

Bericht Datenbanken

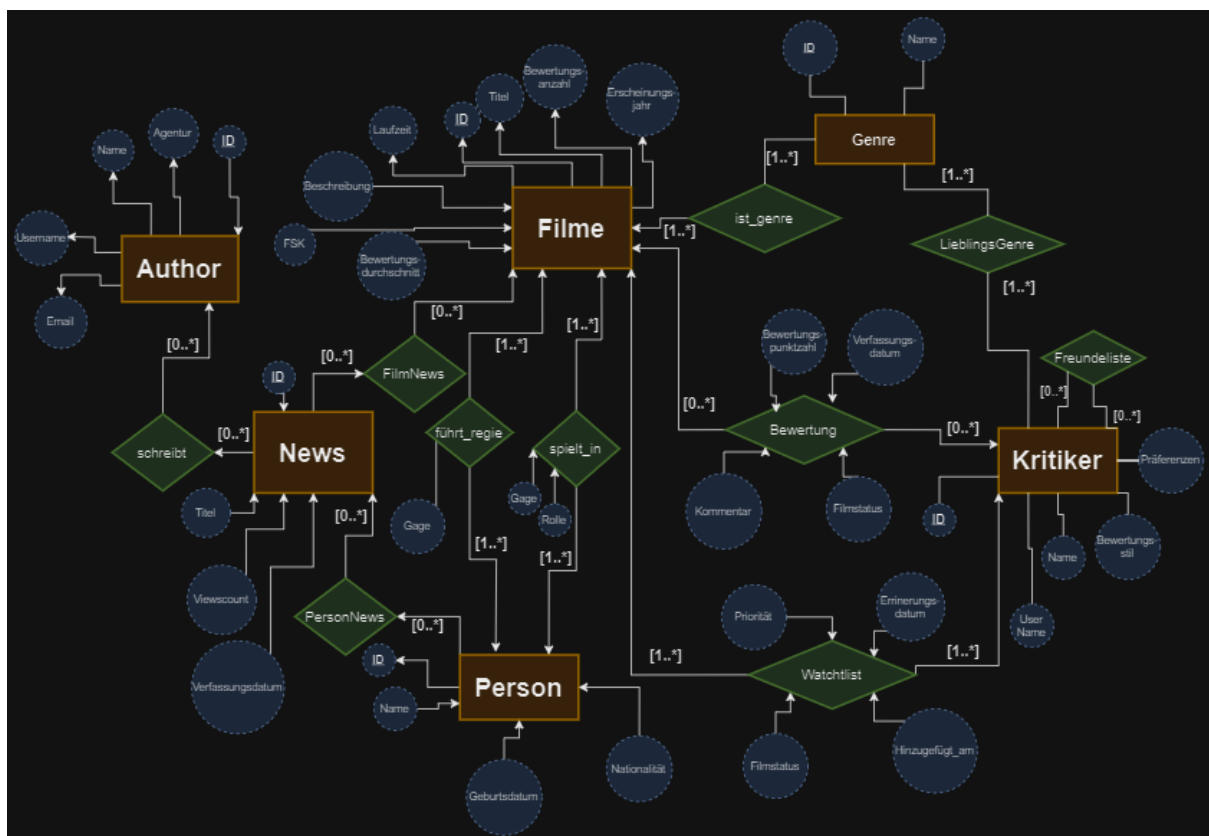
Team 5 (Stephane Sandevski 22 59 802)

1. Vorwort

Dies ist der Bericht, der als Dokumentation der Vierten Iteration für das Datenbankprojekt von Stephane Sandevski, teil des Team 5, dient. In dieser Dokumentation werden die getroffenen Entscheidungen sowie die Gründe für diese Entscheidungen ausführlich erläutert. Dafür werden das ER-Modell, das relationale Schema und die Evaluation weiter geführt.

Im Zuge unseres Projekts haben wir eine Datenbank für eine Filmkritik-Website geplant und umgesetzt. Dabei sollen Nutzer die Möglichkeit haben, Filme zu bewerten, Kritiken zu lesen und Filme zu suchen.

2.ER-Diagramm der vierten Iteration



In der vierten Iteration des ER Modells, habe ich mich dazu entschlossen die Attribute Beschreibung, welches den Film in kurzen Sätzen erläutert und FSK welche die Altersfreigabe angibt.

Dazu habe ich mich entschieden eine Relation zwischen Kritker und Film aufzunehmen namens Watchlist, diese soll Kritikern die Möglichkeit geben noch nicht gesehene Filme auf eine designierte Liste aufzuführen um sie so für einen Späteren Zeitpunkt anschauen zu können.

Zusätzlich habe ich die Option ermöglicht durch der Entität News, dass die neue Entität Author mithilfe der Relationen FilmNews und PersonNews zu diesen Entitäten Berichte verfassen kann.

Anfangs hatte ich die Idee, den Author und den Kritiker zu einer Entität Namens Journalist zu vereinigen. Habe mich aber dann doch dagegen entschlossen, weil dadurch das Problem entsteht dass jeder News verbreiten könnte, das wird verhindert dadurch das Author eine eigene Entität ist. Wodurch nur ausgewählte Personen News verfassen können.

Entität News:

Relation: PersonNews mit Person, FilmNews mit Film

Attribute: ID

Titel: Überschrift des Artikels

Viewcount: Wie oft dieser Artikel gelesen wurde

Verfassungsdatum: Wann der Artikel verfasst wurde

Entität Author:

Relation: schreibt mit News

Attribute:

ID

Agentur: über welcher Firma dieser Author tätig ist,

Name: Name des Authors

Username: Anzeige Name.

Watchlist ist eine Relation zwischen Kritiker und Film.

Attribute der Relation Watchlist:

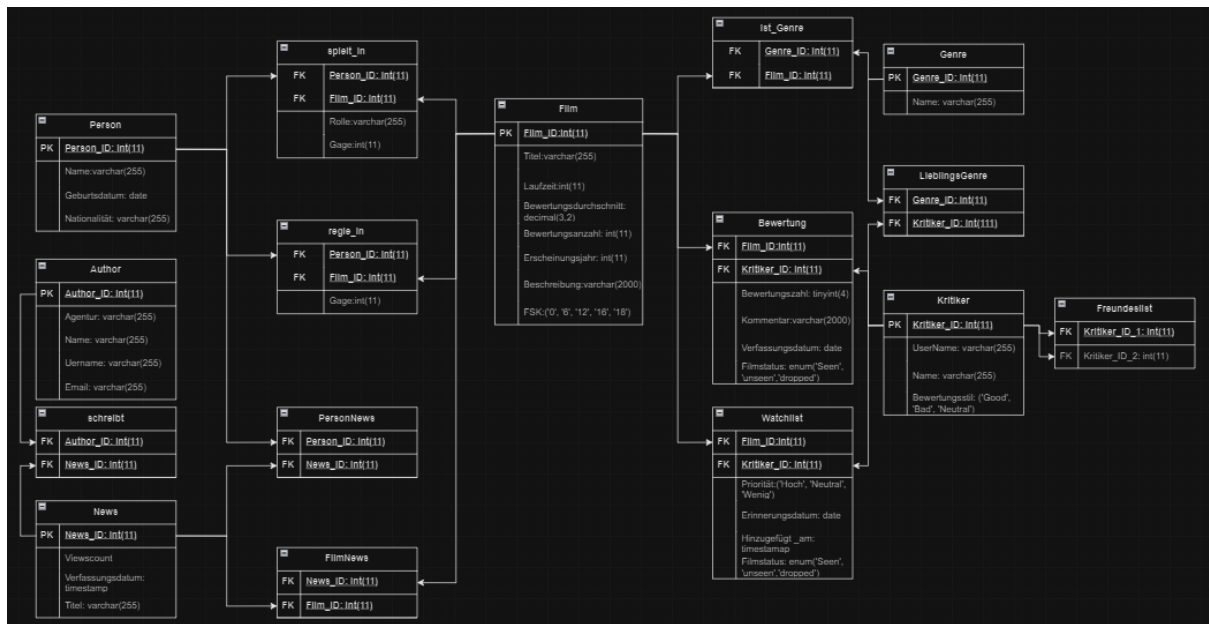
Priorität: Sagt an wie wichtig es ist den Film anschauen zu wollen.

Erinnerungsdatum: Ein Datum festlegen wann man erinnert werden möchte diese Film anzuschauen.

Hinzugefügt_am: Wann dieser Film hinzugefügt wurde.

Filmstatus: Ob man diesen Film gesehen hat oder nicht

3.Relationales Schema



Aus dem ER-Diagramm haben ich zusätzliche die Entitäten "Author" und "News" in das bestehende relationale Schema überführt. Dabei habe ich die Attribute der Entitäten als Spalten in den entsprechenden Tabellen erweitert und definiert.

Kardinalitäten:

[1..*]: Ein Kritiker kann mehrere Filme in der Watchlist haben. Und zudem kann ein Film in mehreren Watchlists enthalten sein.

[0..*]: Author muss keinen Artikel verfasst haben. Kann aber mehrere Artikel pro film oder Person verfassen.

4. Überprüfung der Normalisierung

1NF: Alle Attributwerte sind atomar.

Beispiel:

Die Tabelle **Author** enthält Attribute wie **Agentur**, **Name**, und **Email**, die jeweils nur einen einzelnen Wert pro Datensatz speichern.

2NF: Alle Nicht-Schlüsselattribute voll funktional abhängig voneinander.

Beispiel:

In der Tabelle **Watchlist** ist der Primärschlüssel die Kombination aus **Kritiker_ID** und **Film_ID**. Alle anderen Attribute (**Priorität**, **Erinnerungsdatum**, **Hinzugefügt_am**, **Filmstatus**) hängen vollständig von dieser Kombination ab.

3NF: Keine transitiven Abhängigkeiten zwischen den Attributen bestehen.

Beispiel:

In der Tabelle **News** hängt **Viewscount**, **Verfassungsdatum** und **Title** direkt von **News_ID** ab.

In der Tabelle **Author** hängt **Agentur**, **Name** und **Username** und **Email** direkt von **Author_ID** ab.

5. Überprüfung der Funktionalen Abhängigkeiten

5.1 Author:

Funktionale Abhängigkeiten: **Author_ID** -> **Agentur**, **Name**, **Email**.

Jeder Author wird eindeutig durch seine **Author_ID** identifiziert, und alle anderen Attribute hängen direkt davon ab

5.2 News:

Funktionale Abhängigkeiten: **News_ID** -> **Titel**, **Verfassungsdatum**, **Viewcount**.

Jeder Artikel wird eindeutig durch seine **News_ID** identifiziert, und alle anderen Attribute hängen direkt davon ab

5.3 Watchlist:

Funktionale Abhängigkeiten: (**Kritiker_ID**, **Film_ID**) -> **Priorität**, **Erinnerungsdatum**, **Hinzugefügt_am**, **Filmstatus**

Die Kombination aus **Kritiker_ID** und **Film_ID** identifiziert eindeutig jede Bewertung und deren Attribute.

5.4 Schreibt:

Funktionale Abhängigkeiten: (**Author_ID**, **News_ID**) -> { }

Die Kombination aus **Author_ID** und **News_ID** identifiziert eindeutig die Zuordnung von Autoren zu News.

5.5 PersonNews:

Funktionale Abhängigkeiten: (**Person_ID**, **News_ID**) -> { }

Die Kombination aus **Person_ID** und **News_ID** identifiziert eindeutig die Zuordnung von News zu Personen.

5.6 FilmNews:

Funktionale Abhängigkeiten: (**Film_ID**, **News_ID**) -> { }

Die Kombination aus **Film_ID** und **News_ID** identifiziert eindeutig die Zuordnung von News zu Filmen.