

DBMS PROJECT REPORT

Project Name: NITKUORA, a question answer forum.

Problem description:

A database management system for a question answer forum based on the following requirements-

-The forum has many USERS:

- Each User has a ID, Name, Email, Password, Topic_IDs of Topics followed, Number of Followers
- Each User can follow many Topics
- Each User may also follow other Users
- Each User can post Questions
- Each User posts Questions under a topic
- Each User also associates 3 keywords from the options chosen
- Each User can write Answers
- Each User can write Comments for the Answers
- Each User can Upvote/ Downvote Answers
- Each User can Search Topics through keywords
- Each User has a User Profile associated with it which he can sign into and modify

-The forum has many TOPICS:

- Each Topic has a Topic_ID, Name, and Number of Followers
- Each Topic has many questions under it
- Each Topic has many followers

-Each Topic has many QUESTIONS:

- Each Question has a Question_ID, Question, Topic_ID, User_ID and Keyword_IDs.
- Each Question is posted under a Topic and thus associated with a Topic_ID.

- Each Question has a user who asked that question and thus associated with a User_ID.
- Each Question can have many Answers.
- Each Question can be searched for using a few Keywords. Thus, Keyword_IDs.

-Each Question has many ANSWERS :

- Each answer has an Answer_ID ,User_ID(user who answered), Number of Upvotes, Question_ID, Number of Downvotes, Number of comments, Answer

- Each Question can have many Answers
- Each Answer can have many comments
- Each Answer can have many upvotes
- Each Answer can have many downvotes

-Each Answer has many UPVOTES:

- Each upvote has an Upvote_ID, Answer_ID, User_ID.
- Each Answer can have upvotes. Thus, it is associated with an Answer_ID.
- Each User may upvote an Answer. Thus, it is associated with a User_ID.

-Each Answer has many DOWNVOTES:

- Each downvote has a Downvote_ID, Answer_ID, User_ID
- Each Answer can have downvotes. Thus, it is associated with an Answer_ID.
- Each User may Downvote an answer.Thus, it is associated with User_ID.

-Each Answer can have COMMENTS:

- Each comment has a Comment_ID, Comment, Answer_ID, User_ID.
- Comment can be written for each answer. Thus, it is associated with a Answer_ID.
- Comment is written by a user. Thus, it is associated with a User_ID.

-The application has a FOLLOWER-FOLLOWINGnetwork:

- Each User can follow many Users
- Each User can be followed by many Users
- This table contains UserID1 of the person being followed and UserID2 of the person following them and also a FOLLOW_ID.

-The application also has a KEYWORD Table:

- Each row in the Keyword table contains a Keyword_ID, Keyword, Topic_ID.
- Each Topic has a few Keywords associated with it. Thus, Topic_ID.

-The application also has an option to SEARCH:

- A user can search for a question based on keywords.
- Questions with Answers are retrieved by the application

Major queries:

1. select * from audit where user2_id='\$user_id' or answer_id in (select answer_id from answer where a_question_id in (select question_id from question where q_user_id = '\$user_id')) or comment_id in (select comment_id from comment where c_answer_id in (select answer_id from answer where a_user_id='\$user_id')) or comment_id in (select comment_id from comment where c_answer_id in (select answer_id from answer where a_question_id in (select question_id from question where q_user_id='\$user_id'))))
2. select * from answer where a_question_id="\$q_id_1" and no_upvotes in(select max(no_upvotes) from answer where a_question_id="\$q_id_1")
3. select * from question where question_id in(select question_id from topic_question where topic_id in(select topic_id from topic where topic_name="\$topic_name"))

Major Constraints:

Primary Key constraints-

1. User_ID is the primary key for USER table.
2. Question_ID is the primary key for QUESTION table.
3. Answer_ID is the primary key for ANSWER table.
4. Comment_ID is the primary key for COMMENT table.
5. Upvote_ID is the primary key for UPVOTE table.
6. Downvote_ID is the primary key for DOWNVOTE table.
7. Topic_ID is the primary key for TOPIC table.

8. Topic_ID, Question_ID is the primary key for TOPIC-QUESTION table.
9. User1_ID, User2_ID is the primary key for FOLLOWER-FOLLOWING table.
10. User_ID, Topic_ID is the primary key for FOLLOWER-TOPIC table.

Foreign key constraints-

1. QUESTION table has a foreign key Q_User_ID referencing the USER table.
2. ANSWER table has a foreign key A_Question_ID referencing the QUESTION table.
3. ANSWER table has a foreign key A_User_ID referencing the USER table.
4. COMMENT table has a foreign key C_User_ID referencing the USER table.
5. COMMENT table has a foreign key C_Answer_ID referencing the ANSWER table.
6. UPVOTE table has a foreign key U_User_ID referencing the USER table.
7. UPVOTE table has a foreign key U_Answer_ID referencing the ANSWER table.
8. DOWNVOTE table has a foreign key D_User_ID referencing the USER table.
9. DOWNVOTE table has a foreign key D_Answer_ID referencing the ANSWER table.
10. TOPIC-QUESTION table has a foreign key Topic_ID referencing the TOPIC table.
11. TOPIC-QUESTION table has a foreign key Question_ID referencing the QUESTION table.
12. FOLLOWER-FOLLOWING table has foreign keys User1_ID, User2_ID referencing the USER table.
13. FOLLOWER-TOPIC table has a foreign key User_ID referencing the USER table.
14. FOLLOWER-TOPIC table has a foreign key Topic_ID referencing the Topic table.

Structural Constraints-

1. ASKS, a 1:N relationship type between USER and QUESTION. USER participation is partial. QUESTION participation is total.
2. ANSWERS, a 1:N relationship type between USER and ANSWER. USER participation is partial. ANSWER participation is total.
3. COMMENTS, a 1:N relationship type between USER and COMMENT. USER participation is partial. COMMENT participation is total.
4. UPVOTES, a 1:N relationship type between USER and UPVOTE. USER participation is partial. UPVOTE participation is total.
5. DOWNVOTES, a 1:N relationship type between USER and DOWNVOTE. USER participation is partial. DOWNVOTE participation is total.
6. FOLLOWS, a 1:N relationship type between USER and TOPIC. USER participation is partial. TOPIC participation is total.
7. HAS_ANSWERS, a 1:N relationship type between QUESTION and ANSWER. QUESTION participation is partial. ANSWER participation is total.
8. HAS_COMMENTS, a 1:N relationship type between ANSWER and COMMENT. ANSWER participation is partial. COMMENT participation is total.
9. HAS_UPVOTES, a 1:N relationship type between ANSWER and UPVOTE. ANSWER participation is partial. UPVOTE participation is total.
10. HAS_DOWNVOTES, a 1:N relationship type between ANSWER and DOWNVOTE. ANSWER participation is partial. DOWNVOTE participation is total.
11. HAS_TOPIC, an M:N relationship type between QUESTION and TOPIC. TOPIC participation is partial. QUESTION participation is total. A question may be associated with many topics. A topic can have many questions. A topic need not always have a question.

Other Constraints:

The database has various **transition constraints**. For example, the number of upvotes or downvotes can only increase. There are other **semantic integrity** constraints like if a user follows another user, the number of followers of the other user must get incremented. These type of constraints have been implemented using **triggers**. Also, if a user's question has got an answer, then the user must be notified and a suitable entry must be done in the audit table. The audit table is a dynamic table and an **event** that **triggers** a **procedure** that clears the audit table after certain scheduled time. This prevents overloading the database.