



# Einführung in LAMP

Johannes Franken  
<jfranken@jfranken.de>

## Caution:

Please note: This document is available in german only. An english translation for this document is in progress

## Contents

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)

*Figure: LAMP-Architektur*

## MySQL

# Relationale Datenbank-Management-Systeme (RDBMS)

Figure: 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)

## More about "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)

## More about 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

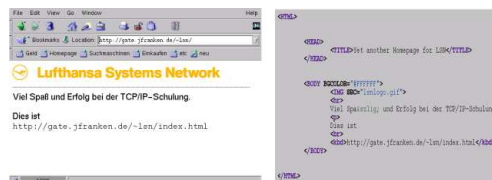


Figure: HTML-Quelltext

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

## Webserver und HTTP

A "webserver" is a program, which receives an HTTP-query (e.g. from some webbrowser over an internet) and sends back data provided for the queried URL. This data can be either

1. static (stored in files) or
2. dynamic (generated by other programs).

The first webserver ("W3 demon") was developed by Tim Berners-Lee in 1989. Today, there are hundreds of implementations, which differ in their

- speed of operation
- dependability
- expandability and
- their supported operating systems.

**More about:**  
see

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

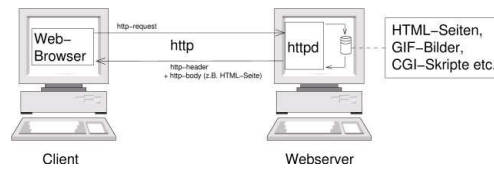


Figure: HTTP Beispiel

```
Autoren: erwin@p1000-01
# telnet gate 80
telnet opened 1...
Connected to gate.jfranken.de.
Escape character is '^]'.
GET http://gate.jfranken.de/franken/mailcom/ HTTP/1.0

HTTP/1.1 200 OK
Date: Fri, 22 Sep 2000 17:07:56 GMT
Server: Apache/1.3.3.6 (Unix) modssl/mod
Last-Modified: Fri, 22 Sep 2000 17:04:07 GMT
ETag: "f4c3-22-26a21c"
Accept-Ranges: bytes
Content-Length: 353
Content-Type: text/html
Connection: close
X-Pad: avoid browser bug

<HTML>
<HEAD><TITLE>Mailcom</TITLE></HEAD>
<BODY bgcolor="#993399" text="#FFFFFF" link="#FFFFFF" vlink="#FFFFFF">
  <P>Mailcom</P>
  <P><A href="mailto:mailcom@jfranken.de" style="color: red; text-decoration: underline; text-decoration-color: red;">Klicken Sie hier, um zu starten!</A>
  <A href="#">Bock</A>
</BODY>
</HTML>
Connection closed by foreign host.
```

Figure: HTTP Beispiel

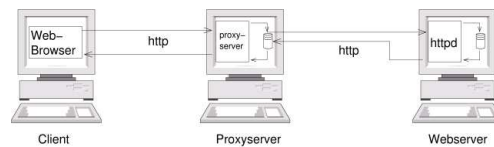


Figure: Proxyserver

## Apache

Apache is open source software, works most efficiently and reliably. Its modular design leads to a wide scope of services, which surpass every other webserver on the market. So it's no wonder, that about 60% of all websites are running on Apache.

**More about "Apache":**

A short overview on the apache webserver is available at [Wikipedia](#) and [Apache Foundation](#).

**More about "Apache-Konfiguration":**

siehe [Debian/Apache HowTo](#)

## CGI-Skripte

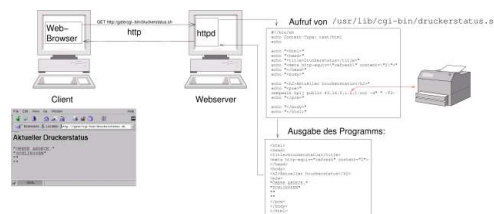


Figure: CGI-Skript

## Module

z.B. PHP

## Virtual Hosting

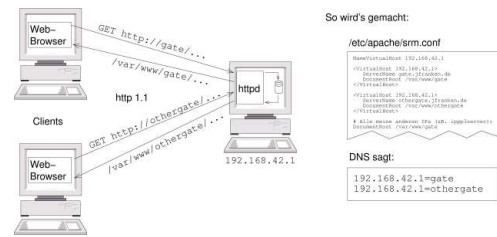


Figure: VirtualHost Definition

## PHP

## Sprache

siehe <http://www.php.net/manual/de/langref.php>

## Kleine Datenbank Anwendung

[phptest.tgz \[73 kB\]](#)

Figure: Includes