

Meilenstein 3 – Werbeseite (mit Datenbank)

Ziel ist die Erweiterung der Werbeseite um dynamische Inhalte, wie die dargestellten Statistiken sowie eine Übersicht von angebotenen Gerichten.

Dazu installieren wir eine Datenbank, legen eine erste Struktur an und stellen erste Datenanfragen an die Datenbank. Abschließend stellen wir Daten aus der Datenbank in der Werbeseite der E-Mensa Werbeseite dar.

Empfehlung VL: Datenbanken Einführung, SQL 1

Aufgabe 1n

Datenbank installieren. Laden Sie das Datenbankmanagementsystem MariaDB „Community“ von der Seite

<https://mariadb.com/downloads/>

herunter und installieren Sie diese auf Ihrem Entwicklungsrechner. Stellen Sie mit einem beliebigen Werkzeug (Empfehlung: PHPStorm) erfolgreich eine Verbindung zur Datenbank her. Verwenden Sie dazu den existierenden Adminaccount (Name: Root).

Aufgabe 2

Datenbank: Strukturen und Daten anlegen. Nach der Installation der Datenbank richten wir die Struktur sowie die Daten ein. Anschließend stellen wir Anfragen an die Datenbank.

Verwenden Sie den nach der Installation existierenden Adminaccount für die weitere Arbeit mit der Datenbank.

*Hinweis: In realen Projekten arbeiten wir **niemals** mit dem Adminaccount direkt aus der Anwendung heraus auf der Datenbank! Dies dient lediglich zur Vereinfachung der Handhabung im Praktikum.*

- 1) Legen Sie eine neue Datenbank (bzw. ein neues Schema) **emensawerbeseite** an. Verwenden Sie den Zeichensatz UTF8mb4 mit Collation utf8mb4_unicode_ci.

Hinweis: Int8 steht für einen 8-Byte Integer.

gericht

Attribut	Typ	Kommentar
id	int8, PKEY	Primärschlüssel
name	varchar(80), not null, unique	Name des Gerichts. Ein Name ist eindeutig.
beschreibung	varchar(800), not null	Beschreibung des Gerichts.
erfasst_am	date, not null	Zeitpunkt der ersten Erfassung des Gerichts.

Attribut	Typ	Kommentar
vegetarisch	boolean, not null	Markierung, ob das Gericht vegetarisch ist. Standard: Nein.
vegan	boolean, not null	Markierung, ob das Gericht vegan ist. Standard: Nein.
preisintern	double, not null	Preis für interne Personen (wie Studierende). Es gilt immer preisintern > 0.
preisextern	double, not null	Preis für externe Personen (wie Gastdozent:innen).
	Nebenbedingung	preisintern ist immer kleiner-gleich preisextern

allergen

Attribut	Typ	Kommentar
code	char(4), PKEY	Offizieller Abkürzungsbuchstabe für das Allergen.
name	varchar(300), not null	Name des Allergens, wie „Glutenhaltiges Getreide“.
typ	varchar(20), not null	Gibt den Typ an. Standard: „allergen“

kategorie

Attribut	Typ	Kommentar
id	int8, PKEY	Primärschlüssel
name	varchar(80), not null	Name der Kategorie, z.B. „Hauptgericht“, „Vorspeise“, „Salat“, „Sauce“ oder „Käsegericht“.
eltern_id	int8	Referenz auf eine (Eltern-)Kategorie. Es soll eine Baumstruktur innerhalb der Kategorien abgebildet werden. Zum Beispiel enthält die Kategorie „Hauptgericht“ alle Kategorien, denen Gerichte zugeordnet sind, die als Hauptgang vorgesehen sind.
bildname	varchar(200)	Name der Bilddatei, die eine Darstellung der Kategorie enthält.

gericht_hat_allergen

Attribut	Typ	Kommentar
code	char(4)	Referenz auf Allergen.
gericht_id	int8, not null	Referenz auf das Gericht.

gericht_hat_kategorie

Attribut	Typ	Kommentar
gericht_id	int8, not null	Referenz auf Gericht.
kategorie_id	int8, not null	Referenz auf Kategorie.

- Erstellen Sie die notwendigen SQL-Statements (DDL) zur Erzeugung der Struktur und führen Sie diese auf der Datenbank aus. Ignorieren Sie zunächst die Prüfungen der referentiellen Integrität (Referenzen) sowie von

Eindeutigkeiten (unique). Geben Sie die verwendeten SQL-Statements mit ab.

- 3) Laden Sie die Daten in die von Ihnen erzeugte Struktur, indem Sie die Datei **werbeseite_daten.sql** einlesen.
- 4) Überprüfen Sie pro Tabelle mit einem SQL-Statement, ob Sie alle Datensätze aus der **werbeseite_daten.sql** eingelesen haben. Verwenden Sie Abfragen, womit Sie die vorhandenen Datensätze pro Tabelle zählen und vergleichen Sie das Ergebnis mit der Anzahl der Datensätze in der Datei. Welche Abfragen verwenden Sie?

Aufgabe 3

Übung. Datenbank: Anfragen. Erstellen Sie die folgenden Anfragen in SQL und führen Sie diese auf der angelegten Datenstruktur durch (z.B. in HeidiSQL oder PHPStorm). Geben Sie sowohl das Statement als auch die Ergebnismenge (z.B. als Screenshot) in Ihrem Dossier mit ab.

- 1) Alle Daten aller Gerichte.
- 2) Das Erfassungsdatum sowie den Namen aller Gerichte.
- 3) Das Erfassungsdatum sowie den Namen (als Attributname Gerichtname im Ergebnis) aller Gerichte absteigend sortiert nach Gerichtname.
- 4) Den Namen sowie die Beschreibung der Gerichte aufsteigend sortiert nach Namen, wobei nur 5 Datensätze dargestellt werden sollen.
- 5) Ändern Sie die vorherige Abfrage so ab, so dass 10 Datensätze dargestellt werden, die nach den ersten 5 Datensätzen folgen. (Die ersten 5 Datensätze werden übersprungen)
- 6) Zeigen Sie alle möglichen Allergen-Typen (typ), wobei Sie keine doppelten Einträge darstellen.
- 7) Namen von Gerichten, deren Name mit einem klein- oder großgeschriebenen „L“ beginnt.
- 8) Ids und Namen von Gerichten, deren Namen ein „suppe“ an beliebiger Stelle enthält absteigend sortiert nach Namen.
- 9) Alle Kategorien, die keine Elterneinträge besitzen.
- 10) Korrigieren Sie den Wert „Dinkel“ in der Tabelle allergen mit dem code a6 zu „Kamut“.
- 11) Fügen Sie das Gericht „Currywurst mit Pommes“ hinzu und tragen Sie es in der Kategorie „Hauptspeise“ ein.

Aufgabe 4

Datenbankanbindung: mysqli. Zur Herstellung einer Datenbankverbindung in PHP stehen einige Funktionen zur Verfügung. Sie erhalten den Foliensatz

DBWT_M3_mysqli.pdf sowie die **connection_example.php**, welche die

Anwendung von mysqli in PHP erklären und zeigen. Setzen Sie eine Datenbankverbindung in PHP um.

Schreiben Sie eine PHP-Seite im Ordner **beispiele** mit dem Namen ...

- 1) **m3_4a_testdatenbank.php**, die eine Verbindung zur Datenbank herstellt, ein beliebiges SELECT Statement aus der vorherigen Aufgabe ausführt und die Ergebnisse in einer Tabelle darstellt.

Aufgabe 5

Werbeseite: Gerichte aus Datenbank. Die Datenbank soll nun die Daten für die Darstellung der Gerichte auf der Werbeseite liefern.

- 1) Laden Sie die Gerichte aus der Datenbank und zeigen Sie die Informationen (Name, Preisintern, Preisextern) auf der Werbeseite an. Sortieren Sie aufsteigend nach Namen. Stellen Sie 5 Gerichte dar.
- 2) Nutzer:innen sollen die Gerichte umsortieren können. Erweitern Sie dazu die neue Darstellung der Gerichte um eine Möglichkeit, die Sortierungsreihenfolge zu wechseln (von aufsteigend nach absteigend, und umgekehrt). Übertragen Sie die Information zur Sortierung via URL-Abfrage zum Webserver.
- 3) Zeigen Sie zusätzlich pro Gericht die enthaltenen Allergene auf der Werbeseite an, wobei Sie nur den Code des Allergens bei den Gerichten in sinnvoller Darstellung anzeigen. Zeigen Sie unter den Gerichten eine Liste der verwendeten Allergene an.
- 4) (Optional) Zeigen Sie 5 zufällige Gerichte an.

Woche 2

Empfehlung VL: SQL 2, DB Entwurf (Relationenmodell für Aufgabe 10b und c)

Aufgabe 6

Übung. Datenbank: Anfragen. Erstellen Sie die folgenden Anfragen in SQL und führen Sie diese auf der angelegten Datenstruktur durch (z.B. in HeidiSQL oder PHPStorm). Geben Sie sowohl das Statement als auch die Ergebnismenge (z.B. als Screenshot) in Ihrem Dossier mit ab.

- 1) Alle Gerichte mit allen zugehörigen Allergenen.
- 2) Ändern Sie die vorherige Abfrage so ab, dass alle existierenden Gerichte dargestellt werden (auch wenn keine Allergene enthalten sind).
- 3) Ändern Sie die vorherige Abfrage so ab, so dass im Ergebnis alle existierenden Allergene dargestellt werden (auch wenn diese nicht einem Gericht zugeordnet sind).
- 4) Die Anzahl der Gerichte pro Kategorie aufsteigend sortiert nach Anzahl.
- 5) Ändern Sie die vorherige Abfrage so ab, dass dabei nur die Kategorien dargestellt werden, die mehr als 2 Gerichte besitzen.
- 6) (optional) Alle Gerichte, die vier oder mehr Allergene aufweisen.

Aufgabe 7

Nebenbedingungen und referentielle Integrität auf der Datenbank. Fügen Sie die noch fehlenden Überprüfungen von Nebenbedingungen (wie UNIQUE) sowie von Fremdschlüsselbeziehungen (Schlüsselwort REFERENCES) aus den Strukturvorgaben der Datenbank für die Werbeseite ein. Notieren Sie Ihre verwendeten Statements im Dossier.

Aufgabe 8 (optional)

Übung. Erstellen Sie zur Übersicht der angelegten Datenstruktur für die Werbeseite ein ERD nach Chen.

Aufgabe 9

Werbeseite: Statistiken aus der Datenbank. Auf der Werbeseite stehen derzeit noch Platzhalter für die Anzahl der Besucher:innen, die Anmeldungen für den Newsletter sowie die Anzahl der Gerichte. Die Werbeseite soll diese Zahlen nun korrekt aus dem Datenbestand (Newsletteranmeldungen aus Datei, Anzahl Gerichte und Anzahl Besuche aus Datenbank) darstellen.

- 1) Stellen Sie die Anzahl der gespeicherten Gerichte unter „Zahlen“ dar.
- 2) Stellen Sie die Anzahl der gespeicherten Anmeldungen zum Newsletter dar.
- 3) Entwickeln Sie einen persistenten Zähler für die Besucher:innen und stellen Sie die Anzahl unter „Zahlen“ dar. Nutzen Sie die Datenbank.

- 4) (Optional) Zählen Sie die Besucher:innen abhängig von der IP. Eine IP soll pro Tag nur einmal gezählt werden.

Aufgabe 10

Peer-Review: Quelltext vorbereiten. Im Rahmen des nächsten Praktikums geben Sie Ihre Inhalte unter dem Ordner **werbeseite** an ein anderes Team im Praktikum, das Ihre Ergebnisse sieht und Ihnen sinnvolle und wertvolle Hinweise und Tipps für mögliche Verbesserungen gibt.

Räumen Sie für das Review Ihren Quelltext auf, indem Sie unter anderem:

- Nicht verwendeten (z.B. auskommentierten) Quelltext entfernen.
- Kommentare einfügen, da wo es Ihnen sinnvoll erscheint.
- Namen von Funktionen und Variablen auf Sinnhaftigkeit und Ausdruckstärke hin überprüfen.
- (Optional). Wenn Sie am Peer-Review teilnehmen möchten, dann melden Sie sich an unter:

ILIAS-Modul > Praktikum > Anmeldung Peer-Review

Teams, die am Review teilnehmen, können eine Aufgabe in Meilenstein 6 einsparen.

Hinweis: Das Review-Team steht noch nicht fest und wird zufällig ermittelt. Sie erhalten den Quelltext des jeweils anderen Teams.

Aufgabe 11

Datenbankdesign. Wir planen eine zukünftige Erweiterung der Werbeseite, die wir praktisch jedoch nicht direkt in SQL umsetzen, sondern zunächst nur konzipieren.

- 1) Aus dem Vertrieb kommt die Frage, warum das Speichern in einfachen Dateien (ohne Einsatz von Datenbanken) direkt aus der Anwendung heraus Probleme bereiten kann. Erklären Sie anhand von mehreren Argumenten.
- 2) Sie erhalten die folgende Beschreibung von der E-Mensa GmbH, wozu Sie das ERD nach Chen entwerfen.

Über die Werbeseite sollen Rückfragen von Kontakten (Interessenten, Kunden, Lieferanten, ...) gestellt werden können. Eine Rückfrage enthält einen Text (die Anfrage), ein Thema, einen Erfassungszeitpunkt und optional einer Dringlichkeit. Zu jeder Rückfrage wird sofort eine automatische eindeutige Rückfragenummer generiert. Das Alter einer Anfrage wird berechnet aus der Differenz zwischen dem aktuellen Zeitpunkt und dem Erstellungszeitpunkt. Als Thema sind Beschwerde, Frage oder Wunsch möglich. Ein Kontakt kann viele Rückfragen stellen, wobei eine Rückfrage nur von genau einem Kontakt gestellt werden kann. Ein Kontakt hat eine E-Mail, einen Vor- und Nachnamen, eine Anrede (Frau, Herr) sowie optional eine Telefonnummer und eine eindeutige Kontaktid. Ein Kontakt besitzt genau eine Anrede. Eine Anrede wird von vielen Kontakten verwendet. Ein Kontakt

kann ein Kunde (Besucher:innen der Mensa) oder ein Zulieferer (Lieferant von Zutaten). Ein Kontakt kann gleichzeitig Kunde und Zulieferer sein. Ein Zulieferer hat Lieferungszeitfenster (morgens, mittags, abends), die pro Wochentag unterschiedlich sein können. Ein Kontakt hat eine Rechnungsadresse und mehrere Hausanschriften, die alle Straße, Hausnummer, Postleitzahl und Ort enthalten. Ein Kontakt kann mehrere Hausanschriften besitzen, die sich in der Art (Hauptanschrift und Zweitwohnsitz) unterscheiden. Ein Kontakt kann nur eine Rechnungsadresse besitzen. Eine Rechnungsadresse ist genau einem Kontakt zugeordnet.

- Entwerfen Sie das ERD nach Chen in elektronischer Form, z.B. mit dem Online-Werkzeug draw.io⁵. Speichern Sie den Entwurf ab unter **/beispiele/m3_11a_rueckfragen.{png, jpeg, ...}**.
- Entwickeln Sie ein passendes Relationenmodell in Relationenschreibweise.
- Markieren Sie alle Primär- (durch Unterstreichung) und Fremdschlüssel (durch gestrichelte Unterstreichung).

Aufgabe

Abgabe. Laden Sie Ihre Ergebnisse als ZIP in unter

<https://git.fh-aachen.de/praktika/dbwt/ws24>

in den Ordner Ihres Teams in den Branch „main“ hoch. Der Name des ZIPs soll sein:

M3.zip

Das Zip umfasst mindestens die Verzeichnisse und Dateien:

- /beispiele
- /werbeseite
- dossier.{txt, pdf, html, ...}
- README

Das ZIP soll alle Ergebnisse des Meilensteins (Antworten zu Freitextaufgaben, Grafiken, Quelltexte, ...) enthalten. Fügen Sie zusätzlich ein README ein, in der Sie Vor-, Nachname und Matrikelnummer der beteiligten Teammitglieder hinterlegen. Eine Abgabe einer Person für das Team reicht aus. Die letzte Abgabe vor dem Abgabezeitpunkt (Freitag, 22:00 Uhr) zählt.

⁵ Draw.io unter <http://draw.io> oder <https://app.diagrams.net> (Letzter Zugriff 24.07.2022)