

## Dokumentation

### Integration von Netflix mit First.Fm

#### Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Allgemeines . . . . .	2
1.2	Formatierung . . . . .	2
<b>2</b>	<b>Pre-Integration Analyse</b>	<b>3</b>
<b>3</b>	<b>Vergleich der Schemata</b>	<b>3</b>
<b>4</b>	<b>Angleichung der Schemata</b>	<b>4</b>
4.1	Genre . . . . .	4
4.2	Comment . . . . .	4
4.3	User . . . . .	4
<b>5</b>	<b>Zusammenführung und Restrukturierung</b>	<b>4</b>
5.1	Diagramm - Paket FIRST.FM . . . . .	6
5.2	Diagramm - Paket NETFLAX . . . . .	7

# 1 Einleitung

## 1.1 Allgemeines

Es wurden die Datenbankschemata von NETFLAX und FIRST.FM integriert. Diese Dokumentation zeigt in chronologisch anschaulicher Art und Weise die Probleme und Herausforderungen auf die dabei aufgetreten sind. Bei der Integration verwenden wir die Vorgehensweise der 4 grundlegenden Schritte einer konzeptuellen Integration. Im Verlauf der Integration vergleichen wir die Schemata, achten auf Homo- und Synonyme, sowie auf strukturelle Konflikte, wie Kardinalitäten und Schlüsselattribute. Zur besseren Übersicht haben wir die dazu gekommenen Klassennamen ins Englische übersetzt.

Am Ende dieser Dokumentation befinden sich die Diagramme, welche das Ergebnis dieser Integration darstellen.

## 1.2 Formatierung

In der Dokumentation sowie im dazugehörigen Diagramm gilt die folgende Formatierung.

### Klassennamen

Der Name einer Klasse ist im Regelfall **fett** formatiert.

### Attribute

Attribute einer Klasse sind folgendermaßen gekennzeichnet

- Reguläres Attribut
- \* Attribut ist Teil des Primärschlüssel
- / Abgeleitetes Attribut
- + Fremdschlüssel

### Relationentypen

- Reguläre Relationen zwischen Klassen sind durch eine einfache schwarze Linie gekennzeichnet.
- Sogenannte XOR-Relationen besitzen an der ausgehenden Klasse ein Rautesymbol (◆). Hierbei darf die Klasse, für die gespeicherte Entität, nur mit einer einzigen der Klassen gleichzeitig in Relation stehen.

## 2 Pre-Integration Analyse

Bei der Betrachtung der gegebenen Schemata ist zu erkennen, dass es nur wenige Schnittpunkte gibt. Dies erlaubt eine relativ problemlose Zusammenführung beider Schemata. Bei den vorhandenen Schnittpunkten handelt es sich um Folgende:

- **User** und **Benutzer**
- **Comment** und **Kommentare**
- **Genre** und **Genres**

## 3 Vergleich der Schemata

Wir werden bei der Integration folgende Entitäten aufeinander abbilden:

- **Benutzer** auf **User**. Hierbei handelt es sich um synonyme Entitäten.
- **Kommentare** auf **Comment**. Diese Entitäten sind teilsynonym, da es sich zwar in jedem Fall um Kommentare handelt, hierbei allerdings unterschiedliche Dinge kommentiert werden.
- **Genres** auf **Genre**. Auch diese Entitäten sind teilsynonym, da es sich in beiden Fällen um Genres handelt, allerdings um Genres von verschiedenen Dingen. Zur Unterscheidung wird ein Diskriminatorattribut (-type) eingeführt werden.

Aufgrund der wenigen Überschneidungen gibt es keinerlei Kardinalitätskonflikte. Allerdings gibt es folgende Primärschlüsselkonflikte.

### **Benutzer \*B-Nummer / User \*id**

Die **Benutzer \*B-Nummer** muss auf **User \*id** abgebildet werden.

### **Genres \*Bezeichnung / Genre \*id**

Die Identifikation von **Genres \*Bezeichnung** wird auf die Identifikation per **Genre \*id** umgestellt. Ein Surrogatschlüssel ist hier aus Speicherplatzgründen sinnvoll.

### **Kommentare / Comment \*id**

**Kommentare** besitzt keinen Primärschlüssel, deshalb wird hier der Surrogatschlüssel **Comment \*id** übernommen.

## 4 Angleichung der Schemata

Im Diagramm von NETFLAX der anderen Gruppe haben wir die Klassen **Filme**, **Rabattaktion**, **Kaufvertrag**, **Leihvertrag**, **Abonnements**, **Episoden** und **Serien** nicht weiter angepasst, außer, wie oben in der Einleitung angemerkt, übersetzt. Des Weiteren haben wir die Kardinalitäten gleich gelassen. Was unverständlich aus dem Diagramm der anderen Gruppe hervorging war, dass ein Kaufvertrag für einen Film nur abgeschlossen werden kann, wenn eine Rabattaktion läuft. Ansonsten kann ein Film nur ausgeliehen werden. Da die Aufgabe nur darin besteht, beide Diagramme zu integrieren, haben wir das Diagramm der anderen Gruppe für Szenario A dabei belassen, jedoch sollte es logischerweise auch möglich sein einen Kaufvertrag für einen Film abzuschließen ohne dass eine Rabattaktion läuft.

### 4.1 Genre

Die Klasse **Genre** wurde um das Diskriminatorattribut -type ergänzt, welches durch einen numerischen Wert angibt, ob es sich dabei um ein Musik- oder Filmgenre handelt. Im NETFLIX-Schema wurden die Genres durch einen Namen (\*Bezeichnung) identifiziert, diese Identifikation soll nun durch einen Surrogatschlüssel (\*genre\_id) übernommen werden.

### 4.2 Comment

Die Klasse **Comment** wurde um das Attribut (-update) ergänzt. Hiermit wird der Anforderung Sorge getragen, dass auch der eventuelle Änderungszeitpunkt eines Kommentars zeitlich gespeichert werden soll. Dies wird nun als neues Feature auch auf der Userbase von FIRST.FM bereitgestellt.

### 4.3 User

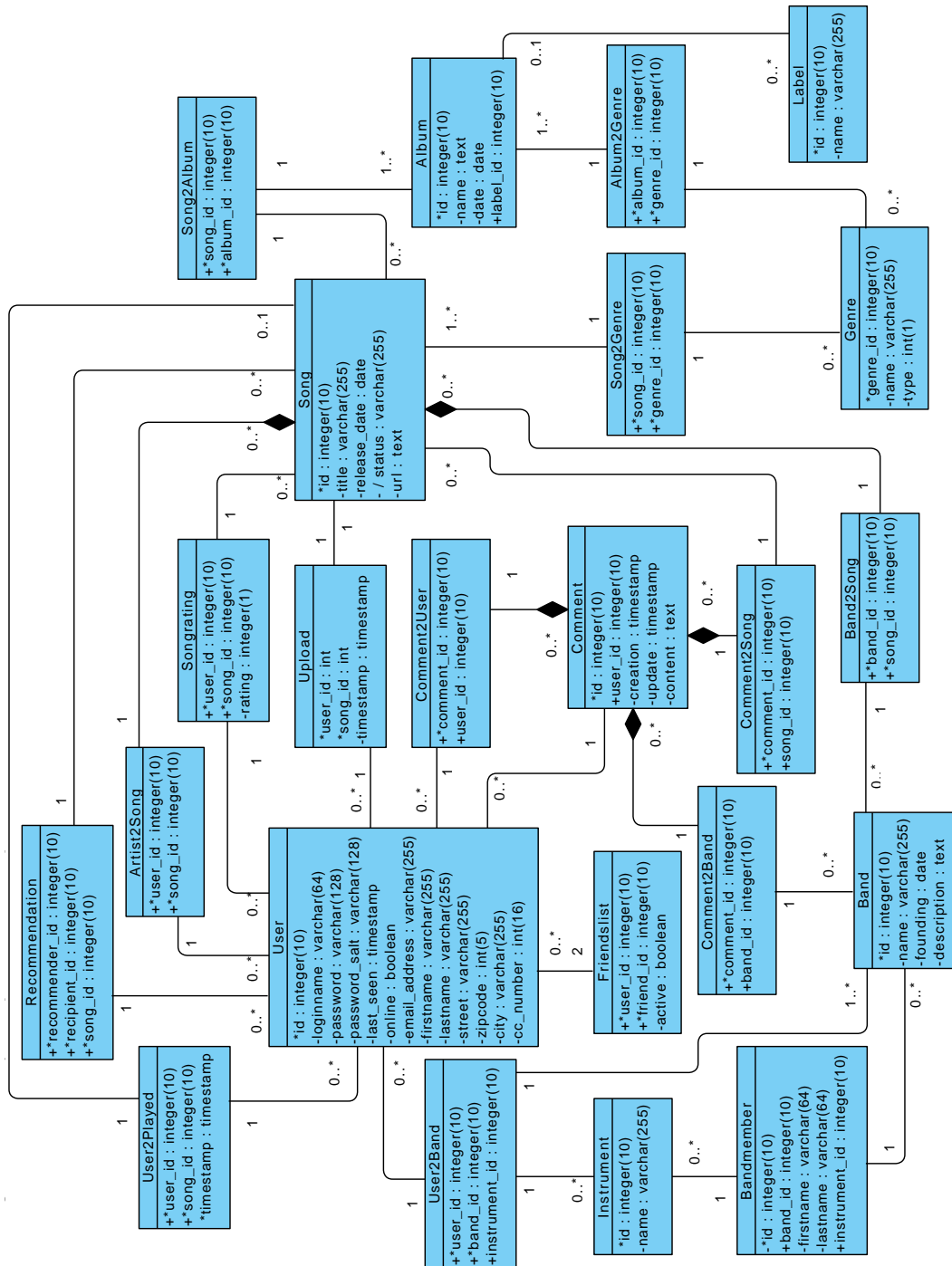
Das Klassendiagramm von NETFLAX sieht vor, dass für Leih- (**LoanContract**) und Kaufverträge (**SalesContract**) Name, Adresse und Kreditkartennummer (-cc\_number) des Vertragspartners (**Benutzer**) gespeichert wird. Für die Zusammenführung war es notwendig dies als optionale Attribute zu **User** hinzuzufügen.

## 5 Zusammenführung und Restrukturierung

Der Übersichtlichkeit und der Konsistenz der Struktur des Diagramms halber wurde das Diagramm in die beiden Pakete NETFLAX und FIRST.FM eingeteilt. Die Überschneidungen sind durch die die selben Klassennamen gekennzeichnet. Wir haben also ein Diagramm zur Beschreibung, dass User sich mit Musik, Bands, Alben, sowie das Kommentieren und das Bewerten dieser beschäftigen können (und /

oder) Filme (ausleihen / kaufen), Episoden kommentieren, Serien bewerten etc..  
Die Schnittstellen sind wie angemerkt die Klasse **User**, **Comment** und **Genre**.

## 5.1 Diagramm - Paket first.fm



## 5.2 Diagramm - Paket netflix

