

# PHP-Programmierkurs [7] - Datenbankabfrage

In diesem Lernpfad, erfährst du, wie du eine dynamische Schnittstelle zwischen dem Betrachter einer Internetseite und den Inhalten einer MySQL-Datenbank erzeugt wird.

## Verbindung

### Verbindung aufnehmen

Zunächst muss eine Verbindung zur Datenbank bestehen:

```
$db_host="localhost";
$db_user="username";
$db_password="deinPasswort";
$db_database="nameDerDatenbank";

//Verbindung aufbauen
$verbindung = mysqli_connect($db_host, $db_user, $db_password, $db_database)
or
die("Keine Verbindung möglich! " . mysqli_error($verbindung));
echo "Verbindung zum Datenbankserver / zur Datenbank hergestellt! <br />";
```

- Die Funktion `mysqli_connect()` öffnet eine Verbindung zum MySQL-Datenbankserver. In den Klammern stehen vier Parameter: Hostname, Benutzername, Kennwort und Tabellename.
- Der Rückgabewert der Funktion `mysqli_connect()` ist eine Referenz auf die Verbindung. Diese Referenz wird anschließend für weitere Funktionen benötigt und daher in der Variablen `$verbindung` gespeichert.

## Abfrage

```
/* SQL-Abfrage ausführen */
$res = mysqli_query($verbindung, "SELECT * FROM tabelle");
```

- Die Funktion `mysqli_query()` führt eine Abfrage mit der SQL-Anweisung `SELECT` in der aktuellen Datenbank aus. Die Abfrage soll alle Datensätze der betroffenen Tabelle liefern. Auch hier wird als erster Parameter die Referenz auf die Verbindung benötigt.
- Falls die Abfrage erfolgreich ist, liefert die Funktion eine Ergebniskennung zurück (hier in der Variablen `$res`). Diese Ergebniskennung wird anschließend benötigt, um die einzelnen Komponenten des Ergebnisses zu ermitteln. Sollten mehrere Abfragen erfolgen, solltest du für jedes Ergebnis eine eigene Variable zur Speicherung der Ergebniskennung verwenden.

# Anzahl Datensätze

```
/* Anzahl Datensätze ermitteln und ausgeben */
$num = mysqli_num_rows($res);
if($num > 0) echo "Ergebnis:<br>";
else echo "Keine Ergebnisse<br>";
```

- Die Funktion `mysqli_num_rows()` wird aufgerufen, wenn du die Anzahl der Datensätze im Abfrageergebnis wissen möchtest. Als Parameter wird die Ergebniskennung übergeben, deren Datensatzanzahl Sie ermitteln möchten.

# Datensätze ausgeben

## Möglichkeit 1

In diesem Beispiel hat die Tabelle fünf Spaltenattribute.

```
/* Datensätze aus Ergebnis ermitteln, */
/* in Array speichern und ausgeben */
while ($dsatz = mysqli_fetch_assoc($res))
{
echo $dsatz["name"] . ", "
. $dsatz["vorname"] . ", "
. $dsatz["personalnummer"] . ", "
. $dsatz["gehalt"] . ", "
. $dsatz["geburtstag"] . "<br>";
}
```

- Die Funktion `mysqli_fetch_assoc()` wird verwendet, um einen Datensatz des Ergebnisses zu ermitteln und ihn in einem assoziativen Feld (hier `$dsatz`) zu speichern. Dabei stellt der Datenbankfeldname den Schlüssel des Felds dar. Die Funktion führt dazu, dass ein sogenannter Datensatzzeiger auf den nächsten Datensatz des Ergebnisses gesetzt wird.
- Die Zuweisung des Datenbankfelds an das assoziative Feld `$dsatz` wird gleichzeitig dazu verwendet, eine `while`-Schleife zu steuern. Die Schleife dient dazu, alle Datensätze des Ergebnisses auszugeben. Falls das Ergebnis aus mindestens einem Datensatz besteht, ist die Zuweisung `$dsatz = mysqli_fetch_assoc($res)` ein wahrer Ausdruck. Daher wird die `while`-Schleife durchlaufen.
- Beachte: Es handelt sich nicht um einen Vergleich, sondern um eine Zuweisung! Es werden also zwei Anweisungen in einem ausgeführt: zuerst die Zuweisung des Felds und anschließend die `while`-Anweisung.
- Der Datensatzzeiger wird durch den wiederholten Funktionsaufruf irgendwann am Ende des Abfrageergebnisses anlangen. Die Funktion liefert in diesem Fall keinen weiteren Datensatz mehr. Damit wird die Zuweisung `$dsatz = mysqli_fetch_assoc($res)` ein unwahrer Ausdruck und die `while`-Schleife beendet.
- Innerhalb der Schleife wird jeweils der Inhalt eines Elements des Felds `$dsatz` ermittelt und ausgegeben. **Die Namen der Schlüssel müssen denen der Datenbankfeldnamen entsprechen.**

## Möglichkeit 2

```
//Ergebnis ausgeben
echo "<table>\n";
$satz = mysqli_fetch_array($res, MYSQL_NUM);
while ($satz)
{
    echo "\t<tr>\n";
    foreach ($satz as $feld)
    {
        echo "\t\t<td>".$feld."</td>\n";
    }
    echo "\t</tr>\n";
    $satz = mysqli_fetch_array($res, MYSQL_NUM);
}
echo "</table>\n";
$anzahl=mysqli_num_rows($res);
echo "Anzahl der Datensätze: $anzahl";
```

## Letzte ID

Mit dieser Abfrage lässt sich die letzte ID abfragen. Das macht vorallem dann Sinn, wenn die ID inkrementiert wird (Auto Increment), also schrittweise hochgezählt wird.

```
//SQL Abfrage Letzte ID
$abfrageLetzteID = "SELECT LAST_INSERT_ID()";
$ergebnisLetzteID = mysqli_query($verbindung, $abfrageLetzteID) or
die("Tabellenabfrage
fehlgeschlagen! " . mysqli_error($verbindung));
//Ergebnis ausgeben
echo "<table>\n";
$satz = mysqli_fetch_array($ergebnisLetzteID, MYSQL_NUM);
while ($satz)
{
    echo "\t<tr>\n";
    foreach ($satz as $feld)
    {
        echo "\t\t<td>Dein Buchungscode: ".$feld."</td>\n";
    }
    echo "\t</tr>\n";
    $satz = mysqli_fetch_array($ergebnisLetzteID, MYSQL_NUM);
}
echo "</table>\n";
```

# IMPLODE

Mit `implode` kannst du Array-Elemente zu einer Zeichenkette verbinden.

```
$abfrage = "SELECT spaltenattribut FROM tabelle WHERE bedingungIstErfüllt;
           $ergebnis = mysqli_query($verbindung, $abfrage) or
die("Tabellenabfrage
fehlgeschlagen! " . mysqli_error($verbindung));
$satz = mysqli_fetch_array($ergebnis, MYSQL_NUM);

//implode verbindet Array-Elemente zu einer Zeichenkette
           $neueZeichenkette = implode(", ", $satz);

//Die Variable ($neueZeichenkette) kann dann wie gewohnt mit der echo-
Anweisung ausgegeben werden.
```

## Verbindung schließen

Die Verbindung zur Datenbank wird durch den Aufruf der Funktion `mysqli_close()` wieder geschlossen.

```
/* Verbindung schließen */
mysqli_close($verbindung);
```

[[informatik](#), [thema](#), [php](#), [html](#), [lernpfad](#), [datenbank](#)]

From: <https://herr-pfeiffer.de/unterrichtswiki/> - Unterrichtswiki - Herr Pfeiffer

Permanent link: <https://herr-pfeiffer.de/unterrichtswiki/informatik:php:php-programmierkurs:php-p7?rev=1522067389>

Last update: 2019/12/16 21:53

