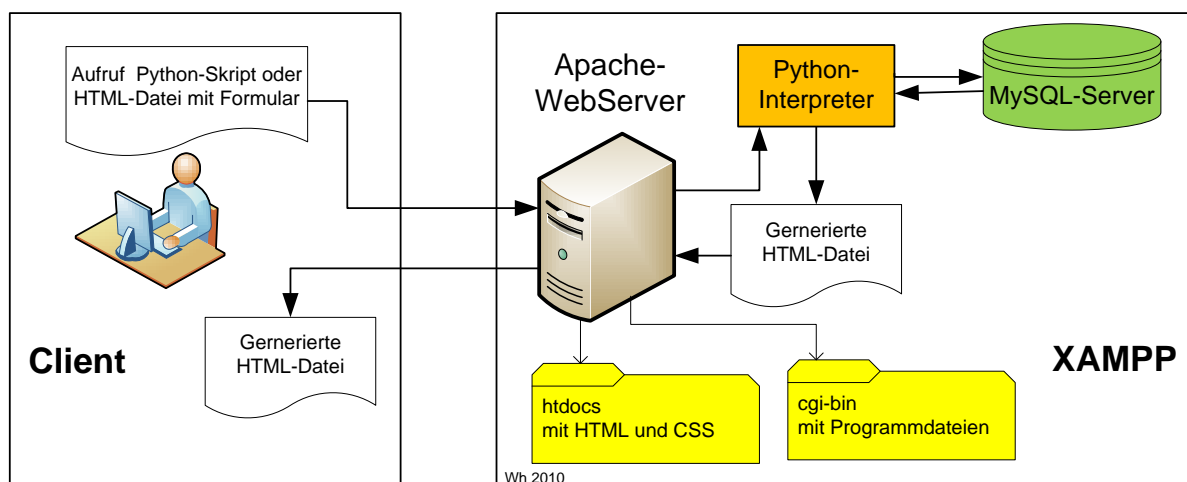


Ein Web-Datenbankprojekt mit CGI-Programmierung, Python und MySQL am Beispiel einer Internetplattform für einen Pizza-Service

Der Umgang mit internetbasierten Datenbanken liegt im Erfahrungsbereich der Lernenden. Sie nutzen soziale Netzwerke, recherchieren u. a. in einem Data-Warehouse-System nach Produkten und bestellen Waren. Ein Webdatenbankprojekt ist daher besonders gut geeignet, an die Lebenswelt der Lernenden anzuknüpfen, die neu erlernten Modelle, Konzepte und Verfahren anzuwenden und mit den Themenfeldern der Einführungsphase und der Qualifikationsphase Q1 in einen gemeinsamen Kontext zu stellen. In Vordergrund steht eigenständiges Erarbeiten im Sinne der Projektmethode im Informatikunterricht. Der Zugriff auf einen Datenbankserver erfolgt mit einer serverbasierten Programmiersprache. Über HTML-Formulare können Datenbestände abgefragt, aktualisiert und eingepflegt werden. Die Lernenden erhalten einen Einblick in die Wirkprinzipien datenbankbasierter Client-Server-Systeme mit dynamischen Webinhalten.

Client-Server-Architektur



Entwicklungsumgebung

Installieren Sie:

1. Die aktuelle Version von XAMPP (XAMP-lite)
2. Python Version 3.3.2 (python-3.3.2.msi)
3. Den MySQL-Connector (mysql-connector-python-1.0.12-py3.3.msi)
4. PyScripter-v2.5.3.zip oder
Eclipse IDE for Java EE Developers mit dem Plugin PyDev
(Eclipse EE herunterladen und PyDev installieren.
Eine Installationsanleitung zu PyDev findet man im Internet)

Ein einfaches Python-CGI-Skript (pizza0.py)

```
#!C:\Python33\python.exe
# Wh 2013 StsGOF

import cgitb; cgitb.enable() # error reporting
import cgi
import time

Zeit = time.localtime()

print ("Content-Type: text/html")
print ()

print ("""
<html>
<head>
<title>Einfaches CGI-Skript</title>
</head>
<body>
<h1>Pizza Einfaches CGI-Skript</h1><br />
Tag %s Monat %s Jahr %s
</body>
</html>
""" % (Zeit[2], Zeit[1], Zeit[0]))
```

Jedes Skript muss zuerst den Python-Interpreter mit `#!C:\Python33\python.exe` aufrufen (dahinter darf kein Kommentar stehen).

Für den WebServer sind die beiden ersten Print-Anweisungen unbedingt notwendig.

Mit `print` wird ein mehrzeiliger, parametrisierter String, der in dreifachen Anführungsstrichen steht, ausgegeben. Er beinhaltet die komplette Webseite. `%s` sind die Platzhalter für den Inhalt der Variablen, die am Ende nach `%` Zeichen in Klammer aufgelistet werden. Die Anzahl und die Reihenfolge muss beachtet werden. `Zeit` ist ein Python-Tupel. Das ganze Tupel wird angezeigt, wenn man nur ein Parameter verwendet und `Zeit` übergibt.

Die Pythonskripte müssen in XAMPP in einem Verzeichnis unterhalb von `cgi-bin` stehen (pizza).

Aufruf mit `http://localhost/cgi-bin/pizza/pizza0.py`.

Zugriff auf die MySQL-Datenbank (pizza1.py)

Legen Sie mit phpMyAdmin die Datenbank *pizzaservice* an und importieren Sie die Datei *pizzaservice.sql*. Ändern Sie das root-Passwort entsprechend.

```
#!C:\Python33\python.exe

import cgi, cgi, mysql.connector

meineDatenbank =
mysql.connector.connect(user="root", password="Offenbach",
                        host="localhost", database="pizzaservice")
meinCursor = meineDatenbank.cursor()
meinCursor.execute("SELECT * FROM angebot")

print ("Content-Type: text/html")
print ()

print ("""
<html>
<head>
<title>Datenbankausgabe</title>
</head>
<body>
<h1>Python-Beispiel: Datenbankausgabe (pizza1.py)</h1><br />
""")

while (1):
    Zeile = meinCursor.fetchone()
    if Zeile == None:
        break
    print (Zeile[0],Zeile[1],Zeile[2],Zeile[3], "</br>")

print ("""
</body>
</html>
""")

meinCursor.close()
meineDatenbank.close()
```

Ausgabe des Angebots in einer HTML-Tabelle (pizza2.py)

Die übliche, aber etwas unschöne while-Schleife in pizza1.py, kann durch

```
for Zeile in meinCursor:
```

ersetzt werden.

```
#!C:\Python33\python.exe
# Wh 2013 StsGOF

import cgitb; cgitb.enable()
import cgi, mysql.connector

meineDatenbank =
mysql.connector.connect(user="root", password="Offenbach",
                        host="localhost", database="pizzaservice")
meinCursor = meineDatenbank.cursor()
meinCursor.execute("SELECT * FROM angebot")

Zeilen = ""
for Zeile in meinCursor:
    Zeilen = Zeilen + """
        <tr>
        <td>%s</td>
        <td>%s</td>
        <td>%s</td>
        <td>%4.2f &euro;</td>
        </tr>
        """ % (Zeile[0],Zeile[1],Zeile[2],Zeile[3])

Angebot = """
<table>
<tr>
    <td>Nr</td>
    <td>Artikelname</td>
    <td>Beschreibung</td>
    <td>Preis</td>
</tr>
    %s
</table>
""" % Zeilen

print ("Content-Type: text/html")
print ()
print ("""
<html>
<head>
<title>Datenbankausgabe</title>
</head>
<body>
<h1>Angebot in einer HTML-Tabelle (pizza2.py) <br/></br>
    %s
</body>
</html>
""" % Angebot)

meinCursor.close()
meineDatenbank.close()
```

In der *for*-Schleife wird der String *Zeilen* mit *Zeilen += ""* ...um die Zeile eines Datensatzes erweitert. Man könnte auch *Zeilen = Zeilen + ""* ... schreiben.

Ausgabe des Angebots mit Link für „in den Warenkorb“ (pizza3.py)

Die Tabellenausgabe enthält jetzt ein Link um einen Datensatz in den Warenkorb zulegen. Über diesen Link wird das Skript erneut aufgerufen, allerdings jetzt mit einer Session-ID (SID) und der Nummer des ausgewählten Datensatzes als Parameter.

Zum Speichern der Auswahl könnte man Cookies verwenden. Meistens werden diese aus Sicherheitsgründen durch die Browsereinstellungen nicht zugelassen. Am besten man speichert die Auswahl in einer Datenbank-tabelle *Session*. Die für die eindeutige Identifizierung notwendige SID erhält man mit

```
z = str(time.time())
```

```
SID = hashlib.sha1(z.encode('utf-8')).hexdigest()
```

aus der genauen Serverzeit, wenn das Skript ohne Parameter zum ersten Mal aufgerufen wird. Wird eine SID mit der Artikelnummer übergeben, so wird der mit der SQL-Anweisung *INSERT INTO* ausgewählte Datensatz in der Tabelle *session* gespeichert. Natürlich müssen diese Einträge nach einer Zerfallszeit wieder gelöscht werden.

```
#!C:\Python33\python.exe
# Wh 2013 StsGOF

import cgi, cgi, mysql.connector
import hashlib, time, os

SID = cgi.FieldStorage().getfirst('SID')

meineDatenbank =
mysql.connector.connect(user="root", password="Offenbach",
                        host="localhost", database="pizzaservice")
meinCursor = meineDatenbank.cursor()
if SID:
    message = "<br> Die ID wurde &uuml;bergeben und die ... ."
    Nr = cgi.FieldStorage().getvalue('Nr')
    SQLInsert = """
        INSERT INTO session (SID, ArtikelNr, ArtikelName, Preis)
        SELECT "%s" AS SID, Nr, ArtikelName, Preis
        FROM Angebot
        WHERE Nr = %s
        """ % (SID, Nr)
    meinCursor.execute(SQLInsert)
    message = message + "<br>" + SQLInsert
else:
    z = str(time.time())
    SID = hashlib.sha1(z.encode('utf-8')).hexdigest()
    message = '<br> Neu Session-ID'

meinCursor.execute("SELECT * FROM Angebot")

Zeilen = ""
for Zeile in meinCursor:
    Zeilen += """
        <tr>
        <td>%s</td>
        <td>%s</td>
        <td>%s</td>
        <td>%4.2f &euro;</td>
        <td> <a href="/.pizza3.py?SID=%s&Nr=%s">in den Wk</a>
        </tr>
        """ % (Zeile[0],Zeile[1],Zeile[2],Zeile[3], SID, Zeile[0])
# & ist im Link das Trennzeichen
```

```
Angebot = """
    <div class="FixedTableHead">
    <table style="height:150px;">
    <thead>
    <tr>
        <td>Nr</td>
        <td>Artikelname</td>
        <td>Beschreibung</td>
        <td>Preis</td>
        <td>in den Wk</td>
    </tr>
    </thead>
    %s
    </table>
    </div>
    %s
    """ % (Zeilen, message)

print ("Content-Type: text/html")
print ()

print ("""
    <html>
    <head>
    <link rel = "stylesheet" type = "text/css" href = "/pizza/pizza.css" />
    <title>Pizza3 DB-Ausgabe mit Warenkorb </title>
    </head>
    <body>
    <h1>DB-Ausgabe "mit in den Warenkorb"</h1><br />
    %s
    </body>
    </html>
    """ % Angebot)

meinCursor.close()
meineDatenbank.close()
```

Das Skript benötigt eine CSS-Datei, die im Verzeichnis *pizza* unterhalb von *htdocs* liegen muss.

Ausgabe des Warenkorbs und mit Bestellformular (pizza4)

```

#
#Der Anfang wurde weglassen!
#
SQLWarenkorb = ""
    SELECT SID, ArtikelNr, ArtikelName, Preis, Anzahl
    FROM Session
    WHERE SID = "%s"
    "" % SID
meinCursor.execute (SQLWarenkorb)
WkZeilen = ""
for WkZeile in meinCursor:
    WkZeilen += ""
    <tr>
    <td><input name="Nr" type="text" value="%s" size="5" readonly> </td>
    <td><input name="Artikel" type="text" value="%s" size="40" readonly></td>
    <td><input name="Preis" type="text" value="%4.2f &euro;" size="5" readly></td>
    <td><input name="Anzahl" type="text" value="%s" size="5"> readonly</td>
    </tr>
    "" % (WkZeile[1],WkZeile[2],WkZeile[3],WkZeile[4])

Bestellung = ""
    <form>
    &nbsp;&nbsp;&nbsp;Sid: &nbsp;&nbsp;&nbsp; &nbsp;&nbsp;&nbsp;
    <input name="ID" type="text" value="%s" size="50">
    <table>
        <tr>
            <td>Name:</td>
            <td><input name="Name" type="text" size="20" maxlength="20"></td>
            <td>Stra&szlig;e:</td>
            <td><input name="Strasse" type="text" size="40" maxlength="40"></td>
            <td>Ort:</td>
            <td><input name="Ort" type="text" size="20" maxlength="20"></td>
            <td>Telefon:</td>
            <td><input name="Telefon" type="text" size="20" maxlength="20"></td>
        <tr>
        </table>
        <table>
        <tr>
            <th>Nr</th>
            <th>Name</th>
            <th>Preis</th>
            <th>Anzahl</th>
        </tr>
        %s
        </table>
    </div>
    <input type="submit" value=" Absenden ">
    <input type="reset" value=" Abbrechen">
    </form>
    "" % (SID, WkZeilen)
# weiter wie bisher

```