FUNZIONI PHP PER POSTGRESQL

Connessione e gestione connessione:

- **pg_pconnect (string $connection_string, int $connect_type])**

  - pg_close() will not close persistent links generated by pg_pconnect().
  - *The connection_string can be empty to use all default parameters, or it can contain one or more parameter settings separated by whitespace. Each parameter setting is in the form keyword = value.*
  - *Restituisce un oggetto di tipo connessione oppure false.*
  - *Es:* `$conn = pg_connect ("host=localhost user=usertest password=userpsw dbname=testdb");`

- **pg_connect (string $connection_string, int $connect_type)**

  - *Restituisce un oggetto di tipo connessione oppure false.*
  - *Es:* `$conn = pg_connect ("host=localhost user=usertest password=userpsw dbname=testdb");`

QUERY e SESSIONI:

- **pg_prepare (resource $connection , string $stmtname , string $query)**

  - **stmtname:** *The name to give the prepared statement. Must be unique per-connection. If "" is specified, then an unnamed statement is created, overwriting any previously defined unnamed statement.*
  - **query:** The parameterized SQL statement. Must contain only a single statement. (multiple statements separated by semi-colons are not allowed.) If any parameters are used, they are referred to as $1, $2, etc.
  - *Restituisce un oggetto di tipo result resource oppure false*
  - *Es:* `$res = pg_prepare($conn,"query",'SELECT * FROM shops WHERE name = $1');`

- **pg_execute (resource $connection , string $stmtname , array $params)**

  - **connection**
  - **stmtname:** The name of the prepared statement to execute.
  - **params:** An array of parameter values to substitute for the $1, $2, etc. placeholders in the original prepared query string.
  - *Restituisce un oggetto di tipo query result oppure false*
  - *Es:* `$res = pg_execute($dbconn, "query", array("Joe's Widgets"));`

- **pg_num_rows (resource $result)**

  - *returns The number of rows in the result*

- **pg_affected_rows (resource $result)**

  - *returns the number of tuples affected by INSERT, UPDATE, and DELETE queries.*

- **pg_fetch_array (resource $result, int $row, int $result_type)**

- **result:** *PostgreSQL query result resource*
- **row:** *Row number in result to fetch. Rows are numbered from 0. If omitted or null, the next row is fetched.*
- **result_type:** **PGSQL_ASSOC, PGSQL_NUM** and **PGSQL_BOTH**
- *Returns An array indexed numerically (beginning with 0) or associatively (indexed by field name), or both. Each value in the array is represented as a string. Database NULL values are returned as null.*
- *Es:*

```
$arr = pg_fetch_array($result, 0, PGSQL_NUM);
        echo $arr[0];   <- Row 1 Author\n";
        echo $arr[1];   <- Row 1 E-mail\n";
```

- **pg_fetch_row (resource $result , int $row=?)**

- fetches one row of data from the result associated with the specified result resource.
- **result:** Resource restituito da un pg_query(), pg_query_params() etc..
- **row:** Row number in result to fetch. Rows are numbered from 0 upwards. If omitted or null, the next row is fetched.
- **returns:** an array indexed from 0 upwards with each value represented as string. Database NULL values are returned as null. False is returned if row exceeds the number of rows in the set, there are no more rows or on any other error
- *Es:*

```
while ($row = pg_fetch_row($result)) {
        echo "First Row: $row[0]  Second Row: $row[1]";
    }
```

- **pg_fetch_assoc (resource $result , int $row=?)**

- returns an associative array that corresponds to the fetched row (records).
- **result:** Resource restituito da un pg_query(), pg_query_params() etc..
- **row:** Row number in result to fetch. Rows are numbered from 0 upwards. If omitted or null, the next row is fetched.
- **returns:** an array indexed associatively by field name. Each value in the array is represented as a string. Database NULL values are returned as null. False is returned if row exceeds the number of rows in the set, there are no more rows or on any other error.
- Es

```
while ($row = pg_fetch_assoc($result)) {
        echo $row['field1'];
        echo $row['field2'];
        echo $row['field3'];
    }
```

- `session_start ( array $options = [] ) : bool`

- creates a session or resumes the current one based on a session identifier passed via a GET or POST request, or passed via a cookie.
- To use a named session, call session_name() before calling session_start().
- This function returns true if a session was successfully started, otherwise false.

RECORD IMPORTANTI DI PHP PER RICHIESTA E SESSIONI:

- **$_POST** = is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_POST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>
</body>
</html>
```

- **$_SESSION** = An associative array containing session variables available to the current script. See the Session functions documentation for more information on how this is used.

```
<?php
session_start();
$_SESSION["newsession"]=$value;
echo $_SESSION["newsession"];
?>
```