12 May 2022

Online Forum Database System

Pengxiang Huang, Han Linghu, Zihan Li, Boyi Chen

School of Data Science, The Chinese University of Hong Kong, Shenzhen

Abstract--These remain the abstract

Index Terms—Remain the index term

—————————— ◆ ——————————

1. Background & Introduction

Our group observe that many student-programmers from CUHKSZ may encounter many familiar programming questions in their project or assignment. Their solutions mainly include: searching online, emailing TAs or professors, uploading questions in WeChat group, or assigning an office hour. Searching online sometimes may not be an efficient way, because the blog or some guidance information may not directly answer the assignment question. Even worse, programmers need to spend much time on filtering the huge amount of information and it becomes hard for them to get answers when the homework question is not relevant to the results on website. Uploading questions on WeChat group could get the detailed guidance and answer. But the new WeChat group will be created every semester for other students who may encounter the same problems. The connection between students already taken this course with the students taking this course right now is broken in this way. Raising questions in office hour is not convenient for programmers to solve their question immediately since they need to make an appointment and wait until that day comes. Therefore, our group would like to take the first step to change the current situation, to provide a online Q&A platform with the support of online forum database.

The main component of our project is the online forum database. There are two main entities in the database, namely "User" and "Blog". The "User" entity stores the account information about the user, including their name, email address, password, etc. The "Blog" entity is the key feature, where users can post a blog to raise a question and get relevant answers from other users. We also implement several relationship schemas to support the interaction between user and blog, like user can like a blog or follow a blog.

To make our database more efficient and space-saving, we made efforts to do normalization on tables and introduce index in our project. Additionally, we built fantastic UI in the frontend and robust backend server to hide the detailed implementation and manipulation of the database. So users can focus on the Q&A procedure, and they don't need to worry about how to get the desired information from the database, as all the queries are carried out by the functions in the frontend and backend.

1. Data Base Structure Design

*A. Requirement & Specification*

*B. Entity-Relation Diagram*

*C. Normalization*

*D. Index & Hashing*

1. SQL Function

This section is mainly about SQL functions. Many SQL requirements and their back-end SQL sentences and corre- sponding results will be demonstrated. They include opera- tional queries as well as analytic queries. All the functions are implemented in our system and can be accessed in the GUI.