


## 07 – Grillparty Teil 1

Im ersten Teil geht es darum, die Datensätze auf der Webseite auszugeben.

Ihnen steht bereits eine Vorlage zur Verfügung.



### Einladung zur Grillparty

**Wir planen eine große Grillparty - sei dabei!**

Hallo, wir planen eine große Grillparty. Es gibt reichlich Essen, Trinken und Musik. Die Feier soll am Samstag in zwei Wochen ab 15 Uhr bei uns im Garten stattfinden. Ihr seid herzlich eingeladen. Wir würden uns sehr über Deine/Eure Zusage freuen. Außerdem fänden wir es toll, wenn jeder eine Kleinigkeit zum Büffet beitragen könnte. (Grillgut, Salat, Brot, Dips, Nachtsch, ...). Grill und Getränke besorgen wir! Bitte tragt euch in die Liste ein!

Liebe Grüße, Anna und Thilo.

**Bisherige Zusagen:**

Nummer	Vorname	Nachname	Ich bringe mit
--------	---------	----------	----------------

Am Ende dieser Lektion sollen in der Tabelle die Datensätze der Tabelle „**gäste**“ aufgelistet werden.

## PHP und MySQL

Die index-Datei enthält lediglich HTML-Elemente. Es fehlen noch die PHP-Befehle, um eine Verbindung zum Datenbankserver aufzubauen und um die Datensätze anzeigen zu lassen.

### Verbindung zur Datenbank aufnehmen

Fügen Sie oberhalb des HTML-Codes folgende PHP-Anweisungen hinzu:

```
1
2  <?php
3
4      /*Verbindung aufnehmen und Datenbank auswählen*/
5      $con = mysqli_connect("", "root", "", "gp");
6
7      /*Falls es einen Verbindungsfehler gibt, soll mir das angezeigt werden.*/
8      if(!$con)
9          echo ("Verbindungsfehler mit der Datenbank");
10
11     /*SQL-Abfrage definieren */
12     $sql = "SELECT * FROM gäste";
13
14     /*SQL-Abfrage ausführen*/
15     $res = mysqli_query($con, $sql);
16
17     /*Anzahl der Datensätze ermitteln*/
18     $num = mysqli_num_rows($res);
19
20
21  ?>
22
23
24  <!DOCTYPE html>
25  <html>
26      <head>
27          <meta charset = "utf-8">
28          <link rel="stylesheet" type="text/css" href="css/style.css" />
29          <title>Grillparty</title>
30      </head>
```

## Zur Erklärung

### ***mysqli\_connect()***

Die Funktion ***mysqli\_connect()*** öffnet eine Verbindung zum MySQL-Datenbankserver. Die Funktion übergibt vier Parameter an den Server.

- Hostname
- Benutzer
- Passwort
- Datenbank

### ***mysqli\_connect(Hostname, Benutzer, Passwort, Datenbank);***

In unserem Fall kann für Hostname und Passwort eine leere Zeichenkette angegeben werden. Das Passwort lautet „**root**“ und die Datenbank lautet „**gp**“ (oder bei Ihnen evtl. „Grillparty“).

Der Rückgabewert der Funktion ist eine Referenz auf die Verbindung. Diese Referenz wird in einer Variablen abgespeichert (hier: **\$con**).

### ***mysqli\_query()***

Diese Funktion führt eine Abfrage aus und benötigt zwei Parameter:

1. Die Referenz der Verbindung (**\$con**)
2. Eine SQL-Anweisung

Es bietet sich an, die SQL-Anweisung vorab in einer Variablen abzulegen (hier: **\$sql**). Sie werden später sehen, dass diese Art komfortabler und eleganter ist.

Es ist aber auch möglich, die SQL-Anweisung mit in die Funktion zu schreiben:

***\$res = mysqli\_query(\$con, "SELECT \* FROM gäste");***

Die Funktion ***mysqli\_query()*** gibt als Rückgabewert alle Datensätze der Tabelle in Form eines Objektes zurück. Das Objekt wird in einer Variablen, hier **\$res**, abgelegt.

### ***mysqli\_num\_rows()***

Mit dieser Funktion kann die Anzahl der gelieferten Datensätze ermittelt werden. Falls die Anfrage nicht erfolgreich war, z.B. wegen eines SQL-Fehlers oder weil die Tabelle keine Datensätze enthält, hat die Variable `$num` den Wert 0.

An dieser Stelle lohnt es sich, den Wert von `$num` mittels `echo` ausgeben zu lassen. Quasi als erste Überprüfung, ob soweit alles funktioniert.

**Wird Ihnen die Zahl 3 ausgegeben?**

### **Datensätze anzeigen**

Im nächsten Schritt geht es darum aus dem Objekt `$res` die einzelnen Datensätze „rauszuholen“. Der grundlegende Befehl hierfür lautet ***mysqli\_fetch\_assoc()***.

Dieser Funktion wird als Parameter das Objekt `$res` übergeben. Die Funktion gibt dann einen kompletten Datensatz (also eine Zeile der Tabelle) in Form eines assoziativen Feldes zurück. Der Rückgabewert wird in einer Variablen (hier: `$dsatz`) abgelegt.

```
/*Anzahl der Datensätze ermitteln*/
$num = mysqli_num_rows($res);

if($num > 0)
{
    $dsatz = mysqli_fetch_assoc($res);

    var_dump($dsatz);
}
```

Lassen Sie sich einmal den Inhalt der Variable `$dsatz` anzeigen.

Sie werden sehen, dass die Spaltennamen (`gast_id`, `vorname`, `nachname`, `beitrag`) die Schlüssel der Array-Felder darstellen.

```
array(4) { ["gast_id"]=> string(1) "1" ["vorname"]=> string(7) "Florian" ["nachname"]=> string(6) "Bertel" ["beitrag"]=> string(10) "Nudelsalat" }
```

Der Zugriff in PHP auf z.B. den Vornamen geschieht dann mit `$dsatz["vorname"]`. Dieses Array-Feld enthält als Wert die Zeichenkette „Florian“.

## Alle Datensätze auslesen

Die Funktion ***mysqli\_fetch\_assoc()*** verwendet intern einen Datensatzzeiger, der nach jedem Funktionsaufruf um 1 erhöht wird und somit auf den nächsten Datensatz (nächste Zeile in der Tabelle) verweist. Dies wird deutlich, wenn man die Funktion ***mysqli\_fetch\_assoc()*** mehrfach aufruft.

```
/*Anzahl der Datensätze ermitteln*/
$num = mysqli_num_rows($res);

if($num > 0)
{
    /* 1. Datensatz*/
    $dsatz = mysqli_fetch_assoc($res);
    var_dump($dsatz);
    echo "<br>";

    /* 2. Datensatz*/
    $dsatz = mysqli_fetch_assoc($res);
    var_dump($dsatz);
    echo "<br>";

    /* 3. Datensatz*/
    $dsatz = mysqli_fetch_assoc($res);
    var_dump($dsatz);
    echo "<br>";

    /* 4. Datensatz gibt es nicht*/
    $dsatz = mysqli_fetch_assoc($res);
    var_dump($dsatz);
}
```

Insgesamt wird die Funktion ***mysqli\_fetch\_assoc()*** hier nacheinander 4-mal aufgerufen. Wie die Ausgabe mittels ***var\_dump()*** zeigt, gibt jeder erneuter Aufruf der Funktion einen neuen Datensatz der Tabelle aus.

```
array(4) { ["gast_id"]=> string(1) "1" ["vorname"]=> string(7) "Florian" ["nachname"]=> string(6) "Bertel" ["beitrag"]=> string(10) "Nudelsalat" }
array(4) { ["gast_id"]=> string(1) "2" ["vorname"]=> string(7) "Kerstin" ["nachname"]=> string(5) "Janke" ["beitrag"]=> string(11) "Herrencreme" }
array(4) { ["gast_id"]=> string(1) "3" ["vorname"]=> string(6) "Robert" ["nachname"]=> string(6) "Wagner" ["beitrag"]=> string(17) "10 Bratwürstchen" }
NULL
```

Erreicht der Datensatzzeiger das Ende der Tabelle, gibt die Funktion den **Wert 0** (NULL) zurück. Diese Tatsache kann genutzt werden, um die Funktion ***mysqli\_fetch\_assoc()*** mit einer **while-Schleife** zu verknüpfen. Solange der Wert von ***\$dsatz*** nicht **NULL** ist, wird die Schleife durchlaufen.

```
<?php

if($num > 0)
{
    while($dsatz = mysqli_fetch_assoc($res))
    {
        /* Ausgabe der Daten */
    }
}
```

## Aufgabe

Bauen Sie diese while-Schleife in den Body-Teil der HTML-Tabelle ein.

```
<h2>Bisherige Zusagen:</h2>
<table>
  <thead>
    <tr><th>Nummer</th><th>Vorname</th><th>Nachname</th><th>Ich bringe mit</th></tr>
  </thead>
  <tbody>

    <?php
      if($num > 0)
      {
        $count = 1;
        while($dsatz = mysqli_fetch_assoc($res))
        {
          /* Ausgabe einer Zeile */
        }
      }
    ?>

  </tbody>
</table>
```

Greifen Sie über die Variable `$dsatz` auf die einzelnen Felder zu und bauen Sie diese in eine neue HTML-Tabellenzeile (`<tr><td> ....</td> ...</tr>`) ein.

Ergebnis soll die Ausgabe einer Tabelle mit den Daten sein (s. unten).

Bisherige Zusagen:

Nummer	Vorname	Nachname	Ich bringe mit
1	Florian	Bertel	Nudelsalat
2	Kerstin	Janke	Herrencreme
3	Robert	Wagner	10 Bratwürstchen