

Twitter ERD

vignesh bhatt | October 13, 2022

The User table stores information about the user. I have a separate userID column as my Primary Key. An argument can be made to have the username as the Primary Key since it's unique, singular and required but I deemed userID to be the better Primary Key because the user table would have rows in the order of ~100 Millions and lookups would be a frequent. Lookups on integer PKs would generally have better performance than lookups on variable length strings. is_verified is a boolean and keeps track of whether the user is verified (blue tick). created_at is a date_time and keeps track of the timestamp when the user created the account. Since there is no column dependency, except on the Primary Key, this table is in BCNF.

User	
• <i>userId</i> 1-1(1)	INTEGER
<i>username</i> 1-1(1)	TEXT
<i>email</i> 1-1(1)	TEXT
<i>password</i> M-1(1)	TEXT
<i>firstName</i> M-1(1)	TEXT
<i>middleName</i> M-1(0)	TEXT
<i>lastName</i> M-1(1)	TEXT
<i>nationality</i> M-1(1)	TEXT
<i>date_of_birth</i>	TEXT
<i>profilePicture</i> M-1(0)	BLOB
<i>bio</i> M-1(0)	TEXT
<i>is_verified</i> M-1(1)	INTEGER
<i>created_at</i> M-1(1)	TEXT

1(1)

M(0)

Followers	
• <i>userId</i> M-1(1)	INTEGER
• <i>followerId</i> M-1(1)	INTEGER

Followers keeps track of a user's followers. userId and followerId can have duplicates individually, but are unique together so they make a composite Primary Key. userId refers to a particular user and followerId is the Id of another user that's following the user. Since there is no column dependency, this table is in BCNF.

Likes keeps track of users who liked a tweet. tweetId and userId can have duplicates individually, but are unique together so they make a composite Primary Key. tweetId refers to the tweet that was liked and userId refers to user who liked the tweet. Since there is no column dependency, this table is in BCNF.

Retweet keeps track of users who retweeted a tweet. tweetId and userId can have duplicates individually, but are unique together so they make a composite Primary Key. tweetId is the Id of the tweet that was retweeted and userId is the Id of the user who retweeted that tweet. Since there is no column dependency, this table is in BCNF.

Retweets	
• <i>tweetId</i> M-1(1)	INTEGER
• <i>userId</i> M-1(1)	INTEGER

M(0)

Likes	
• <i>tweetId</i> M-1(1)	INTEGER
• <i>userId</i> M-1(1)	INTEGER

M(0)

Replies	
• <i>parent_tweetId</i> M-1(1)	INTEGER
• <i>tweetId</i> M-1(1)	INTEGER

M(0)

Replies keeps track of replies on a tweet. parent_tweetId stores the Id of the original tweet and tweetId stores the Id of the reply. tweetId and parent_tweetId can have duplicates individually, but are unique together so they make a composite Primary Key. Both the Ids refer to tweetIds in Tweet table. Since there is no column dependency, this table is in BCNF.

Views	
• <i>tweetId</i> M-1(1)	INTEGER
• <i>userId</i> M-1(1)	INTEGER

Views keeps track of users who viewed a tweet. tweetId and userId can have duplicates individually, but are unique together so they make a composite Primary Key. tweetId is the Id of the tweet that was viewed and userId is the Id of the user who viewed that tweet. Since there is no column dependency, this table is in BCNF.

1(1)

Tweet	
• <i>tweetId</i> 1-1(1)	INTEGER
<i>userId</i> M-1(1)	INTEGER
<i>created_at</i> M-1(1)	TEXT
<i>is_promoted</i> M-1(1)	INTEGER

M(0)

The Tweet table has tweetId as the primary key which is also a unique identifier for each tweet. It contains the userId of the person who made the tweet, a created_at attribute to store the timestamp of creation and an is_promoted boolean to check if the tweet is promoted. Since there is no column dependency, this table is in BCNF.

1(1)

1(1)

1(1)

1(1)

Tweet_content	
• <i>tweetId</i> 1-1(1)	INTEGER
<i>tweetBody</i> M-1(0)	TEXT
<i>picture</i> M-1(0)	BLOB
<i>video</i> M-1(0)	BLOB

The Tweet_content table stores the body of the tweet and has a 1-1 relationship with Tweet table. As such, tweetId is the Primary and the Foreign Key for Tweet_content. A tweet can have a text body, a picture, a video or a combination of the three. I chose to store the body separately because I felt it was more structured to separate out a table for any analyses on the tweet bodies vs analyses on combined tweet metrics. Since there is no column dependency, this table is in BCNF.