***DESIGN FLATS:***

1) Poor design:

The database poorly identifies and defines the entities (tables) and attributes (columns). All the data have relationships to most of the other elements of data so getting each well defined is essential.

2) Ignoring normalization:

Normalizing your data is essential to good performance, and ease of development. The given data is in 1NF but it should atleast be in 3NF for a good overall performance.

3) Inconsistency:

A lot of data in the database is stored in inconsistent manner. For example the hashtag table.

4) If a user makes multiple tweets then each time a new entry with all the information is created. Due to this any deletion made to any tweet of a user will delete all the data on that user.

5) There is a lot of redundancy in the given database.

6) There are a lot of null values in various tables of the database which make the data very inconsistent.

7) Most of the data in the 6 hashtag columns in the database are null throughout the database.

8) There is no mention of status_id in the table but a in_reply_to_status_id column is present. Also the values for in_reply_to_status_id is missing in lots of places.

9) There are a lot of Data Anomalies which occur due to poorly planned, un-normalised database as all the data is stored in one table.

10) retweet_of_tweet_id is created even if the tweet wasn't retweeted and missing in places where a tweet was actually retweeted.

***FUNCTIONAL DEPENDENCIES:***

tweet_id -> created_at ,text ,retweet_count ,tweet_source ,
in_reply_to_screen_name,in_reply_to_status_id ,in_reply_to_user_id ,

tweet_id , hashtag

user_id -> user_name ,user_screen_name ,user_lang ,user_status_count ,user_created_at,
user_location, user_utc_offset ,user_time_zone, user_description,
user_followers_count,user_friends_count

***NEW DATABASE DESIGN:***

1) retweet_of_tweet_id is now created only when tweet is retweeted and not randomly.(A new table Retweet containing retweet_of_tweet_id to do so)
2) in_reply_to_screen_name ,in_reply_to_status_id ,in_reply_to_user_id show trivial dependency on tweet_id and have all been put in a new table Reply.
3) user_utc_offset was dependent on user_time_zone.So created a new table TimezoneTable in which reference to user_id has been made from Users.
4) The data is now in 3NF which improves the overall performance of the database.
5) Data is now consistent which improves the Data Integrity.
6) For every hashtag by the same user a new row is created referring to the respective tweet_id which reduces the inconsistency cause by the 6 different entries taken in the given database.
7) Not all user have description so a new table Description is created to store only the user id's who have a description.
8) A new table Count to store friends and followers count per user with respect to the user id's.
9) Tweet_id and User_id declared as primary keys and the entire database is divided into tables on the basis of the dependencies on these primary keys and the amount of information that is relevant and is required in that particular table.
10) Overall design of the database has been balanced by dividing the data into number of tables.