

Stefan Waidele
Ensisheimer Straße 2
79395 Neuenburg am Rhein
Stefan.Waidele@AKAD.de

AKAD Hochschule Stuttgart
Immatrikulationsnummer: 102 81 71

Modul DBA02 — Praktisches Arbeiten mit Datenbanken
Assignment

DATENBANKGESTÜTZTE PHP–ANWENDUNG FÜR EINE UMFRAGE-WEBSITE

Betreuer: Prof. Dr. Franz–Karl Schmatzer

31. Oktober 2013



AKAD Hochschule Stuttgart

Inhaltsverzeichnis

Abbildungsverzeichnis	ii
1 Einleitung	1
1.1 Aufgabenstellung	1
1.2 Gemeinschaftsarbeit	2
1.3 Aufbau der Arbeit	2
1.4 Abgrenzung	3
2 Grundlagen	4
2.1 Entwurfsmuster: Fassade	4
2.2 Entwurfsmuster: Singleton	4
2.3 PHP-Schnittstelle: Session-Cookies	5
2.4 PHP-Schnittstelle: PDO	5
2.5 HTML-Designframework: Bootstrap	6
2.6 JavaScript-Frontendframework: jQuery	6
3 Datenbank-Schema	7
3.1 Konzeptuelles Datenbankschema: Entity-Relationship Diagram . .	7
3.2 Logisches Datenbankschema: Relationales Datenmodell	8
3.3 Umsetzung in SQL	9
3.3.1 Tabelle: user — Benutzerverwaltung	9
3.4 Tabelle: frage	9
3.5 Tabelle: antwort	9
3.6 Tabelle: geantwortet	10
3.7 Query: Frage – Text	10
3.8 Query: Antwortmöglichkeiten	10
3.9 Query: Anzahl der gegebenen Antworten	11
3.10 Query: Gegebene Antworten und Häufigkeit	11
3.11 Tabelle	11
3.12 Bilder	12
3.13 Syntax Highlighting	12
4 Bewertung	13
4.1 Zusammenfassung	13
4.2 kritische Würdigung	13
4.3 Ausblick	13
4.4 Erfolgsfaktoren	13
Literatur	iii

Abbildungsverzeichnis

1	Entity Relationship Diagram	7
2	Relationales Modell	8
3	SQL: CREATE TABLE user	9
4	SQL: CREATE TABLE frage	10
5	SQL: CREATE TABLE antwort	10
6	SQL: CREATE TABLE geantwortet	10
7	SQL: Text von Frage <n>	11
8	SQL: Mögliche Antworten für Frage <n>	11
9	SQL: Antwortzahl 100% von Frage <n>	11
10	SQL: Gegebene Antworten mit Häufigkeit für Frage <n>	12
11	Akad	12
12	Quellcode: Aufruf von header.php (PHP)	12

1 Einleitung

1.1 Aufgabenstellung

Im Rahmen des Moduls DBA02 war eine datenbankgestützte PHP–Anwendung für eine Website zu erstellen, welche die folgenden Kriterien erfüllt:

- **Frage stellen:** Besuchern der Website soll eine Frage gestellt werden, auf die sie mit einer oder mehreren vorgegebenen Möglichkeiten antworten können.
- **Auswertung:** Nach der Beantwortung der Frage soll dem Besucher eine Auswertung der bisher gegebenen Antworten (Angaben in Prozent) gezeigt werden.
- **Benutzerverwaltung:** Ein Administrator soll sich bei der Anwendung anmelden können. Hierzu soll ein Benutzername und Passwort abgefragt und geprüft werden.
- **Neue Fragen eingeben:** Dem Seitenadministrator soll es über ein Formular möglich sein, neue Fragen mit den zugehörigen Antwortmöglichkeiten einzugeben. Normalen Besucher der Website ist diese Möglichkeit zu verwehren.
- **Datenbank:** Alle benötigten Daten werden in einer MySQL–Datenbank gespeichert.
- **Echtzeitstatistiken:** Die Auswertung der gegebenen Antworten soll unmittelbar vor der Anzeige berechnet werden.
- **XAMP:** Die Anwendung soll mit der Kombination von Apache–Webserver, MySQL–Datenbank und PHP als Programmiersprache lauffähig sein. Das Betriebssystem kann frei gewählt werden.

Desweiteren sollte die Anwendung objektorientiert programmiert werden, Entwurfsmuster verwenden und die HTML-Ausgabe per CSS formatiert werden.

1.2 Gemeinschaftsarbeit

Die Aufgabe war arbeitsteilig in Teamarbeit zu lösen. Das der Anwendung zu Grunde liegende Datenmodell wurde gemeinsam in einer Teambesprechung erarbeitet und festgelegt. Anschließend wurde ein Mockup¹ der HTML-Seiten und die benötigten SQL-Abfragen, gefolgt von einem prozedural programmierten Prototypen erstellt. Hierbei erstellte der Autor die für die Benutzerverwaltung und Frageneingabe notwendige Programmteile. Die Abfrage- und Auswertungsseiten wurden von Yvonne Frezel gefertigt.

Da die im Seminar DBA02 begonnene Umsetzung des Programmcodes in Klassen unterschiedliche Richtungen verfolgte, wurde die enge Teamarbeit anschließend nicht mehr weitergeführt. Aufgrund der Gemeinsamen Datenbasis sind die vom Autor und von Frenzel erstellten PHP-Dateien miteinander kombinierbar, auch wenn sie intern andere Klassen und Zugriffsmethoden nutzen.

Dieses Assignment geht hauptsächlich auf die vom Autor konzipierten Programmteile „Benutzerverwaltung“ und „Neue Fragen hinzufügen“ ein.

1.3 Aufbau der Arbeit

Zunächst wurden die für die Anwendungen relevanten Konzepte, Techniken, und Frameworks beschrieben. Anschließend erfolgte die Beschreibung der Implementierungsdetails und der vom Autor gewählten Lösungsmöglichkeiten

¹engl. für Attrappe. (to mock: nachahmen)

1.4 Abgrenzung

Da eine Datenbankgestützte Web-Anwendung in der Regel einem großen Personenkreis² zur Verfügung steht sind hier unbedingt Sicherheitsaspekte zu beachten. Da eine ausführliche Betrachtung dieser Maßnahmen den Rahmen dieses Dokuments sprechungen würde, werden nur entsprechende Hinweise auf weiterführende Informationen gegeben. Auch Performance-Überlegungen gehen nur in sehr beschränktem Maß in die Implementation ein.

²allen Internet- oder zumindest Intranetnutzern

2 Grundlagen

2.1 Entwurfsmuster: Fassade

Beim Fassaden-Entwurfsmuster gewährt eine Klasse einen einfachen Zugriff auf ein beliebig komplexes System weiterer Klassen. Den Nutzer der Fassadenklasse benötigt kein Wissen über die Funktionsweise der Klassenhierarchie hinter der Fassade, kann jedoch auf diese zugreifen, falls die bereitgestellte Funktionalität nicht ausreicht³.

In der hier erstellten Anwendung wird der Zugriff auf die Datenbank über die Fassaden-Klasse `SQL` realisiert. Diese erstellt das Low-Level Objekt der Klasse `Datenbank`, bereitet die notwendigen SQL-Abfragen vor und gibt die Resultate dann als String oder als Array von Strings zurück. Die Aufrufenden Routinen benötigen kein Wissen über die verwendete Datenbankschnittstelle oder über die Details der Abfragen. Sollten die in der Klassendefinition vorgesehenen Abfragen allerdings nicht ausreichen, kann auch direkt auf die Klasse `Datenbank` zugegriffen werden.

2.2 Entwurfsmuster: Singleton

Das Klasse nach dem Singleton-Entwurfsmuster stellt sicher, dass es in einem Programm von einer Klasse nur ein einziges Mal instanziiert wird. Allen Nutzern der Klasse wird dann eine Referenz auf ebendiese Instanz übergeben, der Zugriff erfolgt jeweils auf die gleichen Daten, die somit global zur Verfügung gestellt werden⁴.

Somit bildet das Singleton-Designpattern eine passende Grundlage für die Nutzerverwaltung, da immer nur ein Benutzer angemeldet sein kann⁵.

³vgl. [Balzert, 2005], Seite 367ff

⁴vgl. [Balzert, 2005], Seite 361ff

⁵Dies gilt jeweils pro Browser-Instanz. In einem weiteren Browserfenster mit eigenen Cookies kann sich ein weiterer Nutzer anmelden, jedoch auch wieder nur einer

Das Singleton-Entwurfsmuster kann aufgrund seiner Eigenschaften als objektorientierte Umsetzung von globalen Variablen mit all deren Vor- und Nachteilen gesehen werden und wird daher auch als „Anti-Pattern“ kritisiert⁶

2.3 PHP-Schnittstelle: Session-Cookies

Da HTTP ein zustandsloses Protokoll ist, wird ein Mechanismus benötigt, mit dem gespeichert werden kann, ob es sich beim Besucher der Website um einen angemeldeten Benutzer handelt oder nicht. Die von PHP bereitgestellten Session-Cookies können eine begrenzte Menge Daten (ca. 4kB), die im Browser gespeichert wird, von Seitenaufruf zu Seitenaufruf weitergeben⁷.

Die hier besprochene Anwendung verwendet diese Möglichkeit um den Anmeldestatus (`angemeldet==TRUE` bzw. `angemeldet==FALSE`) und den Benutzernamen zu speichern.

2.4 PHP-Schnittstelle: PDO

Der Datenbankzugriff wird in der beschriebenen Anwendung über die PDO-Klasse⁸ realisiert. Diese sorgt für den Verbindungsaufbau zur MySQL-Datenbank und stellt Schutzmechanismen gegen SQL-Injection Angriffe bereit. Für einen solchen Angriff müssen die Benutzereingaben ungeprüft in die SQL-Abfrage (z.B. durch einfache Variablen-Substitution) übernommen werden. Diese wird dadurch dann so verändert, dass sensible Daten ausgespäht oder die Daten verändert bzw. gelöscht werden können⁹. Die Überprüfung der Eingabewerte ist somit in jedem Falle empfehlenswert und aufgrund des öffentlichen Charakters der Webanwendung hier besonders wichtig.

⁶vgl. [Hauer, 2010]

⁷vgl. [Theis, 2013], S. 417ff

⁸PDO: PHP Data Object

⁹vgl. [Friedl, 2007]

2.5 HTML–Designframework: Bootstrap

Die hier vorgestellte Anwendung realisiert ihr Aussehen zum großen Teil mit dem CSS–Framework „Bootstrap“, welches moderne Designprinzipien wie variables, mehrspaltiges Layout oder gar Responsive Layout, welches sich an die unterschiedlichen Bildschirmgrößen vom Smartphone bis zum wandfüllenden HD–TV anpassen kann, anbietet¹⁰.

Die Anwendung selbst muss den entsprechenden Elementen lediglich die gewünschten CSS–Klassen zuweisen. Der Entwickler wird somit nicht mit den Design–Details belastet.

Trotz allem ist eine eigene Anpassung des Designs wie in der Datei `.../style.css` zu sehen ist möglich.

2.6 JavaScript–Frontendframework: jQuery

Bei der Realisierung der Eingabeseite für neue Fragen und Antwortmöglichkeiten wird die JavaScript Bibliothek „jQuery“ verwendet, um weitere Antwortmöglichkeiten einfügen oder diese auch wieder löschen zu können. jQuery bietet auch noch weitere Funktionen für interaktive Benutzeroberflächen, die allerdings in der hier beschriebenen Anwendung nicht genutzt werden.¹¹

¹⁰Lizenz: Apache Licence V2.0, Download unter <http://getbootstrap.com/>

¹¹Lizenz: MIT-Licence, Download unter <http://jquery.com/>

3 Datenbank–Schema

3.1 Konzeptuelles Datenbankschema: Entity–Relationship Diagram

Hier werden die Gegenstände der realen Welt modelliert wie in Abbildung 3.1 gezeigt modelliert. Hierbei ist zu beachten, dass die Entität „Frage“ das zentrale Element des Datenmodells darstellt. Die weiteren Entitäten „Antwortmöglichkeit“ bzw. „gegebene Antwort“ sind existenzabhängige Entities. Ohne Antwortmöglichkeit kann ohne zugehörige Frage nicht existieren, eine gegebene Antwort macht nur Sinn, wenn es eine entsprechende Antwortmöglichkeit und Frage gibt.

Die Entität „Benutzer“ steht in keiner Beziehung zu den anderen Entitäten, sie wird auch nur für die Eingabe neuer Fragen und Antwortmöglichkeiten, bzw. zur Authentifizierung benötigt.

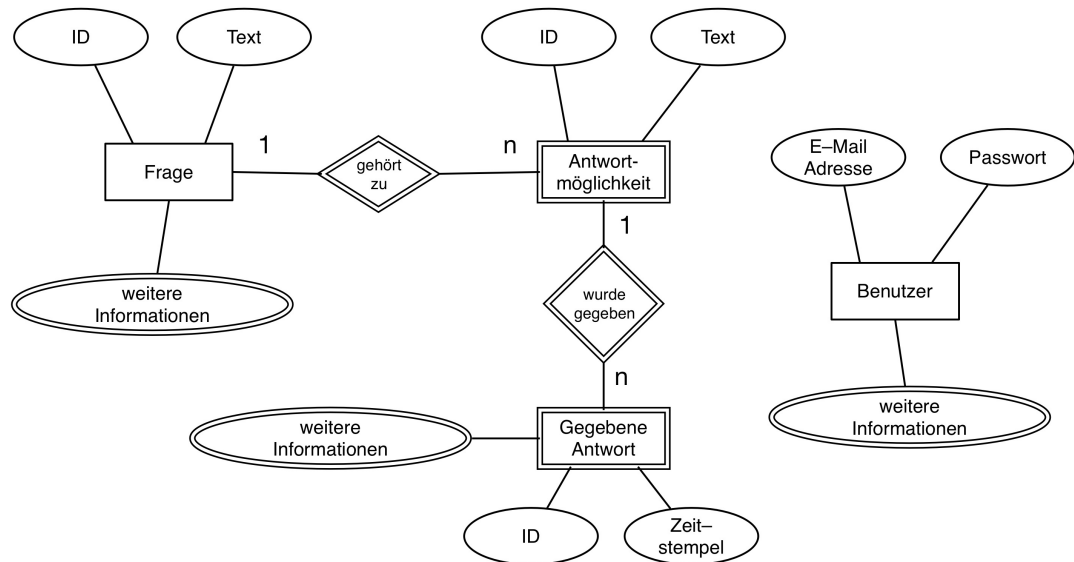


Abbildung 1: Entity Relationship Diagram

Die als zusammengesetzte Attribute dargestellten „weiteren Informationen“ sind Attribute der jeweiligen Entitäten, die für die gestellten Anforderungen nicht not-

wendig sind, jedoch im Produktivbetrieb großen Zusatznutzen bieten könnten. So wäre es möglich die gegebenen Antworten durch die Speicherung der IP-Adresse, Browser-Fingerprinting¹² oder andere Techniken genauer zu identifizieren und somit einem bestimmten Nutzer zuzuordnen. Hierbei sind dann die jeweiligen Datenschutzbestimmungen zu beachten.

Die Datensätze der Entity „Frage“ könnten Zeitangaben enthalten, die festlegen wann bzw. wie lange eine Frage auf der Website angezeigt wird. Die Antwortmöglichkeiten könnten durch entsprechende Angaben zur Sortierreihenfolge geordnet werden.

3.2 Logisches Datenbankschema: Relationales Datenmodell

Aufgrund des in Abbildung 3.1 auf Seite 7 dargestellten Modell ergibt sich die in Abbildung 3.2 dargestellten Relationen. Dieses Modell entspricht der 3. Normalform¹³.

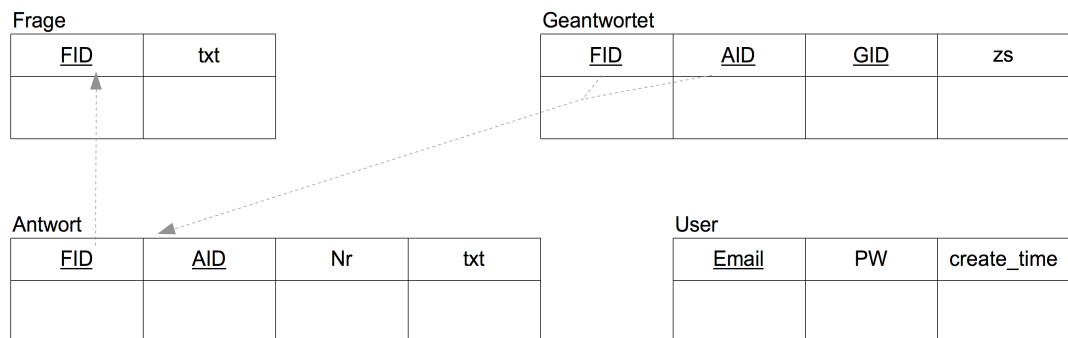


Abbildung 2: Relationales Modell

¹²Die Identifikation eines speziellen Nutzers durch Konfigurationsdetails des Webbrowsers, Bildschirmgröße, Betriebssystemversion, etc. Siehe auch <https://panopticlick.eff.org/>

¹³Kriterien laut [Staud, 2010], Kapitel 3.4

3.3 Umsetzung in SQL

3.3.1 Tabelle: user — Benutzerverwaltung

```
CREATE TABLE user (  
  email VARCHAR(255) NOT NULL,  
  pw CHAR(32) NOT NULL,  
  create_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ('email'));
```

Abbildung 3: SQL: CREATE TABLE user

Als Benutzername und auch als Primärschlüssel wird die E-Mail Adresse des Administrators genutzt. Da diese den Nutzer eindeutig identifiziert wird auf einen separaten Nutzernamen und auch auf eine generierte ID als Primärschlüssel verzichtet.

Das Passwort sollte nicht im Klartext in der Datenbank gespeichert werden. Ein `salted hash`¹⁴ schützt hier das Passwort vor dem Ausspähen durch den Administrator selbst¹⁵ oder durch Angreifer.

Die hier reservierten 32 Byte sind für den in der MySQL-Dokumentation¹⁷ empfohlenen MD5-Hash ausreichend. Da die entsprechende PHP-Dokumentation¹⁸ hier allerdings eine genau entgegengesetzte Empfehlung gibt, ist dieser Sicherheitsaspekt für ein Produktivsystem nochmals genauer zu prüfen.

3.4 Tabelle: frage

3.5 Tabelle: antwort

¹⁴Also der Hashwert des Passworts, welches zuvor mit Applikationsspezifischen Zusatzdaten ergänzt wurde

¹⁵Spätestens seit dem Datenklau bei Vodafone¹⁶ eine dokumentierte Gefahr

¹⁷[?]

¹⁸[?]

```
CREATE TABLE frage (  
  fid INT NOT NULL AUTO_INCREMENT,  
  txt VARCHAR(1024) NOT NULL,  
  PRIMARY KEY ('fid'));
```

Abbildung 4: SQL: CREATE TABLE frage

```
CREATE TABLE antwort (  
  aid INT NOT NULL AUTO_INCREMENT,  
  nr INT NULL,  
  txt VARCHAR(1024) NOT NULL,  
  fid INT NOT NULL,  
  PRIMARY KEY ('aid'),  
  FOREIGN KEY ('fid') REFERENCES frage('fid'));
```

Abbildung 5: SQL: CREATE TABLE antwort

3.6 Tabelle: geantwortet

```
CREATE TABLE geantwortet (  
  gid INT NOT NULL AUTO_INCREMENT,  
  aid INT NOT NULL,  
  zs TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ('gid'),  
  FOREIGN KEY ('aid') REFERENCES antwort('aid'));
```

Abbildung 6: SQL: CREATE TABLE geantwortet

3.7 Query: Frage – Text

Wird für die Abfrage und Auswertung benötigt.

3.8 Query: Antwortmöglichkeiten

Wird für die Abfrage benötigt.

```
select txt from frage where fid = <n>;
```

Abbildung 7: SQL: Text von Frage <n>

```
select antwort.txt
      from antwort
     where fid = <n>
```

Abbildung 8: SQL: Mögliche Antworten für Frage <n>

3.9 Query: Anzahl der gegebenen Antworten

Wird für die Auswertung benötigt: 100%

```
select count(geantwortet.aid)
      from antwort, geantwortet
     where geantwortet.aid = antwort.aid and antwort.fid = <n>;
```

Abbildung 9: SQL: Antwortzahl 100% von Frage <n>

3.10 Query: Gegebene Antworten und Häufigkeit

Wird für die Auswertung benötigt.

3.11 Tabelle

Head1	Head2	Head3
1	2	3
4	5	6
7	8	9
1	2	3
4	5	6
7	8	9

Tabelle 1: Beschreibung. Quelle: Berger, Vorlesung, 2012, München

```
select antwort.txt, count(geantwortet.aid)
  from antwort, geantwortet
 where geantwortet.aid = antwort.aid and antwort.fid = <n>
 group by geantwortet.aid;
```

Abbildung 10: SQL: Gegebene Antworten mit Häufigkeit für Frage <n>

3.12 Bilder



Abbildung 11: Akad. Quelle: www.akad.de

3.13 Syntax Highlighting

```
<?php
$title="Lorem";
$desc = "Lorem Ipsum";
include($_SERVER['DOCUMENT_ROOT'] . '/header.php');
?>
```

Abbildung 12: Quellcode: Aufruf von header.php (PHP)

4 Bewertung

4.1 Zusammenfassung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.2 kritische Würdigung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.3 Ausblick

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.4 Erfolgsfaktoren

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Literatur

- [Balzert, 2005] Balzert (2005). *Lehrbuch der Objektmodellierung*. Spektrum Akademischer Verlag, München.
- [Friedl, 2007] Friedl (2007). SQL Injection Attacks by Example, Abruf am 25.10.2013. <http://www.unixwiz.net/techtips/sql-injection.html>.
- [Fuest, 2013] Fuest (2013). Fall Vodafone zeigt die wahren Sicherheitslücken, Abruf am 25.10.2013. <http://www.welt.de/wirtschaft/webwelt/article119967954/Fall-Vodafone-zeigt-die-wahren-Sicherheitsluecken.html>.
- [Hauer, 2010] Hauer (2010). Das Singleton Design Pattern, Abruf am 11.10.2013. <http://www.philippbauer.de/study/se/design-pattern/singleton.php#nachteile>.
- [Staud, 2010] Staud (2010). *Grundlagen der Datenorganisation: Vom Datenmodell zur Speicherung in Dateien*. AKAD. Die Privathochschulen GmbH, Stuttgart.
- [Strickel, 1991] Strickel (1991). *Datenbankdesign: Methoden und Übungen*. Gabler, Wiesbaden.
- [Theis, 2013] Theis (2013). *Einstieg in PHP 5.5 und MySQL 5.6*. Gabler, Wiesbaden.

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.

tum, Ort)

(Unterschrift) (Da-

— Druckgröße kontrollieren! —

Breite = 100 mm

Höhe = 50 mm

— Diese Seite nach dem Druck entfernen! —