



Erläuterung

Anmerkung: In ArangoDB wird für jede Collection ein `_key` und eine `_id` (`{{collectionName}} + _key`) gesetzt; diese können auch manuelle gesetzt werden.

Die Entitäten 'users' und 'tweets' sollten selbsterklärend sein, bei den Usern haben wir nur einen Namen gespeichert, in einem realen Social Network würden Felder wie "username", "password", "email" etc. sinn ergeben, aber das war ja nicht der Anspruch an die Aufgabe. Die "`_key`"s für die User haben wir aus den Twitter Followerdaten entnommen und gesetzt.

Die Tweets haben wir weitestgehend unverändert eingeladen, wobei die meisten Felder bei den Queries nicht verwendet werden. Bei den Tweets existierte ein Feld "numberOfLikes", dass soll in unserem Fall eine redundante Speicherung der Likezahl sein. Wir wollten die Anzahl der Likes eines Tweets allerdings noch anders darstellen, nämlich wie an einen Graph angelehnt in einer "Kanten" Collection. Diese Collection mit dem Namen "likes" nimmt jeweils eine `_id` eines

Graph angeordnet in einer Kanten Collection. Diese Collection mit dem Namen "likes" nimmt jeweils eine `_id` eines Dokuments einer Collection für die Felder "`_from`" und "`_to`". Also in "likes" wird von einem User auf einen Tweet referenziert. Nach denselben Prinzipien haben wir die Collection "wrote" erstellt, also dass ein Post von einem User geschrieben/gepostet wurde.

Darüber hinaus existiert die Kantencollection "follows", die die Followerbeziehung zwischen zwei Usern definiert, also der "`_from`" User folgt dem "`_to`" User.

Schlussendlich noch die Collection "fanout", die für jeden User die Tweets cached, die auf seiner Startseite angezeigt werden sollen.