Project Report
Milestone 1

Project Description:

The project is a community Q & A website similar to stackoverflow where users can ask and answer questions on the topic of finance. Users are subject to a reputation award process similar to stack overflow. The data is attained from a data dump of the actual website money.stackexchange.com. The data has been cleaned and modified to suit our needs.

The schema is described as follows:

1. The Badges Table

Stores Badges earned by users

Column Name	Datatype	Constraints
id	integer	Primary Key
user_id	integer	NOT NULL, foreign key referencing users(id) on delete cascade on update cascade
class	smallint	NOT NULL
name	varchar(64)	NOT NULL
tag_based	bool	NOT NULL
date	timestamp	NOT NULL

Other Integrity Constraints:
check(class in (1,2,3))

1.1 Attributes

- Id:
  This is the primary key of the table.

- name:
  stores the name of the badge earned.

- date:
  Stores the timestamp when a particular badge was given to a user..

- user_id:
  Stores the user id of the user who the badge was given to. All badges of a user are deleted if user is deleted in users table.

- class:
  Stores integers 1, 2 or 3 depending on whether the badge class is Gold, Silver or Bronze.

- tag_based:
  Stores a boolean value, True if the badge is for a tag, otherwise it's a named badge.

1.2 <u>Functional Dependencies</u>

The attribute "Id" is the primary key,
In all FDs that hold on above table, id must be present in LHS (set of determiners).
There are no other FDs. As id is primary key, all determiners have to be superkeys, thus, above table is in BCNF.
Some of the functional dependencies in the table are:

Id -> name
Id -> date
Id -> user_id
Id -> class
Id -> tag_based

2. Post–History Table

Stores the complete history of activity on all post.

Column Name	**Datatype**	**Constraints**
id	integer	Primary Key
post_id	integer	NOT NULL, foreign key referencing id in posts table, All post history is deleted if post is deleted in posts table
user_id	integer	foreign key referencing users(id)
post_history_type_id	smallint	NOT NULL
user_display_name	varchar(64)	
text	text	
comment	text	
creation_date	timestamp	NOT NULL

2.1 Attributes

- Id:
  This is the primary key of the table.

- post_id:
  The id of the post whose history has been stored.

- user_id:
  The id of the user who made this particular post, NULL if that user has been deleted from users table.

- post_history_type_id:
  Stores integers depending on the kind of update on the post's history type.

- user_display_name:
  Populated if a user has been removed and no longer referenced by user Id, otherwise NULL.

- text:
  Stores a raw version of the new value for a given revision.

- comment:
  This field will contain the comment made by the user who edited a post.

- creation_date:
  Stores the date when the particular post was updated by the user.

2.2 <u>Functional Dependencies</u>

The attribute “Id” is the primary key. In all FDs that hold on above table,
id must be present in LHS (set of determiners). There are no other FDs. As id is primary key, all determiners have to be superkeys, thus, above table is in BCNF. some of the functional dependencies in the table are:

Id -> post_id
Id -> user_id
Id -> post_history_type_id
Id -> user_display_name
Id -> text
Id -> comment
Id -> creation_date

3. Posts Table

Stores all posts (questions and answers) posted by users and related info.

Column Name	Datatype	Constraints
id	integer	Primary Key
owner_user_id	integer	Foreign key referencing users table
last_editor_user_id	integer	Foreign key referencing users table
post_type_id	smallint	NOT NULL
accepted_answer_id	integer	foreign key referencing posts table itself i.e answer must be added to posts table before accepting
score	integer	NOT NULL
parent_id	integer	Foreign key referencing posts table itself i.e questions must be inserted in table before answer
view_count	integer	
answer_count	integer	
comment_count	integer	
owner_display_name	varchar(64)	
last _editor_display_name	varchar(64)	
title	varchar(512)	
tags	varchar(512)	
body	text	
favorite_count	integer	
creation_date	timestamp	NOT NULL

community_owned_date	timestamp	
closed_date	timestamp	
last_edit_date	timestamp	
last_activity_date	timestamp	

Other Integrity Constraints:
check(post_type_id>=1 and post_type_id<=8)

check(creation_date<=community_owned_date and creation_date<=closed_date and creation_date<=last_edit_date and creation_date<=last_activity_date)

3.1 Attributes

- Id:
  This is the primary key of the table. Assigned in serial order in order of creation.

- owner_user_id:
  The id of the owner who wrote the post.

- last_editor_user_id:
  User who last edited the post

- post_type_id:
  Stores integers depending on the kind of post it is (1:question post, 2:answer post).

- accepted_answer_id:
  Only present if post_type_id=1 (A question post), this will store the id of the post which the original poster recognizes as the best answer. Initially NULL, is populated if any answer is accepted by user who posted the question.

- score:
  A score is essentially the difference between the number of upvotes and the number of downvotes that a particular post has.

- parent_id:
  Only present if the post_type_id=2 (An answer post), in which case it stores the post id of the parent post i.e question post.

- view_count:
  Number of people who have viewed that particular post.

- answer_count:
  Number of answers (only populated for question posts)

- comment_count:
  Number of comments on a post

- owner_display_name:
  Populated if a user has been removed and no longer referenced by user Id.

- last_editor_display_name:
  .Populated if last editor user has been removed and no longer referenced by user Id.

- title:
  The title of the post.

- tags:
  Tag names of all the marked tags of that particular post.

- body:
  The content of the post is stored as Html.

- favorite_count:
  The number of people who have saved the post in their favorites.

- creation_date:
  Date when the post was created.

- community_owned_date:
  .

- closed_date:
  The date when the post was closed (exists only when the post is closed).

- last_edit_date:
  The date when the post was edited for the last time.

- last_activity_date:
  The time and date when last activity was performed on the post.

3.2 <u>Functional Dependencies</u>

The attribute “Id” is the primary key, some of the functional dependencies in the table are:

Id -> owner_user_id
Id -> last_editor_user_id
Id -> post_type_id
Id -> accepted_answer_id
Id -> score
Id -> parent_id
Id -> view_count
Id -> answer_count
Id -> comment_count
Id -> owner_display_name
Id -> last_editor_display_name
Id -> title
Id -> tags
Id -> body
Id -> favorite_count
Id -> creation_date
Id -> community_owned_date
Id -> closed_date
Id -> last_edit_date
Id -> last_activity_date

In all FDs that hold on above table,
id must be present in LHS (set of determiners). There are no other FDs. As id is primary key, all determiners have to be superkeys, thus, above table is in BCNF.

4. Post Links Table

Stores all posts related to a certain post.

Column Name	**Datatype**	**Constraints**
id	integer	Primary Key
related_post_id	integer	NOT NULL, foreign key referencing posts tables id with on delete cascade on update cascade
post_id	integer	NOT NULL, foreign key referencing posts(id) with on delete cascade on update cascade
link_type_id	smallint	NOT NULL
creation_date	timestamp	NOT NULL

Other Integrity Constraints:
check(link_type_id in (1,3))

4.1 Attributes

- Id:
  This is the primary key of the table.

- related_post_id:
  Stores the id of the targeted or related post.

- post_id:
  Stores the id of the source post.

- link_type_id:
  Stores integer 1 if the post is linked and 3 if the post is a duplicate of related_post_id.

- creation_date:
  Stores the date for when the link was created.

4.2 Functional Dependencies

Id is primary key. All FDs will have id in LHS (set of determiners). There are no other FDs. Thus, as all FDs have superkey in LHS, table is in BCNF.

Id -> related_post_id
Id -> post_id
Id -> link_type_id
Id -> creation_date

5. Comments Table

Stores all comments made on posts and related info.

Column Name	Datatype	Constraint
id	integer	Primary Key
post_id	integer	NOT NULL, foreign key referencing posts table, If a post is deleted all comments are also deleted
user_id	integer	Foreign key referencing users table, if user is deleted this field is set to NULL
score	smallint	NOT NULL
user_display_name	varchar(64)	
text	text	
creation_date	timestamp	NOT NULL

5.1 Attributes

- Id:
  This is the primary key of the table.

- post_id:
  Stores the id of the post on which comment is made.

- user_id:
  Id of the user who has authored the comment.

- score:
  A score is essentially the difference between the number of upvotes and the number of downvotes that a particular comment has.

- user_display_name:
  Populated if a user has been removed and no longer referenced by user Id, otherwise NULL.

- text:
  A raw version of the content in the comment.

- creation_date:
  Stores the date for when the comment was posted.

5.2 <u>Functional Dependencies</u>

All FDs that hold must contain the primary key ‘id’ in LHS. Also, there are no other candidate keys in the table. As add determiners are superkeys, table is in BCNF.Some FDs are:

Id -> post_id
Id -> user_id
Id -> score
Id -> user_display_name
Id -> text
Id -> creation_date

6. Votes Table

Stores votes cast by users on posts (Ex- upvotes, downvotes etc)

Column Name	**Datatype**	**Constraint**
id	integer	Primary Key
user_id	integer	foreign key referencing users table
post_id	integer	NOT NULL, foreign key referencing posts(id) on delete cascade on update cascade
vote_type_id	smallint	NOT NULL
creation_date	timestamp	NOT NULL

6.1 Attributes

- Id:
  This is the primary key of the table. Assigned in serial order in order of creation.

- post_id:
  Stores the id of the post the vote belongs to.

- user_id:
  Stores the user id who the vote belongs to.

- vote_type_id:
  Stores an integer representing vote type
  1: Accepted by originator, 2 for an upvote, 3 for a downvote.

- creation_date:
  Stores the date for when this vote was cast.

6.2 <u>Functional Dependencies</u>

The attribute “Id” is the primary key. In all FDs that hold on above table,
id must be present in LHS (set of determiners). There are no other FDs. As id is primary key, all determiners have to be superkeys, thus, above table is in BCNF.
Some FDs are:
Id -> post_id
Id -> user_id
Id -> vote_type_id
Id -> creation_date

7. Tags Table

Stores info on tags.

Column Name	Datatype	Constraints
id	serial	Primary Key
excerpt_post_id	integer	foreign key referencing posts(id) on delete set NULL on update cascade
wiki_post_id	integer	foreign key referencing posts(id) on delete set NULL on update cascade
tag_name	varchar(255)	NOT NULL
count	integer	

7.1 Attributes

- Id:
  This is the primary key of the table.

- excerpt_post_id:
  Id of Post that holds the excerpt text of the tag i.e the post where the tag was first created.

- wiki_post_id:
  Id of Post that holds the wiki text of the tag.

- tag_name:
  Name of the tag.

- count:
  count of posts containing the tag. Default value: 0
  .

7.2 <u>Functional Dependencies</u>

The attribute “Id” is the primary key. In all FDs that hold on above table,
id must be present in LHS (set of determiners). There are no other FDs. As id is primary key, all determiners have to be superkeys, thus, above table is in BCNF.

Id -> excerpt_post_id
Id -> wiki_post_id
Id -> tag_name
Id -> count

8. Users Table

Stores information about all the users of the website.

Column name	Datatype	Constraints
id	integer	Primary key
reputation	integer	NOT NULL
views	integer	
down_votes	integer	
up_votes	integer	
display_name	varchar(255)	NOT NULL
location	varchar(512)	
website_url	varchar(255)	
about_me	text	
creation_date	timestamp	NOT NULL
last_access_date	timestamp	NOT NULL

Other Integrity Constraints:
check(creation_date<=last_access_date)

8.1 Attributes

- Id:
  This is the primary key of the table. Unique Id of a user, assigned serially in order of insertion.

- password:
  Stores a safe and hashed value of the user's password.

- reputation:
  It is an integer that is incremented or decremented according to user’s activity on the website (“In some sense, loosely, it represents how much the community trusts you and what privileges you will have”). It is earned and lost based on user activity. Following is scheme,
  You gain reputation when: question is voted up: +10, answer is voted up: +10

  You lose reputation when: your question is voted down: −2, your answer is voted down: −2, your article is downvoted: -2, you vote down an answer: −1, you downvote an article: -1

- views:
  Total number of times the profile is viewed.

- down_votes:
  Total number of downvotes the user has given (to a question, answer or comment).

- up_votes:
  Total number of upvotes the user has given (to a question, answer or comment).

- display_name:
  The name that is visible to all others viewing the account.

- location:
  Most recent location of the user.

- website_url:
  URL of the user’s profile website.

- about_me:
  Details about the user in an HTML format.

- creation_date:
  Non-Null timestamp representing account creation date and time.

- last_access_date:
  Non-Null timestamp representing date and time when user last accessed the website.

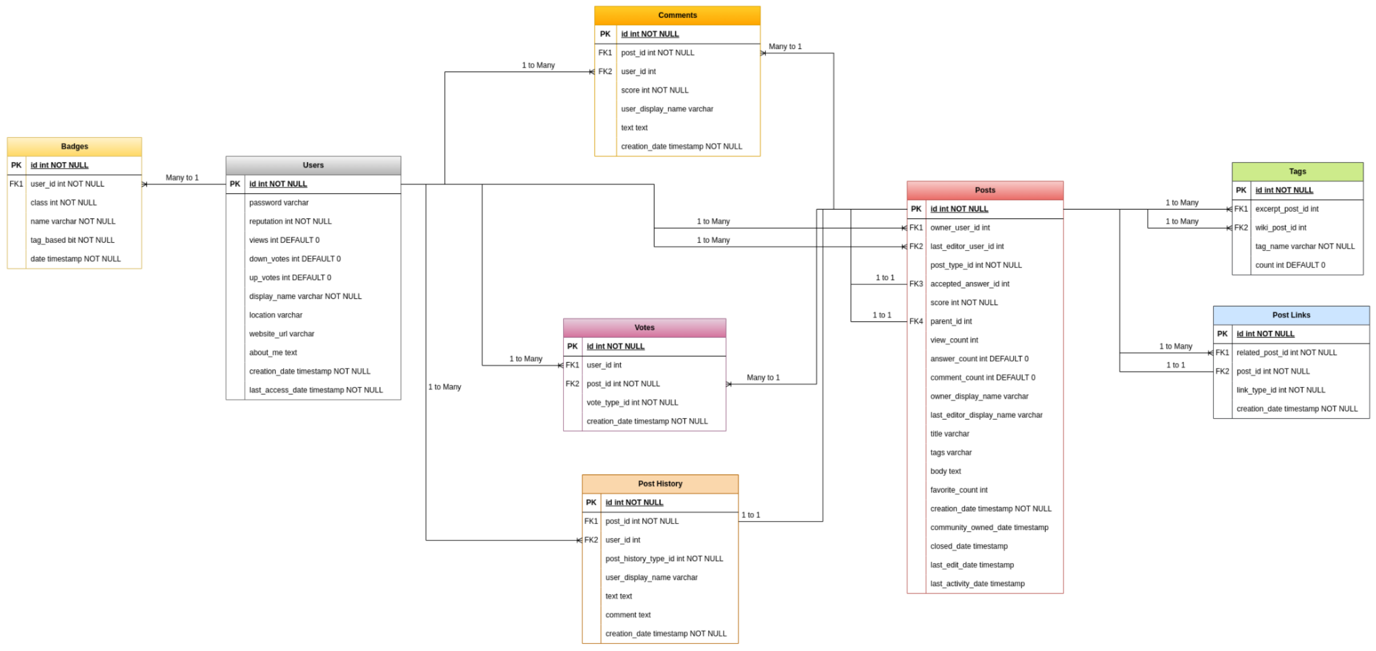
8.2 <u>Functional Dependencies</u>

The primary key i.e ‘id’ determines every other column.
In all FDs that hold on above table, id must be present in LHS (set of determiners).
There are no other FDs. As id is primary key, all determiners have to be superkeys, thus, above table is in BCNF. Some FDs are:

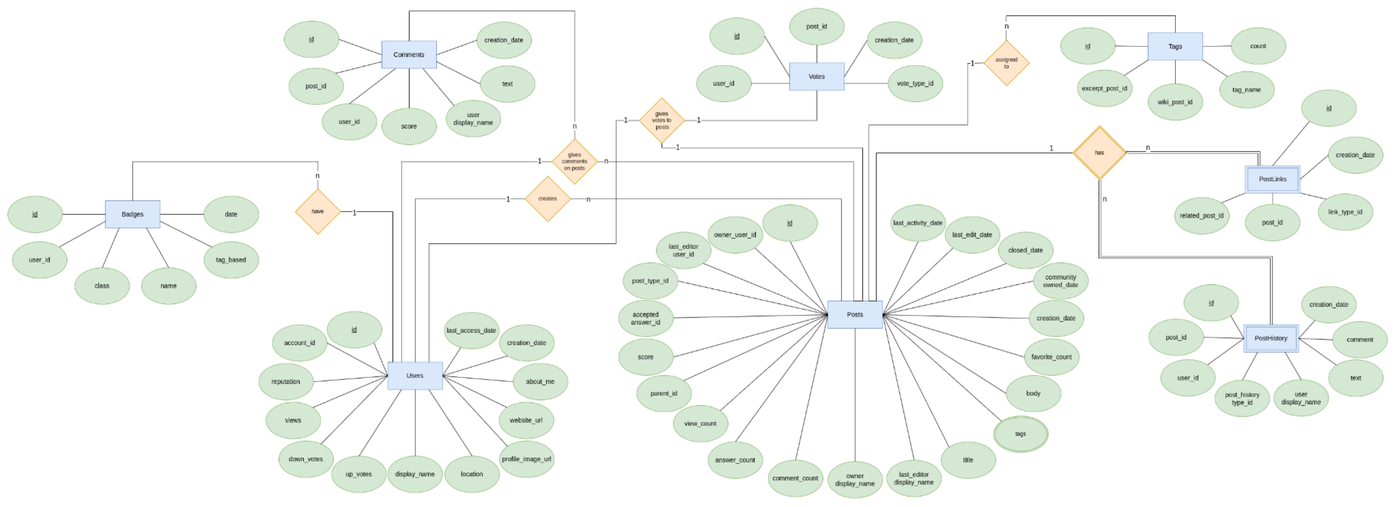
Id -> reputation
Id -> password
Id -> views
Id -> down_votes
Id -> up_votes
Id -> display_name
Id -> location
Id -> website_url
Id -> about_me
Id -> creation_date
Id -> last_access_date

Normalization:
As can be seen from the FDs that hold on above tables, all the determiners (LHS) are superkeys. Thus, all tables are in BCNF. Thus, no further Normalization is required.

9. Schema



10. Entity Relationship Diagram



11. Github Repository

https://github.com/siddharths00/MetaData