Database project

Design flaws and new design

We were initially provided a table containing all attributes (called universal relation), in the course of this report I will show why this form should be broken into smaller tables using functional dependencies and normal form.

Firstly we can see that we have repeating groups, the columns hashtag1, hashtag2 etc. are all storing the same information, this breaks 1nf and makes it difficult to make queries such as how many times a certain hash tag has been used. A more appropriate way to store this information is to have 1 column for hash tags as this will make it easier to query, it would allow us to store more than 6 hash tags and will reduce the number of null values in those columns. However in doing this we have created further redundancy in our table.

Assumptions about the relation schema:

- The tweet id is unique and provides us with information about the tweet
- User_id and user_name are both unique and can provide us with information about the user.

I will now look at functional dependencies to decide whether our table is in 3nf, intuitively I can see that it isn't because of the repetition in the data but in order to achieve a lossless decomposition it is best to inspect the functional dependencies.
To make the functional dependencies easier to write I will refer to the following:
created_at = A
text = B
tweet_id= C
in_reply_to_screen_name= D
in_reply_to_status_id = E
in_reply_to_user_id = F
retweet_count = G
tweet_source = H
retweet_of_tweet_id = I
hashtags = J
user_id = K
user_name = L
user_screen_name = M
user_location = N
user_utc_offset = O
user_time_zone = P
user_followers_count = Q
user_friends_count = R
user_lang = S
user_description = T
user_status_count = U
user_created_at= V

From the table I can see the following functional dependencies (not including trivial functional dependencies):
C->A, B, D, E, F, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V
K->L, M, N, O, P, Q, R, S, T, U, V (right hand side=t)
C, J->A, B, D, E, F, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V
L->K, M, N, O, P, Q, R, S, T, U, V
D->E
E->D
A Candidate key after taking the table into 1nf is C, J but from the functional dependencies we can see that if we remove J the functional dependency still holds therefore we have a partial dependency and our table violates 2nf.
In order to resolve this issue we can split our relational schema R into two parts. To show this in terms of functional dependencies I will write A, B, D, E, F, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V as b, so the above functional dependency will be C->b.
Therefore attribute J=R (relation schema)-(C ∪b). From which we can decompose the giant table into R1=J∪C and R2=C∪b.
Looking at table R2 we can see that we still have a lot of redundancy in the columns related to the user, from our functional dependencies we can see that R2 is not in 3nf due to the fact that C->K and K->t therefore we have a transitive functional dependency.
In order to normalise the table we will again use a lossless decomposition. We now have the following tables:
Tweet, hashtags and users
From our table for tweets we can see that most columns for retweet_of_tweet_id are null this is also the case for information about replies. Due to these null values it may be best to further decompose the tweet table again doing so whilst preserving dependencies.
We now have 5 tables' tweets, hashtags, users, retweets and replies.
The functional dependencies being:
Tweets: C->A, B, G, H, K
Replies: C->D, E, F
D->E
E->D
Retweets: C->I
Hashtags: (C, CJ) ->J new attribute hashtag_id introduced in order to uniquely identify each hashtag, to do this in the table I had to add the primary key after inserting the data.
Users: K-> L, M, N, O, P, Q, R, S, T, U, V
L->K, M, N, O, P, Q, R, S, T, U, V
From the information in replies we can see that all this information is present in other tables, in_reply_to_screen_name and in_reply_to_user_id have the same information as user_screen_name and user_id respectively, and it would be most beneficial to use a foreign key to prevent errors in the table, but in the given data set not all the information present in in_reply_to_user_id is present in user_id therefore this cannot be done. The same goes for tweet_id and in_reply_to_status_id. We also have a functional dependency between in_reply_to_user_id and in_reply_to_screen_name which does break 3nf but there is no way to decompose this whilst still preserving dependencies. In reality there

isn't a point in storing the screen name because that information can be gathered by joining replies and users on in_reply_to_user_id but in this case with incomplete information it isn't possible, therefore I have decided to leave this dependency alone as I don't want to remove data from the table.

<u>Queries</u>

<u>Tweets, users and languages</u>

How many tweets are there in total?

110574

How are these tweets distributed across languages?

```
user_lang | count
-----------+-------
 zh-cn    |   26
 cs       |    2
 pt       | 3977
 th       |  233
 msa      |   14
 ar       | 1405
 ur       |    1
 fi       |    1
 zh-tw    |   14
 ca       |    7
 ru       |  396
 de       |   75
 fr       |  231
 es       | 17910
 nl       |   61
 en       | 74077
 hu       |    1
 el       |    2
 id       | 1423
 ko       |  691
 sv       |    2
```

Compute for each language, the fraction of total tweets that have that language setting, as well as the fraction of the number of users that have that language.

$$\frac{number\ of\ tweets\ in\ each\ language}{number\ of\ tweets} = fraction\_of\_tweets$$

$$\frac{number\ of\ users\ with\ each\ language}{number\ of\ tweets} = fraction\_of\_users$$

```
user_lang |  fraction_of_tweets  |  fraction_of_users
-----------+---------------------+----------------------
 ar       |  0.0127064228480475 |   0.0127954368080781
```

```
ca      |  6.3306021306998e-05 | 6.74458265486044e-05
cs      | 1.80874346591423e-05 | 1.92702361567441e-05
de      | 0.000678278799717836 | 0.000684093383564416
el      | 1.80874346591423e-05 | 1.92702361567441e-05
en      |    0.669931448622642 |    0.667511345351537
es      |    0.161972977372619 |    0.161012458207675
eu      | 9.04371732957115e-06 | 9.63511807837205e-06
fi      | 9.04371732957115e-06 | 9.63511807837205e-06
fil     | 8.13934559661403e-05 | 8.67160627053485e-05
fr      |  0.00208909870313093 |  0.00212936109532022
hu      | 9.04371732957115e-06 | 9.63511807837205e-06
id      |   0.0128692097599797 |    0.013103760586586
it      |  0.00023513665056885 | 0.000240877951959301
ja      |   0.0891800965869011 |   0.0924585930800582
ko      |  0.00624920867473366 |  0.00629173210517695
msa     | 0.000126612042613996 | 0.000125256535018837
nl      |  0.00055166675710384 | 0.000578107084702323
no      | 9.04371732957115e-06 | 9.63511807837205e-06
pl      | 3.61748693182846e-05 | 3.85404723134882e-05
pt      |   0.0359668638197045 |   0.0354957750007226
```

Retweeting habits

What fraction of the tweets are retweets?

$$\frac{number\ of\ tweets\ that\ are\ retweets}{number\ of\ tweets} = 0.193879212111346$$

Compute the average number of retweets per tweet.

$$\frac{sum\ of\ retweet\_count}{number\ of\ tweets} = 121.2742597717365746$$

What fraction of the tweets are never retweeted?

$$\frac{number\ of\ tweets\ where\ retweetcount=0}{number\ of\ tweets} = 0.669207951236276$$

What fraction of the tweets are retweeted fewer times than the average number of retweets (and what does this say about the distribution)?

$$\frac{number\ of\ tweets\ where\ retweetcount\ is\ less\ than\ average}{number\ of\ tweets} = 0.936196574239876$$

From this value we can see that the majority of tweets are retweeted less than the average, therefore the distribution is skewed to the right, and the mean is not an accurate measure for determining retweet_count.

Hashtags

What is the number of distinct hashtags found in these tweets?
10158

What are the top ten most popular hashtags, by number of usages?

```
 hashtag         | hashtagcount
-------------------------+--------------
 ReasonsIFailAtBeingAGirl |        467
 RED                 |        240
 oomf                |        190
 HonestyHour           |        172
 TeamFollowBack         |        139
 EresGuapaSi           |        130
 10PeopleYouTrulyLove    |        126
 TweetLikeAGirl         |        98
 ImSingleBecause        |        97
 WeAllGotThatOneFriend   |        96
```

Write a query giving, for each language, the top three most popular hashtags in that language.

```
user_lang |   hashtag                 | pophashtaglang|    rownum
-----------+-------------------------+----------------+--------
 ar          | سوريا                        |           18|    1
 ar          | بصورة_غرد                    |           13|    2
 ar          | الكويت                       |           11|    3
 ca          | SoyelIngrato               |            1 |    1
 ca          | EseMomentoEnElQue          |            1 |    2
 cs          | continuous                 |            1 |    1
 cs          | fermentation               |            1 |    2
 de          | Hamburg                    |            2 |    1
 de          | Koblenz                    |            1 |    2
 de          | A7                         |            1 |    3
 el          | soundcloud                 |            1 |    1
 en          | ReasonsIFailAtBeingAGirl |        459 |    1
 en          | RED                        |        214 |    2
 en          | oomf                       |        190 |    3
 es          | EresGuapaSi                |        120 |    1
 es          | LoAdmito                   |         50 |    2
 es          | SoloElVenezolano           |         42 |    3
 fil         | NarinigKoNaYan             |          1 |    1
 fil         | food                       |          1 |    2
 fil         | 10TwitterCrushes           |          1 |    3
 fr          | RT                         |          3 |    1
```

<u>Replies</u>

How many tweets are neither replies, nor replied to?

$$\begin{aligned}&number\ of\ tweets\ that\ are\ neither\ replies\ nor\ replied\ to \quad = \\ &number\ of\ tweets - (number\ of\ tweets\ that\ are\ replies + number\ of\ tweets\ that\ are\ replied\ to)\end{aligned}$$

= 84424

If a user user1 replies to another user user2, what is the probability that they have the same language setting?

In order to do this we must find all the user1 user2 combinations where user1 has replied to user2 that exist in the data (where we have information for user2). Counting all the times where the languages are the same/the total number of user1 user2 combinations where user1 has replied to user2 that exist in the data. The probability for this is 0.913404507710557

How does this compare to the probability that two arbitrary users have the same language setting?

The probability that two arbitrary users have the same language is 0.481699735114612 This value was found by first picking a user (user1) and finding the probability of user1's language (user1lang). Then picking a second user (user2) and finding the probability that the second user has a particular language (user2lang). Multiplying the two and again summing to find the probability that two users have the same language setting.

From these probabilities we can make the obvious conclusion that users that reply to each other are way more likely to have the same language setting as opposed to arbitrary users.