

Coursework 2: Group Report

24978 97395

May 9, 2019

Task 1: Schema Design

Task 2: Method Implementation

getUsers

For this method the database is queried to return all the pairs of names and usernames in the database. These pairs are then iterated through and added to the hash map which is then returned.

getPersonView

addNewPerson

getSimpleForums

createForum

getForums

This method uses a single query that retrieves the information for all forums. A complex join was required to retrieve the topic within a forum which has had a post made in it most recently. The results are then iterated through and added to the list of ForumSummaryViews. The topicId was retrieved as a string instead of an integer. This is to deal with the case where a forum has no topics within it. If the topicId string is not null it is then converted to an integer.

getForum

getSimpleTopic

getLatestPost

This method uses a single query to get the information needed to construct the required PostView, joins are used to get the latest post by choosing the post with the largest time posted. Joins are also used to get the number of likes the post has as well as the post number.

createPost

The method first checks that the text and username given as input are not null or empty. A query is used to check that the topic with the given ID exists and another query is used to check that a person with the given username exists. The post is then inserted into the database.

createTopic

This method begins by checking that the title and text arguments are not empty or null. A query is then used to retrieve the persons ID from there username. The topic is then inserted into the database. `RETURN_GENERATED_KEYS` is used

to find the ID of the topic which has been inserted into the database. This key is then used when inserting the initial post of the topic into the Post table in the database.

countPostsInTopic

likeTopic

likePost

The method checks that the person liking the post exists and gets the ID of this person if they exist. The existence of a topic with this ID is then checked. Then a query is done to check whether there is a database entry for this post being liked by this person. If there is and the `like` boolean is true then an error is returned as the post has already been liked. If there is and the `like` boolean is false then the entry is removed from the PostLikes table in the database. If there is no database entry, essentially the reverse is done, if `like` is true, an entry is inserted into the PostLikes table and if `like` is false then an error is returned.

getLikers

getTopic

This method begins by checking the existence of a topic with this ID in the database and if so retrieves the necessary information about the topic and the forum it is contained in. Another query is then used to get the posts contained within the topic, with a left outer join with the PostLikes table being used to get the number of likes each post has. The results are then iterated through and added to the list of PostViews.

getAdvancedPersonView

This method required complex SQL queries in order to avoid querying the database in a loop. In the first query two left outer joins are used to retrieve the the number of likes posts and topics created by this user have received. Another lengthy query is then used to get the rest of the required information. Separate joins were required to get the information about the creation of the topic, the information about the most recent post within the topic, the number of topic likes and number of post likes for all the topics which have been liked by the user. These results are then iterated through and added to the list of TopicSummaryViews.