

# Einführung in LAMP

Johannes Franken  
<jfranken@jfranken.de>

## Inhalt

1. [Übersicht](#)
2. [MySQL](#)
  - a) [Relationale Datenbank-Management-Systeme \(RDBMS\)](#)
  - b) [\(My\)SQL-Befehle](#)
    - i) [Tabellen einrichten \(DDL, DCL, control\)](#)
    - ii) [Mit Tabellen arbeiten \(DML\)](#)
  - c) [Benutzerrechte](#)
3. [Apache](#)
  - a) [HTML](#)
  - b) [Webserver und HTTP](#)
  - c) [Apache](#)
    - i) [CGI-Skripte](#)
    - ii) [Module](#)
    - iii) [Virtual Hosting](#)
4. [PHP](#)
  - a) [Sprache](#)
  - b) [Kleine Datenbankanwendung](#)

## Übersicht

"LAMP" steht für

1. Linux
2. Apache
3. MySQL
4. PHP (oder Perl)

*Abbildung: LAMP-Architektur*

## MySQL

### Relationale Datenbank-Management-Systeme (RDBMS)

*Abbildung: MySQL-Server*

Vorteile DBMS vs. direktem Dateizugriff:

- Zentralisierung des Datenzugriffs (DBMS kümmert sich um Netzzugriff, Locks, Caches u.s.w. -- nicht jede Anwendung selbst)
- Höhere Funktionalität (SQL)
- Unabhängig von physischer Speicherung der Daten
- ...

Konzept RDBMS:

- Datenbanken
- Tabellen
- Spalten (=Eigenschaften =Attribute)
- Werte (=Eigenschaftswerte =Attributswerte)
- Zeilen (=Tupel =Datensätze)
- Referenzen (=Verweise =Fremdschlüssel)
- Abgeschlossenheit (Tabellendefinitionen und Zugriffsrechte innerhalb des DBMS)
- Abfragen (über spezielle Abfragesprache, z.B. SQL):
  - Selektion (zeilenweise)
  - Projektion (spaltenweise)
  - Join (Verbindung mehrerer Tabellen an Hand gemeinsamen Wertes)

**Mehr zum Thema "RDBMS":**

siehe [http://de.wikipedia.org/wiki/Relationale\\_Datenbank](http://de.wikipedia.org/wiki/Relationale_Datenbank)

Referenzen:

- 1:1, 1:c
- 1:m(c)
- m:n (schlecht)

Normalformen:

- Ziel: Schutz vor Redundanzen und Inkonsistenzen
- Hierzu zerlegt man das Datenmodell in mehrere Tabellen, die zueinander in Beziehung stehen. Dabei geht man gewöhnlich in drei Schritten vor:
  1. **Erste Normalform:** Alle Spalten sind atomar, Werte beziehen sich auf einen Primärschlüssel
  2. **Zweite Normalform:** zusätzlich (bei zusammengesetzten Primärschlüsseln): alle Werte nur vom kompletten Primärschlüssel abhängig
  3. **Dritte Normalform:** zusätzlich: transitive Abhängigkeiten beseitigt (Werte nur vom Primärschlüssel abhängig)

**Mehr zum Thema Datenbank-Normalisierung:**

Beispiele auf [http://de.wikipedia.org/wiki/Normalisierung\\_%28Datenbank%29](http://de.wikipedia.org/wiki/Normalisierung_%28Datenbank%29)

## (My)SQL-Befehle

Man unterscheidet zwischen

- Data Definition Language (DDL)
- Data Modification Language (DML)
- Data Control Language (DCL)
- Kommandos zur Steuerung der eigenen Datenbanksitzung (control)

Beispiele:

## Tabellen einrichten (DDL, DCL, control)

- Datenbanken anlegen und löschen:

```
$ mysql -uroot -p
mysql> drop database test;
Query OK, 1 row affected (0.01 sec)

mysql> create database test;
Query OK, 1 row affected (0.00 sec)

mysql> exit
```

*Listing: DROP DATABASE (DDL)*

- Einem User Zugriffsrechte auf eine Datenbank einräumen und entziehen:

```
$ mysql -uroot -p
mysql> grant select,insert,update,delete,drop,alter,create
on test.* to 'testuser'@'localhost' identified by 'testpw';
Query OK, 0 rows affected (0.00 sec)

mysql> revoke alter on test.* from 'testuser'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> exit
```

*Listing: GRANT, REVOKE (DCL)*

- Als User mit dem DBMS verbinden, die erreichbaren Datenbanken auflisten und eine Datenbank auswählen:

```
$ mysql -utestuser -ptestpw
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14374 to server version: 4.1.12

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| test     |
+-----+
1 row in set (0.00 sec)

mysql> use test
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

*Listing: show databases, use (control)*

- Eine neue Tabelle anlegen (innerhalb der Datenbank):

```
mysql> create table tabl (id int, vorname char(30), nachname char(30) );
Query OK, 0 rows affected (0.00 sec)
```

*Listing: CREATE TABLE (DDL)*

- eine Tabelle löschen (inkl. aller ggf. darin enthaltenen Datensätze):

```
mysql> drop table tabl;
Query OK, 0 rows affected (0.00 sec)
```

### Listing: DROP TABLE (DDL)

(bitte jetzt noch nicht ausführen)

- Tabellen der aktuellen Datenbank auflisten (Namen und Eigenschaften):

```
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| tabl           |
+-----+
1 row in set (0.00 sec)

mysql> describe tabl;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | YES  |     | NULL    |      |
| vorname    | char(30)  | YES  |     | NULL    |      |
| nachname   | char(30)  | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Listing: show, describe (control)

## Mit Tabellen arbeiten (DML)

- Datensätze in eine Tabelle einfügen:

```
mysql> insert into tabl (id,nachname,vorname)
values (1,'Mueller','Hans'), (2,'Meier','Gudrun');
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

Listing: INSERT (DML)

- Datensätze aus einer Tabelle abfragen:

```
mysql> select * from tabl;
+-----+-----+-----+
| id  | vorname | nachname |
+-----+-----+-----+
|  1  | Hans    | Mueller  |
|  2  | Gudrun  | Meier    |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select nachname from tabl where vorname='Hans';
+-----+
| nachname |
+-----+
| Mueller  |
+-----+
1 row in set (0.00 sec)
```

Listing: SELECT (DML)

- Datensätze in einer Tabelle verändern:

```
mysql> update tabl set nachname='Meier-Schulze'
where vorname='Gudrun' and nachname='Meier';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

### Listing: UPDATE (DML)

- Datensätze aus einer Tabelle löschen:

```
mysql> delete from tabl where vorname='Alfons' and nachname='Schubert';
Query OK, 0 rows affected (0.00 sec)

mysql> delete from tabl where id=3;
Query OK, 0 rows affected (0.00 sec)

mysql> truncate table tabl;
Query OK, 0 rows affected (0.00 sec)
```

### Listing: DELETE, TRUNCATE (DML)

## Benutzerrechte

mysql-Datenbank

Berechtigungen auf

- Server-Ebene (connect, global)
- Datenbanken
- Tabellen
- (Spalten)

## Apache

## HTML

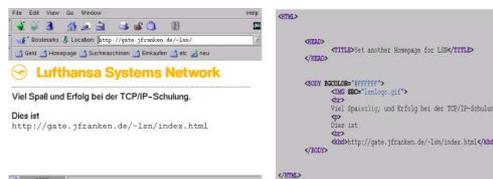


Abbildung: HTML-Quelltext

Das Standardwerk gibt es kostenlos unter <http://de.selfhtml.org/>

## Webserver und HTTP

Als "Webserver" bezeichnet man Programme, die über das HTTP-Protokoll Anfragen entgegennehmen (z.B. von einem Webbrowser über ein Internet) und Daten zurücksenden, die für die in der Anfrage genannte URL vorgesehen sind. Die Daten sind entweder

1. statisch (Dateiinhalte) oder
2. dynamisch (Ausgabe von Programmen).

Der erste Webserver ("W3 demon") wurde 1989 von Tim Berners-Lee programmiert. Inzwischen gibt es hunderte verschiedene Webserver-Implementierungen, die sich in Ihrer

- Arbeitsgeschwindigkeit,
- Zuverlässigkeit,
- Erweiterbarkeit und
- den von ihnen unterstützten Betriebssystemen

unterscheiden.

**Mehr zum Thema:**  
siehe

- <http://www.ietf.org/rfc/rfc2616.txt> (HTTP/1.1-protocol)
- <http://de.wikipedia.org/wiki/Webserver>

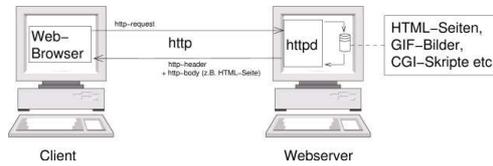


Abbildung: HTTP Beispiel

```
Autbau einer http-Session
# Server: gnu301
# Proxy: 192.168.1.1...
Connected to gnu301.franken.de.
Escape character is '^]'.
http://www.franken.de/~franken/mailcont/ http/1.1
HTTP/1.1 200 OK
Date: Fri, 22 Sep 2000 17:07:16 GMT
Server: Apache/1.3.0 (Ubuntu Debian/00)
Last-Modified: Fri, 22 Sep 2000 17:06:07 GMT
ETag: "/usr/lib/cgi-bin/100"
Accept-Ranges: bytes
Content-Length: 103
Content-Type: text/html
Expires: avoid future
<HTML>
<HEAD> <TITLE>Mailcont</TITLE> </HEAD>
<BODY bgcolor="#FFFFFF" text="#000000" link="#000000" vlink="#000000">
<P ALLIG=CHIFFRE>
<A href="/cgi-bin/main.pl">clap avon</a>

</BODY> </HTML>
Server closed connection
see http://1.0
```

Abbildung: HTTP Beispiel

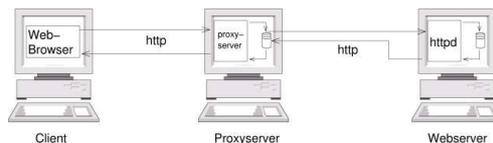


Abbildung: Proxyserver

# Apache

Apache ist Open-Source-Software, arbeitet effizient und zuverlässig und bietet in seinem modularen Aufbau einen Leistungsumfang, der alle anderen Webserver übertrifft. Etwa 60% aller Websites laufen zurzeit unter Apache.

**Mehr zum Thema "Apache":**

Eine kurze Übersicht zum Apache-Webserver finden Sie auf den Webseiten von [Wikipedia](http://de.wikipedia.org/wiki/Wikipedia) und der [Apache Foundation](http://www.apache.org/).

**Mehr zum Thema "Apache-Konfiguration":**

siehe [Debian/Apache HowTo](http://www.debian.org/doc/manuals/apache-howto/)

# CGI-Skripte

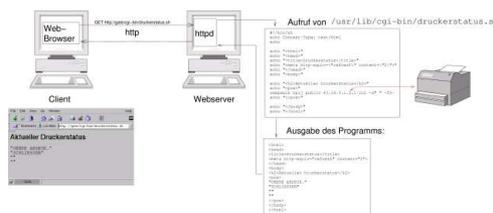


Abbildung: CGI-Skript

