

Jede Collection entspricht einer „Tabelle“.

New\_accounts:

Die txt Datei wurde so verändert, dass jedes Dokument eine Liste an Followern und Accounts, denen gefolgt wird enthält. Als Key dient eine user\_id. So kann die Anzahl an Followern und Accounts denen gefolgt wird direkt über die Arraylänge bestimmt werden

Posts:

Als Key wurde ein Hash bestehend aus dem gesamten Dokument gewählt, damit dieser immer einzigartig ist. Die einzige Änderung die wir im Vergleich zur CSV Datei vorgenommen haben, ist das Hinzufügen einer zufälligen user\_id.

WICHTIG: Die user\_ids wurden komplett zufällig zugeteilt, da nicht ausreichend Promis für die user\_ids vorhanden waren. Deshalb hat jeder Tweet eine andere user\_id. Eine user\_id lässt uns also NICHT auf den Autor schließen.

Likes:

Enthält genau dieselben Einträge wie die Posts Collection, allerdings noch zusätzlich eine Liste an user\_ids der User, die den Post geliket haben. Die Likes rechnen wir aus Performancegründen Modulo 500. Es existieren nur 81305 user\_ids und der meist gelikte Post hat beispielweise 429159 Likes. Der Hash als Key ist selbstverständlich derselbe. Wir haben uns dazu entschlossen den kompletten Post in dieser Collection, anstatt nur den Hash als Key zusammen mit der Liste an user\_ids zu speichern, damit nicht gejoint werden muss wenn man die beiden Informationen angezeigt bekommen möchte.

Starting\_page\_cache:

Erstellt ein Nutzer seine Startseite werden die Elemente hier abgelegt. Die Collection enthält für jeden Nutzer ein Dokument mit den beliebtesten Tweets von user\_ids denen er folgt. Wird ein Post von einem User dem er folgt erstellt, so wird dieser an die Liste angehängt. Die Anzahl Follower und gefolgt Accounts wird nicht gecached, sondern jedes Mal neu berechnet.