

Queries

1. Auflisten der Posts, die von einem Account gemacht wurden, bzw. ihm zugeordnet wurden

queryPostsOfAccount.ts

```
"FOR w IN wrote
  FILTER w._from == '${userId}'
  RETURN {user_id: w._from, post: DOCUMENT(w._to)}"
```

Aus unserer Kantenrelation "wrote" filtern wir nur die Einträge, wo `_from` gleich der Id des Users ist, von dem wir die Posts wollen. Dann geben wir eine Liste mit jeweils der UserId und den Dokumenten der IDs der Posts, die gefiltert wurden.

2. Finden der 100 Accounts mit den meisten Followern

queryTopFollower.ts

```
"FOR f IN follows
  COLLECT id = f._to WITH COUNT INTO count
  SORT count DESC
  LIMIT ${count}
  RETURN { user: DOCUMENT(id), count:count}"
```

Aus der Kantenrelation "follows" zählen wir die Anzahl Einträge, mit den distincten IDs von `_to` (wie ein SQL COUNT). Dann sortieren wir dieses absteigend und geben eine Liste mit jeweils dem Document des `_to` Users und der Anzahl der `_from` Einträge, was der Anzahl seiner Follower entspricht.

3. Finden der 100 Accounts, die den meisten der Accounts folgen, die in 1) [2]) gefunden wurden

queryTopFollowersOfUsersWithTopFollowers

```
"FOR f IN follows
  FILTER f._to IN ${JSON.stringify(top100UserIds)}
  COLLECT id = f._from WITH COUNT INTO count
  SORT count DESC
  LIMIT ${count}
  RETURN { user: DOCUMENT(id), count:count}"
```

Aus der Kantenrelation filtern wir auf Einträge, bei denen `_to` im Array der 100 gefundenen Usern aus 2) liegt. Dann tun wir dasselbe wie in 2, nur mit `_from` statt mit `_to`.

Auflisten der Informationen für die persönliche Startseite eines beliebigen Accounts (am besten mit den in 2) gefundenen Accounts ausprobieren); die Startseite soll Folgendes beinhalten (als getrennte Queries umsetzen)

4. die Anzahl der Follower

queryFollowerCountOfUser.ts

```
"FOR f IN follows
  FILTER f._to == '${userId}'
  COLLECT id = f._to WITH COUNT INTO count
  RETURN { user: DOCUMENT(id), follower_count:count}"
```

Wir filtern in "follows" auf _to mit der UserIDs eines Users und zählen dann die Anzahl der Einträge. Wir geben das Dokument des Users mit der UserID zurück und die Anzahl seiner Follower.

5. die Anzahl der verfolgten Accounts

queryAccountsUserFollows.ts

```
"FOR f IN follows
  FILTER f._from == '${userId}'
  COLLECT id = f._from WITH COUNT INTO count
  RETURN { user: DOCUMENT(id), count_user_follows:count}"
```

Wir tun dasselbe wie in 4., nur mit dem Feld _from, statt mit _to. Damit bekommen wir die Anzahl der Accounts, denen der User folgt.

6. wahlweise die 25 neusten oder die 25 beliebtesten Posts der verfolgten Accounts (per DB-Abfrage)

queryPostsOfFollowedUsers.ts

Caching der Posts für die Startseite (vgl. 4), erfordert einen sog. Fan-Out in den Cache jedes Followers beim Schreiben eines neuen Posts
queryFanOut

Auflisten der 25 beliebtesten Posts, die ein geg. Wort enthalten (falls möglich auch mit UND-Verknüpfung mehrerer Worte)

queryTopLikedTweets.ts

7. Auflisten der 25 beliebtesten Posts, die ein geg. Wort enthalten (falls möglich auch mit UND-Verknüpfung mehrerer Worte)

Die Query geht durch Tweets und filtert den content nach den gesuchten Wörtern. Der Filter läuft über die CONTAINS-Operation und der Content + die gesuchten Wörter werden als Lower-Case zurückgegeben.

Anschließend werden die übriggebliebenen Tweets nach Likes sortiert und auf die TOP 100 reduziert.