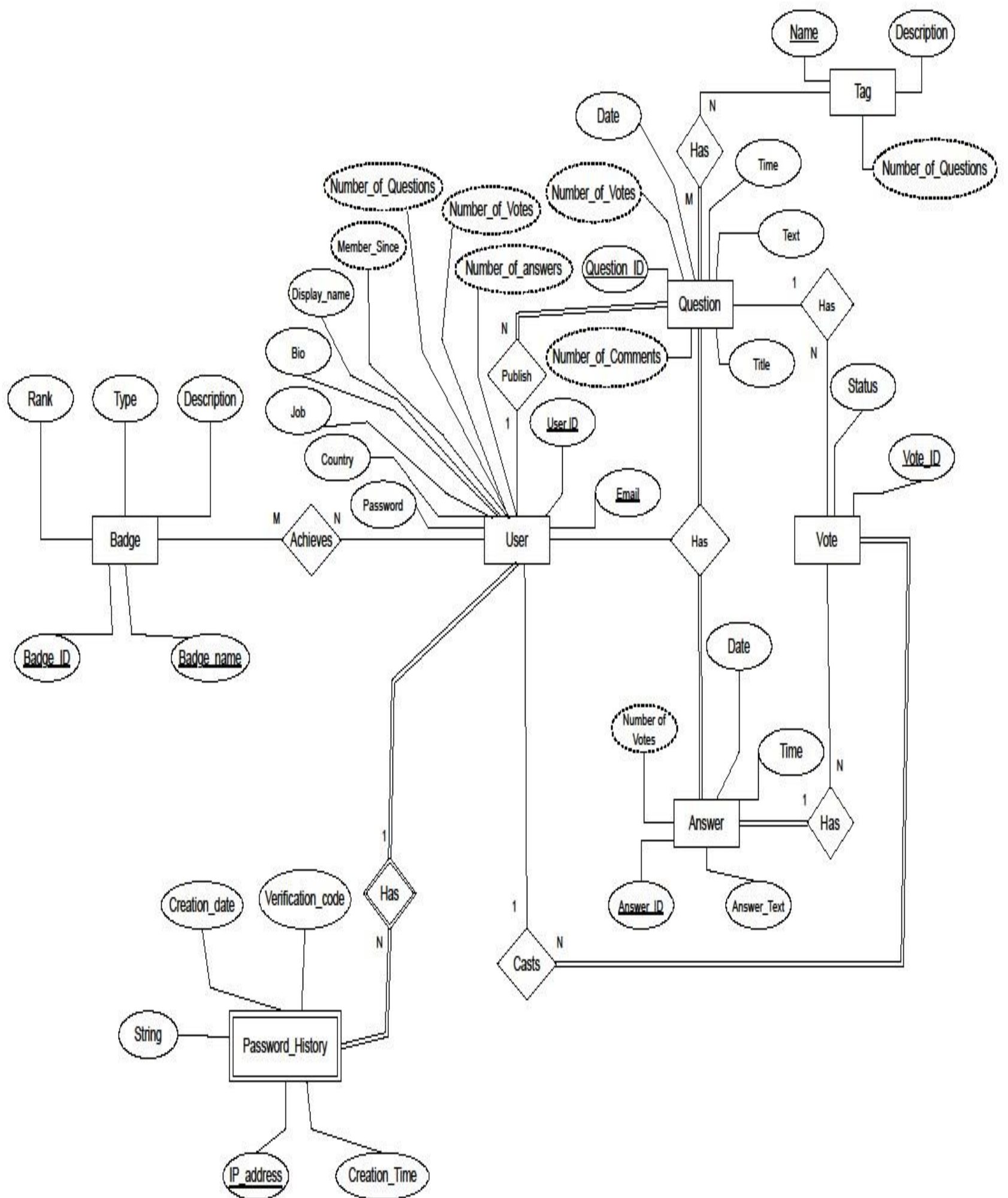


ERD



Relational Schema

USER

<u>user_id</u>	Email	Password	Country	Job	Bio	display_name
----------------	-------	----------	---------	-----	-----	--------------

BADGE

<u>Badge_ID</u>	Badge_name	Rank	Type	Description
-----------------	------------	------	------	-------------

VOTE

<u>Vote_ID</u>	Satutes	User_ID	Question_ID	Answer_ID
----------------	---------	---------	-------------	-----------

ANSWER

<u>Answer_ID</u>	Answer_Text	Date	Time
------------------	-------------	------	------

TAGS

<u>Name</u>	Description
-------------	-------------

QUESTION

<u>Question_ID</u>	Text_Title	Date	Time	User_id
--------------------	------------	------	------	---------

PASSWORD HISTORY

<u>IP_address</u>	<u>User_ID</u>	String	Creation_Date	Verification_Code	Creation_Time
-------------------	----------------	--------	---------------	-------------------	---------------

ACHIEVES

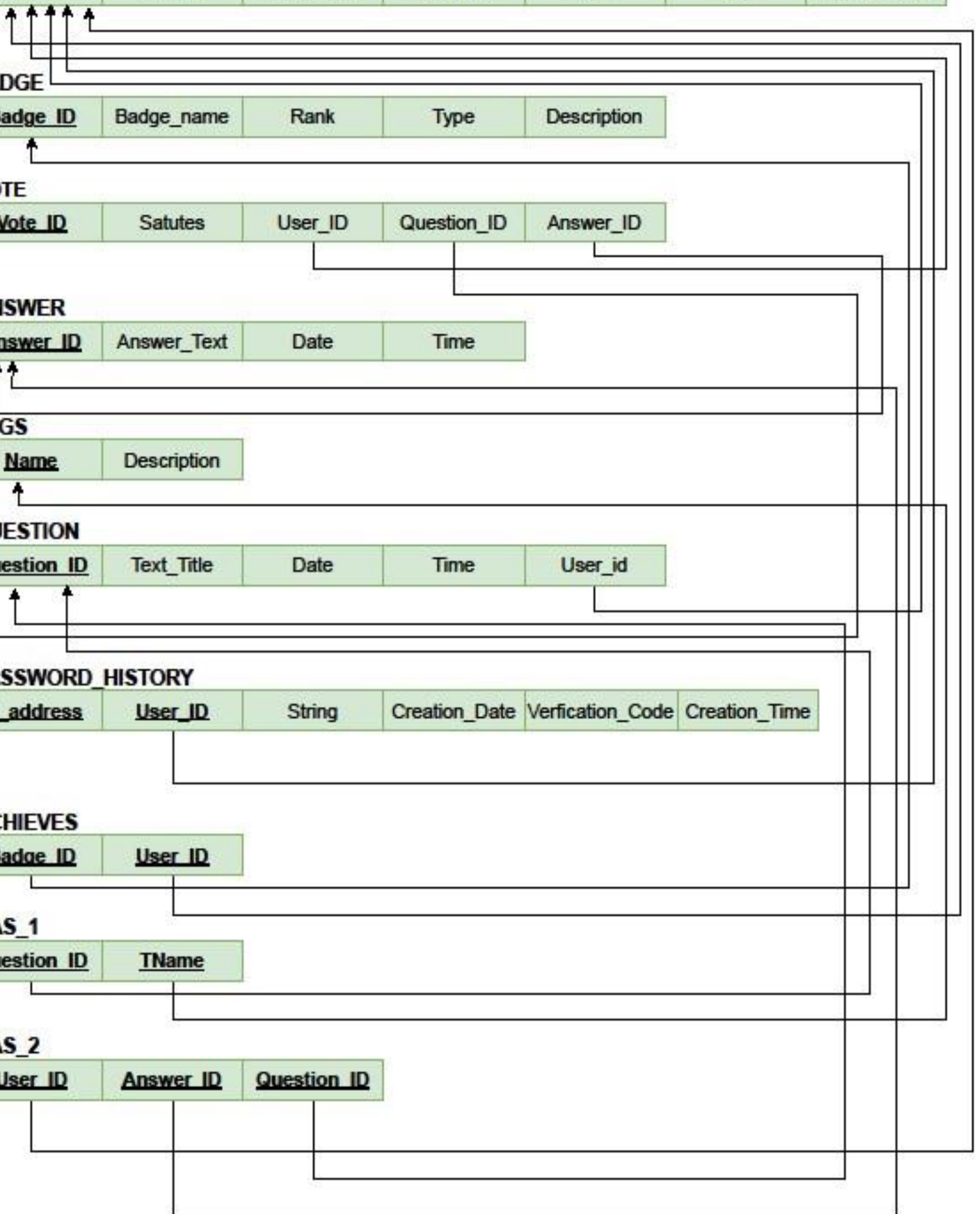
<u>Badge_ID</u>	<u>User_ID</u>
-----------------	----------------

HAS_1

<u>Question_ID</u>	<u>TName</u>
--------------------	--------------

HAS_2

<u>User_ID</u>	<u>Answer_ID</u>	<u>Question_ID</u>
----------------	------------------	--------------------



Project Description

We need to create a database schema design for **Problem Corner (Stack Overflow)**.

Stack Overflow is a question and answer website for professional and enthusiast programmers.

It features questions and answers on a wide range of topics in computer programming.

- We store each user (ID , E-mail , Display name , Bio , Job , Country and Password) .
 - The (Number of Votes , Number of Questions , Number of Answers and the Period that has passed since the user registered on the site) will be calculated .
- We store each Question (ID , title , Text , Date and Time) .
 - The (Number of Votes on the Question and Number of Answers) will be calculated .
- We store each Tag (Name and Description) .
 - The Number of Questions has the Tag will be calculated .
- We Store each Answer (ID , Text , Date and Time) .
 - The Number of Votes on every answer will be calculated .
- We store each old Password (String , Verification code , The IP address of the user who changed it , Creation date and Creation Time) .
- We store each Vote (ID and Status) .
- We store each Badge (ID , Name , Rank , Type and Description) .
- The User may publish Question/s [1 : N relation] (one user can ask more than one question but each questions is asked by only one user)
- The User may has Question/s , The Question must have an asker who is the user , each Answer must be assigned to a Question and must have a publisher who is the user (Question may have many answers , each Question has only one asker and each answer has only one publisher)
- The User must has Password History [1 : N relation] (one User may has many old passwords but the Password history is assigned to only one User) .
- The User may achieve Badge/s [M : N relation] (many users can achieve same badge and one user can achieve more than one Badge)
- The User may cast Vote/s but the vote must have a user to cast it [1 : N relation] (one User can cast more than one vote but each vote is casted by only one user) .
- Answer may has Vote/s [1 : N relation] (one Answer can has many votes but each vote is casted on one Answer only)
- Question may has Vote/s [1 : N relation] (one Question can has many votes but each vote is casted on one question only)
- Question Must has Tags [M : N relation] (One Question may has many tags and same tag may assign to more than one Question)

-
- To open The ERD as Pdf Click [here](#) .
 - To open the Relational Schema as Pdf Click [here](#) .