

Message table

| msg id | parent id | subject |
|--------|-----------|---------|
| 1 | NULL | msg1 |
| 2 | 1 | msg2 |
| 3 | 1 | msg3 |
| 4 | 2 | msg4 |
| 5 | 4 | msg5 |
| 6 | 4 | msg6 |

- msg1
 - msg2
 - msg4
 - msg5
 - msg6
- msg3

```
/*
 * This code shows a very simple method to access and print a threaded
 * discussion that is stored in a single MySQL table.
 *
 * The table definition is:
 *
 * mysql> describe disc;
 * +-----+-----+-----+-----+-----+-----+
 * | Field      | Type      | Null | Key | Default | Extra      |
 * +-----+-----+-----+-----+-----+-----+
 * | name       | text      | YES  |     | NULL    |            |
 * | id         | int(11)   |      | PRI | NULL    | auto_increment |
 * | parent_id  | int(11)   | YES  |     | NULL    |            |
 * +-----+-----+-----+-----+-----+-----+
 * 3 rows in set (0.00 sec)
 *
 * The printThreaded() function takes as arguments a mysql result and an
 * indent level.
 * It then looks for rows in the table that have parent_id = id of the
 * current message,
 * and prints those recursively.
 *
 */
```

```

<?
function printThreaded($result, $indent) {
    while ($row = mysql_fetch_row($result)) {
        print("$indent row: " . $row["name"] . " , " . $row["id"] . " , " .
$row["parent_id"] . "\n");
        $id = $row["id"];
        $new_result = mysql_query("select * from disc where parent_id = $id")
or die("query 2");
        printThreaded($new_result, $indent . " ");
    }
}

$link = mysql_connect("host", "user", "password") or die("mysql conn");
mysql_select_db("database") or die("db");

$result = mysql_query("select * from disc where parent_id is null") or
die("query 1");

printThreaded($result, "");

mysql_close($link);

?>

```

Sessions

- Creating sessions allows you to keep track of the actions of a particular user over the period of time she is viewing your web site.
- That is, the term "session" refers to the time that a user is at a particular web site.
- Sessions automatically set cookies for the session id.

Syntax Example

This is an example of a counter.

- Start a session
- Print the most recent value of the counter
- Increment the counter
- Register the counter

```

session_start(); // sets a cookie if not already there
                // o.w. load variables from session associated with
                // session id from the cookie.

print($counter);
$counter++;
session_register("counter"); // saves in the current session

```

Sessions continued

```
session_destroy(); //removes all the data associated with the session
// call this when someone logs out
```

- You must call session start before session destroy
- Default is to use the root level directory for domain; will get set to any visited page on ieng9, which is not what you want.
- Instead, use the following function, which tells the domain and the path where the cookie is sent back.

```
session_set_cookie_params(domain, path, lifetime);
```

- If the path is not explicitly set, the cookie will be sent to any page within that domain.