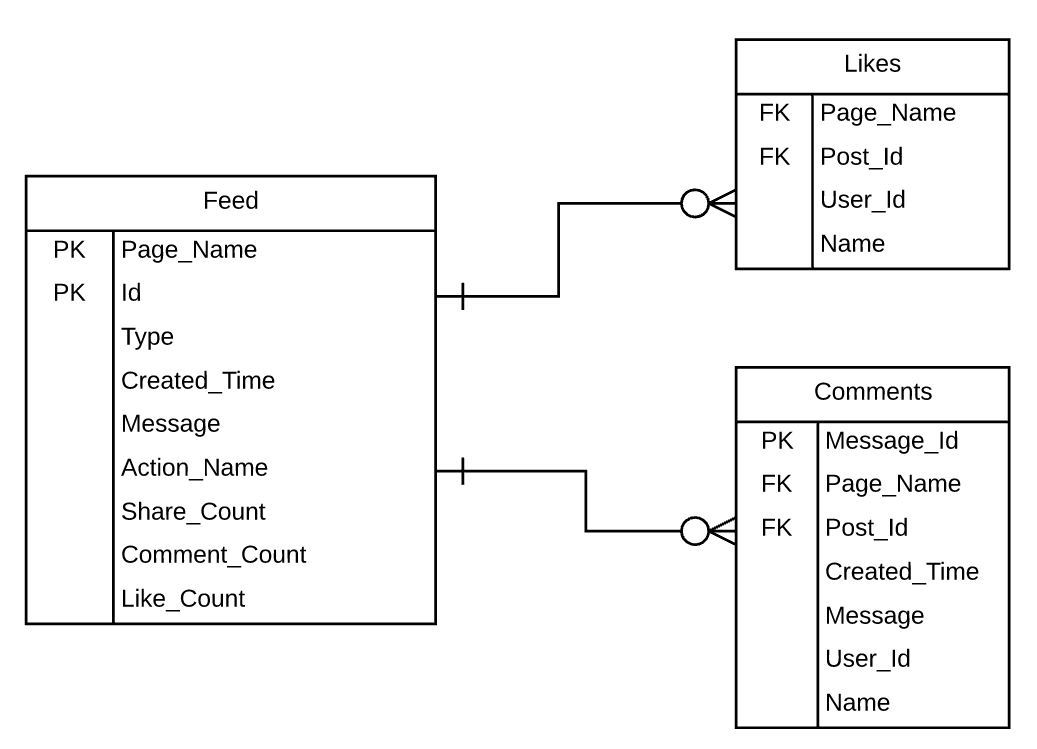
2. Facebook API Feed Data Ingestion

**Introduction**

By using Facebook Graph API, we can get the feed of posts and links published by the specific page, or by others on this page as well as likes and comments (https://developers.facebook.com/docs/graph-api/reference/v2.10/page/feed). I have written a python script to scrape the feed info in the JSON format and turn it into structured tables. Once the data is in the tabular format, we can load it in the relational database or use common analytical tools (like Excel) to do further analysis.

**Data Model**

We can split feed data into 3 tables. Each post has one or many likes and comments. This data model nicely accommodates the one-to-many relationship. In the Feed table, Page\_Name and Id are the composite keys. Likes and Comments can be joined to Feed by the Page\_Name and Post\_Id.



**Facebook Graph API**

Facebook offers different methods for authentication depending on which API function you want to use. In this example, all we need is App ID and App Secret. We can use this neat trick to create access token by concatenating App ID and App Secret with “|”.

First of all, we need to create an app and generate API credentials.

1. Login to Facebook and go to <https://developers.facebook.com/>.
2. Select ‘Add New App’ from the top left corner.
3. Enter Display Name and hit ‘Create App ID’.
4. Get App ID and App Secret From the dashboard
5. Access Token = <App ID>|<App Secret>

Python has Facebook SDK and it works fine. However, I am using the requests and json packages to make API calls and process data. In my opinion, the requests package is the best thing happened for creating REST applications with Python (http://docs.python-requests.org/en/latest/index.html). To make a GET request, we can simply add url and access token as a parameter in the get() function. Then, we convert the response to a JOSN object for further processing.

**Facebook Graph API**

It takes 7 argumenst: Access Token, Page Name (e.g. CocaCola), Json File Name, Feed csv file path, Likes csv file path, Comments csv file path, Since data (from when to pull the data).

**Example Call**

python facebookScrapeFeed.py <Access Token> CocaCola feed.json feed.csv likes.csv comments.csv 2017-10-31

**Key Points**

Since date has to be converted to a unix timestamp. I created the method to convert a regular date string to the unix timestamp, convert\_to\_epochtime().

The maximum number of feed records is 100. To obtain more than 100 records, we loop GET request by incrementing the offset parameter.

The maximum records for Likes and Comments in the Feed json file are 25. If there are more than 25 records, we can use the url in the next node until there is no next url comes back in the data.

The script works for both Python 2.7 and 3.x by changing the few lines to handle Unicode as instructed in the script. This is because each version handles Unicode differently.