

Lars Koenigsmann - 7209439

Mika Bredehoeft - 7209429

Tobias Barthold - 7209370

## Aufgabe 1 a)

### Anwendungskontext

Der von uns gewählte Anwendungskontext ist eine Streamingplattform die Menschen besuchen können um die Live-Übertragungen anderer anschauen zu können, oder um selber eine Live-Übertragung zu starten.

### Anwendungsszenario der Datenbankanwendung

Das Anwendungsszenario unserer Datenbankanwendung setzt sich aus vielen kleinen zusammen. Die Datenbank soll es Nutzern ermöglichen Accounts anlegen und managen zu können, sowie einen Stream starten oder einem anderen zuschauen zu können. Darüber hinaus soll der User anderen Usern folgen und an einem Streamchat teilnehmen können.

### Statische Integritätsbedingungen

```
creation DATE DEFAULT CURRENT_DATE NOT NULL
```

oder anderes Beispiel:

```
message_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
```

Daten dürfen nicht NULL sein allein schon wegen der Semantik, dass es nicht "kein" Datum von der Erschaffung von etwas gibt. Standardmäßig wird das aktuelle Datum verwendet.

```
passwort varchar(64) NOT NULL CHECK(length(passwort) > 5),
```

Das Passwort darf nicht NULL sein da sich sonst jeder in den Account einloggen könnte. Aus sicherheitsgründen darf es auch nicht weniger als 5 Zeichen enthalten, da es sonst zu einfach von entsprechenden Programmen gematched wird, und somit "hackern" leichten zugriff auf den Account verschaffen könnte.

```
mail varchar(80) NOT NULL,
```

Mails dürfen nicht NULL sein weil unsere accounts in erster linie an einer Usermail hängen, ohne die ein User sich nicht identifizieren kann, sollte er mal den username oder das passwort vergessen haben.

```
username varchar(20) UNIQUE NOT NULL,
```

Usernames dürfen nicht NULL enthalten um zu gewährleisten, dass sich User gegenseitig identifizieren können, aus diesem Grund müssen sie auch Unique sein, sonst könnten User Identitätsklau betreiben.

```
tier CHARACTER(1) DEFAULT 1 NOT NULL CHECK (tier BETWEEN '1' AND '3'),
```

Tiers dürfen nicht null sein, weil jeder immer einen Tier/Rank hat, wenn er subscribed ist. Standardmäßig 1.

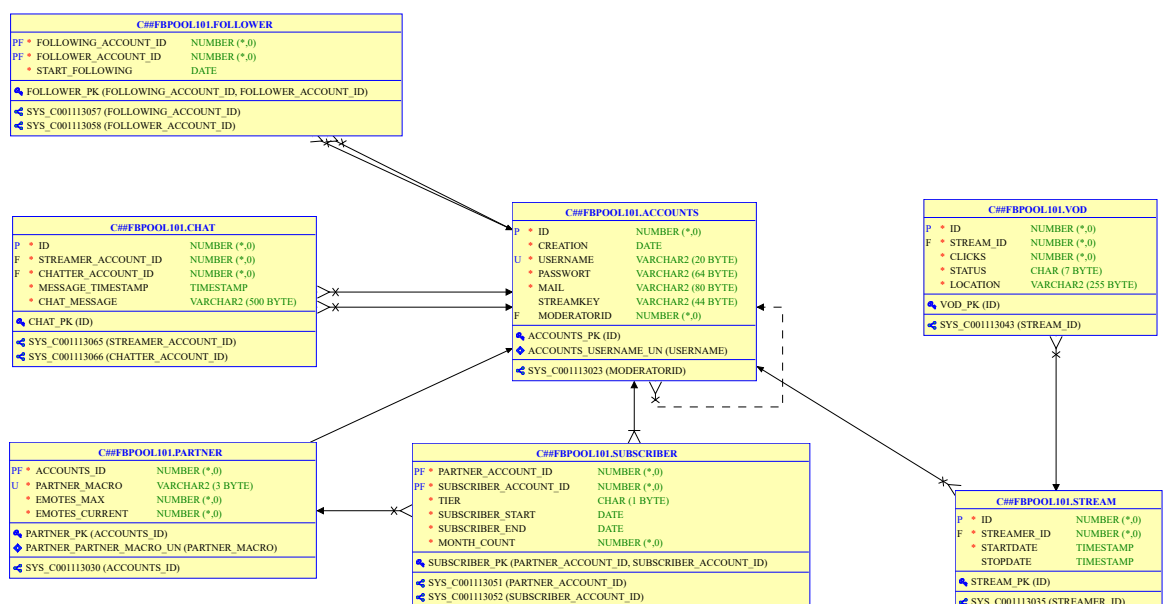
```
chat_message VARCHAR(500) NOT NULL,
```

Chat messages dürfen nicht null sein weil NULL kein Character ist.

```
streamer_id INTEGER NOT NULL,
```

Streamer ID's als Fremdschlüssel im Kontext eines Streams dürfen nicht null sein weil ein Stream nicht ohne Streamer stattfinden kann.

## Aufgabe 1 b)



## Accounts

Accounts dürften relativ selbsterklärend sein bis auf Streamkey. Der Streamkey ist dafür gedacht, dass man jenen generiert und über diesen dann einen Stream starten kann, da wir ja kein Konzept für account gebundene Channels haben.

Die Moderator ID ist ein Fremdschlüssel welcher auf die Account IDs verweist welche diesen Account verwalten dürfen.

## VOD - Video-on-demand

Vergangene Streams. Clicks und Location sind lediglich Datensätze die man am Frontend als Stats anzeigen kann und sonst noch keine weitere Verwendung haben. Status ist dazu gedacht, dass man die Videos z.B. private oder public setzen kann.

## Chat

Streamer-Account ID's werden benötigt da der Chat am Account des streamers hängt.

## Follower

Follower haben einen zusammengesetzten Primärschlüssel mit dem Fremdschlüssel der Account ID des accounts der Followed, da beide Values eindeutig sind und ein Account mehreren anderen Accounts folgen kann aber jedem Account nur einmal. So ergibt sich ein eindeutiges 2er Tupel.

## Partner

Partner von der Plattform kriegen besondere Kapazitäten z.B. mehr Emotes.

## Subscriber

Subscriber können Accounts die Partner sind mit Geld unterstützen was bspw. Monatlich abgebucht wird. Genannt subscriben. Note: Nur Partner können Subscriber enthalten.