

CONNESSIONE POSTGRESQL – PHP

```
// Connection details
$conn_string = "host=localhost port=5432 dbname=World_db user=scg password=MyPa

// Establish a connection with MySQL server
$dbconn = pg_connect($conn_string);
// Check connection status. Exit in case of errors
if(!$dbconn) {
    echo "Error: Unable to open database\n";
} else {
    // echo "Opened database successfully\n";
    $query = "SELECT name, district FROM city WHERE countrycode='ARG'";
    $cities = pg_query($query) or die('Query failed: ' . pg_last_error());
    $myarray = array();
    while ($row = pg_fetch_assoc($cities)) {
        $myarray[] = $row;
    }
}

// Close connection
pg_close($dbconn);
```

Funzioni PHP per la connessione:

1) **pg_connect** apre una connessione a PostgreSQL; richiede come parametri un nome-host, il nome del database, username e password. (tipo “fopen()” sui file)

2) **pg_exec** (o **pg_query**) - esegue un'istruzione SQL.

3) **pg_numrows** - ritorna il numero di record contenuti nel risultato di un'istruzione.
(quante righe)

4) **pg_fetch_array** - ritorna un array che corrisponde ad una riga della tabella
(ritornata da **pg_exec**)

5) **pg_freeresult** - libera le risorse utilizzate nella connessione corrente. (tipo “free() o dispose()”)

6) **pg_close** - chiude la connessione corrente. (tipo “fclose()” sui file)

Assumiamo di avere già a disposizione un database PostgreSQL installato, per il quale abbiamo anche un username e una password validi. Assumiamo inoltre di avere già creato una tabella in quel database, chiamata INVENTARIO_CAFFE. La tabella INVENTARIO_CAFFE è costituita da tre colonne: NOME_CAFFE, TIPO_TOSTATURA e QUANTITA.

NOME_CAFFE	TIPO_TOSTATURA	QUANTITA
Tostatura Francese	Scura	18
Kenya	Scura	6
Harrar Etiopia	Media	35
Sumatra	Scura	8
Colombia	Chiara	12

A questo punto possiamo iniziare a lavorare con il PHP. Prima però è necessario conoscere il nome del server nel quale è contenuto il database, e avere un username ed una password validi. Iniziamo la stesura del codice creando una variabile di connessione:

```
$connessione = pg_connect("host=NomeHost  
                        dbname=NomeDataBase  
                        user=Username  
                        password=XYZ")  
// su localhost la password solitamente viene omessa  
or die("Impossibile stabilire una connessione.");
```

La funzione **die()** viene usata per fermare lo script e mostrare un dato messaggio di errore, nel caso in cui la funzione precedente dovesse fallire (in questo caso nell'eventualità che la connessione dovesse fallire.)

Fino a questo punto abbiamo detto al PHP di connettersi al server e di selezionare un database. Compiuto questo passo siamo già in grado di inviare un'istruzione SQL e di riceverne i risultati!

Usando la tabella INVENTARIO_CAFFE, supponiamo di voler visualizzare l'inventario completo, comprendendo nome della varietà, tipo di tostatura, ordinato in modo da avere per prime le varietà con la maggiore quantità in magazzino. Per fare questo creiamo una variabile che contenga l'istruzione SQL:

```
$sql = "SELECT NOME_CAFFE, TIPO_TOSTATURA, QUANTITA  
FROM INVENTARIO_CAFFE  
ORDER BY QUANTITA DESC";
```

Poi creiamo una variabile che contenga i risultati della query, ricevuti dalla funzione **pg_exec**. La funzione **pg_exec** ha bisogno di due parametri: la stringa di connessione e la variabile che contiene l'espressione SQL che abbiamo appena creato.

```
$sql_risultato = pg_exec($connessione, $sql)  
or die("Impossibile eseguire la query.");
```

A questo punto contiamo il numero di righe contenute nel risultato, usando la funzione **pg_numrows**:

```
$num = pg_numrows($sql_risultato);
```

Al fine di organizzare i risultati attualmente contenuti nella variabile `$sql_risultato`, prima di tutto bisogna dividerli per riga, utilizzando la funzione `pg_fetch_array` all'interno di un loop:

```
<?php
for ($i=0; $i < $num; $i++)
{
    $riga = pg_fetch_array($sql_risultato, $i);

    // altro codice...
}
?>
```

Il loop con while crea un array che si chiama **\$riga**, nel quale ad ogni indice corrisponde un record contenuto nel risultato della query. Per ottenere individualmente i valori corrispondenti ai vari campi (NOME_CAFFE, TIPO_TOSTATURA, QUANTITA), possiamo definire delle variabili specifiche:

```
$nome_caffe      = $riga["NOME_CAFFE"];
$tipo_tostatura  = $riga["TIPO_TOSTATURA"];
$quantita        = $riga["QUANTITA"];
```

Il loop allora deve diventare una cosa simile a questa:

```
<?php
for ($i=0; $i < $num; $i++)
{
    $riga = pg_fetch_array($sql_risultato, $i);
    $nome_caffe      = $riga["NOME_CAFFE"];
    $tipo_tostatura  = $riga["TIPO_TOSTATURA"];
    $quantita        = $riga["QUANTITA"];
    // Aggiungi la visualizzazione dei risultati,
    // eventualmente in tabella HTML
}
?>
```

Per finire, liberiamo le risorse occupate per effettuare la query e chiudiamo la connessione al database. Dimenticarsi questo passo significa causare sprechi di memoria e facilitare eventi fastidiosi che riducono la quantità di risorse a disposizione.

```
pg_freeresult($sql_risultato); // tipo free() o dispose() del C/C++
pg_close($connessione); // tipo fclose() del C
```

L'intero script per effettuare una semplice connessione e una selezione di dati da un database PostgreSQL puoi vederlo anche con la tabella articoli :

```
<html>
<head>
<title> Elenco Articoli </title>
</head>
<center>
<table border=1>
<TH>CODICE</TH><TH>DESCRIZIONE</TH><TH>PREZZO</TH><TR>

<?php
// creo connessione a postgres
$connessione = pg_connect("host=localhost
                           dbname=postgres
                           user=dSACCO
                           ")
                           or die ("Impossibile connettersi.");

$sql="SELECT * FROM articoli";

$risultato=pg_exec($connessione,$sql)
           or die ("Errore lancio query");

$righe=pg_numrows($risultato);

for ($i=0; $i < $righe; $i++)
{
    $riga = pg_fetch_array($risultato,$i);

    printf("<TD>%s</TD>\n",    $riga["codice"] );
    printf("<TD>%s</TD>\n",    $riga["descrizione"] );
    printf("<TD>%s</TD><TR>\n", $riga["prezzo"] );
}
// libero memoria
pg_freeresult($risultato);
// fine connessione
pg_close($connessione);
?>

</table>
</body>
</html>
```