

## Praktikum 1

### Paket 1

#### Zu Aufgabe 1.2

Wenn vom Kunden zu allen acht Mahlzeiten im Shop die zugehörigen Detailseiten gefordert werden müssten, dann:

- müsste für jede Mahlzeit eine eigene HTML-Datei erstellt werden, die die spezifischen Informationen für die Mahlzeit enthält
- müssen gleiche Inhalte verschiedener Dateien (z.B. Navigationselemente) auf jeder Datei neu definiert werden
- müssen Änderungen am Seitenaufbau & -design auf jeder Datei einzeln umgesetzt werden, bedingt durch das statische HTML

#### Zu Aufgabe 1.6

Ein Drop-Down-Menü in einem HTML-Formular kann angeboten werden durch die Benutzung von einem `<select>`-Tag. Einzelne Elemente darin können durch das `<option>`-Tag erzeugt werden.

Mit dem `<optgroup>`-Tag werden Elemente in einer unteren Ebene gruppiert, der Inhalt der oberen Ebene wird dann durch das `label`-Attribut im `<optgroup>`-Tag festgelegt. Diese obere Ebene dient nur der Darstellung und kann deswegen nicht ausgewählt werden. Um untere Elemente ebenfalls als nicht auswählbar zu definieren, wird das `disabled`-Attribut im jeweiligen `<option>`-Tag gesetzt.

Zusätzliche nützliche Attribute können das `multiple`-Attribut zur Auswahl mehrerer Elemente im Menü, oder das `size`-Attribut zur gleichzeitigen Anzeige einer bestimmten Anzahl von Elementen sein.

### Fragen und Antworten während der Praktikumsbearbeitung

Wie lässt sich ein Tab-Menü innerhalb einer HTML-Seite realisieren?

Ein Tab-Menü wird optisch durch ein Feld mit mehreren Buttons darin realisiert. Die Felder werden in der `.css`-Datei mit der `border`-Einstellung ausgestaltet, die Buttons werden dort mit der `float`-Einstellung (hier) an den linken Rand des Feldes gesetzt. Innerhalb der Button-Tags wird mit dem `onclick`-Attribut eine Relation zum Inhalt erstellt. Die Relation kommt durch die beidseitige Nennung der ID zustande. Für die korrekte Funktionsweise muss die HTML-Seite um eine JavaScript-Datei ergänzt werden, sie enthält Informationen darüber, wie mit den Klassen der Elemente während der Interaktion umgegangen wird.

Wie wird durch einen Link ein neues Fenster / Tab geöffnet?

Link-Tags benötigen dazu das Attribut `target`, dies wird gefüllt mit „\_blank“.

Wie werden Bilder ausgegraut?

Man erstellt in der `.css`-Datei eine Klasse, die die Einstellung „-webkit-filter: grayscale(1);“ beinhaltet.

Wie können Links deaktiviert werden?

In der `.css`-Datei wird eine Klasse erstellt, die die Einstellungen „pointer-events: none; cursor: default; text-decoration: none; color: black;“ enthält, sodass sich der Link wie ein normaler Text verhält.

## Praktikum 2

### Paket 2

#### Wie werden die verschiedenen Relationstypen abgebildet?

Eine 1:1-Relation wird durch das Hinzufügen eines Fremdschlüssel-Attributs in einer der betroffenen Tabellen abgebildet. Vorzugsweise wird die Tabelle ausgewählt, die durch den Fremdschlüssel am ehesten NULL-Werte vermeiden kann.

Eine 1:N-Relation wird durch das Einfügen eines neuen Attributs in der N-Tabelle erzielt. Dieses Attribut referenziert auf den Primärschlüssel der 1-Tabelle und bildet diese somit ab.

Eine M:N-Relation wird mit einer zusätzlichen Tabelle in zwei 1:N-Relationen aufgeteilt. Die neue Tabelle bildet zwischen den beiden ursprünglichen Tabellen den N-Relationsteil und enthält somit jeweils den Primärschlüssel beider Tabellen.

#### Was ist der Unterschied zwischen Tabellen- und Spalten-Constraints und wann sind diese sinnvoll?

Im Gegensatz zu Tabellen-Constraints beziehen sich Spalten-Constraints lediglich auf eine einzige Spalte der Tabelle und limitieren nur dort zielgerichtet die Daten bzw. Datentypen. Damit können Tabellen besser strukturiert werden und unterschiedliche Inhalte in verschiedenen Attributen festgehalten werden. Tabellen-Constraints eignen sich gut dafür, völlig ungeeignete Inhalte komplett aus einer Tabelle ausschließen zu können.

#### Welche Constraints dienen in MariaDB welchem Zweck?

- UNIQUE: Betroffene Spalten dürfen Attributwerte nur genau einmal enthalten
- PRIMARY KEY: Spalten mit diesem Constraint werden für die Referenzierung einzelner Zeilen dieser Tabelle genutzt. Diese dürfen deswegen nicht leer sein und dürfen nur eindeutige Werte enthalten (vgl. UNIQUE)
- FOREIGN KEY: Spalten mit diesem Constraint sollen eindeutig auf eine Spalte einer anderen Tabelle referenzieren. Die fremde Spalte muss dafür ein PRIMARY KEY-Constraint enthalten.
- CHECK: Spaltenwerte werden auf die festgelegten Eigenschaften in diesem Constraint überprüft. Bei einer Abweichung wird die betroffene Zeile mit den Zellenwert nicht verändert oder eingefügt.

#### Was sind die Vor- / Nachteile des Aufzählungsdatentyps ENUM und wie kann er per CHECK Constraint in anderen DMBS nachgebildet werden?

Der Aufzählungsdatentyp ENUM eignet sich gut dafür, eine festgelegte Reihe an Werten speichern zu können. Es kann in Datenbanken hilfreich sein, ausschließlich solche festen Werte zu akzeptieren, damit Tabellen redundanzfrei von ähnlichen Attributwerten gehalten werden können. Andererseits kann es auch hinderlich sein, nur bestimmte Werte in ein Datenbank-Attribut einspeichern zu können. In anderen DMBS müssten mehrere CHECK-Constraints für den eingegebenen Wert festgelegt werden, sodass überprüft werden kann, ob dieser Wert einem der zu akzeptierenden Werte entspricht.

#### Was bewirkt das Semikolon am Ende einer Anweisung?

Das Semikolon am Ende einer Anweisung schließt einzelne Statements (bspw. solche mit CREATE, DROP, INSERT oder REPLACE) in der Datei ab, sodass eine Kollision und Fehlinterpretation mehrerer Anweisungen verhindert wird. Es übernimmt standardmäßig die Funktion des Delimiters.

## Paket 3

### Verwendete SQL-Queries in diesem Paket

- Zutatenliste: "SELECT Name, Vegan, Vegetarisch, Glutenfrei, Bio FROM Zutaten ORDER BY Bio DESC, Name ASC"
- Detail-Seite: "SELECT Mahlzeiten.Name, Mahlzeiten.Beschreibung, Mahlzeiten.Verfuegbar, Bilder.AltText, Bilder.Titel, Bilder.Binaerdaten, Preise.Gastpreis, Preise.Studentpreis, Preise.MAPreis FROM Mahlzeiten JOIN MahlzeitenMBilderN ON MahlzeitenMBilderN.MahlzeitenID = Mahlzeiten.ID JOIN Bilder ON MahlzeitenMBilderN.BildID = Bilder.ID JOIN Preise ON Preise.MahlzeitenID = Mahlzeiten.ID WHERE Mahlzeiten.ID = " + mahlZeitID

### Fragen und Antworten während der Praktikumsbearbeitung

#### Wie genau kann die Einbindung eines Bildes in eine Datenbank realisiert werden?

Die Bilddatei muss in das Base64-Format konvertiert werden. Die Zeichenfolge kann dann in einem Attribut des Typs BLOB (Binary Large Object) in der Datenbank gespeichert werden. Mit einem Helper in einer .cshtml-Datei kann die Zeichenfolge dann für das src-Attribut eines <img>-Tags verwendet werden, um den Inhalt der Bilddatei in eine Webseite einzubinden.

#### Was ist der Vorteil davon, auf neue Tabellen für Relationen zu verzichten?

In 1:1-Relationen und 1:N-Relationen können die Relationen durch Fremdschlüssel erzeugt werden. Neue Tabellen müssen dafür nicht erstellt werden. Dies verhindert Redundanzen, da die Primärschlüssel nicht künstlich erzeugt und zusätzlich in einer neuen Tabelle hinterlegt werden müssen.

#### Wie wird eine Spezialisierung in einer Datenbank umgesetzt und welche Vorteile bietet sie?

Eine Spezialisierung kann eine neue Tabelle hervorrufen, die zum einen Attribute der übergeordneten Entität enthält und zum anderen neue, für diese Tabelle spezifische Attribute enthält. In der spezialisierten Tabelle referenziert dann ein neuer Fremdschlüssel auf die ursprüngliche Tabelle. Mit der Spezialisierung können spezifischere Attribute erstellt werden, die in der übergeordneten Tabelle möglicherweise Probleme mit weiteren Datensätzen verursacht hätten.

#### Wie wird das kaskadierende Löschen in den spezialisierten Tabellen abgebildet?

Die mithilfe eines Constraints gebildete Referenzierung zu der übergeordneten Tabelle muss die Schlüsselbegriffe ON DELETE CASCADE enthalten, um das kaskadierende Löschen zu ermöglichen.

## Praktikum 3

### Paket 4

Welche HTTP-Methode wurde für das Speisenfilter-Formular verwendet und warum?

Die GET-Methode wurde für das Speisenfilter-Formular verwendet. So werden die vom Benutzer gesetzten Werte in den Checkboxes / dem Dropdown-Menü ausgelesen. Der definierte Name innerhalb der <select>- oder <input>-Tags wird dann mit seinem ausgelesenen Wert dem URL angefügt. Für weitere Seitenaufrufe kann dann mit dem URL weitergearbeitet werden.

Warum wird ein Cookie gesetzt, nachdem Werte in die Session geschrieben wurden?

Mit dem Cookie kann die Session ID vom Client zum Server mit jedem Request mitgesendet werden. Eine Übertragung vom Server zum Client ist dank des Cookies nur einmal nötig, da die benötigten Daten danach lokal durch den Cookie auf dem Client gespeichert sind.

Was passiert auf der Serverseite nach dem Löschen eines Cookies und einem Absenden eines weiteren Requests?

Der nächste Request wird vom Server als eine erste Anfrage einer neuen Sitzung angesehen. Es wird also eine neue Sitzung gestartet, die alte Sitzung geht dabei verloren. Für die neue Sitzung muss ein neuer Cookie vom Server übertragen werden und für weitere Requests wieder lokal im Client hinterlegt werden.

Wie kann eine Anmeldung auch ohne Cookies realisiert werden?

Die Informationen können mit jedem Seitenaufruf in dem dazugehörigen Uniform Resource Identifier hinterlegt werden. Dabei kann die Session ID entweder einen Teil des Pfades des URIs bilden oder ein GET-Parameter am Ende des URIs sein. Diese Lösung erfordert einen höheren Programmieraufwand und kann durch erhöhte Fehleranfälligkeit zu gelegentlichen Verlusten von Sitzungen führen.

Wie sieht eine Stored Procedure für die Nennung des korrekten Preises zu einer Nutzer-Nummer und einer Mahlzeiten-ID aus?

Für die Ausgabe des korrekten Preises muss die bereits zuvor definierte Stored Procedure zur Ausgabe der Nutzerrolle aufgerufen werden. Danach wird je nach Ausgabe ein anderes Preisattribut aus der Preis-Tabelle selektiert.

```
CREATE PROCEDURE Preis(IN BenutzerNummer INT, IN MahlzeitenID INT)
BEGIN
    CALL Nutzerrolle(BenutzerNummer, @role);
    CASE @role
        WHEN 'Student' THEN SELECT Studentpreis AS Preis FROM Preise WHERE
Preis.MahlzeitenID = MahlzeitenID;
        WHEN 'Mitarbeiter' THEN SELECT MAPreis AS Preis FROM Preise WHERE
Preis.MahlzeitenID = MahlzeitenID;
        ELSE SELECT Gastpreis AS Preis FROM Preise WHERE Preis.MahlzeitenID =
MahlzeitenID;
    END CASE;
END;
```

Was transportiert die Information „sha1:64000:18“ zu Beginn eines Hash-Strings und wie wirkt sich diese Angabe auf Hash & Salt aus?

Der Hash-String soll mithilfe der Variante SHA-1 verarbeitet werden. Dazu wird die Funktion PBKDF2 64000-mal ausgeführt. Der Hash-Wert wird dann eine Größe von 18 Bytes haben.

## Praktikum 4

### Paket 6

#### Zu Aufgabe 6.1

Welche Situationen können sich bei der Nutzung von Triggern für die Reduzierung der Vorräteanzahlen der Mahlzeiten ergeben?

Im Normalfall werden mit der Auslösung des gespeicherten Triggers in der Datenbank alle Vorratsanzahlen der Mahlzeiten in der Mahlzeiten-Tabelle erfolgreich um die Anzahlen in der Bestellung verringert. Zusätzlich kann es sein, dass ein Wert einer Mahlzeit bereits auf 0 gesetzt war und mit der nächsten Bestellung negativ wird. In dem Fall sollte der Bestellprozess abgebrochen werden und dem Benutzer mitgeteilt werden, dass die Bestellung fehlgeschlagen ist.

#### Zu Aufgabe 6.2

Mit welchem Status-Code wurde ein Request bei fehlendem oder fehlerhaftem HTTP-Header zurückgewiesen und warum?

Der Request wurde mit dem Status-Code „401: Unauthorized“ beantwortet, da dieser bereits impliziert, dass keine Anfrage ohne gültige Authentifizierung akzeptiert werden kann. Für nicht gültige Authentifizierungsversuche ist es in Kombination mit diesem Status-Code üblich, dass danach im „WWW-Authenticate“-Header-Feld weitere Informationen über die notwendige Authentifizierung bereitgestellt werden.

Gibt es andere HTTP-Header, die für eine Authentifizierung bereits vorgesehen sind?

Ein weiterer möglicher HTTP-Header für Authentifizierungen kann „Authorization“ sein. Dieser wird standardmäßig für HTTP-Authentifizierungsverfahren verwendet.

Beispiel: `Authorization: Basic QWxhZGRpbjpvGVuIHNLc2FtZQ==`

Für Autorisierungsdaten in Proxys eignet sich zusätzlich der Header „Proxy-Authorization“.

Beispiel: `Proxy-Authorization: Basic QWxhZGRpbjpvGVuIHNLc2FtZQ==`

Welche Möglichkeiten sind besser geeignet, Anfragen von anderen Diensten zu authentifizieren, als per festgelegtem HTTP Header?

Besser geeignete Möglichkeiten zur Authentifizierung sind unter anderem Tokens (insbesondere JSON Web Tokens oder OAuth) und Signaturen außerhalb von Browsern.

Bei JSON Web Tokens werden ein Header, der den Tokentyp und den verwendeten Hashing-Algorithmus enthält, ein Payload und eine Signatur verwendet.

Signaturen können auch (außerhalb von Browsern; bei APIs) zur Verschlüsselung für die gesamte Anfrage genutzt werden. Zur Berechnung des Hashes können dann HTTP-Methode, HTTP-Header, die Checksumme des HTTP-Payloads, der Anfragepfad und ein privater Schlüssel verwendet werden. Damit ist gesichert, dass unautorisierte Benutzer höchstens in der Lage sind, den Datenverkehr auszulesen. Sie können sich aber nicht als ein autorisierter Nutzer ausgeben, da sie nicht in der Lage sind, neue Anfragen selbst zu signieren.

Wie sieht ein Beispielaufruf über diese Methoden aus?

Um eine Signatur für JSON Web Tokens zu generieren, wird in diesem Fall der HMAC SHA256-Verschlüsselungsalgorithmus aufgerufen:

`HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), secret)`