

MySQL-Abfragen

Abfrage senden

Zur Abfrage der Tabellendaten einer bestimmten Datenbank sind die nachfolgenden Befehle notwendig.

mysql_select_db();	Auswahl der Datenbank
mysql_query();	Senden einer MySQL-Anfrage zur aktiven Datenbankverbindung
mysql_db_query();	Auswahl der Datenbank und Senden einer SQL-Anweisung

Nach einer erfolgreichen Verbindung zum MySQL-Server bestimmen Sie mithilfe von `mysql_select_db()`, welche Datenbank genutzt werden soll. Eine Abfrage einzelner Tabellen und deren Werte realisieren Sie über den Befehl `mysql_query()`.

Syntax der `mysql_select_db()`-Anweisung

```
mysql_select_db(Datenbankname [, Verbindungskennung]);
```

- Die Auswahl der zu nutzenden Datenbank erfolgt über deren Namen als Parameterangabe Datenbankname.
- Bestehen über `mysqlconnect()` mehrere Verbindungen zur Datenbank, ist optional die entsprechende Verbindungskennung anzugeben.
- Der Rückgabewert ist je nach Erfolg TRUE oder FALSE.

Syntax der `mysql_query`-Anweisung

```
mysql_query(SQL-Abfrage [, Verbindungskennung]);
```

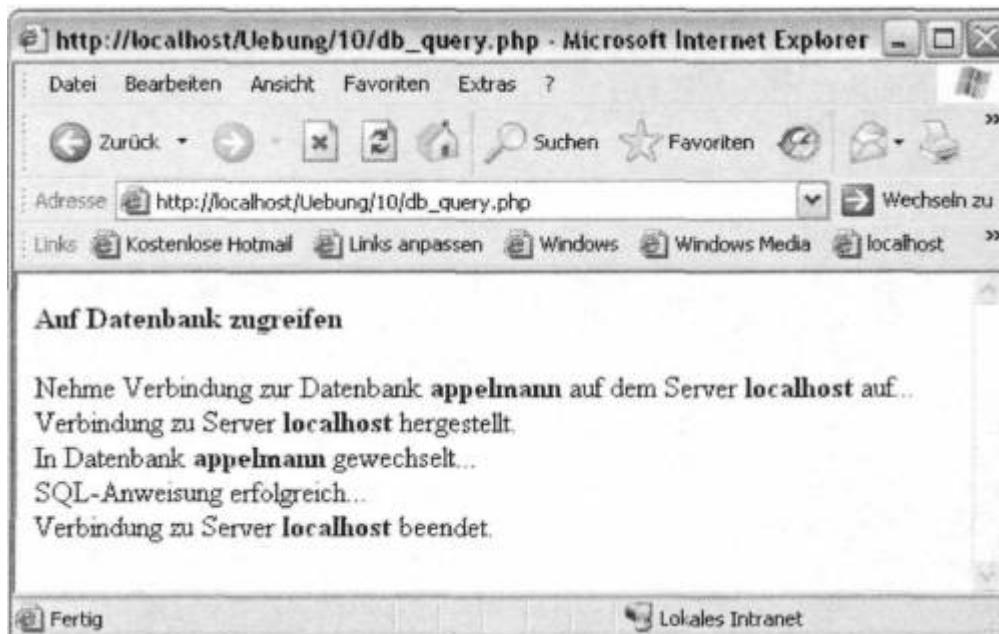
- Eine Abfrage der Daten über die Befehle der SQL-Sprache realisieren Sie über den PHP-Befehl `mysql_query()`.
- Die entsprechende Verbindungskennung ist anzugeben, wenn mehrere Verbindungen zu einem MySQL-Server bestehen. Geben Sie diesen Parameter nicht an, wird die zuletzt aktive Verbindung genutzt.
- Besteht keine Verbindung zum MySQL-Server, versucht PHP, automatisch den Befehl `mysqlconnect()` ohne Parameter-Angaben auszuführen.
- Als Rückgabewert erhalten Sie keine Informationen zur Anzahl der zurückgelieferten Daten, sondern nur, ob die Ausführung des Befehls erfolgreich war (TRUE) oder nicht (FALSE). Die Rückgabe von false erfolgt, wenn ein Fehler in der SQL-Syntax aufgetreten ist, eine angeforderte Tabelle nicht vorhanden ist oder der Zugriff auf eine Tabelle verwehrt wird.

Beispiel: db_query.php

In diesem Beispiel wird eine Verbindung zur Datenbank aufgenommen und eine SQL-Anweisung ausgeführt. Zur Veranschaulichung wird jede Aktion als Meldung im Browser ausgegeben.

```
<?php
$server = "localhost";
$user = "root";
$passwort = "";
$database = "appelmann";
$table = "bestellung";
echo "<br>Nehme Verbindung zur Datenbank <b>$database</b> auf... ";
$verbindung = @mysql_connect($server,$user,$passwort)
            or die("Konnte Verbindung zum Server <b>$server</b> nicht
herstellen!");
echo "<br>Verbindung zu Server <b>$server</b> hergestellt.";
mysql_select_db($database,$verbindung)
            or die("Fehler beim Zugriff auf die Datenbank $database!");
echo "<br>In Datenbank <b>$database</b> gewechselt... ";
$sql = "SELECT * FROM $table";
if(mysql_query($sql))
    echo "<br>SQL-Anweisung erfolgreich... ";
else
    echo "<br>SQL-Anweisung fehlgeschlagen... ";
mysql_close($verbindung);
echo "<br>Verbindung zu Server <b>$server</b> beendet.";
?>
```

- Um auf die Tabelle „bestellung“ der Datenbank appelmann zuzugreifen, werden zu den bereits bekannten Variablen zusätzlich die Variablen \$database und \$table definiert.
- Es wird versucht, eine Verbindung zum SQL-Server herzustellen. Durch die Angabe des Zeichens @ werden alle Fehlermeldungen am Bildschirm unterdrückt. Kommt keine Verbindung zu Stande, wird das Skript mit einer Fehlermeldung abgebrochen.
- Nachdem die Verbindung zum MySQL-Server besteht, wird in die Datenbank appelmann gewechselt. Existiert diese Datenbank nicht, wird das Skript vorzeitig beendet.
- Es folgt die Abfrageanweisung in der Sprache SQL. Sie können die Übersichtlichkeit erhöhen, wenn Sie die SQL-Anweisungen separat in einer Variablen speichern, statt diese direkt in der PHP-Funktion mysql_query() anzugeben.
- Die Anweisung wird über die Variable der Funktion mysql_query() übergeben. Je nachdem, ob die Anweisung erfolgreich ausgeführt werden konnte oder nicht, wird eine entsprechende Meldung angezeigt.
- Zum Schluss wird die Verbindung zum MySQL-Server wieder geschlossen.



Fehler einer SQL-Anfrage anzeigen

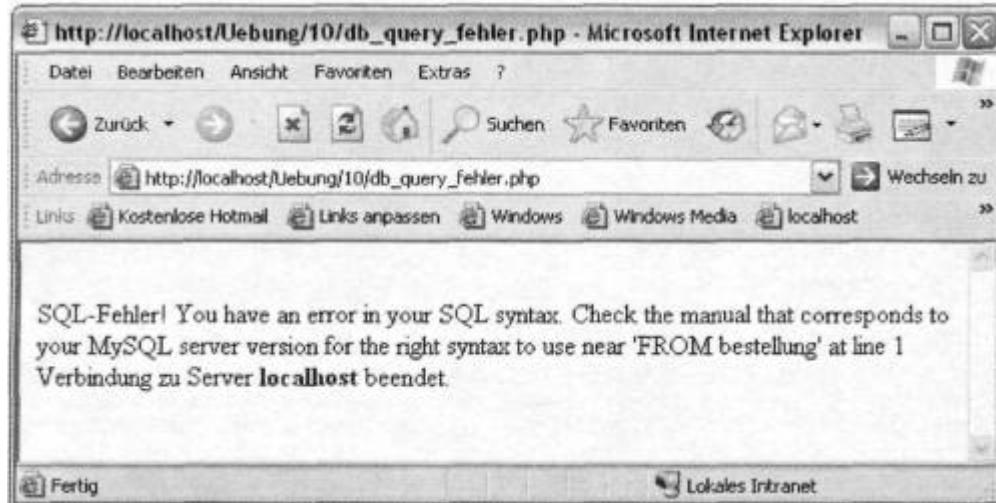
mysql_error(); Ausgabe eines SQL-Fehlers

Eine fehlerhafte SQL-Anweisung, die über die Funktion `mysql_query()` an die Datenbank übergeben wird, gibt als Resultat den Wert FALSE zurück. Um zu erfahren, welcher Fehler aufgetreten ist, verwenden Sie die Funktion `mysql_error()`. Diese leitet die Fehlerrückgabe des MySQL-Servers an das PHP-Skript weiter.

Beispiel: Ausschnitt aus db_query_fehler.php

In diesem Ausschnitt eines PHP-Skripts soll eine Fehlerrückgabe ausgelöst werden. Hierfür wurde bewusst ein syntaktischer Fehler in die SQL-Abfrage eingefügt.

```
$sql = "SELECT FROM $table";
if(mysql_query($sql))
    echo "<br>SQL-Anweisung erfolgreich...";
else
    echo "<br>SQL-Fehler!".mysql_error();
```



From:
<http://elearn.bgamstetten.ac.at/eduwiki/> - **EduDokuWiki**

Permanent link:
http://elearn.bgamstetten.ac.at/eduwiki/doku.php?id=inf:mysql:mysql_abfragen

Last update: **2013/04/27 09:04**