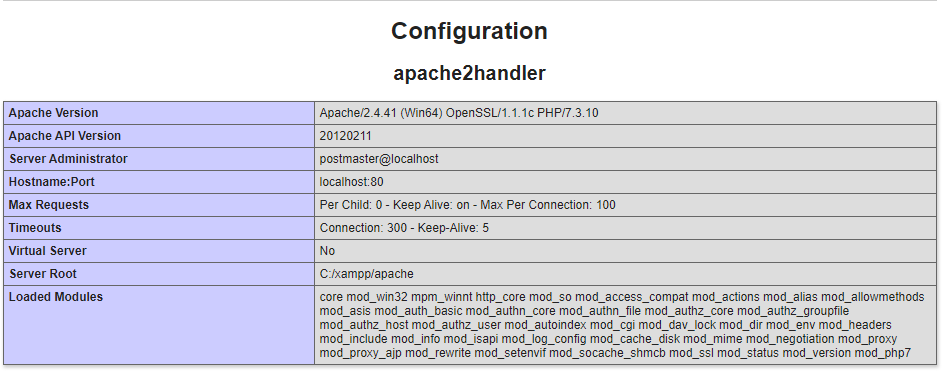
# Vorbereitung

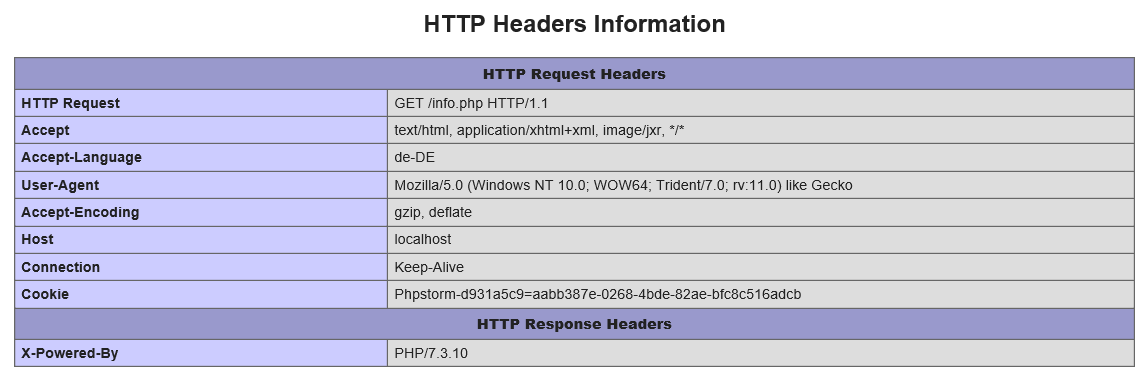
System: Windows NT ADRIANPC 10.0 build 18362 (Windows 10) AMD64

apache2handler: Apache Version: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.10

apache2handler: Server Root: C:/xampp/apache



HTTP Headers Information:



# Meilenstein 1

Beschreiben Sie im Dossier stichpunktartig, wie Sie mit der bisher erlernten Technik (statisches HTML) den Shop umsetzen müssten, wenn

vom Kunden zu allen acht Mahlzeiten die zugehörigen Detail-Seiten gefordert werden.

* jedes eine eigene HTML Seite erstellen
* falle einer design Änderung jede bearbeiten
* Verlinkungen bei jeder Mahlzeit erstellen

# Meilenstein 2

Alter errechnen in PHP, da in der Datenbank alle Einträge überprüft werden müssen und so man beim Aufruf das Alter nur errechnet des aufgeruften Benutzers

Matrikelnummer Überprüfung. Erstmal Datengröße auf 9 Stellen festsetzen, damit es nicht zu viel Platz verbraucht, anschließend mit check testen, ob der einzuspeichernde Wert Zwischen 999999999 und 10000000 liegt.

Aufgaben:

Der 'Studiengang' wird nur die folgenden Werte erlauben: ET, INF, ISE, MCD, WI. Überlegen Sie, welcher Datentyp sich dafür am besten eignen. ✎ Behandeln Sie im Dossier sich ergebende

Vor- und Nachteile.

Studiengang: Enum

Vorteil

* Lässt nur bestimmte zuvor gesetzte Werte zu, und wenn die Speichertypen feste Abkürzungen sind, kann man beim Auflisten der Tabelle schneller sehen, was für ein Studiengang es ist.
* Kein zusätzlicher Test nötig, ob die Werte zugewiesen werden können

Nachteil

* Kein integer Wert, nicht so leicht miteinander zu vergleichen, da am die ganze Zeichenkette durchgehen muss.
* Wenn man die Abkürzung anders schreibt, wird der Wert nicht übernommen.
* Kein integer Wert / Varchar, Tabelle muss neu erstellt werde, wenn ein neuer Studiengang erstellt wird, beim Integer könnte man den Bereich größer machen und zu extern beim Aufrufen den Wert zuweisen.

Welche Relationen lassen sich ohne eigene Tabelle abbilden bzw. welche nicht?

Ohne Zwischentabelle lassen sich 1 zu N Beziehungen abbilden da hier eine der beiden Seite immer nur ein Partner der anderen Seite hat, andersherum jedoch mehrere. Dies lässt sich realisieren durch unterschiedliche Primär keys.

Ohne Zwischentabelle lassen sich 1zu1 Beziehungen verwirklichen, hier haben beide jedoch den Gleichen Primär key. Meist ist es hier schlau, die nachfolgenden mit zu löschen.

N zu M Beziehungen brauchen jedoch Zwischen Tabelle, da eine Liste nicht mehrere Einträge mit gleichen Namen haben kann. Die Fremdschlüssel werden in der Zwischentabelle gespeichert.

Welche Vorteile bietet jene Vorgehensweise?

Ohne Zwischenliste kann man direkt verhindern, das nicht 2 Verbindungen auf ein Element existiert, da der wert im Element ist und in dem falle überschrieben werden würde.

✎ was das Semikolon am Ende einer Anweisung bewirkt

Trennt die SQL Befehle voneinander, wodurch die Befehle hintereinander und nicht als ein ganzer ausgeführt werden.

✎ wie Sie die binären Relationstypen (1:1, 1:N, N:M) abgebildet haben

1 zu 1/N durch FOREIGN KEY in der Tabelle, die eine Beziehung zu einer Bestimmten hat (der FOREIGN KEY ist am Element mit der N Tabelle. Der Hauptschlüssel ist in der 1 Tabelle.

Bei 1 zu 1 ist zusätzlich die Primär Schlüssel beider Tabellen gleich und Referenzieren sich zu einander.

N:M wiederum nutzt Zwischentabellen

✎ den Unterschied zwischen Tabellen- und Spalten-Constraints und wann welche Art sinnvoll ist

Der Unterscheid liegt darin, dass sich die Tabellen Constraints auf die unterschiedlichen Tabellen beziehen und die Spaten Constraints innerhalb einer Tabelle auf die Spalten bezieht

Constrains

✎ wie Sie den Aufzählungsdatentyp ENUM, den MariaDB unterstützt, per CHECK Constraint auch in anderen DBMS nachbilden könnten

In integer Speichern und den Werten einen String zuweisen, Strings zurückgeben. Integer auf einen Bestimmten Werte Bereich festlegen und den Passenden String aus einer Tabelle nehmen.

✎ welche Constraints in MariaDB welchem Zweck dienen

Constrains

* NOT NULL = kein Standartwert, es muss was eingegeben werden
* UNIQUE = eindeutig, Wert darf nur einmal vorkommen, sonst Fehlermeldung
* REFERENCES = fremd Schlüssel, Schlüssel kommt aus anderer Tabelle
* CHECK = Definierte Bedingung prüfen
* PRIMARY KEY = Hauptschlüssel, Spalte, der die Tabelleneinträge benennt
* DEFAULT = setze Standard wert

✎ wieso Sie die Datenbank information\_schema sehen

Die Informationsschemasichten sind in einem speziellen Schema namens INFORMATION\_SCHEMA definiert. Dieses Schema ist in jeder Datenbank enthalten.

Jede Informationsschemasicht enthält die Metadaten für alle in der jeweiligen Datenbank gespeicherten Datenobjekte, wodurch man die Datenbank betrachten kann.

Logindaten HTML

FH Server

<?php $link = mysqli\_connect(  
 "149.201.88.110","s\_af3405s","OYROGPE,4E.","public","3306"  
);

Lokal

<?php $link = mysqli\_connect(  
 "localhost","root","","db3137339",""  
);

Geben Sie in der Überschrift Zutatenliste in Klammern die Anzahl der gefundenen Zutaten an.

m3

Geben Sie in der Überschrift Zutatenliste in Klammern die Anzahl der gefundenen Zutate an.

✎ Welche Möglichkeiten für diese Umsetzung kennen Sie?

Bevor ich den Head reinkopiere, den ich den title als variable übergebe, erstelle ich mit einer Query und den SQL link eine Resultat Liste, welche ich Anschließend mit Mysqli\_num\_rows() durchzähle. Das Ergebnis übergebe ich mit dem Seitennamen den Header

Ab dem 11.11.19 finden in der Datenbank "public" eine Tabelle Zutaten, deren Daten Sie in Ihre eigene Datenbank aus Paket 2 kopieren sollen. ✎ Notieren Sie die dazu erforderliche SQL-Queries im Dossier.

**INSERT** **INTO** db3137339.Zutaten **SELECT** \* **FROM** public.zutaten

Problem, die Reihenfolge der Statements muss gleich sein, sonst landen z.B. Vegan in den Infos für Vegetarisch.

SQL Querrys

Zutaten query:

SELECT \* FROM zutaten ORDER BY Bio DESC, Name;

Produkte query:

SELECT Mahlzeiten.ID, Titel, `Verfügbar`,Beschreibung, Vorrat, Kategorien\_ID, `Name`, `Alt-Text`, `Binärdaten`  
FROM ((Bilder   
INNER JOIN ZT\_Mahlzeiten\_hat\_Bilder ON Bilder.ID=ZT\_Mahlzeiten\_hat\_Bilder.Bilder\_ID)   
INNER JOIN Mahlzeiten ON ZT\_Mahlzeiten\_hat\_Bilder.Mahlzeiten\_ID = Mahlzeiten.ID)  
 ORDER BY `Kategorien\_ID` DESC, `Name`;

Detail querry:

SELECT `Alt-Text`, Mahlzeiten.ID, Titel, `Binärdaten`, `Name`, `Verfügbar`,Beschreibung, Vorrat, Kategorien\_ID, Jahr, Gastpreis, `MA-Preis`, `Student-preis` FROM (((Bilder   
INNER JOIN ZT\_Mahlzeiten\_hat\_Bilder ON Bilder.ID=ZT\_Mahlzeiten\_hat\_Bilder.Bilder\_ID)   
INNER JOIN Mahlzeiten ON ZT\_Mahlzeiten\_hat\_Bilder.Mahlzeiten\_ID = Mahlzeiten.ID )   
INNER JOIN Preise ON Preise.Mahlzeiten\_ID=Mahlzeiten.ID)  
WHERE Mahlzeiten.ID = '.$id.' AND Jahr = '.date("Y").' ORDER BY `Kategorien\_ID`;

Bewerten querry:

SELECT \* FROM `Mahlzeiten` ORDER BY `Name`