**Indexes:**

1. **To determine whether a chat id is grp or dm:**  Create index idx\_chats\_type on chats(chat\_type);
2. **To search for a group by its name:**  Create index idx\_groups\_name on dbms\_endsem\_project.groups(gname);
3. **To search for a user from its name:**  Create index idx\_name\_users on dbms\_endsem\_project.users(Name);
4. **To search for message content:**  Create index idx\_content\_messages on dbms\_endsem\_project.messages(content);
5. **To search for a message’s content inside a particular chat:** Create index idx\_chatcontent\_messages on dbms\_endsem\_project.messages(Chats\_chat\_id, content);
6. **To get receivers of a particular message of a particular chat:** Create index \_idx\_rec\_receive on dbms\_endsem\_project.users\_receive\_messages(Messgaes\_msg\_id, Messgaes\_Chats\_chat\_id);
7. **To check whether a user is an admin or not:** Create index idx\_isadmin\_uhc on alter\_ego.users\_has\_chats(admin);

**Triggers:**

**To populate users\_receive\_messages table whenever a new message entry is made to the message table:**

delimiter |

CREATE TRIGGER receiver\_fill AFTER INSERT ON messages

FOR EACH ROW

BEGIN

INSERT INTO users\_receive\_messages(users\_user\_id, messages\_chats\_chat\_id, messages\_msg\_id)

select distinct c.users\_user\_id, NEW.chats\_chat\_id, NEW.msg\_id from messages, users\_has\_chats as c

where NEW.chats\_chat\_id = c.chats\_chat\_id and c.users\_user\_id <> NEW.sender\_id;

END;

|

delimiter ;