



# Einführung in PHP und MySQL

Handouts zu den interaktiven Folien (<http://t.uzh.ch/phpmysqlfolien>)

Autoren:

Diverse Mitarbeiter der Zentralen Informatik

Letzte Überarbeitung: Daniel Förderer

# Inhalt

## 1. Kursbeispiel und Kursziele

- 1.1 Das Kursbeispiel: eine kleine Mitarbeitenden-Verwaltung
- 1.2 Was brauchen wir?
- 1.3 Ziele Kurs AZIP

## 2. Von Excel nach MySQL

- 2.1 Ziel: Einführung in MySQL
- 2.2 Das EXCEL-Adressblatt
- 2.3 Als csv-Textdatei speichern
- 2.4 MySQL-Datenbanken verwalten mit phpMyAdmin
- 2.5 Excel-Tabellen und Datenbank-Tabellen
- 2.6 Die Tabelle mitarbeitende mit phpMyAdmin erstellen
- 2.7 Dateneingabe und Datenimport mit phpMyAdmin
- 2.8 Das Feld ID zum Primärschlüssel machen
- 2.9 Die auto-increment Funktion

## 3. SQL: insert, select, update und delete

- 3.1 Die Datenbanksprache SQL
- 3.2 Insert: Daten einfügen
- 3.3 Select: Daten abfragen
- 3.4 Where: Abfragen mit Bedingungen
- 3.5 Gross- und Kleinschreibung bei MySQL-SQL
- 3.6 Order by: Abfrageergebnisse sortieren
- 3.7 Update: Daten verändern
- 3.8 Delete: Daten löschen

## 4. PHP

- 4.1 Ziel: Einführung in PHP
- 4.2 Statische Webseiten
- 4.4 Ablauf: Aufruf einer dynamischen Webseite mit PHP
- 4.5 Interaktive Einführung in PHP

## **5. PHP und MySQL**

- 5.1 Ziel: Datenbankbindung mit PHP und MySQL
- 5.2 Konzept: DB-Verbindung und –Abfrage
- 5.3 Beispiel: DB-Verbindung und -Abfrage
- 5.4 Den Inhalt der Tabelle mitarbeitende anzeigen
- 5.5 Beispiel: Abfrage mit einem Parameter
- 5.6 Nur Personen mit einem bestimmten Vornamen anzeigen
- 5.7 Verbindungs-Fehler
- 5.8 SQL-Fehler
- 5.9 User, Passwort und Verbindung in config.php auslagern

## **6. Das Kursbeispiel**

- 6.1 Die Übersichtsseite: alle Mitarbeitenden anzeigen (mitarbeitende3.php)
- 6.2 Die Eingabeseite: eine neue Person aufnehmen (insert.php)
- 6.3 Die Löschseite: eine Person löschen (delete.php)
- 6.4 Eine Person mit Klick auf der Übersichtsseite löschen (mitarbeitende4.php)
- 6.5 Die Änderungsseite: eine Person ändern (update1.php)
- 6.6 Von der Übersichtsseite auf die Änderungsseite springen (mitarbeitende5.php)
- 6.7 Die Änderungsseite fertigstellen (update.php)
- 6.8 Übersichtsseite: Suche einbauen (mitarbeitende.php)
- 6.9 Webpass-Authentifizierung und –Autorisierung

## **7. Zusatzmaterial: das Erweiterte Kursbeispiel**

- 7.1 Die Tabelle abteilungen erstellen
- 7.2 Fremdschlüssel abteilungs\_id in mitarbeitende erstellen
- 7.3 Joins: Abfragen mit mehreren Tabellen
- 7.4 Left Joins und right Joins
- 7.5 Die php-Skripte zum erweiterten Kursbeispiel
- 7.6 Das erweiterte Kursbeispiel CD-Konform



# 1. Kursbeispiel und Kursziele

1.1 Das Kursbeispiel: eine kleine Mitarbeitenden-Verwaltung

1.2 Was brauchen wir?

1.3 Ziele Kurs AZIP

# 1.1 Das Kursbeispiel: Eine kleine Mitarbeitendenverwaltung

Mitarbeitende eingeben

Mitarbeitende suchen, anschauen, bearbeiten und löschen

## 1.2 Was brauchen wir?



das MySQL-  
Datenbanksystem



den Apache Webserver



die php Skriptsprache

## 1.3 Ziele Kurs AZIP

1. MySQL-Datenbanken erstellen, Daten eingeben, ändern und löschen.
2. Einfache PHP-Programme schreiben und auf einem Webserver anzeigen.
3. PHP Benutzerinterfaces für MySQL-Datenbanken schreiben.
4. Eine einfache PHP-MySQL-Anwendung entwickeln.



## 2. Von Excel nach MySQL

2.1 Ziel: Einführung in MySQL

2.2 Das EXCEL-Adressblatt

2.3 Als csv-Textdatei speichern

2.4 MySQL-Datenbanken verwalten mit phpMyAdmin

2.5 Excel-Tabellen und Datenbank-Tabellen

2.6 Die Tabelle mitarbeitende mit phpMyAdmin erstellen

2.7 Dateneingabe und Datenimport mit phpMyAdmin

2.8 Das Feld ID zum Primärschlüssel machen

2.9 Die auto-increment Funktion

## 2.1 Ziel: Einführung in MySQL

1. MySQL-Tabellen erstellen und Daten importieren.
2. Das MySQL-Datenbank-Tool phpMyAdmin kennen lernen.
3. Die wichtigsten SQL-Befehle kennen und anwenden.

## 2.2 Das EXCEL-Adressblatt

	A	B	C	D	E	F
1	id	vorname	name	kuerzel	email	lohn
2	4	Hans Rudolf	Brugger	brug	<a href="mailto:brugger@agg.ch">brugger@agg.ch</a>	6400
3	15	Hans	Müller	mue	<a href="mailto:hans.mueller@agg.ch">hans.mueller@agg.ch</a>	9100
4	14	Nicolas	Mächler	mae	<a href="mailto:nicolas.maechler@agg.ch">nicolas.maechler@agg.ch</a>	5300
5	11	Fritz	Leutwiler	leut	<a href="mailto:leutwiler@agg.ch">leutwiler@agg.ch</a>	

## 2.3 Als csv-Textdatei speichern

1. Menü Datei, Speichern unter...
2. Dateityp: CSV(Trennzeichen getrennt)
3. Dateiname: mitarbeitende.csv
4. Die Datei mitarbeitende.csv in Text-Editor öffnen:

```
id;vorname;name;kuerzel;email;lohn
4;Hans Rudolf;Brugger;brug;brugger@agg.ch;6400
15;Hans;Müller;mue;hans.mueller@agg.ch;9100
14;Nicolas;Mächler;mae;nicolas.maechler@agg.ch;5300
11;Fritz;Leutwiler;leut;leutwiler@agg.ch;
```

## 2.4 MySQL-Datenbanken verwalten mit phpMyAdmin

- MySQL: das Datenbanksystem
- phpMyAdmin: eine Anwendung um MySQL zu verwalten
- Aufruf und Login: <http://idkurs.uzh.ch/phpMyAdmin/>
- phpMyAdmin-Home (Startseite)
  - Linke Seite: Navigationsspalte
  - Rechte Seite: Links und Einstellungen

## 2.5 Excel-Tabellen und Datenbank-Tabellen

<b>Excel-Tabelle</b>	<b>MySQL-Tabelle</b>
Tabellen-Spalten	Tabellen-Felder
Spalten-Namen	Feld-Namen
Tabellen-Zeilen	Datensätze

## 2.6 Die Tabelle mitarbeitende mit phpMyAdmin erstellen

Feld	Typ	Länge	Null
id	int	10	not null
vorname	varchar	25	not null
name	varchar	20	not null
kuerzel	varchar	5	not null
email	varchar	35	not null
lohn	decimal	8,2	null

Wichtig: Unter "Storage Engine" InnoDB auswählen

<http://idkurs.uzh.ch/phpMyAdmin/>

## 2.7 Dateneingabe und Datenimport mit phpMyAdmin

- Dateneingabe in phpMyAdmin
- Daten importieren aus der csv-Textdatei `mitarbeitende.csv`
  - Die erste Zeile mit den Spaltennamen muss gelöscht werden
  - Bei leeren Feldern muss NULL drinstehen
  - Das Semikolon nach dem Salary-Feld muss gelöscht werden
  - Die Reihenfolge der Felder muss mit Reihenfolge in der Tabelle übereinstimmen
  - Feld-Trenner: Semikolon
  - Datensatz-Trenner: Zeilenschaltung
  - Zeichencodierung der Datei: Latin 1



## 2.8 Das Feld ID zum Primärschlüssel machen

- Jede Person soll eine eindeutige ID haben.
- Feld ID ist der Primärschlüssel.
- phpMyAdmin > mitarbeitende auswählen > Structure:



rowse Structure SQL Search Insert Export Import Operations Emp

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(10)			No	None		

- Jetzt auf das Schlüsselssymbol (roter Kreis) klicken.

## 2.9 Die auto-increment Funktion

- Die Auto Increment Funktion generiert die IDs.
- phpMyAdmin > mitarbeitende auswählen > Structure:



Field	Type	Collation	Attributes	Null	Default	Extra	Actio
<input type="checkbox"/>	id	int(10)		No	None		

- Auf das Stiftsymbol (roter Kreis) klicken.
- Checkbox bei AUTO\_INCREMENT anklicken:

Field	id
Type	INT
Length/Values <sup>1</sup>	10
Default <sup>2</sup>	None
Collation	
Attributes	
Null	<input type="checkbox"/>
AUTO_INCREMENT	<input checked="" type="checkbox"/>
Comments	
<input type="button" value="Save"/>	

## 3. SQL: insert, select, update, delete

3.1 Die Datenbanksprache SQL

3.2 Insert: Daten einfügen

3.3 Select: Daten abfragen

3.4 Where: Abfragen mit Bedingungen

3.5 Gross und Kleinschreibung bei MySQL-SQL

3.6 Order by: Abfrageergebnisse sortieren

3.7 Update: Daten verändern

3.8 Delete: Daten löschen

## 3.1 Die Datenbanksprache SQL

Es gibt drei Kategorien von SQL-Befehlen:

1. Definition von Datenbankobjekten wie Tabellen und Views ( Data Definition Language)
2. Datenmanipulation ( Data Manipulation Language):  
Abfragen, Ändern, Einfügen, Löschen von Daten
3. Rechteverwaltung ( Data Control Language)

## 3.2 Insert: Daten einfügen

```
insert into mitarbeitende
(vorname,name,kuerzel,email)
values
('Franz','Blank','bla','f.blank@agg.ch'),
('Hans','Blum','blu','h.blum@agg.ch');
```

## 3.3 Select: Daten abfragen

- Nur die Felder vorname und name abfragen:

```
select vorname, name  
from mitarbeitende;
```

- Alle Felder abfragen:

```
select * from mitarbeitende;
```

## 3.4 Where: Abfragen mit Bedingungen

- Suche nach exakten und ganzen Feldinhalten:

```
select *  
  from mitarbeitende  
 where vorname='Franz'  
       and name='Blank';
```

- Suche nach Teilen von Feldinhalten:

```
select *  
  from mitarbeitende  
 where vorname like '%Fr%';
```

## 3.5 Gross und Kleinschreibung bei MySQL-SQL

- Schlüsselwörter (select, where, order by): spielt keine Rolle
- Datenbankobjekte (Tabellen): spielt eine Rolle
- Tabellenfelder: spielt keine Rolle
- Daten: spielt keine Rolle
- Diese Regeln gelten nur für das SQL von MySQL!



## 3.6 Order by: Abfrageergebnisse sortieren

```
select vorname, name  
from mitarbeitende  
order by name asc, vorname desc;
```

(asc=aufsteigend, desc=absteigend)

## 3.7 Update: Daten verändern

```
update mitarbeitende
  set name='Blanc',
      vorname='Fritz'
  where name='Blank';
```

**ACHTUNG:** Ohne where-Bedingung werden alle Datensätze geändert!!

## 3.8 Delete: Daten löschen

```
delete
  from mitarbeitende
  where name='Förderer';
```

**ACHTUNG:** Ohne where-Bedingung werden alle Datensätze gelöscht!!



## 4. PHP

4.1 Ziel: Einführung in PHP

4.2 Statische Webseiten

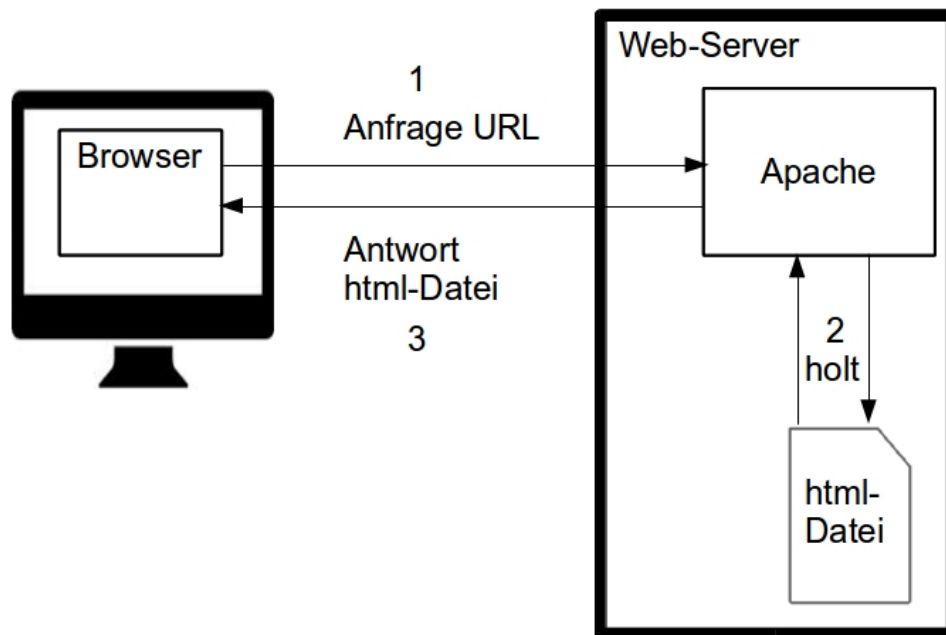
4.3 Dynamische Webseite mit PHP und MySQL

4.4 Interaktive Einführung in PHP

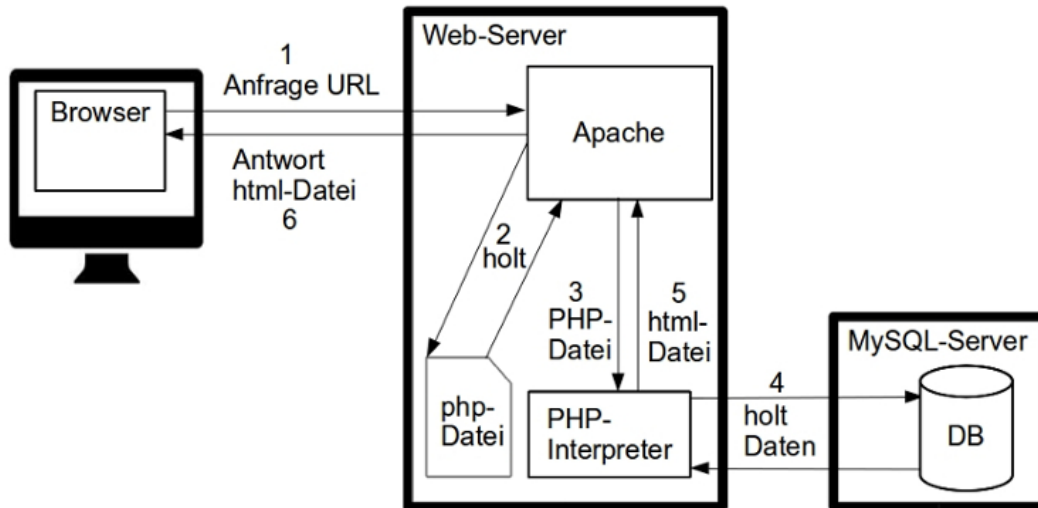
## 4.1 Ziele: Einführung in PHP

1. Statische und dynamische Webseiten.
2. PHP-Grundwissen, um einfache PHP-Anwendungen zu entwickeln.

## 4.2 Statische Webseiten



## 4.3 Dynamische Webseiten mit PHP und MySQL





## 4.4 Interaktive Einführung in PHP

<http://t.uzh.ch/php>

(von David Meier, überarbeitet von Daniel Förderer)



## 5. PHP und MySQL

5.1 Ziel: Datenbankbindung mit PHP und MySQL

5.2 Konzept: DB-Verbindung- und Abfrage

5.3 Beispiel: DB-Verbindung- und Abfrage

5.4 Den Inhalt der Tabelle mitarbeitende anzeigen

5.5 Beispiel: Abfrage mit einem Parameter

5.6 Nur Personen mit einem bestimmten Vornamen anzeigen

5.7 Verbindungs-Fehler

5.8 SQL-Fehler

5.9 User, Passwort und Verbindung in config.php auslagern

## 5.1 Ziel: Datenbankanbindung mit PHP und MySQL

1. Sie können eine MySQL-Datenbank abfragen und die Daten im Web anzeigen.
2. Sie kennen die dazu nötigen PHP-MySQL-Funktionen.

## 5.2 Konzept: DB-Verbindungs- und Abfrage

1. Verbindung zum DB-Server herstellen
2. SQL-Query vorbereiten
3. SQL-Query ausführen
4. Resultat abholen (wenn vorhanden)
5. Resultat anzeigen (wenn vorhanden)

## 5.3 Beispiel: DB-Verbindung- und Abfrage

1. Verbindung herstellen

```
$conn = new PDO("mysql:host=localhost;  
                dbname=mykursXY",  
                "mykursXY",  
                "Passwort" );
```

2. Abfrage vorbereiten

```
$query  
=  
$conn->prepare("select *  
                from mitarbeitende");
```

3. Statement ausführen

```
$query->execute();
```

4. Daten holen (erste Zeile)

```
$person = $query->fetch();
```

5. Daten anzeigen (erste Zeile)

```
var_export($person);
```

Resultat: Beispiel einer Abfrage (abfrage1.php)  
Code: abfrage1.php

## 5.4 Den Inhalt der Tabelle mitarbeitende anzeigen

```
$conn = new PDO( "mysql:host=localhost;
                dbname=mykursXY",
                "mykursXY",
                "Passwort" );

$query
=
$conn->prepare('select * from mitarbeitende');
$query->execute();
while ($person = $query->fetch()){
    $id      = $person['id'];
    $vorname = $person['vorname'];
    $name    = $person['name'];
    $kuerzel = $person['kuerzel'];
    $email   = $person['email'];
    $lohn    = $person['lohn'];
    echo "$id<br>
        $vorname<br>
        $name<br>
        $kuerzel<br>
        $email<br>
        $lohn<br><br>";
}
```

Resultat: Daten von mitarbeitende anzeigen (mitarbeitende1.php)  
Code: Daten von mitarbeitende anzeigen (mitarbeitende1.phps)

## 5.5 Beispiel: Abfrage mit einem Parameter

1. Verbindung herstellen

```
$conn = new PDO("mysql:host=localhost;  
                dbname=mykursXY",  
                "mykursXY",  
                "Passwort" );
```

2. Abfrage vorbereiten mit Parameter

```
$query  
=  
$conn->prepare("select *  
                from mitarbeitende  
                where vorname  
                = :vorname");
```

3. Abfrage mit Vornamen Franz ausführen

```
$query->execute([ 'vorname' => 'Franz' ] );
```

4. Daten holen (1. Zeile)

```
$person = $query->fetch();
```

5. Daten anzeigen (1. Zeile)

```
var_export($person);
```

Resultat: Beispiel einer Abfrage (abfrage2.php)



## 5.6 Nur Personen mit einem bestimmten Vornamen anzeigen

Das Beispiel von Folie 5.4 muss dazu folgendermassen angepasst werden:

```
$query  
=  
$conn->prepare( 'select *  
                from mitarbeitende  
                where vorname  
                =  
                :vorname' );  
  
$query->execute( [ 'vorname' => 'Franz' ] );
```

Resultat: Daten von mitarbeitende anzeigen (mitarbeitende\_param.php)  
Code: Daten von personnel anzeigen (mitarbeitende\_param.phps)

## 5.7 Verbindungs-Fehler

```
<?php
try {
    $conn = new PDO("mysql:host=localhost;
                    dbname=mykursXY",
                    "mykursXY",
                    "FALSCHES PASSWORT");
}
catch(PDOException $e) {
    echo "Verbindungsfehler: "
        . $e->getMessage();
    exit;
}

$query
=
$conn->prepare('select * from mitarbeitende');
$query->execute();
$person=$query->fetch();
var_export($person);
?>
```

Resultat: Falsches Passwort

## 5.8 SQL-Fehler

```
$query
=
$conn->prepare('selecttt *
                from mitarbeitende');
$query->execute();

if ( $query->errorCode() > 0 ){
    $fehler=$query->errorInfo();
    echo "$fehler[2]";
    exit;
}

$person=$query->fetch();
var_export($person);
```

Resultat: SQL-Fehler

## 5.9 User, Passwort und Verbindung in config.php auslagern

Resultat Ausgangsprogramm mitarbeitende1.php ( Code)

- conf/config.php:

```
try {
    $conn = new PDO("mysql:host=localhost;
                    dbname=mykursXY",
                    "mykursXY",
                    "Passwort");
}
catch(PDOException $e){
    echo "Verbindungsfehler: "
        . $e->getMessage();
    exit;
}
```

- config.php in php-Script includen:

```
include('conf/config.php');
$query
=
$conn->prepare('select *
                from personnel');
```

- conf/config.php schützen mit conf/.htaccess:

```
deny from all
```

Resultat verändertes Programm mitarbeitende2.php ( Code)

## 6. Das Kursbeispiel

- 6.1 Die Übersichtsseite: alle Mitarbeitenden anzeigen
- 6.2 Die Eingabeseite: eine neue Person aufnehmen (insert.php)
- 6.3 Die Löschseite: eine Person löschen (delete.php)
- 6.4 Eine Person mit Klick auf der Übersichtsseite löschen
- 6.5 Die Änderungsseite: eine Person ändern (update1.php)
- 6.6 Von der Übersichtsseite auf die Änderungsseite springen
- 6.7 Die Änderungsseite fertigstellen (update.php)
- 6.8 Übersichtsseite: Suche einbauen (mitarbeitende.php)
- 6.9 Webpass-Authentifizierung und -Autorisierung

## 6.1 Die Übersichtsseite: Alle Mitarbeitenden anzeigen

Ausgangsprogramm: [mitarbeitende2.php](#) ( Code)

In der while Schlaufe:

```
echo "<tr>
    <td>$name</td>
    <td>$vorname</td>
    <td>$kuerzel</td>
    <td>$email</td>
    <td>$lohn</td>
</tr>";
```

Resultat: [mitarbeitende3.php](#) ( Code)

## 6.1 mitarbeitende3.php

```
<h1>Mitarbeitende anzeigen</h1>
<table border=1>
  <tr>
    <th>Name</th>
    <th>Vorname</th>
    <th>K&uuml;rzel</th>
    <th>E-Mail</th>
    <th>Lohn</th>
  </tr>
<?php
  include("conf/config.php");
  $sql = "select * from mitarbeitende";
  $query = $conn->prepare($sql);
  $query->execute();
  if ( $query->errorCode() > 0 ){
    $fehler=$query->errorInfo();
    echo "$fehler[2]";
    exit;
  }
  while ($person = $query->fetch()) {
    $id      = $person['id'];
    $name    = $person['name'];
    $vorname = $person['vorname'];
    $email   = $person['email'];
    $kuerzel = $person['kuerzel'];
    $lohn    = $person['lohn'];
    echo "<tr>
      <td>$name</td>
      <td>$vorname</td>
      <td>$kuerzel</td>
      <td>$email</td>
      <td>$lohn</td>
    </tr>";
  }
?>
</table>
```

## 6.2 Die Eingabeseite: eine neue Person aufnehmen (insert.php)

Ausgangsprogramm: form3.php ( [Code](#) )

```
include('conf/config.php');
$sql = "insert into mitarbeitende
      (vorname,name,kuerzel,
       email,lohn)
      values
      (:vorname, :name, :kuerzel,
       :email, :lohn)";

$query = $conn->prepare($sql);

$parameter=[ 'vorname' => $vorname,
              'name'     => $name,
              'kuerzel'  => $kuerzel,
              'email'    => $email,
              'lohn'     => $lohn];

$query->execute($parameter);
```

Resultat: insert.php ( [Code](#) )



## 6.2 insert.php

```
<h1>Eine neue Person aufnehmen</h1>

<?php
include('conf/config.php');
$php_self=$_SERVER['PHP_SELF'];
$vorname = empty($_GET['vorname']) ? '' : $_GET['vorname'];
$name     = empty($_GET['name'])     ? '' : $_GET['name'];
$kuerzel  = empty($_GET['kuerzel']) ? '' : $_GET['kuerzel'];
$email    = empty($_GET['email'])    ? '' : $_GET['email'];
$lohn     = empty($_GET['lohn'])     ? '' : $_GET['lohn'];

echo "<h1>Neue Person aufnehmen</h1>";
if ( empty($vorname) or empty($name) or empty($kuerzel) or
empty($email) ){
    echo "<form action='$php_self' method='get'>
        Vorname: <input name='vorname' value='$vorname'><br>
        Name:     <input name='name' value='$name' ><br>
        Kuerzel:  <input name='kuerzel' value='$kuerzel'><br>
        Email:   <input name='email' value='$email' ><br>
        Lohn:    <input name='lohn' value='$lohn' ><br>
                <input type='submit' name='insert' value='Speichern'>
    </form>";
}
else {
    $sql = "insert into mitarbeitende
            (vorname,name,kuerzel,
             email,lohn)
            values
            (:vorname,:name,:kuerzel,
             :email,:lohn)";

    $query = $conn->prepare($sql);

    $parameter=['vorname' => $vorname,
                'name'     => $name,
                'kuerzel' => $kuerzel,
                'email'   => $email,
                'lohn'    => $lohn];

    $query->execute($parameter);

    if ( $query->errorCode() > 0 ){
        $fehler=$query->errorInfo();
        echo "$fehler[2]";
        exit;
    }

    echo "Danke! Die folgende Person wurde eingegeben:<br>
        Vorname: $vorname<br>
        Name:     $name<br>
        Kuerzel:  $kuerzel<br>
        Email:    $email<br>
        Lohn:     $lohn<br>";
}
?>
```

## 6.3 Die Löschseite: eine Person löschen (delete.php)

```
$id = $_GET['id'];  
$sql = "delete from mitarbeitende  
      where id=:id";  
  
$query = $conn->prepare($sql);  
  
$query->execute(['id' => $id]);  
  
echo 'Die Person wurde geloescht';
```

Code von delete.php

## 6.3 delete.php

```
<?php
include('conf/config.php');
$id    = $_GET['id'];
$sql   = "delete from mitarbeitende where id=:id";
$query = $conn->prepare($sql);
$query->execute(['id' => $id]);
if ( $query->errorCode() > 0 ){
    $fehler=$query->errorInfo();
    echo "$fehler[2]";
    exit;
}
echo 'Die Person wurde gel&ouml;scht!';
?>
```

## 6.4 Eine Person mit Klick auf der Übersichtsseite löschen

Ausgangsprogramm: mitarbeitende3.php ( Code)

- Neue Tabellenspalte Options einfügen
- Der Link nach delete.php:

```
<a href='delete.php?id=$id'>  
  Löschen  
</a>
```

Resultat: mitarbeitende4.php ( Code)

## 6.4 mitarbeitende4.php

```
</table>
<h1>Mitarbeitende anzeigen</h1>
<table border=1>
  <tr>
    <th>Name</th>
    <th>Vorname</th>
    <th>K&uuml;rzel</th>
    <th>E-Mail</th>
    <th>Lohn</th>
    <th>Options</th>
  </tr>
  <?php
    include("conf/config.php");
    $sql = "select * from mitarbeitende";
    $query = $conn->prepare($sql);
    $query->execute();
    if ( $query->errorCode() > 0 ){
      $fehler=$query->errorInfo();
      echo "$fehler[2]";
      exit;
    }
    while ($person = $query->fetch()) {
      $id      = $person['id'];
      $name    = $person['name'];
      $vorname = $person['vorname'];
      $email   = $person['email'];
      $kuerzel = $person['kuerzel'];
      $lohn    = $person['lohn'];
      echo "<tr>
        <td>$name</td>
        <td>$vorname</td>
        <td>$kuerzel</td>
        <td>$email</td>
        <td>$lohn</td>
        <td>
          <a href='delete.php?id=$id'>l&ouml;schen</a>
        </td>
      </tr>";
    }
  ?>
</table>
```

## 6.5 Die Änderungsseite: eine Person ändern (update1.php)

Ausgangsprogramm: insert.php ( Code)

- Das Feld Id in das Formular einfügen:

```
Id: <input name='id'>
```

- insert-Statement durch update-Statement ersetzen

```
update mitarbeitende  
    set vorname    = :vorname,  
        name       = :name,  
        kuerzel    = :kuerzel,  
        email      = :email,  
        lohn       = :lohn  
where id=:id;
```

- Der submit-Knopf bekommt einen Namen:

```
<input type='submit'  
    name='update' value='Speichern'>
```

- Die Variable \$update einführen:

```
$update = empty($_GET['update']) ?'' ...
```

- Die if-Bedingung anpassen:

```
$update=$_GET['update'];  
if (empty($update) or empty($vorname)...
```

Resultat: update1.php ( Code)

## 6.5 update1.php

```
<h1>Eine Person &auml;ndern</h1>
<?php
    include('conf/config.php');
    $php_self=$_SERVER['PHP_SELF'];
    $vorname = empty($_GET['vorname']) ? '' : $_GET['vorname'];
    $name     = empty($_GET['name'])   ? '' : $_GET['name'];
    $kuerzel  = empty($_GET['kuerzel']) ? '' : $_GET['kuerzel'];
    $email    = empty($_GET['email'])  ? '' : $_GET['email'];
    $lohn     = empty($_GET['lohn'])   ? '' : $_GET['lohn'];
    $id       = $_GET['id'];
    $update   = empty($_GET['update']) ? '' : $_GET['update'];
    if ( empty($update) or empty($vorname) or empty($name) or
        empty($kuerzel) or empty($email) ){
        echo "<form action='$php_self' method='get'>
            Id: <input name='id' value='$id'> <br>
            Vorname: <input name='vorname' value='$vorname'><br>
            Lastname: <input name='name' value='$name'> <br>
            Kuerzel: <input name='kuerzel' value='$kuerzel'><br>
            E-Mail: <input name='email' value='$email'> <br>
            Lohn: <input name='salary' value='$lohn'> <br>
                 <input type='submit' name='update' value='Speichern'>
        </form>";
    }
    else{
        $sql = "update mitarbeitende
                set vorname = :vorname,
                    name     = :name,
                    kuerzel  = :kuerzel,
                    email    = :email,
                    lohn     = :lohn
                where id=:id";
        $query=$conn->prepare($sql);
        $parameter=[ 'id' => $id,
                    'vorname' => $vorname,
                    'name' => $name,
                    'kuerzel' => $kuerzel,
                    'email' => $email,
                    'lohn' => $lohn];

        $query->execute($parameter);
        if ( $query->errorCode() > 0 ){
            $fehler=$query->errorInfo();
            echo "$fehler[2]";
            exit;
        }

        echo "Danke! Der Datensatz nach der &Auml;nderung:<br>
            Name: $name<br>
            Vorname: $vorname<br>
            Kuerzel: $kuerzel<br>
            E-Mail: $email<br>
            Lohn: $lohn<br>";
    }
?>
```

## 6.6 Von der Übersichtsseite auf die Änderungsseite springen

Ausgangsprogramm: mitarbeitende4.php ( Code)

Der zusätzliche Link in der Spalte Options:

```
...  
<a href='update1.php?id=$id" .  
    "&name=$name" .  
    "&vorname=$vorname" .  
    "&kuerzel=$kuerzel" .  
    "&email=$email" .  
    "&lohn=$lohn'">" .  
    "Bearbeiten  
</a>  
...
```

Resultat: mitarbeitende5.php ( Code)



## 6.6 mitarbeitende5.php

```
<h1>Mitarbeitende anzeigen</h1>
<table border=1>
  <tr>
    <th>Name</th>
    <th>Vorname</th>
    <th>K&uuml;rzel</th>
    <th>E-Mail</th>
    <th>Lohn</th>
    <th>Options</th>
  </tr>
  <?php
    include("conf/config.php");
    $sql = "select * from mitarbeitende";
    $query = $conn->prepare($sql);
    $query->execute();
    if ( $query->errorCode() > 0 ){
      $fehler=$query->errorInfo();
      echo "$fehler[2]";
      exit;
    }
    while ($person = $query->fetch()) {
      $id      = $person['id'];
      $name    = $person['name'];
      $vorname = $person['vorname'];
      $email   = $person['email'];
      $kuerzel = $person['kuerzel'];
      $lohn    = $person['lohn'];
      echo "<tr>
        <td>$name</td>
        <td>$vorname</td>
        <td>$kuerzel</td>
        <td>$email</td>
        <td>$lohn</td>
        <td>
          <a href='delete.php?id=$id'>l&ouml;schen</a>
          <a href='update1.php?id=$id".
            "&name=$name".
            "&vorname=$vorname".
            "&kuerzel=$kuerzel".
            "&email=$email".
            "&lohn=$lohn'>".
            "bearbeiten
          </a>
        </td>
      </tr>";
    }
  ?>
</table>
```

## 6.7 Die Änderungsseite fertig stellen (update.php)

Ausgangsprogramm: update1.php ( Code)

Ausgangsprogramm: mitarbeitende5.php ( Code)

- Das Feld ID darf nicht geändert werden und wird versteckt:

```
ID: $id<br>  
<input type='hidden  
name='id' value='$id'>
```

- Ändern sie in mitarbeitende3.php update1.php zu update.php.

Resultat: update.php ( Code)

Resultat: mitarbeitende6.php ( Code)

## 6.7 update.php

```
<h1>Eine Person &auml;ndern</h1>
<?php
    include('conf/config.php');
    $php_self=$_SERVER['PHP_SELF'];
    $vorname = empty($_GET['vorname']) ? '' : $_GET['vorname'];
    $name     = empty($_GET['name'])   ? '' : $_GET['name'];
    $kuerzel  = empty($_GET['kuerzel']) ? '' : $_GET['kuerzel'];
    $email    = empty($_GET['email'])  ? '' : $_GET['email'];
    $lohn     = empty($_GET['lohn'])   ? '' : $_GET['lohn'];
    $id       = $_GET['id'];
    $update   = empty($_GET['update']) ? '' : $_GET['update'];
    if ( empty($update) or empty($vorname) or empty($name) or
        empty($kuerzel) or empty($email) ){
        echo "<form action='$php_self' method='get'>
            Id: $id <input type='hidden' name='id' value='$id'><br>
            Vorname: <input name='vorname' value='$vorname'><br>
            Lastname: <input name='name' value='$name'> <br>
            Kuerzel: <input name='kuerzel' value='$kuerzel'><br>
            E-Mail: <input name='email' value='$email'> <br>
            Lohn: <input name='salary' value='$lohn'> <br>
                <input type='submit' name='update' value='Speichern'>
        </form>";
    }
    else{
        $sql = "update mitarbeitende
            set vorname = :vorname,
              name     = :name,
              kuerzel  = :kuerzel,
              email    = :email,
              lohn     = :lohn
            where id=:id";
        $query=$conn->prepare($sql);
        $parameter=['id'      => $id,
                    'vorname' => $vorname,
                    'name'    => $name,
                    'kuerzel' => $kuerzel,
                    'email'   => $email,
                    'lohn'    => $lohn];

        $query->execute($parameter);
        if ( $query->errorCode() > 0 ){
            $fehler=$query->errorInfo();
            echo "$fehler[2]";
            exit;
        }

        echo "Danke! Der Datensatz nach der &Auml;nderung:<br>
            Name:      $name<br>
            Vorname:  $vorname<br>
            Kuerzel:  $kuerzel<br>
            E-Mail:   $email<br>
            Lohn:     $lohn<br>";
    }
?>
```

## 6.8 Übersichtsseite: Suche einbauen

Ausgangsprogramm: mitarbeitende6.php ( Code)

- Suchtext: welcher Text soll gesucht werden?

```
<input name='suchtext'>
```

- Nach dem Abschicken der Suche:

```
$suchtext = $_GET['suchtext'];
```

- Die where-Bedingung:

```
where name      like '%$suchtext%'  
  or vorname   like '%$suchtext%'  
  or kuerzel   like '%$suchtext%'  
  or email     like '%$suchtext%'  
  or lohn      like '%$suchtext%'
```

Resultat: mitarbeitende.php ( Code)

## 6.8 mitarbeitende.php

```
<h1>Mitarbeitende anzeigen</h1>

<?php
    include('conf/config.php');
    $php_self = $_SERVER['PHP_SELF'];
    $suchtext = $_GET['suchtext'];
    echo "<form action='$php_self' method='get'>
        Person suchen: <input name='suchtext' value='$suchtext'>
                <input type='submit' name='suchen' value='Suchen'>
        </form>";
?>

<table border=1>
    <tr><th>Name</th><th>Vorname</th><th>Kuerzel</th><th>E-Mail</th>
        <th>Lohn</th><th>Optionen</th></tr>

<?php
    $sql = "select *
            from mitarbeitende
            where name      like :suchtext
            or vorname    like :suchtext
            or kuerzel    like :suchtext
            or email      like :suchtext
            or lohn       like :suchtext";
    $query = $conn->prepare($sql);
    $parameter=[ 'suchtext' => "%$suchtext%" ];
    $query->execute($parameter);
    if ( $query->errorCode() > 0 ){
        $fehler=$query->errorInfo();
        echo "$fehler[2]";
        exit;
    }
    while ($person = $query->fetch()){
        $id      = $person['id'];
        $name    = $person['name'];
        $vorname = $person['vorname'];
        $kuerzel = $person['kuerzel'];
        $email   = $person['email'];
        $lohn    = $person['lohn'];
        echo "<tr><td>$name</td><td>$vorname</td><td>$kuerzel</td>
            <td>$email</td><td>$lohn</td>
            <td><a href='delete.php?id=$id'>Löschen</a>
                <a href='update.php?".
                    "id=$id".
                    "&name=$name".
                    "&vorname=$vorname".
                    "&kuerzel=$kuerzel".
                    "&email=$email".
                    "&lohn=$lohn'">".
                    "Bearbeiten</a>
            </td>
        </tr>";
    }
?>
</table>
```

## 6.9 Webpass-Authentifizierung und - Autorisierung

Erstellen eines Verzeichnisses ssl-dir mit Datei  
.htaccess darin:

```
AuthType Basic  
# Anschrift Login-Fenster  
AuthName "Webpass-Login"  
#LDAP-Authentifizierung Uni Zürich  
AuthBasicProvider ldap  
# Einschränkung der Berechtigten  
# (Autorisierung)  
<Limit GET POST>  
    require user dfoerd  
    require user walmue  
</Limit>
```

Alle Webpass-User zulassen:

```
.....  
<Limit GET POST>  
    require valid-user  
</Limit>
```

# 7. Zusatzmaterial: Erweiterung Kursbeispiel

7.1 Die Tabelle abteilungen erstellen

7.2 Fremdschlüssel abteilungs\_id in mitarbeitende erstellen

7.3 Joins: Abfragen mit mehreren Tabellen

7.4 Left Joins und right Joins

7.5 Die php-Skripte zum erweiterten Kursbeispiel

7.6 Das erweiterte Kursbeispiel CD-Konform

## 7.1 Die Tabelle abteilungen erstellen

Feld	Typ	Länge	Null
id	int	10	not null
aname	varchar	20	not null
strasse	varchar	25	not null

Wichtig: Unter "Storage Engine" InnoDB auswählen

<http://idkurs.uzh.ch/phpmyadmin/>



## 7.2 Fremdschlüssel `abteilungs_id` in `mitarbeitende` erstellen

1. In Tabelle `mitarbeitende` ein Feld `abteilungs_id` mit folgenden Eigenschaften erstellen:

Field	<code>abteilungs_id</code>
Type <sup>?</sup>	INT
Length/Values <sup>1</sup>	10
Default <sup>2</sup>	None
Collation	
Attributes	
Null	<input checked="" type="checkbox"/>
Index	INDEX <input type="button" value="v"/>

2. Table `mitarbeitende` und dann Tab Structure auswählen
3. Auf "Relation view" klicken:

<b>abteilungs_id</b>	<code>mykurs17.abteilungen.id</code> <input type="button" value="v"/>
ON DELETE	RESTRICT <input type="button" value="v"/>
ON UPDATE	RESTRICT <input type="button" value="v"/>
<input type="button" value="Save"/>	

<http://idkurs.uzh.ch/phpmyadmin/>

## 7.3 Joins: Abfragen mit mehreren Tabellen

```
select m.id,  
       m.vorname,  
       m.name,  
       a.aname,  
       m.abteilungs_id  
from mitarbeitende m  
join abteilungen a  
   on m.abteilungs_id = a.id;
```

Problem: nur Mitarbeitende, die einer Abteilung angehören werden angezeigt.

## 7.4 Left Joins und right Joins

```
select m.id,  
       m.firstname,  
       m.lastname,  
       a.aname,  
       m.abteilungs_id  
from mitarbeitende m  
left join abteilungen a  
       on m.abteilungs_id = a.id;
```

- Auch Mitarbeitende, die keiner Abteilung angehören, werden angezeigt.
- Abteilungen, die keine Mitarbeitenden haben, werden hingegen nicht angezeigt.
- Mit ...right join... wäre es umgekehrt.

## 7.5 Die php-Skripte zum erweiterten Kursbeispiel

Mitarbeitende eingeben ( Code)

Mitarbeitende anschauen, bearbeiten und löschen  
( Code)

Abteilungen eingeben ( Code)

Abteilungen anschauen, bearbeiten und löschen ( Code)

## 7.6 Das erweiterte Kursbeispiel CD-Konform

Das erweiterte Kursbeispiel gemäss dem Corporate Design der Uni Zürich:

Mitarbeitende eingeben ( Code)

Mitarbeitende anschauen, bearbeiten und löschen  
( Code)

Abteilungen eingeben ( Code)

Abteilungen anschauen, bearbeiten und löschen ( Code)