University of British Columbia, Department of Computer Science

CPSC 304

Winter 2016 Term 2

Project Part : Final Project Submission

Group Members:

I don

Name	Student Number	Unix ID	Email Address
Allan Ching Lun Leung	35419143	l1a0b	allanleung1018@gmail.com
Bianca Subion	35132133	w1c0b	ubc.tea@gmail.com
Xu Shang	34372145	j3e0b	nerissasx@hotmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

A script that could be used to create all tables and data in the database such as the one here

Will Submit with Project

A short description of what the project accomplished

The project is a UBC Club Manager which allows club member to post and write message on the website.

A description of how your final schema differed from the schema you turned in. If the final schema differed, why? Note that turning in a final schema that's different from what you planned is fine, we just want to know what changed and why.

In the final schema, we added total user in clubs so we can do the nested average, min and max sql. This was suggested by TA Oliver because our data does not have data to aggregation query.

A list of the SQL queries used

From Clubs

SELECT * FROM Clubs WHERE id =?
SELECT * FROM Clubs
INSERT INTO Clubs SET ?
SELECT * from Members m INNER JOIN Members_Clubs mc ON m.id = mc.member_id INNER JOIN Clubs c ON mc.club_id = c.id AND m.username =?
SELECT m.username from Members m WHERE NOT EXISTS (SELECT * FROM Clubs c WHERE NOT EXISTS (SELECT * FROM Members_Clubs mc WHERE mc.member_id = m.id AND mc.club_id=c.id))
SELECT COUNT(*) FROM Clubs
SELECT COUNT(1) FROM Members_Clubs mc WHERE mc.member_id = ? AND mc.club_id = ?
INSERT INTO Members_Clubs SET ?
INSERT INTO Members_Clubs SET ?
select avg(total_members) from Clubs
select max(total_members) from Clubs
select min(total_members) from Clubs
select mc.member_id, avg(c.total_members) from Clubs c, Members_clubs mc where mc.club_id=c.id group by mc.member_id
select mc.member_id, max(c.total_members) from Clubs c, Members_clubs mc where mc.club_id=c.id group by mc.member_id
select mc.member_id, MIN(c.total_members) from Clubs c, Members_clubs mc where mc.club_id=c.id group by mc.member_id

From Events
SELECT * from Events e INNER JOIN Clubs c ON c.id = e.club_id AND c.id = ?
SELECT * Events from Events

From Members
SELECT * FROM Members WHERE username =?
INSERT INTO Members SET ?
select * from Clubs c INNER JOIN Members_Clubs mc ON mc.club_id = c.id INNER JOIN Members m ON mc.member_id = m.id AND c.name = ?
UPDATE Members SET student_id=? WHERE student_id=?

From Message

select * from Messages msg, Members m WHERE (msg.sender_id = m.id OR msg.receiver_id = m.id) AND m.username = ?
INSERT INTO Messages SET ?

From Posts

select p.id, p.body, m.name FROM Posts p INNER JOIN Clubs c ON c.id = p.club_id INNER JOIN Members m ON m.id = p.member_id AND c.id = ?
delete p FROM Posts p INNER JOIN Members_clubs mc ON p.member_id = mc.member_id AND p.club_id = mc.club_id WHERE id = ?
INSERT INTO Posts SET ?
select count(id) from posts p where p.member_id = ? and club_id = ?

List all functional dependencies that are applicable to the table (including the ones involving the primary key). For each functional dependency, briefly describe its meaning in English.

In our project, we have 6 tables and they are

Tables	Primary Key	Foreign Key
Clubs	id	
Members	id	
Events	id	club_id
Posts	id	Club_id, member_id
Members_Clubs	member_id	club_id
Messages	id	Sender_id, receiver_id

Functional Dependencies

- Events
    - Id -> club_id, date, description
    - ID is atomic and is generated when a member is created
    - Club_ID is created when a Club gets created
    - Date is a int
    - Description is a string so the user can describe the post
- Posts
    - Id -> club_id, body
    - Club_ID is the post for the club ID
    - Body is the body of text the post is about.
- Member_Clubs
    - member_id, club_id -> admin
    - Admin is a boolean, true means they created the Club, false means they are just norml member
- Messages
    - Message_id -> sender_id, receiver_id, body
    - Sender_ID is the ID
    - Receiver_ID is the ID the
- Members
    - Id -> firstname, lastname, department, student_id
    - Firstname is first name of the user
    - Lastname is last name of the user
    - Department is the departmemt of the user
    - Student_id is the student number of the user
- Clubs
    - Id -> name, total_members
    - Name is the name of the club
    - Total_members is the total member of the club