE, REPLACE or DELETE query associated with *link_identifier*.

Parameters

link_identifier

The MySQL connection. If the link identifier is not specified, the last link opened by mysql_connect() is assumed. If no such link is found, it will try to create one as if mysql_connect() was called with no arguments. If by chance no connection is found or established, an **E_WARNING** level warning is generated.

Return Values

Returns the number of affected rows on success, and -1 if the last query failed. If the last query was a DELETE query with no WHERE clause, all of the records will have been deleted from the table but this function will return zero with MySQL versions prior to 4.1.2. When using UPDATE, MySQL will not update columns where the new value is the same as the old value. This creates the possibility that **mysql_affected_rows()** may not actually equal the number of rows matched, only the number of rows that were literally affected by the query. The REPLACE statement first deletes the record with the same primary key and then inserts the new record. This function returns the number of deleted records plus the number of inserted records.

Examples

Example : mysql_affected_rows() example

```
<?php
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
```

```
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
mysql_select_db('mydb');

/* this should return the correct numbers of deleted records */
mysql_query('DELETE FROM mytable WHERE id < 10');
printf("Records deleted: %d\n", mysql_affected_rows());

/* with a where clause that is never true, it should return 0 */
mysql_query('DELETE FROM mytable WHERE 0');
printf("Records deleted: %d\n", mysql_affected_rows());
?>
```

The above example will output something similar to:

```
Records deleted: 10
Records deleted: 0
```