DynamoDB Table Descriptions

Note: all fields are strings

Follows

This table holds all the follow relation information for Tweeter. The followee is the person being followed, the follower is the person doing the following. We use this table to check to see if a user is following anther, and to see which feeds need to be updated. Two sort keys can be used to retrieve a given user's followees or followers.

- follower_handle - *primary partition key*
- followee_handle - *primary sort key*

follows_index: followee_handle (primary partition key), follower_handle (primary sort key)

Users

This table stores all the user's information. The password will always be hashed, and the profile picture will always be an S3 link to the '*tweeter-profile*' bucket. Through this table we are able to verify a user's login credentials, as well as retrieve the user's information from their alias (the primary partition key).

- alias - *primary partition key*
- password
- first_name
- last_name
- profile_picture_url

AuthTokens

This table holds the information for the Auth Tokens. Each Auth Token is generated at run time with a timestamp. When we validate Auth Tokens we check to see if the Auth Token belongs to the user using it and check to see if it hasn't expired. Current expiration time is two hours (this is configurable).

- auth_token - *primary partition key*
- user_alias
- timestamp

Feeds

This table holds the feeds for all Tweeter users. This means when a status gets posted all the poster's follower's feeds get updated with that status. Use the primary partition key, *user_feed_alias*, to get all of that user's feed. Posts are sorted by timestamp, so that they can be retrieved in order. This table also utilizes denormalization, meaning we store the poster's information in the table now, because it won't change, and it will increase read times.

- user_feed_alias - *primary partition key*
- status_message
- status_timestamp - *primary sort key*
- poster_alias
- poster_first_name
- poster_last_name
- poster_profile_picture_url

Stories

This table store all the information for a given user's story (aka, their own posts). Given a *user_alias* (the primary partition key) a query can be down to get the user's posts. Posts are sorted by timestamp, so that they can be retrieved in order. This table also utilizes denormalization, meaning we store the user's information (such as name and the profile picture) in the table, because it won't change, and it will increase read times.

- user_alias - *primary partition key*
- status_message
- status_timestamp - *primary sort key*
- user_first_name
- user_last_name
- user_profile_picture_url