

## **Assignment**

# **Realisierung einer Datenbank mit zugehörigem Anwenderprogramm für Online-Umfragen**

Datum: 12.11.2020

Modulbezeichnung: DBA20

Thema: Thema 3 – Online Umfrage

Name: Jannis Willwater

Immatrikulationsnummer: 8092313

Adresse: Karl-Marx-Str. 21, 38104 Braunschweig

E-Mail: jannis.willwater@gmx.de

## **Eidesstattliche Erklärung**

Ich versichere, dass ich das beiliegende Assignment selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlich oder sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet habe.

Jannis Willwater

# Inhaltsverzeichnis

1. Einleitung	1
2. Grundlagen	1
2.1. Online-Umfragen	1
2.2. Datenbanken und MySQL	2
2.3. PHP	3
3. Ausarbeitung der Datenbank mit zugehörigem Anwenderprogramm	3
3.1. Konzeptuelles Datenmodell	4
3.2. Logisches Datenmodell	5
3.2.1. Relationsmodell	5
3.2.2. Realisierung in MySQL	7
3.3. Realisierung des Anwenderprogramms mit PHP	8
4. Zusammenfassung und Ausblick	12
Literaturverzeichnis	13

# 1. Einleitung

In diesem Assignment soll eine Online-Umfrage unter dualen Studenten entworfen werden. Wir beginnen mit den Grundlagen, welche die immer weiter steigenden Relevanz von Online-Umfragen beleuchten wird [1]. Zusätzlich werden das hier verwendete Datenbankverwaltungssystem MySQL und PHP, welches für das Anwenderprogramm benötigt wird, vorgestellt. Nach den Grundlagen wird mit einem konzeptuellen Datenmodell gestartet, um die Realität möglichst logisch und vollständig in ein Datenmodell zu übertragen. Dabei wird hier das E/R-Modell verwendet. Dieses lässt sich in einem zweiten Schritt einfach in ein logisches Datenmodell übertragen und zu guter Letzt wird dann eine Datenbank in MySQL erstellt.

Zusätzlich zu der Datenbank soll auch noch ein Anwenderprogram erstellt werden, welches ein Eingeben der Fragen, Durchführen der Umfrage und Auslesen der Ergebnisse ermöglicht. Im letzten Abschnitt wird also das Programm vorgestellt und auf die technische Umsetzung eingegangen.

## 2. Grundlagen

In dem ersten Abschnitt der Grundlagen soll die Bedeutung von Online-Umfragen vorgestellt werden und auf die entscheidenden Vorteile eingegangen werden. Die nachfolgenden Abschnitte beschäftigen sich dann mit dem Datenbankverwaltungssystem MySQL und der Skriptsprache PHP, welche für die Realisierung der Datenbank mit zugehöriger Eingabemaske verwendet wurden.

### 2.1. Online-Umfragen

Online-Umfragen sind mit insgesamt 9,2 Mio.<sup>1</sup> Befragungen im Jahre 2019 die am meistgenutzte Befragungsart noch vor Face-to-Face- (4,6 Mio.) und Telefon-Interviews (4,2 Mio.) [1]. Die vielfältigen Verwendungen von Online-Umfragen in der kommerziellen

---

<sup>1</sup> Die Zahlen beziehen sich dabei auf alle in Deutschland ausgeführten Umfragen von Mitgliedern des Arbeitskreis Deutscher Markt- und Sozialforschungsinstitute e.V. (ADM) [1].

Marktforschung, Evaluationen oder der Sozialforschung ist ein wichtiger Faktor bei der schnellen Verbreitung [2]. Zusätzlich liefert die schnelle und automatisierbare Auswertung neue Optionen, welche besonders im Zuge von Big-Data und Machine-Learning neue Möglichkeiten eröffnet. Weitere wichtige Vorteile wie die Kostenfrage und zeitlich/örtliche Unabhängigkeit ermöglicht eine vielseitige Einsetzung [2]. Es können unterschiedliche Fragetypen verwendet werden, um Online-Umfragen weiter zu personalisieren. In dem in dieser Arbeit zu erstellenden Umfragemodul, werden Multiple-Choice- und Eingabefragen verwendet. Es muss also entweder zwischen schon definierten Antworten ausgewählt werden (Multiple-Choice-Frage) oder die Antwort kann frei eingetragen werden (Eingabefrage).

Wichtig ist auch noch, dass rechtliche Rahmenbedingungen beachtet werden, welche für Deutschland im BDSG (Bundesdatenschutzgesetz) aufgeführt werden [3]. So muss bei einer anonymen Online-Umfrage sichergestellt werden, dass keine Daten gespeichert werden, welche einen Rückschluss auf die Person zulassen [3].

Aufgrund der oben genannten Vorteile soll in dieser Arbeit eine Online-Umfrage unter dualen Studenten erstellt werden. Dafür wird eine Eingabemaske zur Verfügung gestellt, damit der Nutzer selbständig Fragen einreichen kann und so die Umfrage nach seinem/ihrer Wunsch gestalten kann.

## **2.2. Datenbanken und MySQL**

In der heutigen Zeit entstehen jede Sekunde Massen von Daten. Um all diese Daten strukturiert zu sammeln und auszuwerten, muss eine geeignete Infrastruktur bereitstehen. In Datenbanken werden zusammengehörende Daten gesammelt. Wir beschränken uns hier auf relationale Datenbanken, welche ausschließlich auf Tabellen (Relationen) und deren Beziehungen zueinander bestehen [4]. Für die Verwaltung ist dann das sogenannten Database Management System (DBMS) zuständig. Der Nutzer kann dann über eine logische Schnittstelle direkt auf das Database Management System zugreifen oder es ist noch zusätzlich ein Anwendungsprogramm zwischengeschaltet [5].

Die wichtigste Schnittstelle ist dabei SQL (Structured Query Language), welche ab 1970 bei IBM entwickelt wurde [5]. Der große Vorteil von SQL ist dabei, dass es sich um eine

standardisierte und deklarative Programmiersprache handelt [6]. Es gibt also festgelegte Standards für SQL (aktuell: SQL3), welche in allen SQL-Datenbanken Anwendung finden. In eine deklarative Programmiersprache wird nur eine WAS-Anfrage benötigt und das WIE muss nicht genauer definiert werden [6]. Soll zum Beispiel eine Tabelle in SQL kreiert werden, dann geht das mit dem Befehl CREATE TABLE. Die Datenbank legt dann die Tabelle automatisch an.

In dieser Arbeit wurde die Open-Source-Datenbanken MySQL verwendet.

## **2.3. PHP**

Wie schon erwähnt, kommunizieren die meisten Nutzer nicht direkt mit einer Datenbank, sondern mit einem zwischengeschalteten Anwenderprogramm [5]. Hier wurde dieses Anwenderprogramm als Website mit PHP programmiert. PHP wurde 1995 entwickelt und ist eine serverseitige Skriptsprache [7]. PHP-Programme können also nur mit einem Webserver ausgeführt werden. Dabei wird HTML-Code für die Darstellung verwendet und, in dem hier vorgestellten Projekt, sorgt das PHP-Programm dann dafür, dass die Datenbank ausgelesen wird bzw. neue Daten in die Datenbank eingetragen werden.

Um die Programme kostenlos zu testen bieten sich das Programmpaket XAMPP an. Es beinhaltet Apache-Webserver mit PHP und MySQL, sowie einige weitere Zusatzprogramme [8]. Damit haben wir die wichtigsten Grundlagen zur Erstellung einer Online-Umfrage mit MySQL und PHP behandelt und können jetzt mit der eigentlichen Aufgabe beginnen.

## **3. Ausarbeitung der Datenbank mit zugehörigem Anwenderprogramm**

In diesem Kapitel soll es jetzt um die eigentliche Realisierung der Online-Umfrage gehen. Dabei widmen sich die ersten beiden Abschnitte der Datenbank und der letzte Abschnitt der Programmierung des Anwenderprogramms mit PHP.

### 3.1. Konzeptuelles Datenmodell

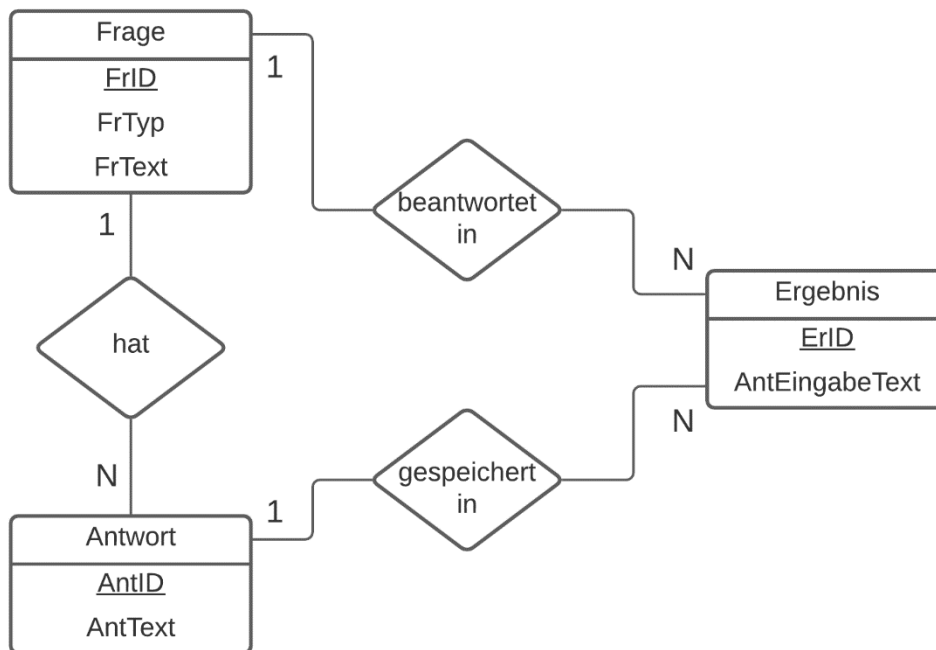


Abbildung 1: E/R-Modell für die Umfragen-Datenbank. Es werden insgesamt drei Entitäten benötigt, welche jeweils durch Beziehungen verknüpft werden.

Um eine Übersicht der Realität zu erhalten, bietet es sich immer an mit einer grafischen Darstellung zu beginnen. So können früh mögliche Fehler erkannt und vermieden werden. Dabei wird hier die Entity-Relationship Methode (oder auch E/R-Modell) verwendet. Das E/R-Modell für die Online-Umfrage-Datenbank ist in Abbildung 1 zu sehen. Die Objekte aus der realen Welt werden mit Entitäten und zugehörigen Attributen beschrieben [9] und befinden sich in Abb. 1 in den Kästen. Für die Umfrage-Datenbank werden die drei Entitäten Frage (mit den Attributen FrID, FrTyp und FrText), Antwort (AntID, AntText) und Ergebnis (ErID, AntEingabeText) benötigt. Gleichartige Entitäten werden zu Entitätstypen zusammengefasst [9]. Um die Entitäten eindeutig zu unterscheiden gibt es den Primärschlüssel, welches das unterstrichende Attribut in Abb. 1 ist. Jede Frage ist also zum Beispiel eindeutig durch die FrID zu identifizieren. Das Attribut FrTyp soll die Art der Frage (Multiple-Choice- oder Eingabefrag) unterscheiden und in dem Attribut FrText wird dann die eigentliche Frage gespeichert. Sollte es sich um eine Multiple-Choice-Frage handeln, dann müssen Antworten vorgegeben werden. Diese sind in der Entität Antwort enthalten, welche die Attribute AntID (Primärschlüssel) und

den eigentlichen Text der Antwort: `AntText` enthält. Zu guter Letzt speichert die Entität `Ergebnis` die von dem User ausgewählte Antwort. Dabei gibt es neben dem Primärschlüssel `ErID` noch das Attribut `AntEingabeText`, welches Antworten von Eingabefragen speichert.

Ein weiterer wichtiger Bestandteil im E/R-Modell sind die durch eine Raute dargestellten Beziehungen. So steht die Entität `Fragen` über die Beziehung `hat` mit der Entität `Antwort` zusammen. Die beiden weiteren Beziehungen sind, dass die Frage beantwortet wird in `Ergebnis` und die Antwort ist gespeichert in `Ergebnis`.

Abschließend müssen wir uns noch mit dem Komplexitätsgrad (oder auch Kardinalität) beschäftigen, welcher die Mengenmäßigkeit einer Beziehung angibt [9]. So ist die Beziehung zwischen `Frage` und `Antwort` eine 1:N-Beziehung, da jede Frage mehr als eine Antwort haben kann. Wichtig ist hier noch, dass N gleich Null sein kann, da es möglich ist, dass eine Frage keine Antwort vorgibt (Eingabefrage). Eine ähnliche Beziehung besteht auch zwischen `Antwort` und `Ergebnis`. Die Beziehung `beantwortet in` ist ebenfalls eine 1:N-Beziehung, hier kann N aber nicht Null sein, da jede Frage beantwortet werden muss und es damit auch zu jeder Frage ein Eintrag in `Ergebnis` geben muss.

Mit dem vollständigen konzeptuellen Datenmodell kann diese jetzt in ein logisches Datenmodell übertragen werden und die Datenbank angelegt werden.

## **3.2. Logisches Datenmodell**

Wie in Abschnitt 2.2. schon erläutert, soll eine relationale Datenbank erstellt werden. Dazu wird im ersten Abschnitt das Relationsmodell betrachtet und im zweiten Abschnitt die Realisierung mit MySQL.

### **3.2.1. Relationsmodell**

Aus dem E/R-Modell kann nach den Regeln aus [10] direkt ein Relationsmodell erstellt werden.



Als erster Schritt werden alle Entitätstypen zu Relationen und die Attribute der Entitätstypen werden zu Attributen der Relationen. Es ergibt sich also:

**Fragen** (FrID: Numerisch, FrTyp: Numerisch, FrText: String)

**Antwort** (AntID: Numerisch, AntText: String)

**Ergebnis** (ErgID: Numerisch, AntEingabeText: String)

Zusätzlich zu den Relationen und Attributen wurde hier auch schon der allgemeine Datentyp der Attribute angegeben.

Im nächsten Schritt werden auch die Beziehungen zu Relationen:

**hat** (FrID: Numerisch, AntID: Numerisch)

**beantwortet in** (FrID: Numerisch, ErID: Numerisch)

**gespeichert in** (AntID: Numerisch, ErID: Numerisch)

Der dritte und letzte Schritt ist jetzt die Vereinfachung der 1:N- und 1:1-Beziehungen. Da es in der Umfrage-Datenbank, wie schon in Abschnitt 3.1. gesehen und begründet, nur 1:N-Beziehungen gibt, können alle Beziehungs-Relationen den jeweiligen N-Relationen zugeordnet werden. Damit bleiben also am Ende genau drei Tabellen in der Datenbank.

Im nächsten Schritt soll die Datenbank in MySQL angelegt werden. Bevor die eigentlichen Befehle besprochen werden, muss noch ein Blick auf die Datentypen geworfen werden. In Tabelle 1 sind die Relationen nochmal aufgeführt mit den MySQL-Datentypen und zusätzlichen Bedingungen. Für die Primär- und Fremdschlüssel wurde INT als Datentyp gewählt, um sicher zu stellen, dass genug Fragen, Antworten und Ergebnisse in die Datenbank eingetragen werden können. Für FrTyp wurde als Datentyp TINYINT gewählt. Der TINYINT-Datentyp hat eine Reichweite von -128 bis 127 [11] und hat damit ausreichend Platz um unterschiedliche Fragentypen zu definieren. Die Fragen- und Antworttexte werden als TEXT gespeichert. Eine weitere wichtige Bedingung ist, dass die Attribute FrTyp, FrText und AntText nicht leer (Not Null) sein dürfen, da es sonst die Zuordnung des Fragentyps unmöglich macht oder die Frage leer ist.

*Tabelle 1: Darstellungen aller Relationen des Relationsmodells mit zugehörigen Datentypen und dem MySQL-Code zur Initialisierung.*

Relation	Attribut	Datentyp	Bedingung	MySQL-Code
Frage	FrID	INT	Primärschlüssel	CREATE TABLE Frage ( FrID INT NOT NULL AUTO_INCREMENT, FrTyp TINYINT NOT NULL, FrText TEXT NOT NULL, PRIMARY KEY (FrID));
	FrTyp	TINYINT	Not Null	
	FrText	TEXT	Not Null	
Antwort	AntID	INT	Primärschlüsse	CREATE TABLE Antwort ( AntID INT NOT NULL AUTO_INCREMENT, FrID INT NOT NULL, AntText TEXT NOT NULL, PRIMARY KEY (AntID));
	FrID	INT	Fremdschlüssel	
	AntText	TEXT	Not Null	
Ergebnis	ErgID	INT	Primärschlüssel	CREATE TABLE Ergebnis ( ErgID INT NOT NULL AUTO_INCREMENT, FrID INT NOT NULL, AntID INT NULL, AntEingabeText TEXT NULL, PRIMARY KEY (ErgID));
	FrID	INT	Fremdschlüssel	
	AntID	INT	Fremdschlüssel	
	AntEingabeText	TEXT		

### 3.2.2. Realisierung in MySQL

Jetzt soll das Relationsmodell in MySQL übertragen werden. Dafür muss zu erste die Datenbank mit dem Befehl „CREATE DATABASE Umfrage;“ erstellt werden. Jetzt können die Tabellen nach den Befehlen in Tabelle 1 angelegt werden. Zusätzlich zu den schon erwähnten Bedingungen, erhalten die Primärschlüssel noch den Zusatz „AUTO\_INCREMENT“. Die Primärschlüssel werden also automatisch hochgezählt und müssen nicht händisch vorgegeben werden.

Als letztes müssen noch die Fremdschlüssel definiert werden. Falls zum Beispiel der Primärschlüssel einer Frage geändert werden sollte, dann muss auch das Attribut FrID

in den Relationen Antwort und Ergebnis geändert werden. Es werden also noch die folgenden Befehle ausgeführt, um die Fremdschlüssel einzurichten:

```
ALTER TABLE Antwort ADD FOREIGN KEY (FrID) REFERENCES Frage(FrID) ON  
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Ergebnis ADD FOREIGN KEY (FrID) REFERENCES Frage(FrID) ON  
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Ergebnis ADD FOREIGN KEY (AntID) REFERENCES  
Antwort(AntID) ON DELETE CASCADE ON UPDATE CASCADE;
```

Der Befehl „CASCADE“ sorgt dafür, dass wenn der Primärschlüssel geändert oder gelöscht wird, automatisch auch der Fremdschlüssel geändert oder gelöscht wird. Zusätzlich wurden durch die Fremdschlüssel auch gleich wichtige Indexe erstellt, um zum Beispiel ein schnelles Filtern der Ergebnisse nach Fragen zu ermöglichen.

Die hier aufgeführten Befehle können entweder direkt der Datenbank übergeben werden oder über eine Eingabemaske, welche mit PHP programmiert wurde. Darum soll es im nächsten Abschnitt gehen.

### **3.3. Realisierung des Anwenderprogramms mit PHP**

Das Anwenderprogramm soll das Eintragen der Fragen und Antworten und das Abfragen der Ergebnisse ermöglichen. Um die Datenbank zu initialisieren, kann das Programm „init.php“ verwendet werden. Nach Betätigen des Buttons „Umfrage erstellen“ wird die Datenbank erzeugt.

Für die eigentliche Umfrage werden drei weitere PHP-Dateien<sup>2</sup> „Umfrage\_einrichten.php“, „umfrage.php“ und „ergebnisse.php“ erstellt. In Abbildung 2 sind die kompilierten Webseiten gezeigt. Im Folgenden sollen die Programme durchgegangen werden und an einige technische Fragen eingegangen werden.

---

<sup>2</sup> Zusätzlich wurde noch eine Datei „abgeschlossen.php“ erstellt. Da diese nur eine Zeile mit „Vielen Dank für Ihre Teilnahme!“ nach der Umfrage zeigt und keinen relevanten Code enthält, wird auf eine genauere Betrachtung verzichtet.

Als erstes betrachten wir das Programm „Umfrage\_einrichten.php“ (s. Abb. 2 (a)). Dieses Programm ist nur für den Umfragen-Administrator, um Fragen einzurichten. Zuerst muss der Fragentyp ausgewählt werden. Dafür stehen die Wahlmöglichkeiten Eingabefeld und Multiple-Choice in einer Dropdown-Liste zur Verfügung (s. Zeilen 26-28 in „Umfrage\_einrichten.php“). Wie schon im vorherigen Abschnitt gesehen, wird der Fragentyp als Attribut FrTyp mit dem Datentyp TINYINT in der Relation Frage gespeichert. Eine „1“ in dem Attribut FrTyp steht dabei für die Auswahl Eingabefeld und eine „2“ für Multiple-Choice. Als nächstes kann die Frage in ein Eingabefeld eingetragen werden (s. Zeile 31 in „Umfrage\_einrichten.php“). Das wird dann in der Datenbank als FrText gespeichert. Sollte es sich um eine Multiple-Choice-Frage handeln, müssen zusätzlich noch Antworten eingetragen werden (s. Zeilen 33-44 in „Umfrage\_einrichten.php“). Dafür stehen zehn Eingabefelder für Antworten zur Verfügung. Diese werden nach Einreichen der Frage in das Attribut AntText der Relation Antwort eingetragen. Sollte eine Eingabefrage ausgewählt worden sein, müssen keine Antworten eingetragen werden. Durch drücken des Buttons „Einreichen“ (s. Zeile 45 in „Umfrage\_einrichten.php“) wird die Frage und mögliche Antworten in die Datenbank gespeichert. Der My-SQL-Befehl wird dabei erstmal in einem String \$sqlFr aufgebaut:

```
$sqlFr = "INSERT INTO `frage` (`FrID`, `FrTyp`, `FrText`) VALUES ('" . $frId_next . "',  
'1', '" . $_POST['Frage'] . "')";
```

Die Variable \$frId\_next steht für die FrID und \$\_POST['Frage'] für FrText in der Relation Frage. Mit „MySQLi“ kann dann auf die Datenbank zugegriffen werden und die gewünschten Informationen eingespeichert werden. Zusätzlich zu dem „Einrichten“-Button gibt es noch einen Button um zu der Umfrage zu kommen und zu den Ergebnissen.

Die eigentliche Umfrage wurde dann in „umfrage.php“ programmiert. Die Website mit einigen beispielhaften Fragen und Antworten ist in Abbildung 2 (b) zu sehen. Bevor es um die Gestaltung der Website für die Umfrage geht, müssen die Fragen und Antworten aus der Datenbank gelesen werden. Dafür werden die Relationen Frage und Antwort mit dem „SELECT“-Befehl ausgewählt und jeweils in einem Array gespeichert (s. Zeilen 3-28 in „umfrage.php“). Um die Fragen und Antworten dann dazustellen werden die Arrays in for-Schleifen durchgegangen und die jeweiligen Frage- und Antworttexte dargestellt.

Dabei wird bei jeder Frage erst geschaut, um welchen Typ es sich handelt (s. Zeile 48 in „umfrage.php“). Entweder wird also ein Eingabefeld oder mehrere Auswahlmöglichkeiten gezeigt. Bei den Multiple-Choice-Fragen muss genau eine Antwort ausgewählt werden. Sollten auch Fragen mit mehreren Antworten gewünscht sein, dann muss noch ein zusätzlicher Fragentyp, definiert werden. Durch das Drücken des Buttons „Absenden“ werden die ausgewählten Antworten in die Relation *Ergebnis* in der Datenbank gespeichert (s. Zeilen 85-102 in „umfrage.php“).

Als letztes wird die Website zum Anzeigen der Ergebnisse betrachtet, welche in Abbildung 2 (c) zu sehen ist. Analog zu dem vorherigen Programm wird damit begonnen, dass die Fragen und Antworten aus der Datenbank abgefragt werden (s. Zeilen 3-28 in „ergebnisse.php“). Dann werden die Fragen und Antworten wieder durchgegangen. Um die Ergebnisse der Eingabefragen zu erhalten, werden alle Ergebnisse aus der Datenbank rausgesucht die die passende FrID haben und dargestellt (s. Zeilen 48-58 in „ergebnisse.php“). Für die unterschiedlichen Antworten der Multiple-Choice-Fragen wird in der Ergebnis-Relation nach allen Ergebnissen mit der passenden FrID und AntID gesucht und mit dem Befehl „count(“) gezählt. So können alle Ergebnisse mit der entsprechenden Antwort gezählt werden.

Damit ist das Anwenderprogramm fertig gestellt und kann zum Erstellen und Auswerten von Umfragen unter dualen Studenten verwendet werden.

(a)

Bitte geben Sie im Folgenden die neue Frage ein:

Fragentyp:   
Frage:

Antworten:

Wenn dies eine Frage mit Eingabefeld sein soll, dann müssen keine Antworten vorgegeben werden!

1:

2:

3:

4:

5:

6:

7:

8:

9:

10:

(b)

Bitte beantworten Sie die folgenden Fragen!

Frage 1: Was ist Ihr Geschlecht?

- ☐ M  
☐ W  
☐ D

Frage 2: In welcher Altersgruppe sind Sie?

- ☐ 16 - 25  
☐ 25 - 35  
☐ Alter als 35

Frage 3: Wie gefällt Ihnen das duale Studium?

- ☐ Sehr gut  
☐ Gut  
☐ Mittel  
☐ Schlecht  
☐ Sehr schlecht

Frage 4: Was gefällt Ihnen gut?

Frage 5: Was stört Sie?

Frage 6: Was können wir noch verbessern?

(c)

Das sind die Ergebnisse der aktuellen Umfrage:

Frage 1: Was ist Ihr Geschlecht?

M: 3  
W: 0  
D: 0

Frage 2: In welcher Altersgruppe sind Sie?

16 - 25: 3  
25 - 35: 0  
Alter als 35: 0

Frage 3: Wie gefällt Ihnen das duale Studium?

Sehr gut: 3  
Gut: 0  
Mittel: 0  
Schlecht: 0  
Sehr schlecht: 0

Frage 4: Was gefällt Ihnen gut?

Alles, Sehr viel, Onlineportal

Frage 5: Was stört Sie?

Nichts, Aufgaben, Aufgaben

Frage 6: Was können wir noch verbessern?

Aktualität, Informationen

Abbildung 2: Darstellung des Anwenderprogramms beruhend auf (a) „Umfrage\_einrichten.php“, (b) „umfrage.php“ und (c) „ergebnisse.php“

## 4. Zusammenfassung und Ausblick

Damit wurde eine Online-Umfrage mithilfe von MySQL und PHP erfolgreich aufgebaut. Durch die konzeptuelle Modellierung mit Hilfe eines E/R-Modells konnten die wichtigen Entitäten und Beziehungen ausgearbeitet werden. Das wurde dann in ein Relationsmodell übertragen. Es wurde sich dazu entschieden mit drei Relationen zu arbeiten. In der Frage-Relation werden die Fragen gespeichert, in der Antwort-Relation die möglichen Antworten bei Multiple-Choice-Fragen und in der Ergebnis-Relation werden alle Antworten von ausgeführten Umfragen gespeichert. Dieses Modell wurde dann anschließend mittel MySQL in eine Datenbank umgewandelt. Zu guter Letzt wurde ein Anwendungsprogramm mit PHP realisiert. Dabei gibt es eine Maske zur Eingabe von Fragen, die eigentliche Umfrage-Website und noch zusätzlich eine Seite, um sich die Ergebnisse anzeigen zu lassen.

Es wurde hier ein relativ einfaches Modell zum Erstellen und Durchführen von Online-Umfragen entwickelt. Durch zusätzliche Komponenten könnte das Umfragen-Modul noch erweitert werden. Zum Beispiel kann noch die IP-Adresse überprüft werden, um sicherzustellen, dass niemand die Umfrage zweimal durchführt. Dabei müssten die Adressen in einer eigenständigen Tabelle gespeichert werden, damit die Ergebnisse nicht mit der IP-Adresse in Verbindung gebracht werden kann. Auch eine Implementierung von weiteren Fragetypen ist denkbar.

Ein Modul um Online-Umfragen unter dualen Studenten zu ermöglichen, wurde also vollständig mit MySQL und PHP entworfen.

## Literaturverzeichnis

- [1] vgl. Arbeitskreis Deutscher Markt- und Sozialforschungsinstitute e.V. (Hrsg.): Jahresbericht 2019, Berlin, 2020
- [2] vgl. Welker, Martin; Matzat, Uwe (2009): Online-Forschung: Gegenstände, Entwicklung, Institutionalisierung und Ausdifferenzierung eines neuen Forschungszweiges, in: Zerback, Thomas; Schoen, Harald; Jakob, Nikolaus (Hrsg.): Sozialforschung im Internet – Methodologie und Praxis der Online-Befragung, 1. Auflage, Wiesbaden, S. 13-39.
- [3] vgl. Faulkbaum, Frank (2019): Methodische Grundlagen der Umfrageforschung, 1. Auflage, Duisburg, S. 70-82.
- [4] vgl. Schicker, Edwin (2017): Datenbanken und SQL - Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL, 5. aktualisierte und erweiterte Auflage, Regensburg 2017, S. 12.
- [5] vgl. Schicker, Edwin (2017): Datenbanken und SQL - Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL, 5. aktualisierte und erweiterte Auflage, Regensburg, S. 2-6.
- [6] vgl. Blaschka, Markus (o.J.): SQL – Structured Query Language, AKAD-Studienbrief DBA103, o.O, S. 7f.
- [7] vgl. Bühler, Peter; Schlaich, Patrick; Sinner, Dominik (2018): Webtechnologien – JavaScript – PHP – Datenbank, 1. Auflage, S. 54
- [8] vgl. Stöckl, Andreas; Bongers, Frank (2006): Einstieg in TYPO3 4.0: Installation, Grundlagen, TypoScript; [inkl. Einführung in TypoScript; mit TypoScript-Referenz], 2. aktualisierte und erweiterte Ausgabe, Bonn, S. 467.
- [9] vgl. Thimm, Heiko; Blaschka, Markus (o.J.): Datenbankentwurf, AKAD-Studienbrief DBA102, o.O, S. 11-14.
- [10] vgl. Thimm, Heiko; Blaschka, Markus (o.J.): Datenbankentwurf, AKAD-Studienbrief DBA102, o.O, S. 36.



- [11] vgl. Oracle Corporation (Hrsg.): 11.1.2 Integer Types (Exact Value) - INTEGER, INT, SMALLINT, TINYINT, MEDIUMINT, BIGINT, <https://dev.mysql.com/doc/refman/8.0/en/integer-types.html> (Datum des Zugriffs: 08.11.2020)