



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Digitale Videothek

Bager Bingöl | 2239338

Irena Becker | 2238833

Betreuer/in: Prof. Dr. -Ing. Katrin Wolf

27. Januar 17

Relationale Datenbanken

Studiengang Media Systems (B.Sc.)

Hochschule für angewandte Wissenschaften Hamburg /
Hamburg University of Applied Sciences

Department Medientechnik

Fakultät Design, Medien und Information

Inhalt

1. Vorstellung der Projektidee	3
1.1 User Stories	3
1.2 Motivation	4
2. Mockup der Visualisierung des Frontends	4
2.1 Startseite	4
2.2 Filme von A-Z	5
2.3 Auswertungen	6
3. Informationen zu den Daten	6
4. Chen-Diagramm.....	7
4.1 Chen-Diagramm zur Digitalen Videothek	7
5. Crow's Foot Diagramm	8
5.1 Crow's Foot Diagramm zur Digitalen Videothek	8
6. Normalisierung der Tabellen	10
6.1 Erste Normalform	10
6.2 Zweite Normalform	10
6.3 Dritte Normalform	11
7. Implementierung der Datenbank und SQL-Abfragen.....	12
7.1 Datenbankimplementierung	12
7.2 SQL-Abfragen	13
8. Implementierung des Frontends.....	16
9. Anbindung der Datenbank.....	16
10. Finale Umsetzung	17
11. Arbeitsaufteilung	18
12. Fazit.....	18

1. Vorstellung der Projektidee

Zu Beginn der Veranstaltung *Relationale Datenbanken* überlegten wir die Essgewohnheiten der Studierenden und Angestellten an unserem Campus zu analysieren und auszuwerten. Jeder Speise sollte ein Gesundheitswert zugeordnet werden. Somit wären Aussagen über die allgemeinen Präferenzen und Einstellungen zu einer bewussten Ernährung möglich gewesen. Wir traten in Kontakt mit den Leitern der Mensa und mussten leider feststellen, dass es uns nicht möglich sein würde an die Daten zu kommen, da diese streng vertraulich behandelt werden.

Da uns Filmdaten im Netz in großen Mengen zur Verfügung standen und der Zugang zu solchen Daten sehr unkompliziert ist, entschieden wir uns alternativ für die Umsetzung einer digitalen Videothek. Die Anforderung an unsere Videothek schrieben wir in Form von User Stories auf.

1.1 User Stories

- Als Nutzer möchte ich eine direkte Übersicht aller Filme in alphabetischer Reihenfolge haben.
- Als Nutzer möchte ich zu einem Film, den Titel, das Genre, das Erscheinungsjahr, die Laufzeit, den Regisseur, das Produktionsland und die Bewertung sehen.
- Als Nutzer möchte ich zu einem Film das Poster und den Trailer sehen können.
- Als Nutzer möchte ich sehen, wie viele Filme zu einem Initialbuchstaben gefunden wurden
- Als Nutzer möchte ich Filme nach Titel und/oder Genre und/oder Regisseur suchen können.
- Als Nutzer möchte ich die Genres mit durchschnittlicher Bewertung aufgelistet haben.
- Als Nutzer möchte ich die Regisseure mit durchschnittlicher Bewertung aufgelistet haben.
- Als Nutzer möchte ich die Erscheinungsjahre mit durchschnittlicher Bewertung aufgelistet haben.
- Als Nutzer möchte ich wissen, welcher Film die längste Laufzeit hat.
- Als Nutzer möchte ich eine übersichtliche Benutzeroberfläche bedienen.

1.2 Motivation

Zum einen war uns wichtig den Umgang mit großen Datenmengen und Datenbanken zu erlernen. Beide Projektteilnehmer hatten zu Beginn der Veranstaltung keine Kenntnisse über die Realisierung und Anbindung von Datenbanken.

Zum anderen entschieden wir uns für eine Filmdatenbank, da wir die Filmsuche anhand verschiedener Kriterien erleichtern wollten. Durch die Einpflegung weiterer Daten, zum Beispiel Daten von unbekannten Filmen, ermöglichen wir diesen Filmen den eigenen Bekanntschaftsgrad zu erweitern und den Filmerfolg zu steigern. Nutzer unserer Webseite können so Filme finden, die ihnen noch nicht bekannt sind und die Vorschläge aus unserer Datenbank tragen zur Unterhaltung des Nutzers bei.

2. Mockup der Visualisierung des Frontends

2.1 Startseite

Die Startseite der Homepage soll hauptsächlich der schnellen Suche eines Filmes anhand von verschiedenen Kriterien dienen. Die Anzeige der Ergebnisse wird in Listenform unter dem Label der Suche dargestellt. Von dieser Seite kommt man über das Menü in der oberen Leiste auf alle weiteren Seiten. Das Projekt wird in der oberen Hälfte der Seite kurz erklärt.

Home	Filme von A-Z	Auswertung
------	---------------	------------

Titel und Projektbeschreibung

Suche:	Kriterium1	Kriterium2	Kriterium3	Button
--------	------------	------------	------------	--------

von Irena Becker und Bager Bingöl

Abbildung: Mockup der Startseite der Homepage, eigene Graphik

2.2 Filme von A-Z

Auf der Seite *Filme von A-Z* findet man Tabs mit allen Buchstaben des Alphabets. Klickt man auf einen Buchstaben werden alle Filme, die mit dem ausgewählten Buchstaben beginnen, in einer Liste unterhalb der Buchstabenleiste angezeigt. Die Anzahl der Ergebnisse wird vor der Liste ausgegeben. Die Informationen zu den Filmen enthalten Titel, Genre, Erscheinungsjahr, Produktionsland, Dauer, Bewertung, Regisseur und das Poster. Klickt der Nutzer auf das Bild des jeweiligen Filmposters, wird der Trailer zum Film in einem weiteren Tab des Browsers geöffnet. Über das obere Menü gelangt man wieder auf alle weiteren Seiten.









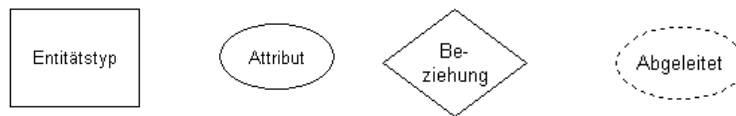
<div> Home Filme von A-Z Auswertung </div>							
<div> A B C D E F G H I J K L M N O P Q R S T U V W X Y Z </div>							
Anzahl der Suchergebnisse: 8							
Poster	Filmtitel	Genre	Erscheinungsjahr	Land	Dauer	Bewertung	Regisseur
	Matrix	Action	1999	USA	144	8,7	Lilly Wachowski
	Matrix	Action	1999	USA	144	8,7	Lilly Wachowski
	Matrix	Action	1999	USA	144	8,7	Lilly Wachowski
	Matrix	Action	1999	USA	144	8,7	Lilly Wachowski
	Matrix	Action	1999	USA	144	8,7	Lilly Wachowski
	Matrix	Action	1999	USA	144	8,7	Lilly Wachowski
	Matrix	Action	1999	USA	144	8,7	Lilly Wachowski
	Matrix	Action	1999	USA	144	8,7	Lilly Wachowski

Abbildung: Mockup der Seite *Filme von A-Z*, eigene Graphik

4. Chen-Diagramm

Das Chen-Diagramm ist eine Methode zur graphischen Darstellung der Beziehungen zwischen Entitäten. Dieses Diagramm wurde nach dem Informatiker Peter Chen benannt und wurde 1976, zusammen mit dem Entity- Relationship- Modell, zur Darstellung von Datenmodellen eingeführt.

Allgemein werden die Elemente eines Chen-Diagramm folgendermaßen dargestellt:



Abbildungen: Chen-Notationen, Quelle: https://de.wikipedia.org/wiki/Chen-Notation#/media/File:Darstellung_chen_neu.gif, aufgerufen am 25.01.2017

Hinzu kommen die Kardinalitäten, Mengenangaben, die für jeden Beziehungstyp festgelegt werden. Schlüsselattribute werden durch Unterstreichung gekennzeichnet.

4.1 Chen-Diagramm zur Digitalen Videothek

Unsere Datenbank besteht aus fünf Entitäten, mit jeweils einigen Attributen. Beziehungen verbinden die Entitäten jeweils untereinander.

Filme haben die Attribute Erscheinungsjahr, deutscher Titel (Primärschlüssel), Land, Laufzeit und Bewertung. Die Attribute Bild_Id, Regisseur_Id, Trailer_Id und Genre_Id sind Sekundärschlüssel und weisen auf weitere Entitäten.

Jeder Film aus der Datenbank hat ein Bild, einen Trailer und ein Genre und wurde von einem Regisseur geleitet. Die Kardinalitäten unserer Datenstrukturen lauten:

- N Filme : 1 Regisseur (Ein Regisseur kann mehrere Filme geleitet haben, ein Film hat immer nur einen Regisseur)
- N Filme : 1 Genre (Mehrere Filme können das gleiche Genre haben, ein Film kann immer nur ein Genre haben)
- 1 Film : 1 Bild (Ein Film hat jeweils nur ein Bild)
- 1 Film : 1 Trailer (Ein Film hat jeweils nur einen Trailer)

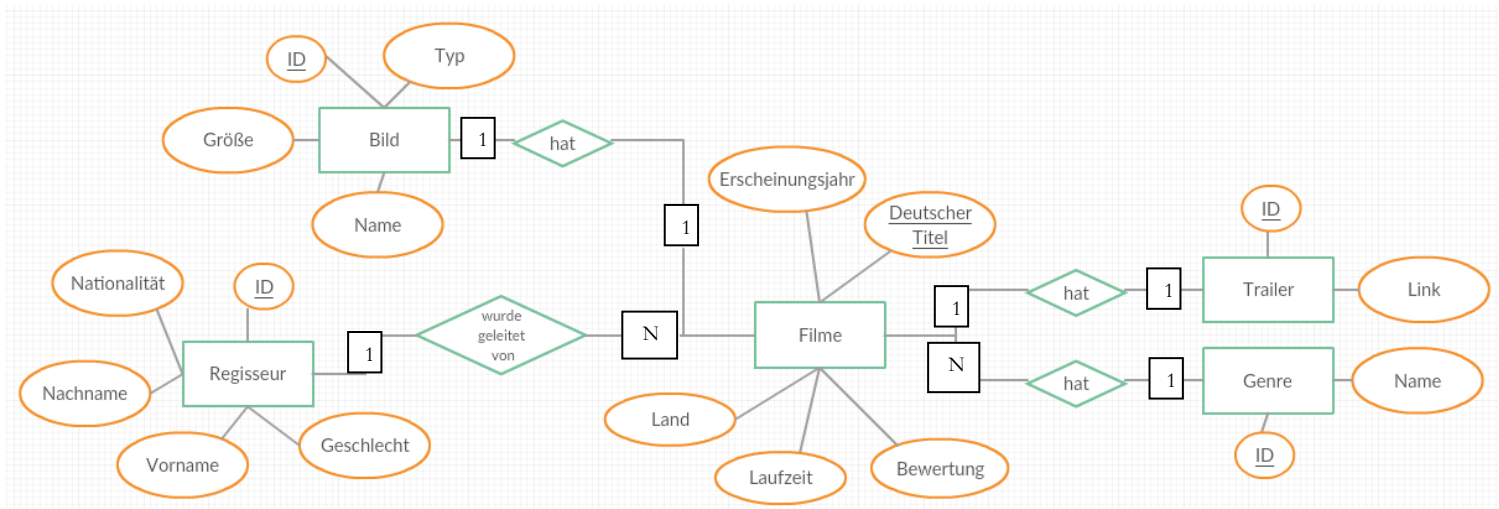


Abbildung: Chen-Diagramm der Strukturen der Filmdatenbank, eigene Graphik

5. Crow's Foot Diagramm

Das Crow's Foot Diagramm ist eine graphische Notation zur semantischen Datenmodellierung. Wie das Chen Diagramm handelt es sich um ein Entity-Relationship Modell, das die Beziehungen der Entitäten zueinander visualisiert, jedoch handelt es sich hierbei um ein vereinfachteres Modell. Das Diagramm wurde nach James Martin, Bachmann und Odell benannt.

Allgemein wird das Crow's Foot Diagramm folgendermaßen dargestellt:

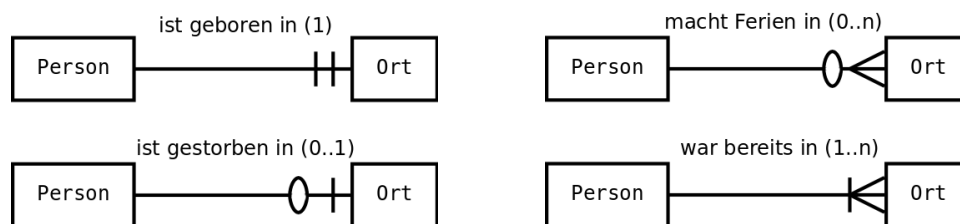


Abbildung: Kräbenfußnotationen, Quelle: <https://de.wikipedia.org/wiki/MartinNotation#/media/File:MartinOdell.png>, aufgerufen 25.01.2017

5.1 Crow's Foot Diagramm zur Digitalen Videothek

Wie das Chen Diagramm visualisiert das Crow's Foot Diagramm, wie die Entitäten zueinanderstehen. Jeder Film hat ein Bild, einen Trailer, ein Genre und ein Regisseur hat die Produktion zu einem Film geleitet. Allerdings können mehrere Filme den gleichen Regisseur haben oder auch aus dem gleichen Genre stammen. Bei den Bil-

dern und Trailern ist die Beziehung eindeutig und total. Als Schlüsselattribute haben alle Sekundärschlüssel numerische IDs, das primäre Schlüsselattribut in der Filmdatenbank ist der deutsche Titel.

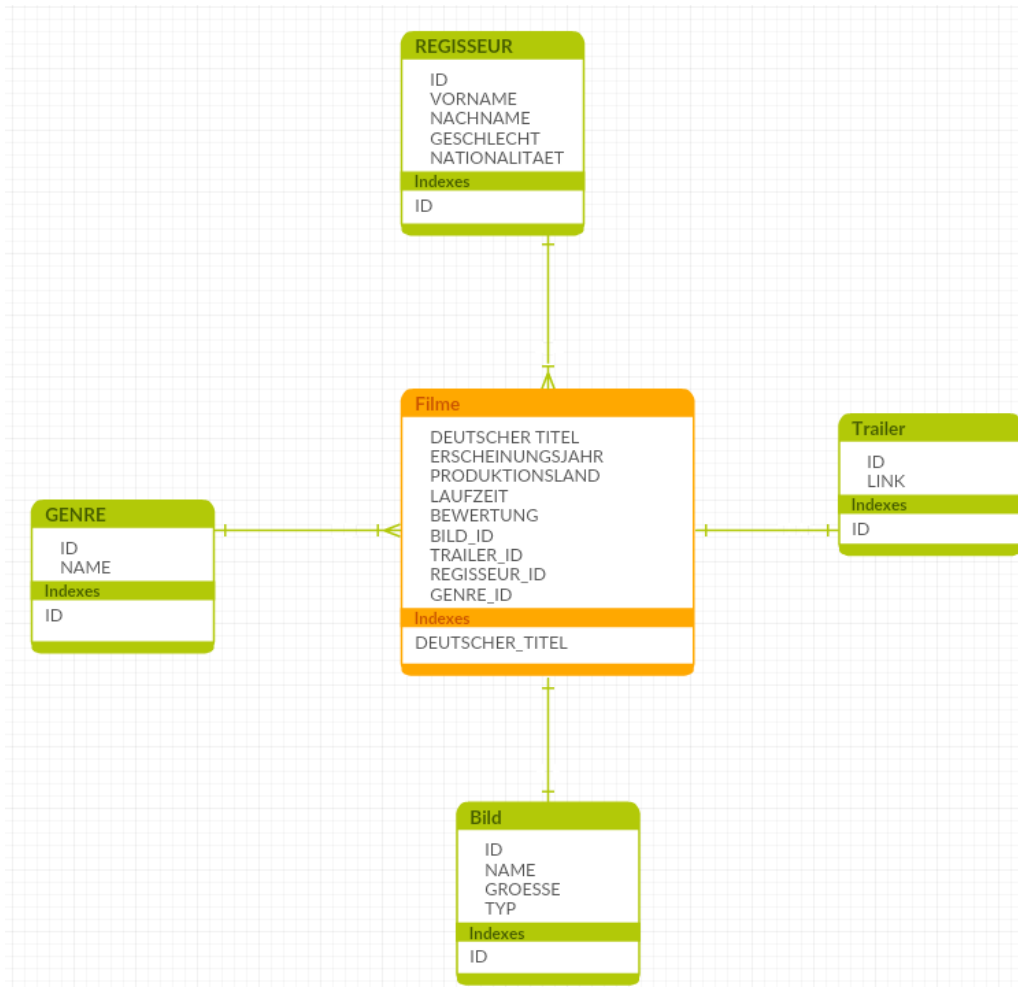


Abbildung: Crow's Foot Diagramm der Strukturen der Filmdatenbank, eigene Graphik

6. Normalisierung der Tabellen

	A	B	C	D	E	F	G	H	I
1	DEUTSCHER TITEL	GENRE	ERSCHEINUNGSJAHR	PRODUKTIONSLAND	LAUFZEIT [MIN]	BEWERTUNG	BILDDATEI	TRAILER	REGISSEUR
2	Alles steht Kopf	Animation	2015	Vereinigte Staaten	95	8,2	n1.jpg	https://www.youtube.com/watch?v=K107_03D8k0	Tomas Alfredson
3	Bowling for Columbine	Dokumentation	2002	Vereinigte Staaten	120	8,0	n2.jpg	https://www.youtube.com/watch?v=K107_03D8k0	Roger Allers
4	Brazil	Komoedie	1985	Vereinigtes Koenigreich	130	8,0	n3.jpg	https://www.youtube.com/watch?v=K107_03D8k0	Darren Aronofsky
5	City of God	Drama	2002	Brasilien	130	8,7	n4.jpg	https://www.youtube.com/watch?v=K107_03D8k0	Luc Besson
6	Das Fest	Drama	1998	Daenemark	106	8,1	n5.jpg	https://www.youtube.com/watch?v=K107_03D8k0	Tim Burton
7	Der Koenig der Loewen	Animation	1994	Vereinigte Staaten	89	8,5	n6.jpg	https://www.youtube.com/watch?v=K107_03D8k0	Damien Chazelle

Abbildung: Tabelle vor der Normalisierung

6.1 Erste Normalform

- 1) Zunächst gilt es die nichtatomaren Werte aufzutrennen:

REGISSEUR_NAME wird zu REGISSEUR_VORNAME
und REGISSEUR_NACHNAME

- 2) Ein eindeutiger Primärschlüssel wird bestimmt:

DEUTSCHER_TITEL

- 3) Es sind keine Wiederholungsgruppen vorhanden

6.2 Zweite Normalform

- 1) Die Fremdschlüssel werden festgelegt:

REGISSEUR, BILD, TRAILER, GENRE

- 2) Nicht-Schlüsselattribute, die nicht voll funktional vom zusammengesetzten Primärschlüssel abhängen, werden in eine neue Tabelle ausgelagert:

- REGISSEUR enthält REGISSEUR_ID, REGISSEUR_NAME, REGISSEUR_VORNAME, REGISSEUR_NATIONALITÄT, REGISSEUR_GESCHLECHT
- BILD enthält BILD_ID, DATEINAME, DATEIGROESSE, TYP
- TRAILER enthält TRAILER_ID, LINK
- GENRE enthält GENRE_ID, GENRE_NAME

6.3 Dritte Normalform

Nicht-Schlüsselattribute sind nicht von anderen NSA abhängig. Es existiert somit keine transitive Abhängigkeit der NSA vom Primärschlüssel.

	A	B	C	D	E	F	G	H	I
1	DEUTSCHER TITEL	GENRE	ERSCHEINUNGSJAHR	PRODUKTIONSLAND	LAUFZEIT	BEWERTUNG	BILD ID	Trailer ID	REGISSEUR ID
2	Alles steht Kopf	Animation	2015	Vereinigte Staaten	95	8,2	1	1	11
3	Bowling for Columbine	Dokumentation	2002	Vereinigte Staaten	120	8,0	2	2	27
4	Brazil	Komoedie	1985	Vereinigstes Koenigreich	130	8,0	3	3	15
5	City of God	Drama	2002	Brasilien	130	8,7	4	4	25
6	Das Fest	Drama	1998	Daenemark	106	8,1	5	5	39
7	Der Koenig der Loewen	Animation	1994	Vereinigte Staaten	89	8,5	6	6	2
8	Der Pate	Drama	1972	Vereinigte Staaten	175	9,2	7	7	9
9	Die Jagd	Drama	2012	Daenemark	120	8,3	8	8	39
10	Die sieben Samurai	Action	1954	Japan	159	8,7	9	9	21
11	Django Unchained	Western	2012	Vereinigte Staaten	165	8,5	10	10	37
12	Fight Club	Thriller	1999	Vereinigte Staaten	139	8,8	11	11	14
13	Free to Run	Sport	2016	Belgien	90	7,7	12	12	28
14	Gran Torino	Drama	2008	Vereinigte Staaten	116	8,2	13	13	12

	A	B	C	D	E
1	REGISSEUR ID	NACHNAME	VORNAME	NATIONALITAET	GESCHLECHT
2	1	Alfredson	Tomas	Schweden	m
3	2	Allers	Roger	Vereinigte Staaten	m
4	3	Aronofsky	Darren	Vereinigte Staaten	m
5	4	Besson	Luc	Frankreich	m
6	5	Burton	Tim	Vereinigte Staaten	m
7	6	Cameron	James	Kanada	m
8	7	Chazelle	Damien	Vereinigte Staaten	m
9	8	Coen	Ethan	Vereinigte Staaten	m
10	9	Coppola	Francis Ford	Vereinigte Staaten	m
11	10	Del Toro Gómez	Guillermo	Mexiko	m
12	11	Docter	Pete	Vereinigte Staaten	m
13	12	Eastwood	Clint	Vereinigte Staaten	m

TRAILER ID	LINK
1	https://www.youtube.com/watch?v=wWTQZkD5FAQ
2	https://www.youtube.com/watch?v=CgpyaUVivkA
3	https://www.youtube.com/watch?v=ZKPF8DA9_8
4	https://www.youtube.com/watch?v=dC7lnv9gY7k
5	https://www.youtube.com/watch?v=L8nSgxvBrhQ
6	https://www.youtube.com/watch?v=2oRNjgMgM_g
7	https://www.youtube.com/watch?v=yR3ntn7LEoY
8	https://www.youtube.com/watch?v=KfMyYyhRYjs
9	https://www.youtube.com/watch?v=zpxaNTVP9_c
10	https://www.youtube.com/watch?v=0RjoVdtE2nk
11	https://www.youtube.com/watch?v=G7tr7xcUCFA
12	https://www.youtube.com/watch?v=AhEWIAQ-CmE
13	https://www.youtube.com/watch?v=AZ4zF6RAQOw

	A	B	C	D
1	BILD ID	DATEINAME	GROESSE	DATEITYP
2	1	n1	34,9	JPG
3	2	n2	43,6	JPG
4	3	n3	70	JPG
5	4	n4	35	JPG
6	5	n5	53,4	JPG
7	6	n6	789	JPG
8	7	n7	76,9	JPG
9	8	n8	126	JPG
10	9	n9	40,1	JPG
11	10	n10	223	JPG
12	11	n11	18,7	JPG
13	12	n12	1760	JPG

Abbildungen: Tabellen nach der Normalisierung

7. Implementierung der Datenbank und SQL-Abfragen

7.1 Datenbankimplementierung

Die Datenbank wurde zunächst über die MySQL Workbench, wie in der Vorlesung besprochen, implementiert. Dazu wurde auf die fünf bereits existierenden .csv-Dateien zugegriffen.

```
USE abx393_filmDB;

LOAD DATA LOCAL
INFILE "Z:\\ViewProfile\\Dokumente\\Datenbank-0.1.csv"
INTO TABLE FILME
CHARACTER SET UTF8
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\n'
IGNORE 1 LINES ;
```

Abbildung: Implementierung der Datenbank mit MySQL

Da sich im Verlauf der PHP-Programmierung herausstellte, dass das Tool phpMyAdmin die Anbindung der Datenbank vereinfachen würde, entschieden wir uns dieses Tool für das weitere Projekt zu nutzen.

DEUTSCHER_TITEL	GENRE_ID	ERSCHINUNGSJAHR	PRODUKTIONSLAND	LAUFZEIT	BEWERTUNG	BILD_ID	TRAILER_ID	REGISSEUR_ID
Alles steht Kopf	1	2015	Vereinigte Staaten	95	8.20	1	1	11
Bowling for Columbine	5	2002	Vereinigte Staaten	120	8.00	2	2	27
Brazil	6	1985	Vereinigtes Königreich	130	8.00	3	3	15
City of God	4	2002	Brasilien	130	8.70	4	4	25
Das Fest	4	1998	Dänemark	106	8.10	5	5	39
Der König der Löwen	1	1994	Vereinigte Staaten	89	8.50	6	6	2
Der Pate	4	1972	Vereinigte Staaten	175	9.20	7	7	9
Die Jagd	4	2012	Dänemark	120	8.30	8	8	39
Die sieben Samurai	3	1954	Japan	159	8.70	9	9	21
Django Unchained	10	2012	Vereinigte Staaten	165	8.50	10	10	37
Fight Club	11	1999	Vereinigte Staaten	139	8.80	11	11	14
Free to Run	9	2016	Belgien	90	7.70	12	12	28
Gran Torino	4	2008	Vereinigte Staaten	116	8.20	13	13	12
Hero	3	2002	China	96	7.90	14	14	42
Inception	11	2010	Vereinigte Staaten	148	8.80	15	15	30
Inglourious Basterds	4	2009	Vereinigte Staaten	154	8.30	16	16	37

Abbildung: Screenshot des Tools phpMyAdmin

7.2 SQL-Abfragen

Zu den SQL Abfragen gehören dynamische und statische Abfragen. Zu den dynamischen Abfragen zählen die SQL-Abfragen, die der Nutzer direkt beeinflussen kann. Die statischen Abfragen haben wir im Vorhinein programmiert, die Ausgabe ändert sich somit nur, wenn sich die Daten in der Datenbank ändern, neue Daten hinzukommen oder Daten gelöscht werden.

- Statische Abfragen: Auswertungen (Seite Auswertungen)

Genres mit den jeweiligen durchschnittlichen Bewertungen:

```
$query = "SELECT GENRE.NAME as genre, AVG(FILME.BEWERTUNG) as rating
FROM GENRE, FILME
WHERE FILME.GENRE_ID = GENRE.GENRE_ID
GROUP BY GENRE.NAME asc
LIMIT 13;";
```

Regisseure mit den jeweiligen durchschnittlichen Bewertungen:

```
$query = "SELECT REGISSEUR.NACHNAME as reg, AVG(FILME.BEWERTUNG) as rat
FROM FILME, REGISSEUR
WHERE FILME.REGISSEUR_ID = REGISSEUR.REGISSEUR_ID
GROUP BY FILME.BEWERTUNG desc
LIMIT 10;";
```

Erscheinungsjahre mit den jeweiligen durchschnittlichen Bewertungen:

```
$query = "SELECT FILME.ERSCHEINUNGSJAHR as erjahr, AVG(FILME.BEWERTUNG) as rating
FROM FILME
GROUP BY FILME.BEWERTUNG desc
LIMIT 10;";
```

Der Film mit der längsten Laufzeit:

```
$query = "SELECT FILME.DEUTSCHER_TITEL as titel, MAX(FILME.LAUFZEIT) as playtime
FROM FILME
```

Filme nach Anfangsbuchstaben ausgeben:

```

45 function getlist($db) {
46     $list = '<div class="bs-example bs-example-tabs" role="tabpanel" data-example-id="toggleable-tabs">';
47     $tabs = '<ul id="myTab" class="nav nav-tabs" role="tablist">';
48     $listtabs = '<div id="myTabContent" class="tab-content"><div role="tabpanel" class="tab-pane fade in active" id="home" aria-labelledby="home-';
49         <div class="agile-news-table">
50             <div class="w3ls-news-result">
51                 <h4>Suchergebnisse : <span>0</span></h4>
52             </div>
53             <table id="table-breakpoint">
54                 <thead>
55                     <tr>
56                         <th>Poster</th>
57                         <th>Filmtitel</th>
58                         <th>Genre</th>
59                         <th>Erscheinungsjahr</th>
60                         <th>Land</th>
61                         <th>Dauer</th>
62                         <th>Bewertung</th>
63                         <th>Regisseur</th>
64                     </tr>
65                 </thead>
66             </table>
67         </div>
68     </div>;
69
70     $query = "SELECT
71         FILME.DEUTSCHER_TITEL as title,
72         GENRE.NAME as genre,
73         FILME.ERSCHEINUNGSAHR as year,
74         FILME.PRODUKTIONS LAND as country,
75         FILME.LAUFEIT as playtime,
76         FILME.BEWERTUNG as rating,
77         FILME.BILD_ID as pic_id,
78         FILME.TRAILER_ID as trailer_id,
79         FILME.REGISSEUR_ID as reg_id,
80         REGISSEUR.NACHNAME as nachname,
81         REGISSEUR.VORNAME as vorname,
82         TRAILER.ID as trailer,
83         BILD.DATEINAME as poster
84     FROM FILME
85     INNER JOIN REGISSEUR ON FILME.REGISSEUR_ID = REGISSEUR.REGISSEUR_ID
86     INNER JOIN TRAILER ON FILME.TRAILER_ID = TRAILER.TRAILER_ID
87     INNER JOIN BILD ON FILME.BILD_ID = BILD.BILD_ID
88     INNER JOIN GENRE ON FILME.GENRE_ID = GENRE.GENRE_ID
89     ORDER BY FILME.DEUTSCHER_TITEL";
90
91     $result = $db->query($query) or die($db->error);
92
93     if ($result->num_rows > 0) {
94         while($row = $result->fetch_assoc()) {
95             $a = substr(strtolower($row["title"]),0,1);
96             $index = "";
97             if ($a != $index) {
98                 $movies[$a][] = $row;
99             }
100         }
101         return "Nothing found..";
102     }
103
104     foreach ($movies as $a => $movie_array) {
105         $count = count($movie_array);
106         $tabs .= '<li role="presentation">
107             <a href="#'.$a.'" role="tab" id="'.$a.'" data-toggle="tab" aria-controls="'.$a.'">'.strtoupper($a). '</a>';
108         </li>';
109     }
110     $listtabs .= '<<< EOF';
111
112     <div role="tabpanel" class="tab-pane fade" id="$a" aria-labelledby="$a-tab">
113         <div class="agile-news-table">
114             <div class="w3ls-news-result">
115                 <h4>Suchergebnisse : <span>$count</span></h4>
116             </div>
117             <table id="table-breakpoint1">
118                 <thead>
119                     <tr>
120                         <th>Poster</th>
121                         <th>Filmtitel</th>
122                         <th>Genre</th>
123                         <th>Erscheinungsjahr</th>
124                         <th>Land</th>
125                         <th>Dauer</th>
126                         <th>Bewertung</th>
127                         <th>Regisseur</th>
128                     </tr>
129                 </thead>
130             </table>
131         </div>
132
133         foreach ($movie_array as $movie) {
134             $title = $movie["title"];
135             $genre = $movie["genre"];
136             $year = $movie["year"];
137             $country = $movie["country"];
138             $playtime = $movie["playtime"];
139             $rating = $movie["rating"];
140             $reg_vor = $movie["vorname"];
141             $reg_nach = $movie["nachname"];
142             $trailer = $movie["trailer"];
143             $poster = $movie["poster"];
144
145             $listtabs .= '<<< EOF';
146             <tr>
147                 <td><a href="$trailer" target="_blank"></a></td>
148                 <td>$title</td>
149                 <td>$genre</td>
150                 <td>$year</td>
151                 <td>$country</td>
152                 <td>$playtime</td>
153                 <td>$rating</td>
154                 <td>$reg_vor $reg_nach</td>
155             </tr>
156         }
157     }
158 }

```

Abbildung: Code aus Klasse list.php, Markierung: Main-Query

- Dynamische Abfragen: Filmsuche mit komplexer Abfrage. Es kann nach Titel und / oder Genre und / oder Regisseur gesucht werden:

```

2
3 function getGenreOptions($db) {
4     $query = "";
5
6
7     $options = '<select class="domains valid" name="genre">';
8
9     $options .= '<option selected="selected" value="">Wähle Genre</option>';
10
11     $query = "SELECT * FROM GENRE ORDER BY NAME";
12     $result = $db->query($query) or die($db->error);
13
14     while ($row = $result->fetch_assoc()) {
15         $name = $row["NAME"];
16         $id = $row["GENRE_ID"];
17         $options .= '<option value="' . $id . '">' . $name . '</option>';
18     }
19
20     $options .= '</select>';
21     return $options;
22 }
23
24
25 function getRegOptions($db) {
26     $options = '<select class="domains valid" name="regisseur">';
27     $options .= '<option selected="selected" value="">Wähle Regisseur</option>';
28
29     $query = "SELECT * FROM REGISSEUR ORDER BY NACHNAME";
30     $result = $db->query($query) or die($db->error);
31
32     while ($row = $result->fetch_assoc()) {
33         $name = $row["NACHNAME"];
34         $vorname = $row["VORNAME"];
35         $id = $row["REGISSEUR_ID"];
36         $options .= '<option value="' . $id . '">' . $name . ', ' . $vorname . '</option>';
37     }
38
39     $options .= '</select>';
40     return $options;
41 }
42

```

Abbildungen: php-Methoden `getGenreOptions()` und `getRegOptions()`, aus der Klasse `list.functions.inc.php`. Hier werden die optionalen Suchwerte Genre und Regisseur an die Main-Query weitergeleitet

```

154 <?php
155 if(isset($_POST["title"])&& isset($_POST["genre"])&& isset($_POST["regisseur"])){
156     $title = $_POST["title"];
157     $genre = $_POST["genre"];
158     $regisseur = $_POST["regisseur"];
159     $query = "SELECT FILME.DEUTSCHER_TITEL as title,
160             FILME.ERSCHEINUNGSJAHR as year,
161             FILME.PRODUKTIONSLAND as country,
162             FILME.LAUFZEIT as playtime,
163             FILME.BEWERTUNG as rating,
164             FILME.GENRE_ID as genre_id,
165             FILME.BILD_ID as pic_id,
166             FILME.TRAILER_ID as trailer_id,
167             FILME.REGISSEUR_ID as reg_id,
168             REGISSEUR.NACHNAME as nachname,
169             REGISSEUR.VORNAME as vorname,
170             TRAILER.LINK as trailer,
171             BILD.DATEINAME as poster,
172             GENRE.NAME as genre
173     FROM FILME
174             INNER JOIN REGISSEUR ON FILME.REGISSEUR_ID = REGISSEUR.REGISSEUR_ID
175             INNER JOIN TRAILER ON FILME.TRAILER_ID = TRAILER.TRAILER_ID
176             INNER JOIN BILD ON FILME.BILD_ID = BILD.BILD_ID
177             INNER JOIN GENRE ON FILME.GENRE_ID = GENRE.GENRE_ID
178
179     WHERE
180             FILME.DEUTSCHER_TITEL LIKE '%$title%';
181     if (!empty($_POST["genre"]) && is_numeric($_POST["genre"])) {
182         $query .= "AND FILME.GENRE_ID = '$genre'";
183     }
184     if (!empty($_POST["regisseur"]) && is_numeric($_POST["regisseur"])) {
185         $query .= "AND FILME.REGISSEUR_ID = '$regisseur'";
186     }
187
188
189     $query .= "ORDER BY FILME.DEUTSCHER_TITEL";
190
191     $result = $db->query($query) or die($db->error);
192     $count = $result->num_rows;
193
194
195 }

```

Abbildung: die gekapselte Suchanfrage, Klasse `index.php`

8. Implementierung des Frontends

Bereits zu Beginn der Veranstaltung wurde durch Frau Wolf die Aufgabe erteilt das Frontend vorläufig zu implementieren. Da im Mittelpunkt der Veranstaltung die Implementierung der Datenbank stand, sahen wir uns zur Inspiration diverse Webseitenvorlagen an, die online erhältlich sind. Wir stießen auf eine Vorlage von w3layouts (<https://w3layouts.com/one-movies-entertainment-category-bootstrap-responsive-web-template/>) und änderten diese nach unseren Vorstellungen ab.

9. Anbindung der Datenbank

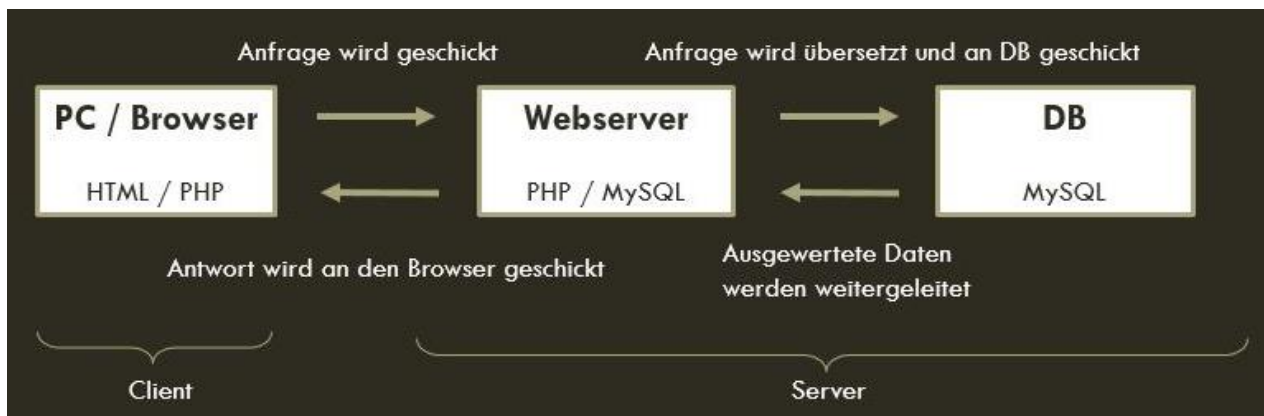


Abbildung: Anbindung der Datenbank, eigene Graphik

Die oben abgebildete Graphik stellt die Funktionsweise unserer Datenbankanbindung dar.

Zunächst wird dem Nutzer (Client) die Webseite im Browser angezeigt. Gibt der Nutzer nun Suchkriterien ein, wonach beispielsweise eine Liste mit passenden Filmen angezeigt werden soll, wird die Anfrage über den Webserver (HAW Server) an die Datenbank geschickt. Diese gibt die passenden Daten zurück an den Server. Die Antwort der Anfrage wird im nächsten Schritt vom Server an den Browser des Nutzers geschickt und dort visualisiert.

10. Finale Umsetzung

Unsere finale Umsetzung der Homepage ist unter <http://abx393.pstud0.mt.haw-hamburg.de/index.php> zu finden.



Abbildung: Startseite mit gekapselter Suchfunktion

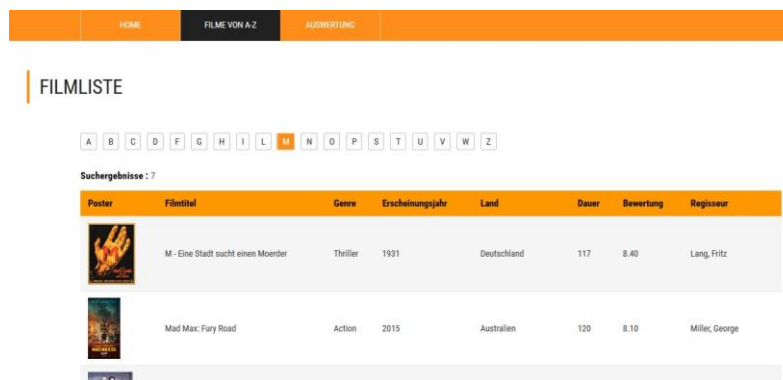


Abbildung: Filmliste mit alphabetischer Abfrage



Abbildung: Abfragen zu den durchschnittlichen Bewertungen u.a. nach Genre und Regisseur

11. Arbeitsaufteilung

Die Implementierung des Frontends, die Erstellung und Anbindung der Datenbank, die SQL-Abfragen, sowie die Erstellung der abschließenden Dokumentation sind zu gleichen Teilen von beiden Projektteilnehmern umgesetzt worden.

12. Fazit

Die erste eigene relationale Datenbank wurde von beiden Projektpartnern motiviert und interessiert realisiert. Bis zu diesem Zeitpunkt war die Anbindung einer Datenbank an ein digitales Frontend im Studium noch nicht thematisiert worden, dennoch war uns bewusst, dass Datenbanken ein essenzieller Bestandteil von Programmen und Applikationen sind. Deswegen war uns umso wichtiger eine Datenbank einmal selber zu implementieren um den Umgang mit Daten zu erlernen.

Die Teamarbeit hat sehr gut funktioniert. Dank des Lehrbeauftragten Herrn Wagener war es uns möglich auf unseren persönlichen Rechnern eine Anbindung der Entwicklungsumgebung Eclipse an den HAW Server herzustellen. So konnten wir gleichermaßen am Projekt arbeiten.

Da wir zu Beginn der Veranstaltung keinerlei Kenntnisse über PHP besaßen, sind wir auch an dieser Stelle Herrn Wagener für die spontane Hilfestellung bei der Programmierung der PHP-Klassen sehr dankbar.

Zusammenfassend lässt sich sagen, dass das Projekt die Vorlesung, die die theoretischen Aspekte der relationalen Datenbanken thematisierte, sehr gut ergänzte. Das Projekt war für uns eine gute Möglichkeit uns erstmals praktisch mit Datenbanken auseinanderzusetzen und von Fehlern und Problemen bei der Umsetzung zu lernen.