

Interaktive Webseiten mit PHP und MySQL



Vorlesung 4: PHP & MySQL

Sommersemester 2003

Martin Ellermann

Heiko Holtkamp

MySQL

- Hier noch ein wenig zu (My)SQL:
 - SHOW
 - INSERT
 - SELECT
 - ORDER BY
 - GROUP BY
 - LIKE / BETWEEN / IN
 - Aliase
 - Funktionen
 - Joins
 - DELETE FROM
 - UPDATE
 - ALTER TABLE
 - Benutzer in MySQL

PHP & MySQL: Erste Schritte

- MySQL wird von PHP explizit unterstützt.
- Für die Nutzung stehen Funktionen zur Verfügung, die direkt auf die Datenbank zugreifen.
- Neben MySQL bietet PHP auch die Unterstützung für weitere Datenbanken, u.a.
 - mSQL
 - Postgres
 - Informix
 - MS SQL Server
 - Oracle
 - „ODBC“

PHP & MySQL: Erste Schritte

- Mit dem MySQL DBMS verbinden:

```
resource mysql_connect ([string Server[:Port] [, string  
Benutzername [, string Benutzerkennwort [, bool  
neue_Verbindung [, int client_flags]]]])
```

Beispiel:

```
<?php  
    $link = mysql_connect ("localhost", "mysql_user",  
        "mysql_password") or die ("keine Verbindung möglich");  
    print ("Verbindung erfolgreich");  
    mysql_close ($link);  
?>
```

- Verbindung zum MySQL DBMS schließen:

```
bool mysql_close ([resource Verbindungs-Kennung])
```

PHP & MySQL: Erste Schritte

- Persistente Verbindung zu MySQL:

```
resource mysql_pconnect ( [string Server[:port] [,  
string Benutztername [, string Benutzerkennwort [, int  
client_flags]]])
```

- vor dem Verbindungsaufbau wird zunächst versucht eine offene (persistente) Verbindung zum gleichen Host, mit dem gleichen Benutzernamen und Benutzerkennwort zu finden
- die Verbindung zum SQL Server wird beim Beenden des PHP-Skripts nicht geschlossen

- Auswahl einer MySQL Datenbank:

```
bool mysql_select_db ( string Datenbankname [, resource  
Verbindungs-Kennung])
```

PHP & MySQL: Erste Schritte

- Eine Anfrage an das MySQL DBMS senden:

```
resource mysql_query (string Anfrage [, resource  
Verbindungs-Kennung [, int Ergebnis-Modus]])
```

Beispiel:

```
<?php  
    $result = mysql_query ("SELECT * FROM my_tbl")  
        or die ("Invalid query");  
?>
```

- Daneben gibt es auch die Funktion:

```
resource mysql_db_query (string Datenbank, string  
Anfrage [, resource Verbindungs-Kennung])
```

- Diese Funktion sollte aber nach Möglichkeit nicht mehr verwendet werden (seit PHP 4.0.6 als veraltet gekennzeichnet)

PHP & MySQL: Erste Schritte

- Eine Zeile aus einer DB-Abfrage als indiziertes Array liefern:

```
array mysql_fetch_row ( resource Ergebnis-Kennung)
```

- Eine Zeile aus einer DB-Abfrage als assoziatives und indiziertes Array liefern:

```
array mysql_fetch_array ( resource Ergebnis-Kennung [, int Ergebnistyp])
```

```
<?php
    mysql_connect("localhost", "user", "password") or
        die("Keine Verbindung möglich");
    mysql_select_db("mydb");
    $result = mysql_query("SELECT id, name FROM mytable");
    while ($row = mysql_fetch_array($result)) {
        echo $row["id"];      // oder $row[0]
        echo $row["name"];   // oder $row[1]
    }
?>
```

- Eine Zeile aus einer DB-Abfrage als assoziatives Array liefern:

```
array mysql_fetch_assoc ( resource Ergebnis-Kennung)
```

PHP & MySQL: Erste Schritte

- Anzahl der Datensätze im Ergebnis:

```
int mysql_num_rows (resource Ergebnis-Kennung)
```

```
<?php
    $link = mysql_connect("localhost", "user", "password");
    mysql_select_db("database", $link);
    $result = mysql_query("SELECT * FROM table1", $link);
    $num_rows = mysql_num_rows($result);
    echo "$num_rows Rows\n";
?>
```

- Anzahl der betroffenen Datensätze einer vorhergehenden Operation:


```
int mysql_affected_rows ([resource Verbindungs-Kennung])
```

```
<?php
    /* ... */
    mysql_query("DELETE FROM mytable WHERE id < 10");
    printf ("Datensätze gelöscht: %d\n", mysql_affected_rows());
?>
```


PHP & MySQL: Erste Schritte

- `mysql_create_db`
- `mysql_errno`
- `mysql_error`
- `mysql_insert_id`

Nützliche Bibliotheken

- PHPLib – PHP Base Library 
 - <http://phplib.sourceforge.net> (früher: <http://phplib.netuse.de>)
 - PHPLib ist eine Sammlung von in PHP geschriebenen Klassen
 - Datenbankabstraktion (mSQL, MSSQL, MySQL, Oracle8, Postgres ...)
 - Sessionhandling
 - Authentifizierung
 - Rechteverwaltung
 - Templates
 - **Aber:** die PHPLib wird nicht mehr weiter entwickelt.
 - Entwicklung seit Version 7.2d im Stillstand
 - Dennoch eine der wichtigsten und immer noch am meisten genutzten PHP-Bibliotheken
 - Funktionalität der PHPLib wird schrittweise in das "PEAR-Projekt" integriert

PHPLib – Ein Beispiel: Datenbankabstraktion

- Verbindung zu MySQL (bzw. einem DBMS):

```
<?php
class Worksphere_DB extends DB_Sql {
    var $Host      = "localhost";
    var $Database  = "worksphere";
    var $User      = "heiko";
    var $Password  = "secret";
}
?>
```

- Ein Datenbankobjekt erzeugen:

```
$db = new Worksphere_DB;
```

- Eine Anfrage stellen:

```
$db->query("SELECT id, name FROM table");
```

Nützliche Bibliotheken



- PEAR – PHP Extension and Application Repository

- <http://pear.php.net>
- Klassen und Erweiterungen für PHP
 - "A structured library of open-sourced code for PHP users"
 - "A system for code distribution and package maintenance"
 - "A standard style for code written in PHP"
 - "The PHP Foundation Classes (PFC)"
 - "The PHP Extension Code Library (PECL)"
- In PHP und C programmiert
- Zur Zeit verfügbare Funktionen (Auswahl):
 - Authentifikation
 - Datenbankabstraktion
 - Verschlüsselungsfunktionen
 - Log-Mechanismen
 - Dateisystemfunktionen