COSC 3P32: Introduction to Database Systems

Group Project

Winter 2024

Checkpoint: Form your team (**up to 5 people**) and upload to Brightspace per team**.**

Due date: April 30th, 2024, at 23:59. Submissions received after the deadline will not be accepted**!**

Project Specification

This document describes the functional and non-functional requirements of Message Hub, a mobile instant messaging application. Message Hub is designed to leverage the contacts on a user's mobile device to facilitate text messaging services, supporting a variety of message types, including text, audio, video, and documents (PDF, WORD, etc.).

Requirements:

- **User Management**: Users have the ability to import contacts, send messages individually or in groups, and join or create up to 100 groups. Each user is identified by a unique ID, and their personal information such as phone number, name, bio, email, profile pictures, location, and availability status. Users may also choose a unique username for contact purposes without disclosing their phone number or ID.
- **Group and Channel Dynamics**: Groups, with a capacity of up to 2000 members, allow for message pinning and general participation. Channels, designed for larger audiences, limit posting privileges to creators and admins but offer unlimited membership. Each message within a channel features a view counter and maintains the channel's identity. Both group types support multiple message formats.
- **Account Management**: Users can permanently delete their accounts, which eradicates all personal messages and contacts. However, any groups or channels initiated by the user will remain accessible to their designated administrators.
- **Interaction and Notification Controls**: MessageHub enables users to interact with messages, track conversation threads, and manage unread

message counts across chat types. Additionally, users can tailor notification settings for private chats, groups, and channels.

- **Message Editing and Deletion**: Post-publication, users retain the ability to edit or completely remove messages. Deleting a message will delete it for everyone.
- **Administrative Functions**: Group creators can assign administrative roles, offering varying levels of group management capabilities. Unique usernames and join links facilitate easy group access.
- **Server-Side Architecture**: The application operates without a local database, relying on server-side storage for all data transactions.

Tasks:

You must complete all of the following steps:

1. Develop an ER model detailing entity sets, relationship sets, keys, and constraints. Explain any necessary constraints that your ER diagram cannot model. **(15 pts)**
2. Based on the ER model, design a relational schema for the database. **(15 pts)**
3. For each relation, identify all functional dependencies that hold on the fields of that table. For each table, specify if that table is in BCNF, 3NF, or neither. If a table is not in BCNF, then attempt to find a BCNF decomposition that is both lossless-join and dependency-preserving. If this is not possible, then a lossless-join, dependency-preserving 3NF decomposition is acceptable. Clearly specify the resulting relational schema. **(5 extra pts)**
4. Write the SQL statements necessary to create the tables for the above database, capturing as many constraints as possible. **(20 pts)**
    a. You must ensure that all domain constraints, primary key constraints, and foreign key constraints are enforced.
    b. Your project must be able to support insertions, deletions, and updates for all data stored in the database while also ensuring that constraints are satisfied.
    c. Input some data into your tables. You should input enough data that it is possible to verify any query you might have, and also any constraints that must hold.

        i. The Users table (and possibly Groups, etc.) should contain your own information (your name, email, etc.) as well as some other random info.

5. The project must support the following queries: **(50 pts)**
    a. Retrieve the list of all users.
    b. Retrieve the list of all online users.
    c. Given a user (by phone number or unique ID or username), retrieve all information of the user.
    d. Given a user (by phone number, unique ID or username) retrieve all his/her chats (private chats, normal groups and channels)
    e. For a given chat, retrieve its metadata (chat title, bio, join link (if applicable), etc.)
    f. For a given chat, retrieve all its users.
    g. For a given chat, retrieve all its online users.
    h. For a given chat, retrieve its creator.
    i. For a given chat, retrieve all its admins (including the creator).
    j. For a given chat admin, retrieve his/her permissions.
    k. For a given chat, retrieve all its message history
    l. For a given chat, retrieve its message during a specific date-time range.
    m. For a given chat, retrieve all messages posted by a user during a specific date-time range.
    n. For a given chat, retrieve its unread messages.
    o. For a given chat, retrieve the last n (say 100) message.
    p. For a given message ID, retrieve all its information.
    q. Important note concerning queries:
        i. The queries must be implemented either in PostgreSQL, MySQL, MS SQL Server, or Oracle.

Submission Requirements:

1. You must submit a report containing all of the following information.
    a. The cover page of the report including the names and ID numbers of all members in your group.
    b. The database schema and a list of all data in your tables.
    c. Screenshots of the queries as well as their results.
    d. A note on the division of labour within your group.
2. The report must be submitted as a single PDF file to Brightspace (under the Assignments section). Only one submission per group is allowed.