

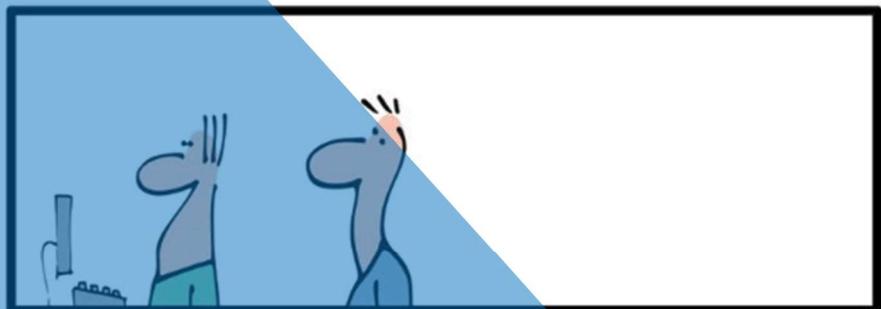
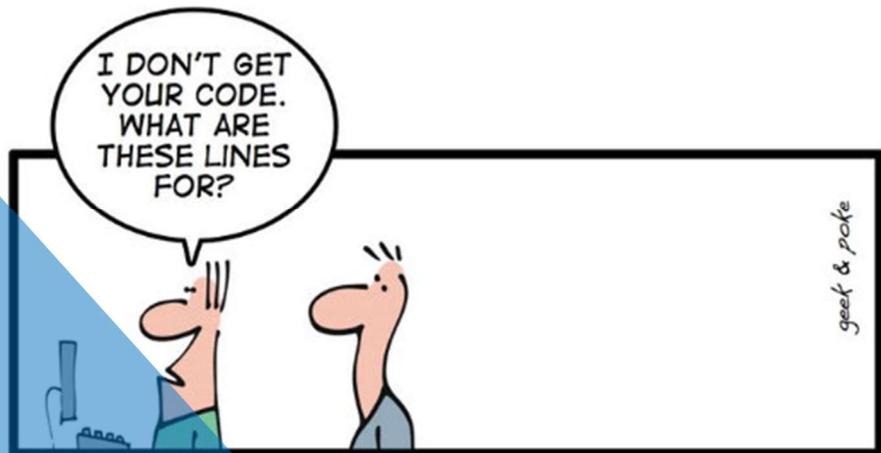


Auktionsbeispiel

einfache Datenbank in Webanwendung einbinden

Gerd Gesell

Version 2.0





Inhalt

1 Ziel	3
2 Datenbasis	4
3 Codebasis für die 1. Version	5
3.1 Details im Basiscode	5
3.2 Codeerweiterung für die 2. Variante	6
3.3 Codeerweiterung für die 3. Variante – mit Link für die Detailansicht	6
3.4 Erweiterungsmöglichkeiten:	9
4 Abgrenzung	10

1 Ziel

Dies ist eine Anleitung, um eine einfache Webapplikation mit Datenbankanbindung zu erstellen. Es sollen Daten aus einer Datenbank wahlweise als Tabelle oder als einzelner Datensatz in einem Browserfenster (html-Dokument) dargestellt werden. Dabei soll es möglich sein, aus der Tabelle einen Datensatz auszuwählen und diesen dann einzeln und detaillierter auf einer neuen Seite zu präsentieren (siehe Dialogbeispiel weiter unten).

Die „Applikation“ besteht am Anfang nur aus einer Tabelle, aber an ihr können Sie die Einzelschritte einfach nachvollziehen. Als Beispiel dient eine Internetauktion. Für die generelle Anbindung PHP und MySQL reicht das aus. Sie können das Beispiel ja nach eigenen Wünschen weiter ausbauen (Datenerfassung, weitere Tabellen usw.).



2 Datenbasis

Erstellen Sie eine Datenbank mit dem Namen «auktion». Es geht vermutlich am besten, wenn Sie ein GUI-Tool wie zum Beispiel PHPMYAdmin verwenden.

Field	Type	Null	Key	Default	Extra
artikel_id	int(11)	NO	PRI	NULL	auto_increment
artikel_titel	varchar(50)	NO		NULL	
artikel_beschreibung	text	YES		NULL	
preis_min	decimal(10,2)	YES		NULL	
time_end	datetime	YES		NULL	
time_start	datetime	YES		current_timestamp()	

Sie können die Tabelle Artikel entweder selber erstellen oder den folgenden SQL-Code verwenden:

```
CREATE TABLE artikel (  
    artikel_id int(11) NOT NULL AUTO_INCREMENT,  
    artikel_titel varchar(50) NOT NULL,  
    artikel_beschreibung text DEFAULT NULL,  
    preis_min decimal(10,2) DEFAULT NULL,  
    time_end datetime DEFAULT NULL,  
    time_start datetime DEFAULT current_timestamp(),  
    PRIMARY KEY (artikel_id)  
) ENGINE=MyISAM
```

Damit Sie nicht nur die leere Datenbank haben hier noch ein paar Daten:

```
INSERT INTO `artikel` (`artikel_id`, `artikel_titel`,  
`artikel_beschreibung`, `preis_min`, `time_end`,  
`time_start`) VALUES  
(1, 'Laptop', 'Yoga 2', '550.00', '2020-05-01 13:00:00',  
'2020-10-07 00:00:00'),  
(2, 'Boot', 'Ruderboot', '230.00', '2019-05-28 14:30:41',  
'2019-05-18 14:30:38'),  
(3, 'Festset Garten', '1 Klapptisch Kiefer und 2 Bänke  
Kiefer; Metallgestell', '50.00', '2019-12-09 00:00:00',  
'2020-03-09 00:00:00'),  
(4, 'Holzkohlegrill Gigant', 'Ständer mit grosser  
Feuerschale', '290.00', '2020-04-01 00:00:00', '2020-07-  
30 00:00:00'),  
(5, 'Thermometer', 'Satz von 5 Aussenthermometern und  
einer Zentralstation für USB-Anschluss', '55.00', '2020-  
01-23 08:00:00', '2020-04-30 17:00:00');
```

3 Codebasis für die 1. Version

Erzeugen Sie ein Unterverzeichnis im htdocs-Verzeichnis ihres Webservers mit der Bezeichnung `auktion` und legen darin eine Textdatei mit der Bezeichnung `index.php` ab.

Hinweis für Anwender von XAMPP: Sie finden diesen Ordner unter `.../xampp/htdocs`. Darin können Sie sämtliche Webauftritte ablegen. Wenn Sie den Apache Webserver gestartet haben, können Sie im Browser als URL `localhost/auktion` eingeben und sollten ein erstes Resultat sehen.

```

<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Auktion</title>
</head>
<body>
  <h1>Artikelliste</h1>
  <?php
    //DB-Verbindung herstellen
    $linkID = mysqli_connect('localhost', 'root', '',
'auktion') or die ("Ging nicht");
    //Abfrage absetzen
    $result = mysqli_query( $linkID, "SELECT artikel_titel,
preis_min, time_end FROM artikel ORDER BY time_end ASC");
    //Ausgabe des Resultats Variante 1
    print "<h2>Variante1</h2>";
    while($zeile = mysqli_fetch_object($result)){
      print "$zeile->artikel_titel, Preis: CHF $zeile-
>preis_min, $zeile->time_end<br>\n";
    } // ende while
    //Ende Ausgabe des Resultats
  ?>
</body>
</html>

```

Das Ergebnis ist nicht besonders schön, aber es zeigt immerhin schon etwas dynamisch an!

Vorausgesetzt es läuft der Webserver und der Datenbankserver.

3.1 Details im Basiscode

```

//DB-Verbindung herstellen
$linkID = mysqli_connect('localhost', 'root', '',
'auktion') or die ("Ging nicht");

```



Nach dem üblichen HTML-Code wird hiermit die Verbindung zur Datenbank aufgebaut. Dazu benötigen Sie

- die IP des Datenbankservers (oder eben «localhost»),
- den Benutzernamen und das Passwort (als root-User und ohne Passwort geht natürlich nur in der Entwicklungsumgebung)
- und den Datenbanknamen.

Falls etwas schiefgeht – Sie können ja zum Test mal den Datenbanknamen ändern – wird die Meldung nach `or` die ausgegeben. Diese Verbindung wird dann in einer Variablen gespeichert – hier `$linkID`.

```
$result = mysqli_query( $linkID, "SELECT artikel_titel,
preis_min, time_end FROM artikel ORDER BY time_end ASC");
```

Mit dieser Zeile führen wir eine SQL-Abfrage aus. Meistens wird das SQL-Statement selber auch noch in einer Variablen gespeichert. Darauf verzichten wir hier. Das Ergebnis – null, einer oder viele Datensätze – wird in der Variablen `$result` abgespeichert.

In der `while`-Schleife werden dann die Datensätze aus der Ergebnisvariablen geholt und die Werte ausgegeben.

3.2 Codeerweiterung für die 2. Variante

Wir wollen im nächsten Schritt die Ausgabe etwas übersichtlicher gestalten und geben die Ergebnisse in einer HTML-Tabelle aus:

```
//Ausgabe des Resultats Variante 2
print "<h2>Variante2</h2>";
print "<table border>";
print "<th>Artikel</th><th>Preis</th><th>Gueltigkeit</th>";
while($zeile = mysqli_fetch_assoc($result)){
    print "<tr>";
    print
"<td>$zeile[artikel_titel]</td><td>$zeile[preis_min]
    </td><td>$zeile[time_end]</td>\n";
    print "</tr>";
} // ende while
print "</table>";//Ende Ausgabe des Resultats
```

3.3 Codeerweiterung für die 3. Variante – mit Link für die Detailansicht

In dieser Tabelle stellen wir ja nur wenige Informationen pro Artikel zur Verfügung. Wir wollen nun je Zeile einen Link einfügen, mit dem wir dann auf eine neue Seite gelangen, auf der alle Informationen zu einem Artikel angezeigt werden.

```
//Ausgabe des Resultats Variante 3
```

```

print "<h2>Variante3</h2>";
print "<table border>";
print
"<th>Angebot</th><th>Preis</th><th>Gueltigkeit</th><th>View<
/th>";
while($zeile = mysql_fetch_array($result)){
    print "<tr>";
    print "<td>$zeile[angebot_titel]</td><td>$zeile[preis_min]
        </td><td>$zeile[time_end]</td>";
    print "<td><a href =
\"./artikel_details.php?angebot_id=$zeile[angebot_id]\">
        Details</a></td>\n";
    print "</tr>";
} // ende while
print "</table>";

```

Es sieht zwar schon toll aus, funktioniert aber noch nicht, da wir die neue Seite `artikel_details.php` erst erzeugen müssen.

Erstellen Sie in einem Editor also eine neue Datei mit der Endung `php` und versuchen Sie aus der Version 1 dieses Beispiels selber alle Daten eines Artikels anzuzeigen – oder nehmen Sie den folgenden Code, aber versuchen Sie ihn zu verstehen!

```

<!DOCTYPE html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Auktion - Artikel</title>
</head>
<body>
<h2>Artikel Details</h2>
<?php
//Ueberprüfen, ob die Variable $angebot_id gesetzt ist,
sonst Fehlermeldung und Abbruch des Scripts
$artikel_id = $_GET['artikel_id'];
if(!isset($artikel_id)){
    print "artikel_id nicht gesetzt<br>";
    exit();
}
//DB-Verbindung herstellen
$linkID = mysqli_connect('localhost', 'root', '', 'auktion')
or die ("Ging nicht");

```



```
//Abfrage absetzen, dabei die übergebene Variable in der
WHERE-Clausel als Kriterium //einsetzen
$result = mysqli_query($linkID, "SELECT * FROM artikel WHERE
artikel_id = $artikel_id");
print "<h2>Artikel</h2>";
//Ueberprüfen, ob ein Resultat vorliegt, Ausgabe in Tabelle,
sonst Fehlermeldung.
//Liegt kein Resultat vor, liefert die Funktion
mysql_fetch_array() 0 zurück = false
if($zeile = mysqli_fetch_assoc($result)){
    print "<TABLE border>";
    print "<TR>";
    print
"<td><b>Artikelnummer</b></td><td>$zeile[artikel_id]</td>";
    print "</TR>";
    print "<TR>";
    print
"<td><b>Artikel</b></td><td>$zeile[artikel_titel]</td>";
    print "</TR>";
    print "<TR>";
    //nl2br ersetzt Newlines durch <br>
    print
"<td><b>Beschreibung</b></td><td>$zeile[artikel_beschreibung
]</td>";
    print "</TR>";
    print "<TR>";
    print "<td><b>Mindestpreis</b></td><td>$zeile[preis_min]
</td>";
    print "</TR>";
    print "<TR>";
    print
"<td><b>Erfassungsdatum</b></td><td>$zeile[time_start]
</td>";
    print "</TR>";
    print "<TR>";
    print "<td><b>Gültigkeitsdatum</b></td><td>$zeile[time_end]
</td>";
    print "</TR>";
}
else{
    print "kein Datensatz vorhanden<br>";
}
```

```
print "</TABLE>";
?>
<p><a href="./index.php">zurück zur Liste</a></p>
</body>
</html>
```

3.4 Erweiterungsmöglichkeiten:

- Erstellen Sie einen weiteren Link (wie den «Detail»-Link) mit dem Sie Datensätze löschen können.
- Sie können ein Eingabeformular einbauen, mit dem neue Artikel erfasst werden können.
- Anschliessend können Sie über dieses Formular vorhandene Artikel verändern – editieren.

Dann haben Sie im Prinzip bereits eine CRUD-Applikation erstellt. CRUD steht für CREATE, READ, UPDATE, DELETE. Das sind die typischen immer wiederkehrenden Funktionen in Datenbankapplikationen. Viele Frameworks bieten bereits Automatismen an, um von einem vorhandenen Datenmodell automatisch den Code für CRUD-Applikationen zu erstellen.



4 Abgrenzung

Sie können dieses kleine Beispiel natürlich nach Belieben weiter ausbauen und bei Bedarf sogar die Tabellenstruktur erweitern. Sie müssen sich aber klar machen, dass es sich hierbei nur um ein Lernbeispiel handelt. Für eine produktive Applikation setzt man immer eine Softwarearchitektur ein, die verschiedene Layer beinhaltet und den Code sauber aufteilt und nicht in einer Datei vermischt. Auch zum Thema Sicherheit ist hier nichts gesagt und gemacht, sondern es geht nur darum an einem kleinen Beispiel den funktionalen Einsatz von Applikation und Datenbank zu zeigen.

Auch die Fehlertoleranz des Beispielcodes ist nicht wirklich beispielhaft. Die verwendete Funktion `or die` sollte durch die Verwendung einer Error-Nummer ersetzt werden.