

Paket 1:

Lessons Learned

Problem: „<fieldset>“ hat keinen Rand.

Lösung: Bootstrap entfernt Ränder, Padding & Margin von „<fieldsets>“ & „<legend>“. Manuell in CSS einfügen.

Problem: Lupen-Icon in Suchleiste einfügen, sodass es immernoch angezeigt wird, wenn die Leiste gefüllt ist.

Lösung: Bootstrap hat eine Möglichkeit Inputs zu gruppieren („input-group“). Button wird mit einem div der Klasse „input-group-prepend“ vor das Inputfeld eingefügt („input-group-append“ zum dranhängen).

Aufgaben

Aufgabe 1.2:

Es wäre ziemlich aufwendig für jede einzelne Mahlzeit (hier acht) eine eigene Seite zu machen und somit die Arbeit quasi verachtfachen. Außerdem würde dies viel Aufwand erfordern, wenn eine neue Mahlzeit hinzugefügt wird.

Aufgabe 1.6:

Mit dem „<select>“-Tag fängt man ein Drop Down Menü an. Einzelne Optionen kann man mit dem „<option>“-Tag hinzufügen.

Mit dem „<optgroup>“-Tag kann man eine Ebene bzw. Kategorie zur besseren Navigation einbauen.

Um eine Option nicht auswählbar zu machen, kann man das „disabled“ Attribut auf „disabled“ stellen.

Die „multiple“, „required“ und „size“ Attribute scheinen sehr hilfreich zu sein, wenn man mit größeren bzw. mehreren Drop Down Menüs arbeitet.

Paket 2:

Aufgabe 2.1:

Das Feld „Studiengang“ in der Tabelle Studenten soll nur die Werte ET, INF, ISE, MCD und W annehmen.

Hierfür kann man ein Enum erstellen („Studiengang ENUM('ET', 'INF', 'ISE', 'MCD', 'WI')“) oder eine zweite Tabelle, die dann referenziert werden kann (mit FOREIGN KEY).

Ersteres ist vorteilhafter, wenn das Enum nur einmal genutzt werden soll, da dies keine neue Tabelle erfordert.

Zweiteres ist jedoch bei häufiger Benutzung der Werte zu empfehlen, da somit nicht bei der Definition jedes Feldes eine lange Enum Liste eingefügt werden muss.

Aufgabe 2.2:

1:1 & 1:N Relationen kann man ohne eigene Tabellen erzeugen.

Bei N:M Relationen hingegen benötigt man eine Referenztafel.

Für die Spezialisierung nutzt man in den spezialisierten Tabellen einen Primary Key, der jedoch auch ein Foreign Key ist, der den Primärschlüssel der generalisierten Tabelle referenziert. Ein Nachteil bei dieser Vorgehensweise ist, dass man nicht direkt einen spezialisierte Eintrag erstellen kann, sondern erst einen Eintrag in der generalisierten Tabelle erstellen muss.

Kaskadiertes Löschen kann man mit ON DELETE CASCADE in der spezialisierten Tabelle erzeugen.

Das Semikolon am Ende eines Statements zeigt dem Parser, dass das Statement endet. In manchen Fällen kann es auch ausgelassen werden (z.B. wenn nichts darauf folgt).

Paket 4:

Aufgabe 4.2:

Sessions speichern eine einzigartige SessionID in den Cookies des Clients.

Diese dient zur Identifizierung des Clients, wenn dieser sich zu einem späteren Zeitpunkt wieder verbinden will.

Wenn man diesen Cookie nun löscht und einen weiteren Request sendet, kann der Server nicht den Client wiedererkennen (da keine SessionID in seinen Cookies existiert) und behandelt ihn als nicht angemeldeten Client.

Session ohne Cookies: Indem man einer Session eine einzigartige ID zuweist und dem Client übergibt. Dieser kann dann per POST die ID schicken und daran erkannt werden.

Stored Procedure PreisfürNutzer: Siehe P3.sql

Sha1:64000:18

Sha1 beschreibt den benutzten Hashing Algorithmus (Secure Hash Algorithm).

64000 beschreibt die Iterationen des Algorithmus.

18 die Menge der Bytes (144 Bits) des Hashes.

Paket 6:

Aufgabe 6.1:

Trigger Situationen: Before/After Insert/Delete/Update

Aufgabe 6.2:

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

Authorization/Proxy-Authorization

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

Z.B. OAuth