



by M  YFLOWER

## Einführung in PHP 5 und IBM DB2

Webapplikationen mit PHP 5.2 und IBM DB2

PHP Conference 2006 - 08.11.2006



# Themenübersicht

---



by MAYFLOWER

- PHP 5.2 und Datenbanken
- PDO
- Warum IBM DB2?
- PHP und IBM DB2
- Installation / Konfiguration
- Vorstellung der ibm\_db2 Extension
- Funktionen und Besonderheiten



# Vorstellung

---



- Thorsten Rinne
- Jahrgang 1977
- Diplom-Informatiker (FH)
- Software-Entwickler bei der Mayflower GmbH
  - Ratinganwendungen nach Basel II
  - Businesskritische Bankapplikationen
  - Rating- und Reportinganwendungen
- PHP-Programmierung seit 1999
- Erster Kontakt mit IBM DB2 im Jahr 2002
- Hauptentwickler der Open Source FAQ-Managementsoftware **phpMyFAQ**
- Portierung phpMyFAQ auf IBM DB2 in 2005 mit inoffizieller Unterstützung von IBM

# PHP 5.2 und Datenbanken: Allgemeines



- PHP unterstützt alle wichtigen Datenbanken
  - MySQL
  - PostgreSQL
  - Microsoft SQL Server
  - Oracle
  - SQLite
  - Firebird / Interbase
  - Sybase
  - Informix
  - ODBC
  - ...und natürlich IBM DB2 ☺
- Nachteil: Jede Datenbank braucht eine eigene Extension



by MAYFLOWER

## PHP und Datenbanken: PHP 5.2

---

- PDO bietet eine gemeinsame und effiziente Schnittstelle
- Spezielle Extensions wie z.B. ext/mysqli
- Support für PEAR
- Unterstützung von
  - dateibasierter Datenbanken wie SQLite
  - verbinden mit internen und externen Datenbankservern
  - ODBC
- Neu in PHP 5.2
  - Filter,JSON und ZIP Extension
  - Neuer Memory Manager für die ZEND Engine
  - Viele Verbesserungen und Bugfixes

## Vorstellung: PDO

---

- Abstrahierung des Datenzugriffs (einheitliche API)
- Plugins für mehrere Datenbanksysteme
  - MySQL, PostgreSQL, SQLite, ODBC usw.
- Objektorientiert
- Support für Iteratoren
- Schnell, da in C programmiert
- PDO...
  - ... ist über PECL in PHP 5.0 einfügbar
  - ... ist seit PHP 5.1 fest integriert
  - ... Windows-DLLs sind verfügbar



by MAYFLOWER

## Die Features von PDO

---

- Unterstützung von Prepared Statements
- Rückgabe von SQL Fehlercodes
- Eigenschaften für Datenbank-Portabilität
- Support für Transaktionen
- Scrollable Cursors
- Unterstützt das PHP Error-Handling oder auch Exceptions
- Geplant: Unterstützung für LOBs



# Verwenden von PDO



by MAYFLOWER

```
<?php
try {
    $db = new PDO('odbc:DRIVER={IBM DB2 ODBC DRIVER};HOSTNAM
E=localhost;PORT=50000;DATABASE=FAQ;PROTOCOL=TCPIP;UID=db2ad
min;PWD=db2admin');
    $stmt = $db->prepare('SELECT * FROM faquser');
    $stmt->setFetchMode(PDO::FETCH_NAMED);
    $stmt->execute();
    while ($row = $stmt->fetch()) {
        print_r($row);
    }
    $dbh = null;
} catch (PDOException $e) {
    print "Error: " . $e->getMessage() . "<br/>";
    die();
}
```



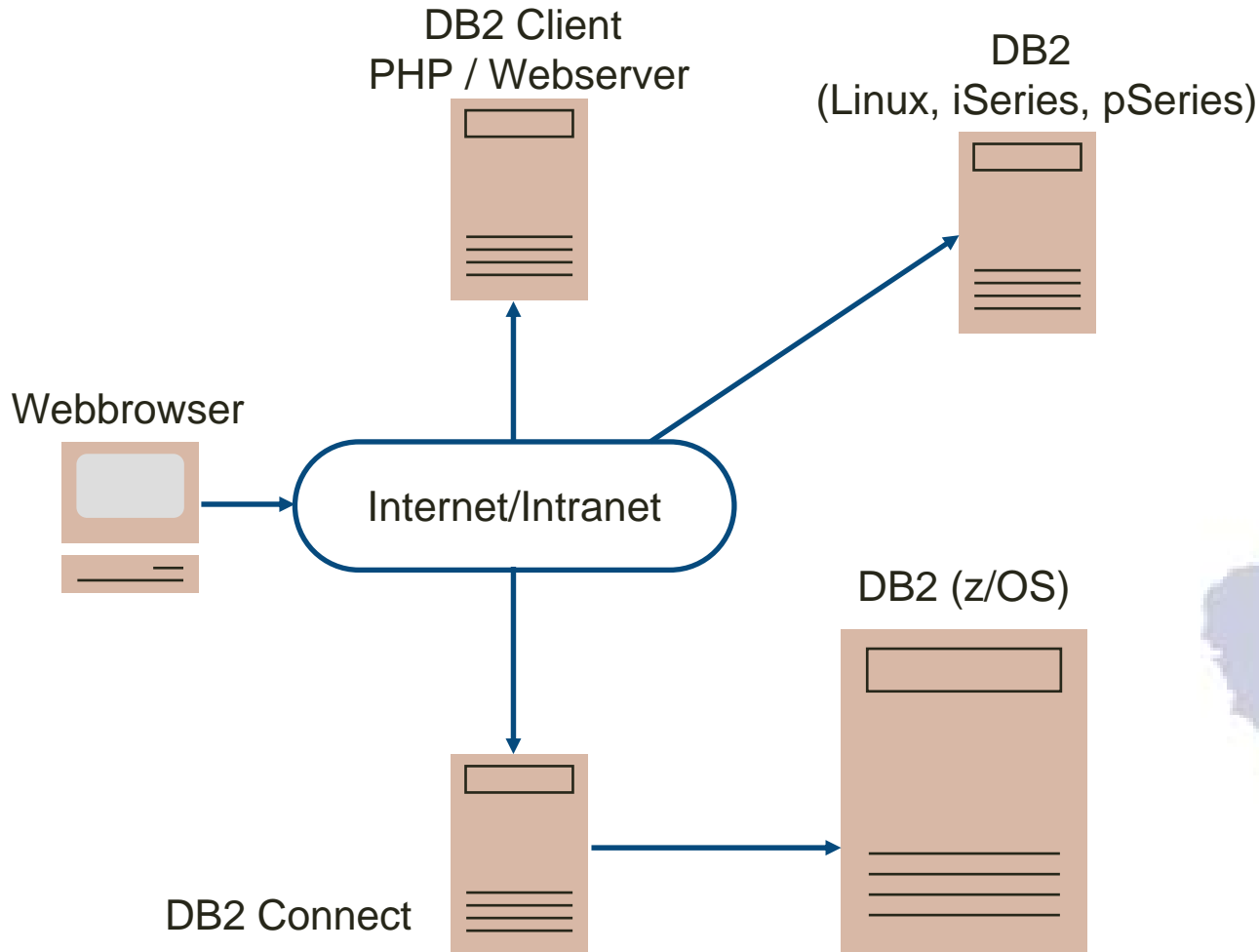
by MAYFLOWER

## Warum IBM DB2?

---

- IBM DB2 ist **die** Datenbank im Enterprise-Umfeld
- Unterstützt z/OS, iSeries, Linux, Unix und Windows
- Hochskalierbar
  - ... beginnend mit IBM DB2 Express-C für 2 CPUs
  - ... über Systeme mit mehr als 64 Prozessoren
  - ... bis Cluster mit bis zu 1024 Knoten
- Führend unter Linux bei TPC-H
- Aktuelle Version: IBM DB2 V.9.1
  - Für relationale und XML-Daten
  - Abfrage per SQL und XQuery
  - SAP zertifiziert

# IBM DB2 im Netz





by MAYFLOWER

## Konfiguration von IBM DB2

---

- Sicherstellung des korrekten Pfads zu Verzeichnis der Extensionen
- Anpassung der php.ini

```
extension=ibm_db2.so
```

bzw.

```
extension=ibm_db2.dll
```

- Einbinden der IBM DB2 Umgebung
- Anpassung der Apache Webserver Umgebung

# Zend Core for IBM DB2

---



- Kostenloser Download unter [www.zend.com](http://www.zend.com)
- Installiert folgende Komponenten
  - IBM DB2 Client
  - PHP
  - ibm\_db2 Extension
  - ODBC Extension
  - IBM DB2 Express-C
- Beinhaltet eine webbasierte Verwaltungsoberfläche
- Verfügbar für
  - Linux (x86, x64, PowerPC)
  - Windows
  - AIX
  - i5/OS

# PHP und IBM DB2

---



- PHP unterstützt drei Möglichkeiten
  - ODBC (ext/odbc)
  - Extension für IBM DB2 und Cloudscape
  - PDO\_ODBC
- Open Source
- Download: <http://www.php.net>

## PHP und IBM DB2: ODBC (ext/odbc)

- Im PHP Core integriert
- Normalerweise mit einem generischen ODBC-Treiber kompiliert
- wird mit der IBM DB2 Bibliothek kompiliert werden (./configure --with-ibm-db2)
- Nachteile:
  - Scrollable Cursors sind langsam
  - Kein Support für IN/INOUT Stored Procedures

# PHP und IBM DB2: ibm\_db2 Extension

---



- Erhältlich im PECL Repository
- Entwickelt und unterstützt von IBM
- Volle Unterstützung für Stored Procedures und LOBs
- Schnell!
- Nachteil:  
Relativ neue Extension, damit noch nicht 100%ig fehlerfrei

# PHP und IBM DB2: PDO\_ODBC



- Schnelle, leichte und C-basierte Standardschnittstelle für PHP
  - PDO\_ODBC wird direkt mit der generischen IBM DB2 Bibliotheken kompiliert
  - Standardisierte API vereinfacht die Software-Entwicklung
  - Schnelle objektorientierte Schnittstelle
- Nachteil:
  - Nur für PHP 5.x verfügbar
  - Trotz PDO 1.0 noch in der Entwicklung



by MAYFLOWER

## Erster Kontakt mit ibm\_db2

```
<?php
$database = "faq";
$user     = "db2admin";
$password = "db2admin";

$sql = "SELECT realname, email FROM faquser WHERE id
> ?";
$conn = db2_connect($database, $user, $password);
$stmt = db2_prepare($conn, $sql);
$res = db2_execute($stmt, array(0));
while ($row = db2_fetch_assoc($stmt)) {
    print $row['REALNAME'] . '<br />';
    print $row['EMAIL'] . '<br />';
}
```



by MAYFLOWER

## Erste Beobachtungen

---

- Die erste Verbindung dauert sehr lange ...
  - Bei der ersten Verbindung weist sich das System zunächst Systemressourcen zu
  - Lösung:

```
ACTIVATE DATABASE faq
```

- Persistente Verbindungen?

Gefährlich, aber nützlich bei lesenden Zugriffen



by MAYFLOWER

## Beenden der ersten Verbindung

---

- Beim Beenden des PHP-Skripts wird automatisch die Verbindung geschlossen
- Verbindungstrennung auch durch

```
db2_close( $conn );
```

```
?>
```



by MAYFLOWER

## Fehlerbehandlung mit ibm\_db2

---

### ■ Rückgabe des SQLSTATE Wertes

- `db2_conn_error()`

- `db2_stmt_error()`

### ■ Rückgabe der SQL-Fehlermeldung

- `db2_conn_errormsg()`

- `db2_stmt_errormsg()`





by MAYFLOWER

## Prepared Statements mit ibm\_db2

- Die ibm\_db2 Extension unterstützt Prepared Statements
- Mit dem Platzhalter „?“ werden die Werte an die gewünschte Position gebunden:

```
$sql = "SELECT id FROM faquser WHERE realname = '?'";  
$stmt = db2_prepare($conn, $sql);  
$res = db2_execute($stmt, array('Thorsten Rinne'));
```



by MAYFLOWER

## Zeilenweise Datenabfrage

---

- Indiziert mit der Spaltennummer

```
$row = db2_fetch_array($stmt);
```

- Indexiert mit dem Spaltennamen

```
$row = db2_fetch_assoc($stmt);
```

- Indiziert mit Spaltennummer und -name

```
$row = db2_fetch_both($stmt);
```

- PHP Objekt, Spaltennamen sind die Objekte

```
$row = db2_fetch_object($stmt);
```



by MAYFLOWER

## Unterschiede zu MySQL (I)

- IBM DB2 gibt bei SELECTs keine Anzahl der betroffenen Zeilen zurück
- Lösungen:

- Langsam: Scrollable Cursors

```
$stmt = db2_exec($conn, "SELECT realname FROM faquser", array('cursor' => DB2_SCROLLABLE));  
$num = db2_num_rows($stmt);
```

- Besser:

```
$stmt = db2_exec($conn, "SELECT COUNT(*) FROM faquser");  
list($num) = db2_fetch_array($stmt);
```

- Am besten:

```
$stmt = db2_exec($conn, "SELECT realname, realname FROM faquser");  
while ($row = db2_fetch_assoc($stmt)) { ... }
```



by MAYFLOWER

## Unterschiede zu MySQL (II)

- IBM DB2 unterstützt kein LIMIT
- Lösungen
  - ROW\_NUMBER OVER ( ) Syntax nach ANSI SQL
  - Scrollable Cursors
  - Sequenzen und automatisch erstellte Spalten für ein einfaches WHERE
- Spaltennamen sind standardmäßig immer in GROSSBUCHSTABEN:

```
print $row[ 'REALNAME' ] ;
```

```
print $row[ 'realname' ] ; // Notice!
```



by MAYFLOWER

## Abfragen per XQuery

- IBM DB2 V.9.1 speichert nativ Daten in XML
- Beispiel:

```
$xquery = '$document/customer/phone' ;  
  
$res = db2_exec($conn,  
"SELECT  
    XMLQUERY( '$xquery'  
PASSING  
    INFO AS \"document\" )  
FROM  
    COSTUMER " ) ;
```

# Noch Fragen?

---



by MAYFLOWER



„PHP 5 und IBM DB2“



by M  YFLOWER

**Vielen Dank für Ihre Aufmerksamkeit!**

Thorsten Rinne, Dipl.-Inf. (FH)

Mayflower GmbH

Sendlinger Str. 42a

80311 München

+49 (89) 24 20 54 – 31

[rinne@mayflower.de](mailto:rinne@mayflower.de)

