

Praktikum zur Vorlesung Datenbanken
BAI, Komedia, ISE CSCE/CE,
Mathematik/Technomathematik und Lehramt
Informatik

Alfred Sliwa, M.Sc.
Ioannis Karatassis, M.Sc.
Iryna Rushchak
Dakai Men
Huthayfa Ghunaim

Wintersemester 2019/2020

Datum	
Team (Account)	
Passwort	

Aktuelle Informationen, Ansprechpartner, Materialien und Uploads unter:
http://www.is.inf.uni-due.de/courses/db_ws19/

v1.0

Inhaltsverzeichnis

0	Block 0: Einführung in Linux	3
1	Block 1: Modellierung der Datenbank für die Crowdfunding-Plattform „ProjectFunder“ (12 Punkte)	4
2	Block 2: SQL mit DB2 (15 Punkte)	6
3	Block 3: Implementierung der Crowdfunding-Plattform „ProjectFunder“ (18 Punkte)	9
A	Spezifikation der Website	11
A.1	Beschreibung der Miniwelt	11
A.2	Beschreibung der zu erstellenden Website	12
A.2.1	Hauptseite (ID: <code>view_main</code>)	12
A.2.2	Projekt erstellen (ID: <code>new_project</code>)	13
A.2.3	Detailseite eines Projekts (ID: <code>view_project</code>)	14
A.2.4	Projekt editieren (ID: <code>edit_project</code>)	16
A.2.5	Projekt Spende (ID: <code>new_project_fund</code>)	18
A.2.6	Benutzerprofil (ID: <code>view_profile</code>)	18
A.2.7	Kommentare hinzufügen (ID: <code>new_comment</code>)	19
A.2.8	Bonusaufgabe (1 Punkt)	21
B	Beschreibung der IMDB-Datenbank	23
B.1	UML-Diagramm der IMDB-Datenbank	23

0 Block 0: Einführung in Linux

Keine Abgabe

Aufgabe 1: Einrichten der Arbeitsumgebung (0 Punkte) Bereitet zunächst Eure Arbeitsumgebung vor, so dass Ihr in den nächsten Wochen damit komfortabel arbeiten könnt. Die meisten Einstellungen können direkt über das Menü in der rechten oberen Ecke (eine Sprechblase mit Eurem Login-Namen) und dann **Systemeinstellungen** vorgenommen werden.

Aufgabe 2: Linux-Umgebung (0 Punkte) Zur Eingewöhnung in die Arbeitsumgebung werden einige vom Praktikumsleiter vorgegebene, einfache Aufgaben bearbeitet.

Am Ende der Aufgabe solltet Ihr in der Lage sein, mit der Linux-Shell Verzeichnisse zu navigieren, Verzeichnisse anzuzeigen und Dateien anzusehen. Ihr solltet wissen, wo Ihr einen Text-Editor, Dateibrowser oder Webbrowser aufrufen könnt und wie Ihr aus einer Textdatei ein PDF erzeugt.

1 Block 1: Modellierung der Datenbank für die Crowdfunding-Plattform „ProjectFunder“ (12 Punkte)

Abgabe bis 01.12.2019, 23:59 Uhr

Aufgabe 3: Datenbank-Modellierung (6 Punkte) In diesem Semester soll „ProjectFunder“ – eine Web-Applikation für die Finanzierung von Projekten – entwickelt werden. Die Plattform ist sehr vielseitig und soll verschiedene Funktionen unterstützen, u.A. folgende:

- Benutzer können Projekte erstellen, aktualisieren und löschen.
- Benutzer können Projekte mit einer Spende unterstützen.
- Benutzer können Projekte kommentieren.

In dieser Aufgabe soll zunächst die zugrundeliegende Datenbank durch ein **Entity-Relationship-Diagramm (ER-Diagramm)** modelliert werden.

Im Anhang A.1 auf Seite 11 wird die Funktionsweise der zu erstellenden Web-Applikation vorgestellt. Das zu modellierende ER-Diagramm soll **nur** auf der Beschreibung der Miniwelt basieren. Es gilt zu beachten, dass nicht jede Anforderung im ER-Diagramm umgesetzt werden kann.

Es ist dringend anzuraten vor dem Entwurf, neben der Beschreibung der Miniwelt, auch die vollständige Spezifikation (siehe Anhang A.2 auf Seite 12) zu lesen und zu verstehen.

- Verwendet für die Erstellung des ER-Diagramms das Programm Dia (<http://dia-installer.de/>).
- Falls ihr während der Modellierung mehr Attribute als angegeben verwenden möchtet, so ist dies – an sinnvollen Stellen – natürlich erlaubt.
- Gibt Kardinalitäten in der Chen-Notation (1:1, 1:N, N:M) an.

Abgabe:

- ER-Diagramm als PDF-Datei

Aufgabe 4: Datenbank-Erstellung (6 Punkte) Erzeugt für die in Aufgabe 3 entworfene Datenbank benötigte `CREATE TABLE`- sowie `INSERT INTO`-Statements, welche die Datenbank-Tabellen in der DB2 anlegen und befüllen. Es sollen nur die Daten eingefügt werden, die in der Beschreibung der Miniwelt explizit spezifiziert wurden. Wendet die Techniken aus der Datenbanken-Vorlesung an, um euer ER-Diagramm aus Aufgabe 3 zunächst in ein Datenbankenschema zu überführen.

Die Abgabe soll enthalten:

- `CREATE TABLE`-Statements inkl. sinnvoller Integritätsbedingungen für sämtliche Attribute (4,5 Punkte)
- `INSERT INTO`-Statements (0,5 Punkte)
- Einen Trigger, der beim Einfügen einer Spende prüft, ob das Finanzierungslimit für das jeweilige Projekt erreicht wurde. Falls ja, soll der Status des Projekts auf „geschlossen“ geändert werden. (1 Punkt)

Abgabe:

- `CREATE TABLE`-Statements, `INSERT INTO`-Statements, und `CREATE Trigger`-Statement als Textdatei (.txt)

2 Block 2: SQL mit DB2 (15 Punkte)

Abgabe bis 22.12.2019, 23:59 Uhr

Für die folgenden Aufgaben steht eine Beschreibung der verwendeten Datenbank im Anhang B auf Seite 23 zur Verfügung.

Hinweis: Block 2 ist unabhängig von Block 1 und 3. Letztere bauen aufeinander auf.

Aufgabe 5: Katalogisieren (1 Punkt) In diesem Praktikum werden wir die DB2 aus der Kommandozeile heraus ansprechen. Es gibt auch grafische, Java-basierte Werkzeuge, auf die wir jedoch hier verzichten.

Die Datenbank, mit der wir arbeiten werden, befindet sich auf einem entfernten Rechner und muss daher in eurem Rechner importiert werden. Macht euch kurz mit den unterschiedlichen Modi vertraut, die in den Unterlagen beschrieben werden. Katalogisiert dann die Instanz `dbmaster` auf dem Rechner `bijou.is.inf.uni-due.de` mit Server-Port 50005.

Katalogisiert die Datenbank IMDB lokal unter einem Alias. Anschließend, lasst euch das Node Directory und das Database Directory anzeigen.

Abgabe:

- Einträge für Instanz (node) und Datenbank (database) als Textdatei (.txt)

Aufgabe 6: Sichten (2 Punkte)

- (a) Erstellt eine Sicht `dbpXXX1.avgRatedMovies`, welche Filme beinhalten soll, die besser als durchschnittlich bewertet wurden.

★ Die Sicht soll 4 Spalten umfassen: `MOVIEID`, `TITLE`, `YEAR` und `RANK`.

- (b) Erstellt eine Sicht `dbpXXX.waltDisneyMovies`, die alle Filme enthält, die der Produzent „Disney, Walt“ produziert hat. Darüber hinaus sollen die Genres der Filme angezeigt werden. Hat ein Film mehr als ein Genre, so soll pro Genre eine Zeile angezeigt werden.

★ Die Sicht soll 3 Spalten umfassen: `MOVIEID`, `TITLE`, und `GENRE`.

Abgabe:

- SQL-Befehle als Textdatei (.txt)

Aufgabe 7: Verwendung von Sichten (2 Punkte)

- (a) Verwendet die Sicht `dbpXXX.avgRatedMovies` aus Aufgabe 6, um für jedes Jahr die Filme mit der jeweils höchsten Bewertung auszugeben. (Es kann durchaus mehrere Filme pro Jahr mit der höchsten Bewertung geben.)

★ Die Ausgabe soll 3 Spalten umfassen: `YEAR`, `TITLE`, und `RANK`.

¹Dabei steht „XXX“ für eure Gruppennummer, z.B. `dbp001`.

- (b) Verwendet die Sicht `dbpXXX.waltDisneyMovies` aus Aufgabe 6, um Filme auszugeben, die nicht mehr als 2 Genres besitzen.

★ Die Ausgabe soll 2 Spalten umfassen: `TITLE`, `AMOUNT`.

Abgabe:

- SQL-Befehle (SELECT Statements) als Textdatei (.txt)

Aufgabe 8: SQL-Anfragen (7 Punkte) Formuliert folgende Anfragen in SQL:

- (a) Wie viele Schauspieler (Actors) hat ein Horrorfilm durchschnittlich?

★ Die Ausgabe soll 1 Spalte umfassen: `AMOUNT`

- (b) Ermittelt für jeden Regisseur (Director) den durchschnittlichen Profit (gross) pro Film. Sortiert die Ausgabe absteigend nach dem Profit.

★ Die Ausgabe soll 2 Spalten umfassen: `NAME` und `MEAN_GROSS`

- (c) Welcher Film hat die meisten Schauspieler? (Es kann mehrere Filme geben.) Gebt neben dem Titel des Films auch die Anzahl der Schauspieler aus.

★ Die Ausgabe soll 2 Spalten umfassen: `TITLE`, `AMOUNT`

- (d) Welche Regisseure haben mindestens einmal selbst in einem ihrer eigenen Filme mitgespielt? Wir gehen mal davon aus, dass ein Regisseur (Director) und ein Schauspieler (Actor) ein und dieselbe Person sind, wenn sie den gleichen Namen besitzen.

★ Die Ausgabe soll 1 Spalte umfassen: `NAME` (Regisseur)

- (e) Welche Autoren (Writer) haben **nur** an Filmen mitgewirkt, die zwischen 1920 und 1929 veröffentlicht wurden.

★ Die Ausgabe soll 1 Spalte umfassen: `NAME` (Writer)

- (f) Ermittle für jeden Film die mitwirkenden Schauspieler und Regisseure (Rolle). Pro Film und Mitwirkender soll eine Zeile ausgegeben werden. Seine Rolle (Actor/Director) soll ebenfalls ausgegeben werden. Die Ausgabe soll absteigend nach dem Filmtitel sortiert werden.

★ Die Ausgabe soll 4 Spalten umfassen: `TITLE` (Film), `YEAR` (Film), `NAME` (Mitwirkender) und `ROLE` (Actor/Director)

- (g) Berechne für jedes Genre den relativen Marktanteil in Prozent gemessen an den Filmen, die das jeweilige Genre besitzen. Sind beispielsweise von 80 Filmen 10 davon Horrorfilme, so beträgt der Marktanteil für Horrorfilme 12.5%. Filme, die kein Genre haben, sollen grundsätzlich nicht berücksichtigt werden.

★ Die Ausgabe soll 3 Spalten: `GENRE`, `MOVIES_PER_GENRE`, `MARKET_SHARE`.

Abgabe:

- SQL-Befehle als Textdatei (.txt)

Aufgabe 9: Rekursive Anfrage (1 Punkt) Das Spiel „Six Degrees of Kevin Bacon“ geht so (Beschreibung von <http://www.cs.virginia.edu/oracle>):

The object of the game is to start with any actor or actress who has been in a movie and connect them to Kevin Bacon in the smallest number of links possible. Two people are linked if they've been in a movie together. We do not consider links through television shows, made-for-tv movies, writers, producers, directors, etc. For example, you might wonder how Alfred Hitchcock can be connected to Kevin Bacon. One answer is that:

Alfred Hitchcock was in Show Business at War (1943) with Orson Welles, and Orson Welles was in A Safe Place (1971) with Jack Nicholson, and Jack Nicholson was in A Few Good Men (1992) with Kevin Bacon!

*Then we can count how many links were necessary and assign the actor or actress a **Bacon number**.*

Findet alle Schauspieler und Schauspielerinnen mit einer maximalen Bacon-Nummer von zwei. Als Gedankenübung schreibt ein rekursives SQL-Statement, dass die Schauspieler und Schauspielerinnen für beliebige Bacon-Nummern findet.

Tipp: In SELECT-Statements kann man sich das Ergebnis von Berechnungen zurückliefern lassen. Beispiel: `SELECT 1+3 FROM title;`

Abgabe:

- SQL-Befehle als Textdatei (.txt)

10: Trigger (2 Punkte) Erstellt zunächst die Tabelle `dbpXXX.frequentRatedMovies` (ersetzt XXX wieder durch eure Gruppennummer):

```
CREATE TABLE dbpXXX.frequentRatedMovies (  
  movieID INTEGER NOT NULL,  
  title VARCHAR(400),  
  year VARCHAR(100),  
  PRIMARY KEY (movieID)  
);
```

Schreibt nun zwei Trigger mit dem Namen `dbpXXX.trigger1` und `dbpXXX.trigger2`, die folgendes leisten:

- **trigger1:** Sobald ein Film seine 100. Bewertung erhält, soll er in die Tabelle `dbpXXX.frequentRatedMovies` mit den angegebenen Attributen aufgenommen werden.
- **trigger2:** Sobald ein Rating zu einem Film gelöscht wird und dieser Film dadurch weniger als 100 Bewertungen hat, soll der Film aus der Tabelle `dbpXXX.frequentRatedMovies` entfernt werden.

Abgabe:

- SQL-Befehle als Textdatei (.txt)

3 Block 3: Implementierung der Crowdfunding-Plattform „ProjectFunder“ (18 Punkte)

Fertigstellung bis zum Abnahmetermin (27.01.2020 – 31.01.2020)

11: Korrektur des Entwurfs (0 Punkte) Wenn Ihr die Korrekturen zu Block 1 erhalten habt, überarbeitet euren Entwurf und beseitigt evtl. Fehler und Probleme in eurer Datenbank. Setzt letztere anschließend auf und befüllt diese mit Beispieldaten. Wir empfehlen jedoch, die Musterlösung zum Aufsetzen der Datenbank zu verwenden. Diese wird nach Ablauf des 1. Blocks auf der Veranstaltungs-Webseite veröffentlicht.

Abgabe:

- Keine.

12: Implementierung (18 Punkte) Ziel des 3. Blocks ist die Implementierung einer Web-Applikation, die die beschriebenen Anforderungen (siehe Beschreibung A.2 im Anhang A) umsetzt. Die Struktur der Daten ist bereits aus der Beschreibung der Miniwelt (siehe Anhang A.1) bekannt. Als Datenbank dient das in Block 1 erstellte Datenbankenschema.

Folgende Anforderungen sind zu beachten:

- Verwendung von DB2 als DBMS. Andere DBMS (wie z.B. MySQL oder MariaDB) sind nicht zugelassen und führen bei Verwendung zum Nichtbestehen. Wir empfehlen, DB2 auf unseren Rechnern zu verwenden. Die Anmeldedaten erhalten ihr zu Beginn des Praktikums.
- Die Wahl der Programmiersprache ist auf Java und Python beschränkt. Von euch wird eine Webanwendung verlangt, die die Daten aus der DB2 liest / bzw. in die DB2 schreibt und anzeigt. Beispielprojekte in Java sowie Python werden bereitgestellt, welche als Grundlage für die anstehende Implementierung verwendet werden können.

Es gilt zu beachten, dass etwaige Frameworks/Bibliotheken bezüglich SQL-Statements nicht erlaubt sind. Verwendet werden sollten die jeweiligen Standardfunktionen der Programmiersprache (z.B. Statements in Java).

- Korrekte Verwendung von Transaktionen.
- Schließen von DB-Ressourcen an sinnvollen Stellen.
- Verhinderung von SQL-Injection.
- Zum Testen der Website sowie für die Abnahme, sollten in jeder erzeugten Tabelle genug Beispieldaten vorhanden sein.
- Alle Interaktionen sollen von dem aktuell angemeldeten Benutzer erfolgen. Eine Login-Funktion ist nicht notwendig. Der Benutzer kann hartcodiert in den Programmcode eingefügt werden.

13: Bonusaufgabe (1 Punkt) Im Anhang A.2.8 befinden sich eine Bonusaufgabe. Ihr habt die Möglichkeit, einen Bonuspunkt zu sammeln, indem ihr die Anforderungen vollständig implementiert. Teilpunkte werden nicht vergeben.

Abgabe:

- Keine. Die Web-Anwendung wird bei einem separaten Abnahmetermin zusammen mit den Veranstaltern vorgestellt und überprüft.

Hinweise zur Abnahme von Block 3: Beim Abnahmetermin gibt es zwei Möglichkeiten:

- (a) Ihr könnt eure Lösung auch auf einem Rechner im Computerpool vorführen. Dazu sollte sich die Endversion eures Projekts im Workspace eures Accounts befinden.
- (b) Ihr könnt eure Lösung abnehmen lassen, indem Ihr sie auf eurem eigenen Laptop etc. vorführt. Dazu bringt ihr das Gerät zum Termin mit und denkt bitte an Netzteil und/oder an genug Akkuladung. Testet die Vorführung bitte vorher. Zur Not sollte sich im Workspace eures Accounts eine Kopie eures Endprojekts befinden.

Wir empfehlen Variante (a), da wir immer noch eingreifen können, falls etwas nicht funktionieren sollte. In beiden Fällen werden wir uns sowohl die Funktionalität des Programms als auch den Quellcode anschauen! Für jedes Feature, was korrekt implementiert wurde, werden Teilpunkte vergeben. Daher lohnt sich eine Anmeldung für die Abnahme in jedem Fall.

Wichtig: Um die Abnahme erfolgreich zu bestehen, müssen folgende Bedingungen erfüllt werden:

- Sowohl die Web-Applikation als auch die Datenbank müssen gestartet werden können. Etwaige Änderungen können nicht während der Abnahme durchgeführt werden!
- Die Hauptseite der Web-Applikation muss aufrufbar sein.
- Alle anzuzeigende Daten (außer Bilddateien) müssen in der Datenbank vorhanden sein. Die Web-Applikation muss in der Lage sein, Daten zu lesen, zu schreiben und zu löschen.
- Das Projekt muss von eurer Gruppe erstellt worden sein. Täuschungsversuche (z.B. hartkodierte Daten statt dem Lesen aus der Datenbank) oder Plagiate führen zum Ausschluss vom Praktikum und werden dem Prüfungsausschuss mitgeteilt.

Trifft eine Bedingung nicht zu, so wird Block 3 mit 0 Punkten bewertet. Testet daher vorher ausgiebig euer Projekt und verzichtet auf (ungetestete) Änderungen in letzter Sekunde!

Anhang

A Spezifikation der Website

A.1 Beschreibung der Miniwelt

- Benutzer haben einen Namen, eine eindeutige E-Mail Adresse sowie eine optionale Profilbeschreibung.
- Alle Benutzer verfügen über ein Konto, welches ihr Guthaben auf der Plattform verwaltet. Das Konto ist eindeutig über den Benutzer identifizierbar und verfügt neben einem Guthaben über eine Geheimzahl.
- Ein Benutzer kann mehrere Projekte erstellen. Jedes Projekt wird von genau einem Benutzer erstellt.
- Ein Projekt besitzt eine eindeutige Kennung (z.B. eine fortlaufende Nummer), einen Titel, eine Beschreibung (optional) und ein Finanzierungslimit.
- Benutzer können Projekte durch Spenden unterstützen. Wir speichern natürlich auch den gespendeten Betrag (Spendenbetrag). Um die Privatsphäre von Spendern zu schützen, wird zudem festgehalten, ob die Spende anonym oder öffentlich erfolgen soll. Eine Spende kann nur durchgeführt werden, wenn genug Geld auf dem Konto des Benutzers vorhanden ist.
- Bei Guthaben, Finanzierungslimit und Spendenbetrag handelt es sich um Fließkommazahlen mit je zwei Nachkommastellen. Grundlegend wird auf der Plattform die Währung Euro verwendet.
- Ein Projekt besitzt ein Attribut Status, welches nur die Werte „offen“ und „geschlossen“ annehmen kann. Wird ein Projekt erstellt, so nimmt der Status standardmäßig den Wert „offen“ an. Geschlossen hingegen bedeutet, dass das Finanzierungslimit erreicht wurde und keine weiteren Spenden getätigt werden können.
- Projekte können in einer Vorgänger-Nachfolger-Beziehung stehen. Ein Projekt kann mehrere Nachfolger haben aber maximal einen Vorgänger; wir kennzeichnen beide Rollen eindeutig!

Beispiel: Das Projekt namens „Ubuntu Touch“ hat als Nachfolger „Ubuntu Touch Light“ und „Ubuntu Touch Pro“.

- Kategorien besitzen jeweils eine ID, einen Namen, und ein Icon. Wir bieten folgende Kategorien an: „Health & Wellness“, „Art & Creative Works“, „Education“ und „Tech & Innovation“. Jedes Projekt ist eindeutig einer Kategorie zugeordnet.
- Zu jedem Projekt können Benutzer Kommentare erstellen. Ein Kommentar besteht aus einer ID, einer Textnachricht und dem Erstellungsdatum. Außerdem kann der Benutzer die Sichtbarkeit des Kommentars (privat oder öffentlich) angeben. Private Kommentare verbergen – im Gegensatz

zu öffentlichen – den tatsächlichen Benutzernamen und zeigen stattdessen den Namen “Anonym” an.

A.2 Beschreibung der zu erstellenden Website

Zu erstellen ist eine **Web-Applikation** namens „ProjectFunder“. Diese basiert auf der Datenbank aus Block 1. Unsere Web-Applikation soll mehrere Seiten umfassen, um u.a. Projekte zu erstellen oder zu unterstützen. Alle Seiten, die für Block 3 implementiert werden müssen, werden in den Abschnitten A.2.1 bis A.2.7 beschrieben.

Jede zu erstellende Seite besitzt eine eigene ID, über die sie aufgerufen werden kann. Darüber hinaus existiert für jede Seite eine graphische Darstellung (Mockup). Verweise auf andere Seiten werden in den Mockups durch blaue Pfeile gekennzeichnet. In jedem Fall ist ein Link oder Button gefordert, der den Benutzer auf die entsprechende Seite weiterleitet. Es gilt zu beachten, dass das vorgegebene Design in den Mockups nicht 1 zu 1 nachgebaut werden muss. Es wird lediglich die Funktionalität eurer Web-Applikation bewertet. Für jede geforderte Funktionalität, die erfolgreich umgesetzt wurde, werden Teilpunkte vergeben.

Zum besseren Verständnis wird für jede zu erstellende Seite die Funktionalität im Namen mitangegeben. Wir verwenden die Präfixe **view**, **new** sowie **edit**:

- **view** dient hauptsächlich zum Anzeigen von Daten.
- **new** dient zum Erstellen von Daten.
- **edit** dient zur Aktualisierung bereits vorhandener Daten.

Beispiel: Die Seite **view_project** dient zum Anzeigen der Details eines Projekts während **new_project** das Erstellen von Projekten ermöglicht. Über **edit_project** können wir ein bereits vorhandenes Projekt aktualisieren.

A.2.1 Hauptseite (ID: view_main)

- Der Einstieg soll über die Hauptseite mit der ID **view_main** erfolgen (siehe Abbildung 1). Auf der Hauptseite sollen alle offene und geschlossene Projekte separat angezeigt werden. Für jedes aufgelistete Projekt, sollen Icon, Titel, Benutzername des Erstellers und die aktuelle Spendensumme (Aktuell) angezeigt werden. Bei dem Icon handelt es sich um das Icon der entsprechenden Kategorie. Darüber hinaus enthält die Hauptseite noch zwei Buttons: „Mein Profil“ und „Projekt erstellen“.
- Klickt man auf den Button „Mein Profil“, so wird man auf das eigene Profil (Seite **view_profile** A.2.6) weitergeleitet.
- Klickt man auf den Button “Projekt Erstellen“, so wird man auf die Seite **new_project** (A.2.2) weitergeleitet.
- Klickt man auf den Titel eines Projekts, so soll die entsprechende Detailseite (**view_project** A.2.3) aufgerufen werden.

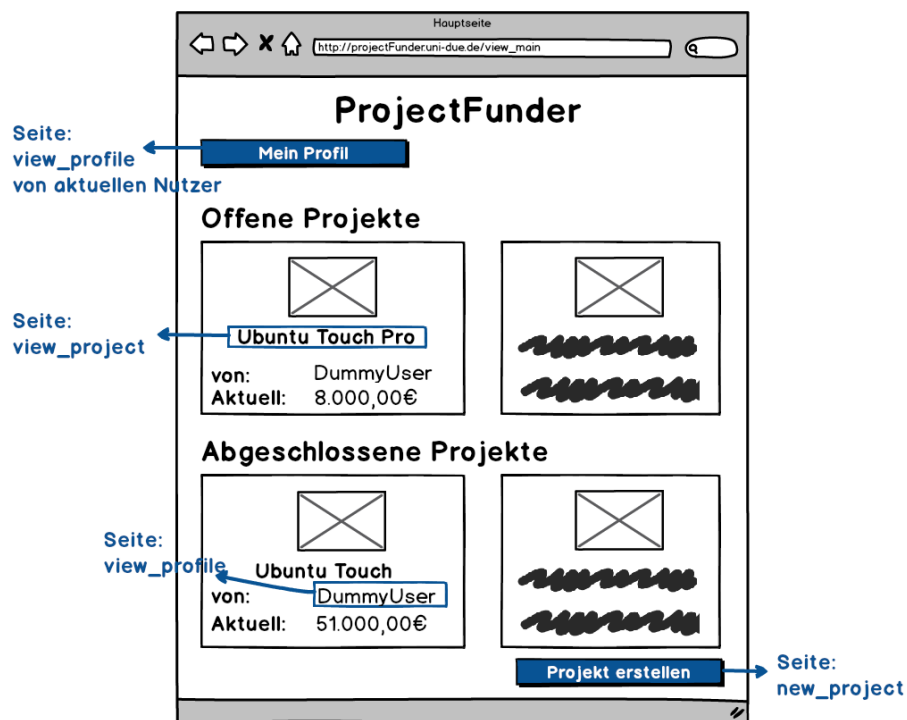


Abbildung 1: Übersicht aller Projekte (ID: view_main)

- Klickt man auf den Benutzernamen (Ersteller eines Projekts), so soll die entsprechende Profilseite (view_profile A.2.6) aufgerufen werden.

A.2.2 Projekt erstellen (ID: new_project)

- Über die Seite new_project soll ein neues Projekt erstellt werden können (siehe Abbildung 2).
 - Titel und Finanzierungslimit sollen jeweils über ein Textfeld angegeben werden.
 - Für unsere vier Kategorien „Health & Wellness“, „Art & Creative Works“, „Education“ und „Tech & Innovation“ verwenden wir Radio-Buttons.
 - Für das Vorgänger-Projekt kann ein Projekt aus der Liste der bereits erstellten Projekte vom (und **nur** vom) aktuellen Benutzer selektiert werden. Die Auswahl erfolgt über Radiobuttons. Es gibt auch die Option „Kein Vorgänger“ (Standardauswahl), falls der Benutzer das zu erstellende Projekt nicht verknüpfen will.

Beispiel: Wenn der Benutzer zuvor schon die Projekte „Ubuntu Touch“ und „Ubuntu Touch Pro“ erstellt hat, so werden diese beiden Projekte zur Auswahl des Vorgänger Projekts des neu zu erstellenden Projekts „Ubuntu Touch Light“ zur Verfügung gestellt.

Projekt erstellen

Titel

Finanzierungslimit €

Kategorie ☐ Health & Wellness ☐ Art & Creative Works
☐ Education ☒ Tech & Innovation

Vorgänger ☒ Ubuntu Touch
☐ Ubuntu Touch Pro
☐ Kein Vorgänger

Beschreibung

Erstellen

Abbildung 2: Seite zum Erstellen von Projekten (ID: new_project)

- Die Beschreibung soll über eine Textarea angegeben werden.
- Klickt der Benutzer auf den Button „Erstellen“, so wird das Projekt in die Datenbank eingetragen. Für die Kennung kann eine fortlaufende Nummer (ID) verwendet werden. Treten bei der Erstellung eines Projekts Fehler auf, so soll eine entsprechende Fehlermeldung angezeigt und das Projekt nicht erstellt werden. Folgende Fehler müssen abgefangen werden:
 - Wenn die Länge des eingegeben Titels 30 überschreitet oder leer ist.
 - Wenn kein numerischer Wert oder ein Wert kleiner als 100 für das Finanzierungslimit eingegeben wurde.
 - Wenn keine Kategorie ausgewählt wurde.
 - Wenn keine Option für Vorgänger ausgewählt wurde.
 - Allgemein Fehler mit der Datenbank.

A.2.3 Detailseite eines Projekts (ID: view_project)

Abbildung 3 illustriert die Detailseite eines Projekts. Diese umfasst 4 Abschnitte:

Informationen

- Anhand der Kennung eines Projekts sollen folgende Informationen aus der Datenbank gelesen und anschließend angezeigt werden: Icon der Kategorie, Titel, Benutzername des Projekterstellers, Beschreibung (falls vorhanden), Finanzierungslimit, die aktuelle Spendensumme und der Status.
- Klickt der Benutzer auf den Benutzernamen vom Ersteller, so soll die Profilseite des Erstellers (A.2.6) aufgerufen werden.
- Es soll zudem eine Verlinkung zum Vorgänger-Projekt mitangegeben (A.2.3) werden, falls eine Vorgänger-Beziehung existieren sollte. Falls dies nicht zutrifft, dann soll ein einfacher Text mit dem Inhalt „Kein Vorgänger vorhanden“ erscheinen.

Aktionsleiste Der Benutzer kann verschiedene Aktionen auf der Detailseite eines Projekts ausführen. Es werden die Aktionen „Spenden“, „Editieren“ und „Löschen“ in Form von Buttons/Links unterstützt.

- Klickt der Benutzer auf den Button/Link „Spenden“, so wird man auf die Seite `new_project_fund` (A.2.5) mit dem entsprechenden Parameter (Kennung) weitergeleitet.
- Klickt der Benutzer auf den Button/Link „Projekt Editieren“, so wird man auf die Seite `edit_project` (A.2.4) mit dem entsprechenden Parameter (Kennung) weitergeleitet.
- Klickt der Benutzer auf den Button „Projekt Löschen“, so sollen die folgenden Aktionen durchgeführt werden, aber nur, wenn es sich beim aktuellen Benutzer um den Ersteller handelt. Sonst wird eine Fehlermeldung angezeigt.
 - Die zum Projekt zugehörigen Kommentare sollen gelöscht werden.
 - Die zum Projekt zugehörigen Spenden sollen gelöscht werden.
 - Die Benutzer, die dieses Projekt unterstützt haben, erhalten ihre Spende zurück. Das bedeutet, dass die Spenden auf die Konten der jeweiligen Benutzer gutgeschrieben werden.
 - Das Projekt soll aus der Datenbank gelöscht werden.
 - Der aktuelle Benutzer wird anschließend auf die Hauptseite weitergeleitet (A.2.1).

Falls ein Fehler während des Vorgangs auftritt, so soll der gesamte Vorgang abgebrochen und eine entsprechende Fehlermeldung angezeigt werden.

Liste der Spender

- Im mittleren Bereich der Webseite sollen die zum Projekt zugehörigen Spender aus der Datenbank geladen und angezeigt werden. Neben den Namen der Spender soll der jeweilige Spendenbetrag angezeigt werden.

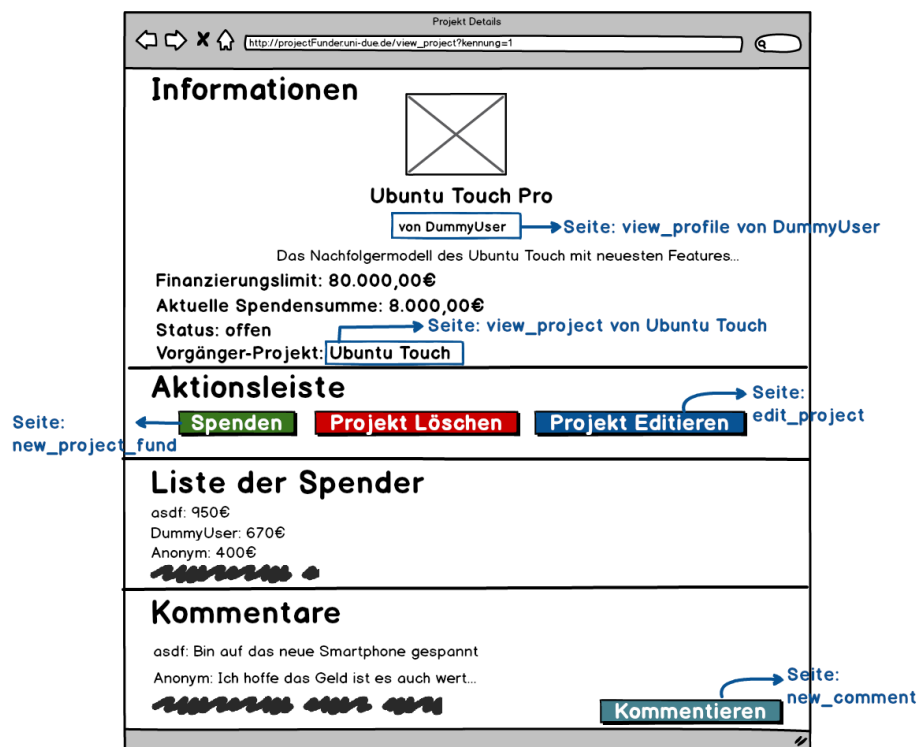


Abbildung 3: Detailseite eines Projekts (ID: view_project)

- Die Liste der Spender soll absteigend nach dem Spendenbetrag (größte zuerst) sortiert werden.
- Falls ein Nutzer anonym gespendet hat, so soll der Name „Anonym“ in dieser Liste angezeigt werden und nicht sein tatsächlicher Benutzername.

Kommentare

- Im unterem Bereich der Webseite sollen die zum Projekt zugehörigen Kommentare aus der Datenbank geladen und angezeigt werden. Neben dem Kommentar soll der Benutzername des Verfassers (oder der Name „Anonym“, falls der Kommentar nicht öffentlich ist) angezeigt werden.
- Die Kommentare sollen anhand des Erstellungsdatums absteigend (neuere zuerst) sortiert werden.
- Klickt man auf den Button/Link „Kommentieren“, so wird man auf die Seite new_comment (A.2.7) weitergeleitet.

A.2.4 Projekt editieren (ID: edit_project)

Die Seite zum Editieren eines Projektes hat den gleichen Aufbau wie die Seite zum Erstellen eines Projektes (siehe A.2.2). Zunächst werden die Daten zu dem

Projekt editieren

Titel

Finanzierungslimit €

Kategorie ☐ Health & Wellness ☐ Art & Creative Works
☐ Education ☒ Tech & Innovation

Vorgänger ☒ Ubuntu Touch
☐ Ubuntu Touch Pro
☐ Kein Vorgänger

Beschreibung

Aktualisieren

Abbildung 4: Das Editieren eines Projekts (ID: edit_project)

ausgewählten Projekt aus der Datenbank geladen und anschließend angezeigt (siehe Abbildung 4). Folgende Bedingungen müssen beachtet werden sobald der Button „Aktualisieren“ angeklickt wurde:

- Ein Projekt darf **nur** vom Ersteller editiert werden und auch nur, wenn der Status „offen“ ist.
- Der Titel muss eine Länge zwischen 1 und 30 Zeichen haben.
- Das Finanzierungslimit muss eine Zahl sein. Zudem kann man nur das Limit erhöhen: Das neue Limit muss größer (oder gleich) als das vorherige Finanzierungslimit sein.
- Es muss eine Kategorie ausgewählt sein.
- Es muss eine Option für das Vorgänger-Projekt (oder die Option „Kein Vorgänger“) ausgewählt sein.
- Im Allgemeinen Fehler mit der Datenbank.

Klickt man auf „Aktualisieren“ und werden die Anforderungen erfüllt, so soll die Datenbank aktualisiert und der Benutzer auf die Detailseite des Projekts (A.2.3) weitergeleitet werden. Ansonsten wird eine Fehlermeldung ausgegeben; die Details werden dann auch nicht aktualisiert.

A.2.5 Projekt Spende (ID: new_project_fund)

Auf dieser Seite (siehe Abbildung 5) kann der Benutzer ein ausgewähltes Projekt finanziell durch eine Spende unterstützen. Zunächst wird der Titel des Projekts angezeigt. Der Benutzer kann über ein Textfeld einen numerischen Betrag eingeben, um das Projekt zu unterstützen. Zusätzlich kann er über eine Checkbox angeben, ob die Spende anonymisiert erfolgen soll. Über den Button „Spenden“ bestätigt der Nutzer die Eingabe. Folgendes gilt es zu beachten:

- Zunächst wird überprüft, ob der Status des Projekts auch offen ist. Nur offene Projekte sind unterstützbar.
- Der eingegebene Wert für die Spende muss ein numerischer Wert und größer als 0 sein.
- Es soll überprüft werden, ob der Spendenbetrag nicht das Guthaben des Benutzers überschreitet.
- Man darf eine Spende für ein Projekt nur dann abgeben, falls man dieses zuvor nicht unterstützt hat. Mehrfaches Spenden bezüglich eines Projekts ist nicht möglich.

Sollte eine der erwähnten Bedingungen nicht erfüllt sein, so soll eine entsprechende Fehlermeldung erscheinen und der Vorgang abgebrochen werden.

Falls sämtliche Anforderungen erfüllt werden, so sollen folgende Aktionen durchgeführt werden:

- Der Spender wird mit dem entsprechendem Spendenbetrag in die Datenbank eingetragen.
- Falls der Benutzer anonym spendet, dann soll dies auch in der Datenbank vermerkt werden.
- Das Guthaben auf dem Konto des Benutzers wird um den entsprechenden Spendenbetrag verringert.
- Am Ende soll noch sichergestellt werden, dass der Status von „offen“ auf „geschlossen“ aktualisiert wird, sofern das Finanzierungslimit durch die aktuelle Spende erreicht wurde.
- Der Benutzer soll anschließend auf die Detailseite des Projekts (A.2.3) weitergeleitet werden.

A.2.6 Benutzerprofil (ID: view_profile)

Das Benutzerprofil ist in drei Abschnitten aufgeteilt (siehe Abbildung 6):

Kopfbereich Im Kopfbereich sollen die Informationen des entsprechenden Benutzers aus der Datenbank gelesen und angezeigt werden. Die E-Mail-Adresse sollte zur Ermittlung der korrekten Daten verwendet werden. Angezeigt werden sollen folgende Informationen: E-Mail-Adresse, Benutzername, und die Anzahl erstellter sowie unterstützter (egal ob anonym oder öffentlich) Projekte.

Projekt Spende

http://projectFunder.uni-due.de/new_project_fund?kennung=1

Ubuntu Touch Pro

Spendenbetrag (€): 899,99

☐ Anonym spenden?

Spenden

Abbildung 5: Spendeseite (ID: new_project_fund)

Erstellte Projekte In diesem Abschnitt sollen alle Projekte angezeigt werden, die vom ausgewählten Benutzer erstellt wurden. Für jedes Projekt sollen das Icon der Kategorie, Titel, die aktuelle Spendensumme sowie der Status („offen“ oder „geschlossen“) angezeigt werden.

Wird auf den Titel eines Projekts geklickt, so wird man auf die entsprechende Detailseite weitergeleitet (A.2.3).

Unterstützte Projekte In diesem Abschnitt werden alle Projekte angezeigt, die vom ausgewählten Benutzer öffentlich unterstützt wurden. Anonyme Spenden sollen also nicht berücksichtigt werden. Angezeigt werden jeweils das Icon der Kategorie, Titel, Finanzierungslimit, der Status und zu guter Letzt der vom Benutzer gespendete Betrag.

Wird auf den Titel eines Projekts geklickt, so wird man auf die entsprechende Detailseite weitergeleitet (A.2.3).

A.2.7 Kommentare hinzufügen (ID: new_comment)

Auf dieser Seite (siehe Abbildung 7) kann der Benutzer einen Kommentar zum ausgewählten Projekt erstellen. Der Titel des Projektes wird als Information angezeigt. Die Eingabe des Kommentars erfolgt über eine Textarea. Über eine Checkbox kann weiterhin angegeben werden, ob der Kommentar anonym gespeichert wird.

Sobald der Benutzer auf den Button „Kommentar hinzufügen“ klickt, soll der entsprechende Kommentar in die Datenbank eingefügt werden. Folgendes gilt es zu beachten:

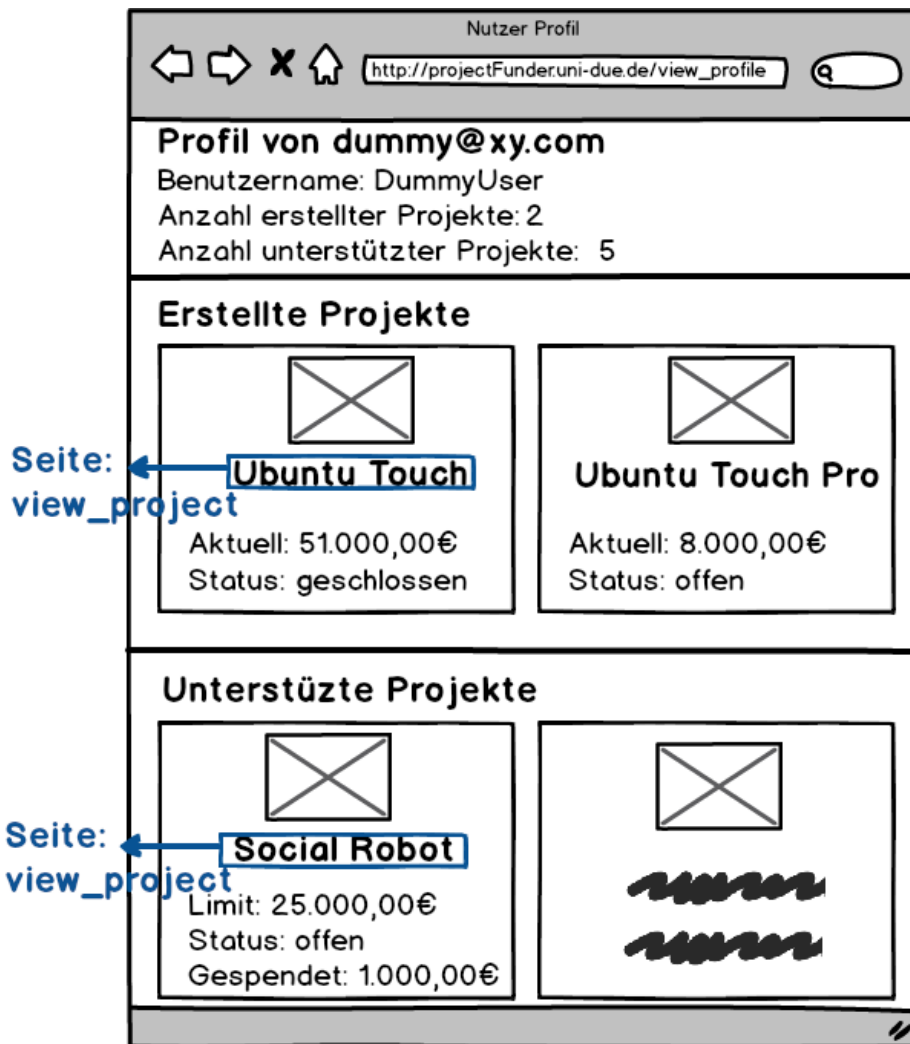


Abbildung 6: Das Profil vom Benutzer (ID: view_profile)

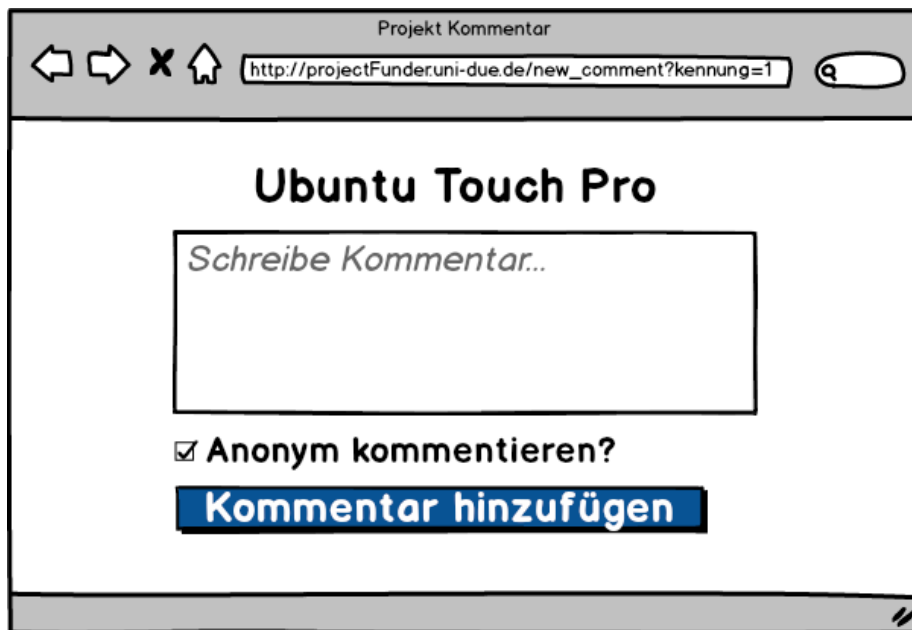


Abbildung 7: Hinzufügen eines Kommentars zu einem Projekt (ID: new_comment)

- Der Kommentar darf nicht leer sein. In dem Fall soll eine Fehlermeldung ausgegeben und der Vorgang abgebrochen werden.
- Wird ein Kommentar erfolgreich erstellt, so soll man auf die Detailseite des Projekts (A.2.3) weitergeleitet werden.

A.2.8 Bonusaufgabe (1 Punkt)

Die folgende Aufgabe ist optional. Durch die Bearbeitung kann ein weiterer Punkt gesammelt werden.

Suchen (ID: search)

Eine Suchfunktion darf in unserer Web-Applikation natürlich nicht fehlen. Auf der neuen Seite hat der Benutzer die Möglichkeit, Projekte anhand eines Titels zu suchen (siehe Abbildung 8).


- Klickt der Benutzer auf den Button „Suchen“, so sollen Projekte gefunden werden, deren Titel mit der angegebenen Suchanfrage beginnt. Groß- und Kleinschreibung können ignoriert werden. Die Suchoperation muss auf Datenbankebene erfolgen!
- Für jedes gefundene Projekt sollen das Icon der Kategorie, der Titel, der Benutzername des Erstellers, die aktuelle Spendensumme, und der Status („offen“ oder „geschlossen“) angezeigt werden.

Projekt Suche


http://projectFunder.uni-due.de/search

Titel:


Suchergebnisse



Ubuntu Touch Pro
 von: DummyUser
 Aktuell: 8.000,00€
 Status: offen



Ubuntu Touch Light
 von: DummyUser
 Aktuell: 1.000,00€
 Status: offen



Ubuntu Touch
 von: DummyUser
 Aktuell: 51.000,00€
 Status: geschlossen

Abbildung 8: Bonusaufgabe: Suchen

- Falls die Suchanfrage leer ist oder zu keinen Ergebnissen führt, so soll stattdessen eine entsprechende Meldung angezeigt werden.

Beispiel: Wenn der Benutzer nach „Ubuntu Touch“ sucht, so können folgende Projekte zurückgegeben werden: „Ubuntu Touch“, „Ubuntu Touch Light“ und „Ubuntu Touch Pro“.

Projekte, deren Titel nicht mit „Ubuntu Touch“ beginnt, sollen nicht angezeigt werden, z.B. „Ubuntu Tablet“ oder „Windows Touch“.

B Beschreibung der IMDB-Datenbank

Für die SQL-Aufgaben ist eine Beispieldatenbank mit Filmdaten vorgegeben. Die Datenbank IMDB solltet Ihr in der ersten Woche unter einem lokalen Alias bereits katalogisiert haben.

Der relevante Ausschnitt des Schemas der Datenbank ist im Anschluss beschrieben. Ihr könnt den Aufbau der Tabellen auch direkt von der Datenbank erfragen. Die Daten der Tabellen geben teilweise kein vollständiges Bild der Welt wieder, sollen aber für unsere Zwecke genügen.

B.1 UML-Diagramm der IMDB-Datenbank

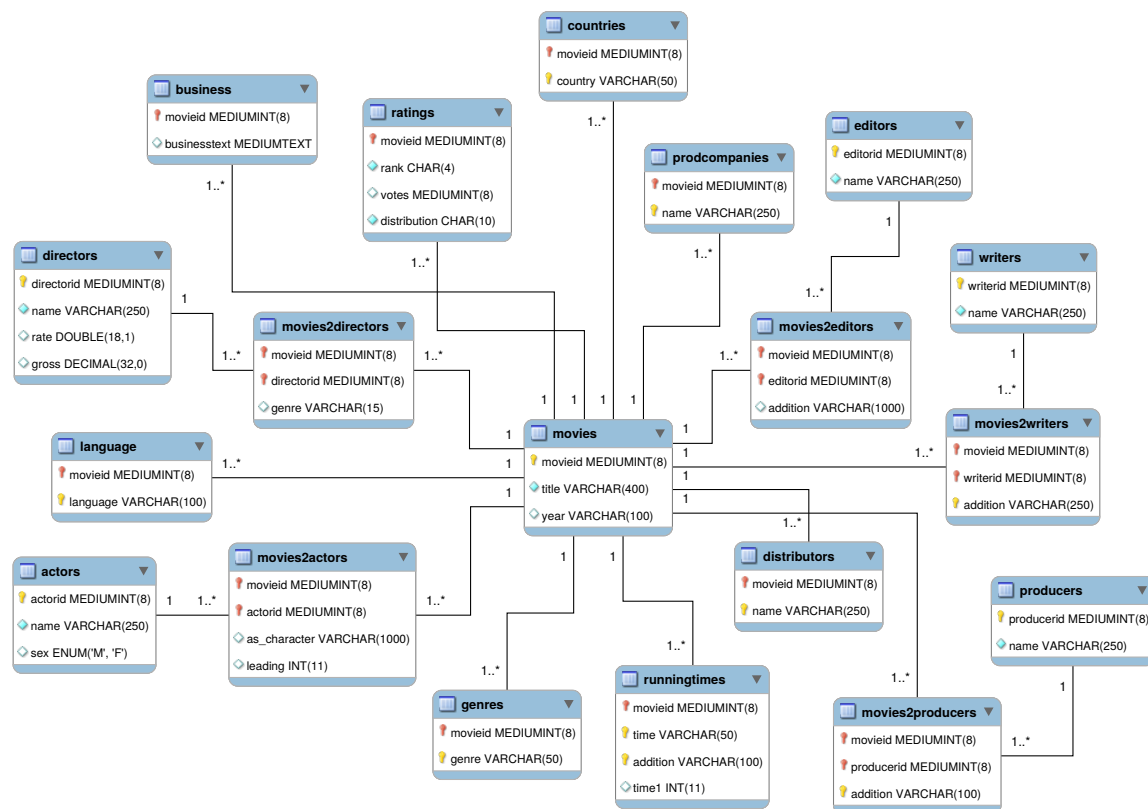


Abbildung 9: UML-Diagramm der IMDB-Datenbank