

Vorbereitung

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

System	Windows NT ADRIANPC 10.0 build 18362 (Windows 10) AMD64
Build Date	Sep 24 2019 11:51:12

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

apache2handler: Server Root: C:/xampp/apache

Configuration

apache2handler

Apache Version	Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.10
Apache API Version	20120211
Server Administrator	postmaster@localhost
Hostname:Port	localhost:80
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	No
Server Root	C:/xampp/apache
Loaded Modules	core mod_win32 mpm_winnt http_core mod_so mod_access_compat mod_actions mod_alias mod_allowmethods mod_asis mod_auth_basic mod_authn_core mod_authn_file mod_authz_core mod_authz_groupfile mod_authz_host mod_authz_user mod_autoindex mod_cgi mod_dav_lock mod_dir mod_env mod_headers mod_include mod_info mod_isapi mod_log_config mod_cache_disk mod_mime mod_negotiation mod_proxy mod_proxy_ajp mod_rewrite mod_setenvif mod_socache_shmcb mod_ssl mod_status mod_version mod_php7

HTTP Headers Information:

HTTP Headers Information

HTTP Request Headers	
HTTP Request	GET /info.php HTTP/1.1
Accept	text/html, application/xhtml+xml, image/jxr, */*
Accept-Language	de-DE
User-Agent	Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding	gzip, deflate
Host	localhost
Connection	Keep-Alive
Cookie	Phpstorm-d931a5c9=aabb387e-0268-4bde-82ae-bfc8c516adcb
HTTP Response Headers	
X-Powered-By	PHP/7.3.10

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 aufgerufenen 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 100000000 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 man die ganze Zeichenkette durchgehen muss.
- Wenn man die Abkürzung anders schreibt, wird der Wert nicht übernommen.
- Kein integer Wert / Varchar, Tabelle muss neu erstellt werden, 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, dass nicht 2 Verbindungen auf ein Element existiert, da der wert im Element ist und in dem fälle 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 Spalten Constraints innerhalb einer Tabelle auf die Spalten bezieht

Constraints

🔪 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

Constraints

- NOT NULL = kein Standardwert, 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 query:

```

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 query:

```

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

```

Meilenstein 3 | Paket 4

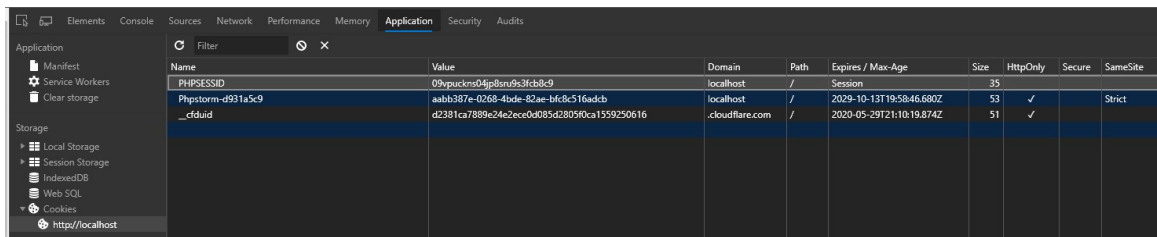
🔑 Notieren Sie im Dossier, wie Ihre SQL Query für die Kategorien abgeändert werden muss, wenn dort nur noch Kategorien mit mindestens einer verfügbaren Mahlzeit zu finden sein werden.

Mahlzeiten Joinen und die Anzahl der Mahlzeiten zählen, Anschließend wie Beim Vegan nur mit größer 1 anzeigen

```
SELECT a.ID, a.Bilder_ID, b.Bezeichnung AS `Ober_Kategorie`, a.Bezeichnung AS `Unter_Kategorie`, COUNT(m.ID) AS Mahlzeit_Anzahl
FROM Kategorien AS a
INNER JOIN Kategorien AS b on a.Kategorien_ID = b.ID
LEFT JOIN Mahlzeiten AS m ON m.Kategorien_ID = a.ID
GROUP BY Unter_Kategorie
HAVING Mahlzeit_Anzahl > 0
ORDER BY Ober_Kategorie, Unter_Kategorie;
```

🔑 ... weshalb Sie plötzlich einen Cookie gesetzt bekommen nachdem Sie Werte in die Session schreiben. Lassen Sie sich den Cookie einmal anzeigen im Browser (Entwickler-Tools).

Die Cookies werden genutzt um verschlüsselt die Session Variablen im Browser zu speichern. Wir erstellen die via Session ID im Login interface. PHP erstellt diese Automatisch.



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite
PHPSESSID	09vpuckms04jp8sruf83fcb8c9	localhost	/	Session	35			
Phipstorm-d931a5c9	aab1387e-0208-4bde-82ae-bfc8-516adcb	localhost	/	2029-10-13T19:58:46.690Z	53	✓		Strict
__cfduid	d2381ca7889e24e2ece0d085d28050ca1559250616	cloudflare.com	/	2020-05-29T21:10:19.874Z	51	✓		

🔑 ... die Situation auf Serverseite, wenn Sie den Cookie löschen und einen weiteren Request absenden.

Wird der Cookie Client seitig gelöscht, verliert der Server seine Session. Ohne diese, ist man nicht mehr als angemeldet gespeichert

🔑 ... wie eine Anmeldung auch ohne Verwendung von Cookies realisiert werden könnte. Recherchieren Sie.

Man kann die PHPSESSID auch in der URL übergeben und entgegennehmen. Über diese Methode kann man die Session auch ohne Cookies aufrechterhalten.

✎ ... wie die Stored Procedure aussieht, die Ihnen den korrekten Preis zu einer genannten Nutzer-Nummer und einer Mahlzeiten-ID zurückgibt. (oder verwenden Sie sie direkt in der Anwendung).

```
DELIMITER /

CREATE PROCEDURE Preisabfrage (
IN NutzerNummer INT(10),
IN Mahlzeiten_ID INT(10),
IN Jahr YEAR
)
BEGIN
SELECT CASE
WHEN (SELECT R.Rolle FROM Rolle R WHERE R.Nummer = NutzerNummer) =
'Mitarbeiter' THEN `MA-Preis`
WHEN (SELECT R.Rolle FROM Rolle R WHERE R.Nummer = NutzerNummer) = 'Gast'
THEN `Gastpreis`
WHEN (SELECT R.Rolle FROM Rolle R WHERE R.Nummer = NutzerNummer) =
'Student' THEN `Student-preis`
END
AS Preis
FROM `preise` P
JOIN Mahlzeiten M ON P.Mahlzeiten_ID = M.ID
WHERE M.ID = Mahlzeiten_ID
AND P.`Jahr` = Jahr;
END;

DELIMITER ;
```

Ein Beispielhafter Aufruf der Stored Procedure wäre :

```
CALL `Preisabfrage`('$SESSION['UserID']', '$row['Mahlzeiten_ID']',
'.date("Y").'
)
```

Dieser würde lediglich den Preis zu der Rolle des Nutzers von der bestimmten Mahlzeit ausgeben.

```
$_SESSION['user'] = Namevoll (vorname nachname
$_SESSION['role'] = Student, Mitarbeiter, Gast
```

```
CREATE VIEW Rolle AS
WITH temp AS (SELECT S.`Student_ID` AS Nummer, 'Student' AS Rolle FROM
`studenten` AS S
UNION SELECT G.`Gäste_ID` AS Nummer, 'Gast' AS Rolle FROM `gäste` AS G
UNION SELECT M.`Mitarbeiter_ID` AS Nummer, 'Mitarbeiter' AS Rolle FROM
`mitarbeiter` AS M
UNION SELECT F.`FHAngehörig_ID` AS Nummer, 'FH-Angehörige' AS Rolle FROM `fh
angehörige` AS F)
SELECT temp.Nummer AS Nummer, temp.Rolle AS Rolle from temp group by
temp.Nummer;
```


Meilenstein 3 | Paket 5

✎ Notieren Sie im Dossier, welche verschiedenen Fehler durch Ihre Constraints entstehen können. Erzeugen Sie diese Fehler per SQL einmal und notieren Sie sich die Fehlermeldung im Dossier. Es geht vorrangig um das Anlegen von Studierenden.

Möglicher Fehler Nr.1:

Den neuen Studenten zuerst oder nur in die Tabelle Studenten einzufügen.

```
INSERT INTO studenten (Student_ID,Matrikelnummer,Studiengang) VALUES (12,'12345678','INF');
```

Liefert folgenden SQL Fehler:

```
SQL Fehler (1452): Cannot add or update a child row: a foreign key constraint fails (`db3137339`.`studenten`, CONSTRAINT `studenten_sind_FHAngehoerige` FOREIGN KEY (`Student_ID`) REFERENCES `fh_angehoerige` (`FHAngehorig_ID`) ON DELETE CASCADE ON UPDATE CASCADE)
```

Damit dieser Fehler nicht eintritt, muss man den Studenten zuerst in die Tabelle `benutzer`, dann `fh_angehoerige` und zum Schluss in `studenten` einfügen.

(kann über mysqli_multi_query() erfolgen, jedoch ist es Einzel besser, da während dessen PHP weiter arbeitet)

Möglicher Fehler Nr.2:

Den neuen Studenten eine nicht UNIQUE Matrikelnummer einzutragen.

Wenn man eine schon vorhandene Matrikelnummer einfügt

```
INSERT INTO studenten (Student_ID,Matrikelnummer,Studiengang) VALUES (6,'31373390','INF');
```

Liefert folgenden SQL Fehler:

```
SQL Fehler (1062): Duplicate entry '31373390' for key 'MatrikelnummerEindeutig'
```

Möglicher Fehler Nr.3:

Den neuen Studenten eine nicht UNIQUE Email Adresse zuordnen.

```
INSERT INTO Benutzer (Vorname, Nachname, Nutzernamen, `E-Mail`) VALUES ('Adrian', 'Freytag', 'AF123', 'Adrian.Freytag@alumni.fh-aachen.de');
```

Liefert ähnlichen Fehler wie bei Nr.2

```
SQL Fehler (1062): Duplicate entry 'Adrian.Freytag@alumni.fh-aachen.de' for key 'E-MailEindeutig'
```

✎ ... welche Probleme Ihnen aufgefallen sind bei der Umsetzung des MVC-Pattern.

Unwissenheit beim Umsetzen allgemein, auch, ob es richtig ist, Verständnis.

Übergeben von verschiedenen Datentypen- einfache Variable oder array

✎ ... welche Situationen es geben kann, wenn wirklich viele Benutzer gleichzeitig diese Art von Registrierung durchführen.

Dass die ID durch **AUTO_INCREMENT** doppelt vergeben wird, wenn mehrere Benutzer gleichzeitig eine **INSERT INTO** Transaktion starten.

Das die Überprüfung des Anmeldenamen nach der Überprüfung doch schon vergeben wird

✎ ... welcher Aspekt in der E-Mensa sich noch für Transaktionen anbieten könnte.

Vorbestellen der Mahlzeiten