**Lektion 19: Daten in eine Datenbank einfügen**

In dieser Lektion schauen wir uns an, wie man direkt aus PHP-Scripten Daten in Datenbanken einfügen kann.

**Daten mit SQL einfügen**

Sie benutzen SQL auf die gleiche Weise, um Daten in eine Datenbank einzufügen, wie Sie dies für das Erstellen der Datenbank und Tabellen getan haben. Die Syntax der SQL-Query ist:

INSERT INTO Tabellenname(Spalte1, Spalte2, ...) VALUES(Wert1, Wert2, ...)

Wie Sie sehen, können Sie mehrere Spalten in einem Statement aktualisieren, indem Sie diese mit Komma getrennt auflisten. Es ist aber natürlich auch möglich, nur eine Spalte mit einem Wert festzulegen. Die Spalten, die im SQL-Statement nicht erwähnt werden, bleiben einfach leer.

**Beispiel: Eine neue Person in die Tabelle einfügen**

In diesem Beispiel benutzen wir die Datenbank aus [Lektion 18](http://de.html.net/tutorials/php/lesson18.php). Nehmen wir an, wir möchten der Datenbank eine Person hinzufügen. Dies könnte z.B. eine Person namens *Franz Gans* mit der Telefonnummer *99887766* und *1964-04-20* als Geburtsdatum.

Das SQL-Statement sähe dann wie folgt aus:

$strSQL = "INSERT INTO Personen(Vorname,Name,Telefon,Geburtsdatum) VALUES('Franz','Gans','99887766 ','1964-04-20')";

mysql\_query($strSQL) or die(mysql\_error());

Wie Sie sehen, können die SQL-Statements ziemlich lang werden und man kann ziemlich schnell den Überblick verlieren. Darum kann es von Vorteil sein, lange Statements in einer leicht abgewandelten Weise aufzuschreiben:

strSQL = "INSERT INTO Personen(";

strSQL = strSQL . "Vorname, ";

strSQL = strSQL . "Name, "

strSQL = strSQL . "Telefon, ";

strSQL = strSQL . "Geburtsdatum) ";

strSQL = strSQL . "VALUES (";

strSQL = strSQL . "'Franz', ";

strSQL = strSQL . "'Gans', ";

strSQL = strSQL . "'99887766', ";

strSQL = strSQL . "'1964-04-20')";

mysql\_query($strSQL) or die(mysql\_error());

So ist das SQL-Statement aus mehreren kleineren Stücken aufgebaut, die in der Variable **$strSQL** zusammengeführt werden.

In der Praxis macht es keinen Unterschied, welche Methode Sie wählen. Wenn Sie aber beginnen, mit größeren Tabellen zu arbeiten, ist es wichtig, dass Sie immer den Überblick behalten. Suchen Sie sich also die Vorgehensweise aus, mit der Sie am besten zurechtkommen.

Versuchen Sie mal, den nachfolgenden Code auszuführen und Franz Gans in die Datenbank einzufügen:

<html>

<head>

<title>Daten in eine Datenbank einfügen</title>

</head>

<body>

<?php

// mit dem Datenbankserver verbinden

mysql\_connect("mysql.ihrserver.de", "benutzername", "ihrpasswort") or die (mysql\_error ());

// Datenbank auswählen

mysql\_select\_db("ihredatenbank") or die(mysql\_error());

// das SQL-Statement wird aufgebaut

$strSQL = "INSERT INTO Personen(";

$strSQL = $strSQL . "Vorname, ";

$strSQL = $strSQL . "Name, ";

$strSQL = $strSQL . "Telefon, ";

$strSQL = $strSQL . "Geburtsdatum) ";

$strSQL = $strSQL . "VALUES(";

$strSQL = $strSQL . "'Franz', ";

$strSQL = $strSQL . "'Gans', ";

$strSQL = $strSQL . "'99887766', ";

$strSQL = $strSQL . "'1964-04-20')";

// das SQL-Statement wird ausgeführt

mysql\_query($strSQL) or die (mysql\_error());

// schließe die Datenbankverbindung

mysql\_close();

?>

<h1>Die Datenbank wurde aktualisiert!</h1>

</body>

</html>

**Speichern von Nutzereingaben in einer Datenbank**

Oft möchte man Eingaben vom Benutzer in eine Datenbank speichern.

Wie Sie sich vielleicht schon denken können, kann man dies mit einem Formular, wie in [Lektion 11](http://de.html.net/tutorials/php/lesson11.php) beschrieben, machen - wo die Werte aus den Formularfeldern in das SQL-Statement eingefügt werden. Stellen Sie sich vor, Sie haben ein einfaches Formular wie dieses:

<form action="einfuegen.php" method="post">

<input type="text" name="Vorname" />

<input type="submit" value="Speichern" />

</form>

Das Formular übermittelt seine Angaben an die Datei **einfuegen.php**, wo man, wie in Lektion 11 gezeigt, mit den Nutzereingaben weiterarbeiten kann. In diesem speziellen Beispiel könnte ein SQL-Statement so aussehen:

strSQL = "INSERT INTO Personen(Vorname) values('" . $\_POST["Vorname"] . "')"

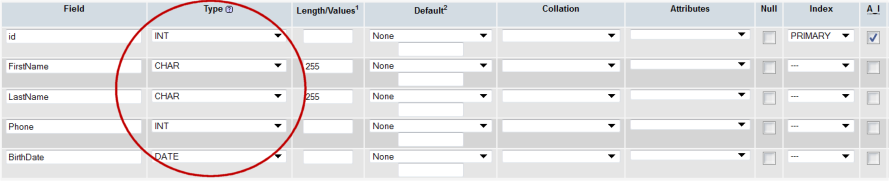
Auf die gleiche Weise kann man auch Daten aus Cookies, Sessions, Query-Strings usw. weiterverarbeiten.

**"Beliebteste" Anfängerfehler**

Am Anfang werden Sie wahrscheinlich viele Fahlermeldungen bekommen, wenn Sie versuchen Ihre Datenbank zu aktualisieren. Aber es ist kein Platz für den kleinsten Fehler, wenn Sie mit Datenbanken arbeiten. Ein Komma an der falschen Stelle und Ihre Datenbank wird nicht aktualisiert, sondern Sie bekommen stattdessen eine Fehlermeldung. Nachfolgend beschreiben wir Anfängerfehler, die immer wieder vorkommen.

**Falsche Datentypen**

Es ist wichtig, dass Art der Daten und der Typ, der der Spalte zugewiesen wurde, übereinstimmen. Jede Spalte kann einen Datentyp zugewiesen bekommen. Der folgende Screenshot zeigt die Datentypen für die Tabelle "Personen" unseres Beispiels.



Ein Fehler passiert, wenn Sie z.B. versuchen Text oder Zahlen in ein Datumsfeld einzufügen. Versuchen Sie deshalb die Datentypen so genau wie möglich festzulegen.

Nachfolgend nochmal die gebräuchlisten Datentypen in einer Übersicht:

|  |  |  |
| --- | --- | --- |
| **Einstellung** | **Datentyp** | **Größe** |
| CHAR | Text oder Kombination aus Text und Zahlen. Kann auch für Zahlen benutzt werden, die nicht in Berechnungen verwendet werden (z.B. Telefonnummern). | Bis zu 255 Zeichen - oder die Länge, die bei "Länge/Length" definiert wurde. |
| TEXT | Längerer Text oder Kombination aus Text und Zahlen. | Bis zu 65.535 Zeichen. |
| INT | Numerische Daten für mathematische Berechnungen. | 4 Bytes. |
| DATE | Datum im Format JJJJ-MM-TT | 3 Bytes. |
| TIME | Zeit im Format hh:mm:ss | 3 Bytes. |
| DATETIME | Datum und Zeit im Format JJJJ-MM-TT hh:mm:ss | 8 Bytes. |

**SQL-Statements mit Anführungszeichen oder Backslash**

Wenn Sie versuchen, Text einzufügen, der die folgenden Zeichen enthält - einfache Anführungszeichen ('), doppelte Anführungszeichen (") oder Backslash (\) - dann werden Ihre Angaben nicht in die Datenbank eingefügt. Die Lösung hierfür ist, dass ein Backslash vor dem jeweiligen Zeichen in einer Query eingefügt wird.

Dieses kann mit der Function [[Dokumentation](http://php.net/manual/en/function.addslashes.php)addslashes](http://php.net/manual/en/function.addslashes.php), wie folgt, gemacht werden:

<?php

$strText = "Ist Ihr Name O'Reilly?";

$strText = addslashes($strText);

?>

Allen einfachen Anführungszeichen ('), doppelten Anführungszeichen (") und Backslashes (\) wird ein extra Backslash vorgestellt. Dies ist nur, um die Daten in die Datenbank zu bekommen. Das zusätzliche \ wird nicht mit eingefügt. Bitte beachten Sie, dass PHP [[Dokumentation](http://php.net/manual/en/function.addslashes.php)addslashes](http://php.net/manual/en/function.addslashes.php) standardmäßig bei allen $\_GET-, $\_POST- und $\_COOKIE-Daten ausführt. Verwenden Sie daher [[Dokumentation](http://php.net/manual/en/function.addslashes.php)addslashes](http://php.net/manual/en/function.addslashes.php) nicht auf Strings, wo diese Ersetzung (engl. Begriff "escaping a character") bereits vorgenommen wurde.

In der folgenden Lektion werden Sie lernen, wie Sie Daten aus Ihrer Datenbank abrufen. Probieren Sie aber erst noch, ein paar mehr Leute in Ihre Datenbank einzufügen (wie Franz Gans im obigen Beispiel).