CS 5318: Database Management Systems Project

Project Name:

Database for Girl's Scout Troop

Project Partner:

Thomas Deleon

Shamima Haque

Melissa Raje

Abstract:

The database created is a tool that is helpful for members, and everyone who wants to access the database. In the current system, all the activities are done manually which is time-consuming and costly. In the software, we can register either as a member, parent, or troop leader (technical support). The troop leader/technical support has the power to add members and edit or delete users. A member can register as a user and can view their meeting, view events, badge requirements, badge earned, etc. However, a member can't manipulate their profile. Parents can query about their children's performance and can view the meetings, events

MySQL, google PowerPoint, and Microsoft word was used as supporting application to work on this project.

The team-viewer application was used by members for collaboration

Normalization and SQL statements were used to generate the database functionalities

Link to the database demo:
https://drive.google.com/file/d/1MrxQHF-Momt2zCM-n9DWd76DZsL3L7GA/view?usp=sharing

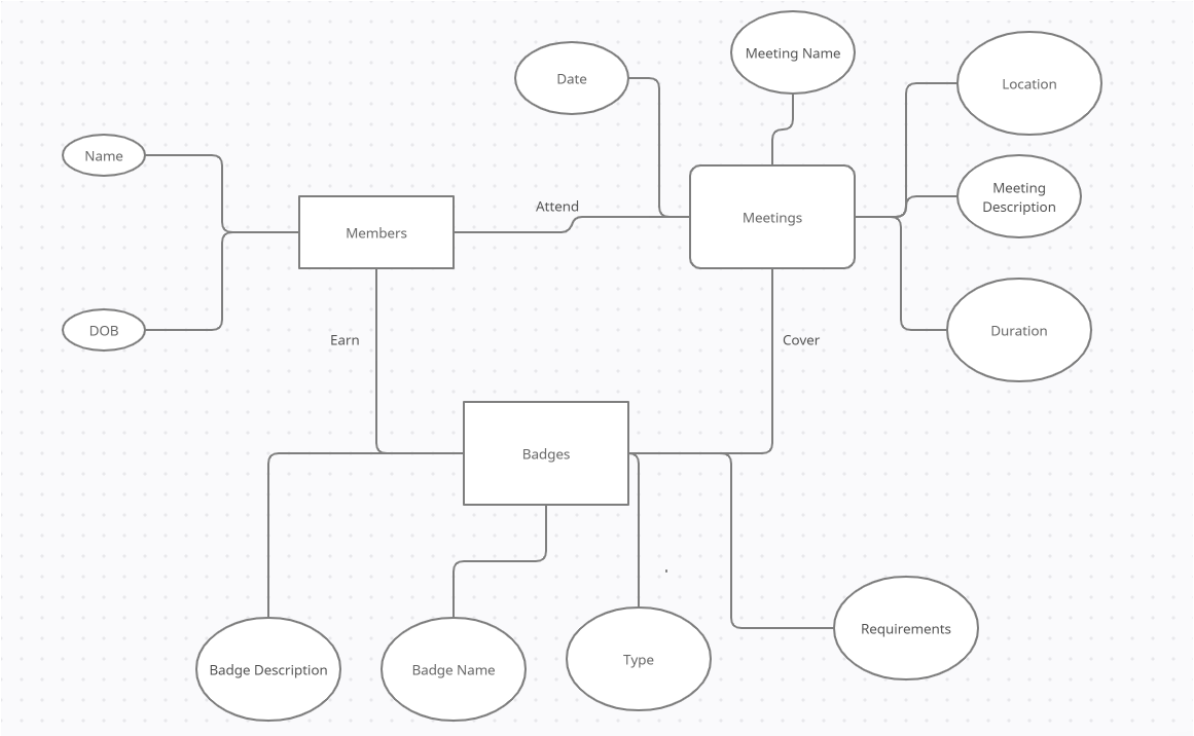
Mission Statement:

The purpose of a database system for a girl scout troop is to create a centralized location for the meetings, badges, troop roster, and attendance for the troop.

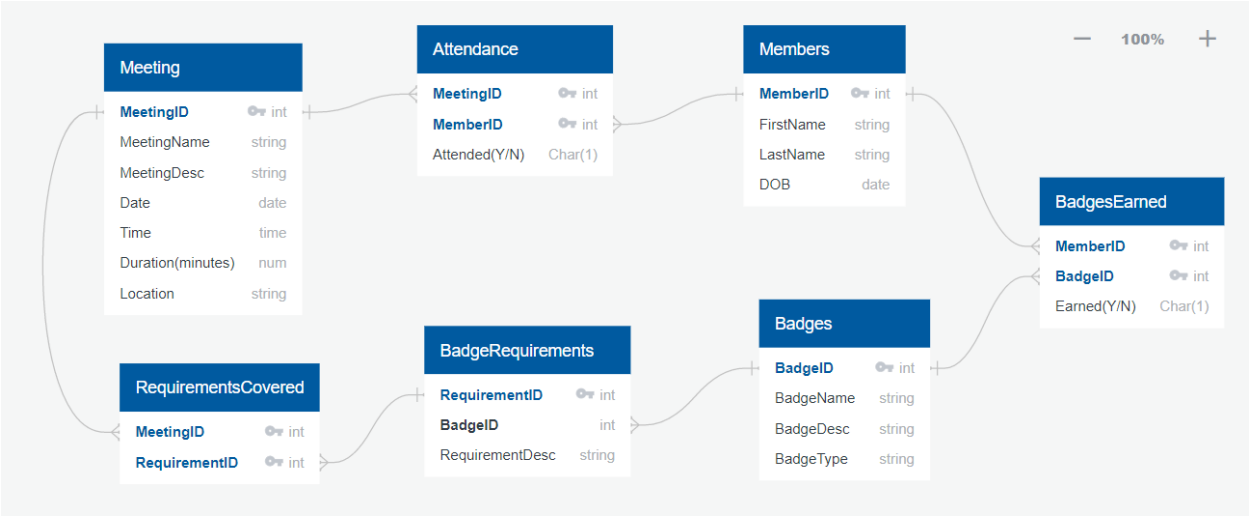
Mission Objectives: the objective of this project is to

- Record meeting details
- Records of the troop roster (girls and adults)
- Show attendance of the meetings and the badge requirements covered in each meeting
- Keep track of the badges earned by each member, by having attended all required meetings or performed the activities on their own
- Have parents view their child's meeting and badge activity separately

E/R diagram:

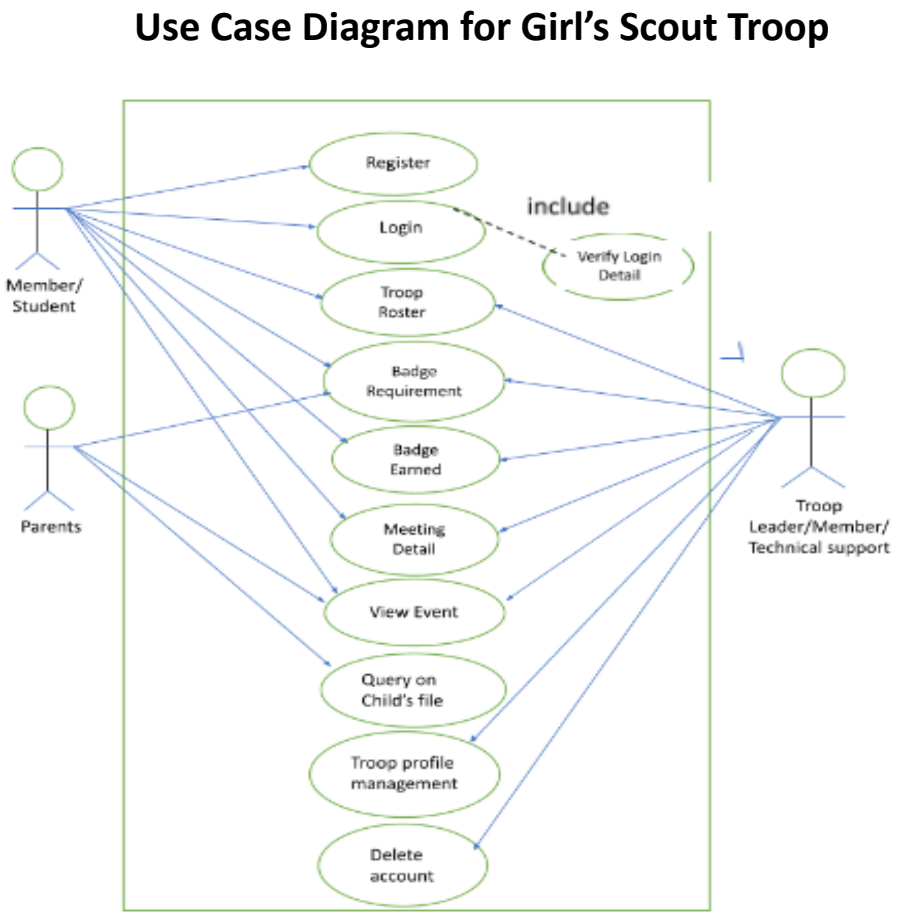


Normalization:



These 7 tables are in 3NF form. They are in 3NF because there are no dependencies and all information is in its own column.

Use Case:



STEP BY STEP EXPLANATION OF USE CASE DIAGRAM

IDENTIFYING THE SYSTEM ACTORS: Member/students, parents, technical support/member/troop leader.

IDENTIFYING THE GOALS OF THE ACTORS:

- **Member/Students:** They are the reasons the database exists**.** Members/students who register can access their database as long as they have the required username and password. They can have access to the following items: troop roster, meeting detail, badge requirement, badge earned, view meeting, and view event.

- **Parents:** Parents can view their children's profiles. They can only view necessary information applicable to them like event view, badges earned by their child, and badge requirements.

- **Troop Leader/Member/ Technical support:** Troop Leader/ Technical support can have access to their own personal profile, they can update the troop roster, can manage troop profile records, and they can have access to their meeting records, badges earned records, and even modify or delete any record/profile. creates or publishes news and announcements, and views all information in the system.

SQL Project Test Queries

#Query Join and Aggregate

```
SELECT BadgesEarned.BadgeID, BadgeName,  Count(MemberID) As Num_of_Members
FROM Scoutsdb1.BadgesEarned
Left Join Badges On BadgesEarned.BadgeID = Badges.BadgeID
Where Earned= 'Yes'
Group By BadgeID;
```

#SelectAll Attendance Table

```
SELECT * FROM Scoutsdb1.Attendance;
```

#Insert for Attendance table

```
INSERT INTO `Scoutsdb1`.`Attendance` (`MeetingID`, `MemberID`, `Attended Yes/No`)
VALUES ('104', '2', 'no');
```

#Update Attendance Table

```
UPDATE `Scoutsdb1`.`Attendance` SET `MeetingID` = '110' WHERE (`MeetingID` = '104') and
(`MemberID` = '2');
```

#Deletion on Attendance Table

DELETE FROM `Scoutsdb1`.`Attendance` WHERE (`MeetingID` = '104') and (`MemberID` = '2');

#SelectAll BadgeRequirements Table

SELECT * FROM Scoutsdb1.BadgeRequirements;

#Insert BadgeRequirements Table

INSERT INTO `Scoutsdb1`.`BadgeRequirements` (`RequirementID`, `BadgeID`, `RequirementDesc`) VALUES ('Req.2.4', 'S002', 'Test Insert');

#Update BadgeRequirments table

UPDATE `Scoutsdb1`.`BadgeRequirements` SET `RequirementDesc` = 'Test update' WHERE (`RequirementID` = 'Req.2.4');

#Deletion BadgeRequirements Table

DELETE FROM `Scoutsdb1`.`BadgeRequirements` WHERE (`RequirementID` = 'Req.2.4');

#SelectAll Badges Table

SELECT * FROM Scoutsdb1.Badges;

#Insert Badges Table

INSERT INTO `Scoutsdb1`.`Badges` (`BadgeID`, `BadgeName`, `BadgeDesc`, `BadgeType`) VALUES ('S003', 'Test', 'Test', 'Test');

#Update Badges Table

UPDATE `Scoutsdb1`.`Badges` SET `BadgeDesc` = 'Update test' WHERE (`BadgeID` = 'S003');

#Deletion Badges Tables

```
DELETE FROM `Scoutsdb1`.`Badges` WHERE (`BadgeID` = 'S003');
```

#SelectAll BadgesEarned Table

```
SELECT * FROM Scoutsdb1.BadgesEarned;
```

#Insert BadgesEarned Table

```
INSERT INTO `Scoutsdb1`.`BadgesEarned` (`MemberID`, `BadgeID`, `Earned`) VALUES ('4', 'S003', 'No');
```

#Update BadgesEarned Table

```
UPDATE `Scoutsdb1`.`BadgesEarned` SET `MemberID` = '5' WHERE (`MemberID` = '4') and (`BadgeID` = 'S003');
```

#Deletion BadgesEarned Table

```
DELETE FROM `Scoutsdb1`.`BadgesEarned` WHERE (`MemberID` = '5') and (`BadgeID` = 'S003');
```

#SelectAll Meeting Table

```
SELECT * FROM Scoutsdb1.Meeting;
```

#Insert Meeting Table

```
INSERT INTO `Scoutsdb1`.`Meeting` (`MeetingID`, `MeetingName`, `MeetingDesc`, `Date`, `Duration(min)`, `Location`) VALUES ('104', 'Bagdge 3.1', 'test', '2022-03-01', '10', 'test');
```

#Update Meeting Table

```
UPDATE `Scoutsdb1`.`Meeting` SET `MeetingDesc` = 'update test', `Location` = ' update test' WHERE (`MeetingID` = '104');
```

#Deletion Meeting Table

```
DELETE FROM `Scoutsdb1`.`Meeting` WHERE (`MeetingID` = '104');
```

#SelectAll Members Table

```
SELECT * FROM Scoutsdb1.Members;
```

#Insert Members Table

```
INSERT INTO `Scoutsdb1`.`Members` (`MemberID`, `FirstName`, `LastName`, `DOB`)
VALUES ('4', 'Test', 'Test', '2009-02-02');
```

#Update Members Table

```
UPDATE `Scoutsdb1`.`Members` SET `FirstName` = 'Update' WHERE (`MemberID` = '4');
```

#Deletion Members Table

```
DELETE FROM `Scoutsdb1`.`Members` WHERE (`MemberID` = '4');
```

#SelectAll RequirmentsCovered Table

```
SELECT * FROM Scoutsdb1.RequirementsCovered;
```

#Insert RequirmentsCovered Table

```
INSERT INTO `Scoutsdb1`.`RequirementsCovered` (`MeetingID`, `RequirmentID`) VALUES
('104', 'Test');
```

#Update RequirmentsCovered Table

```
UPDATE `Scoutsdb1`.`RequirementsCovered` SET `RequirmentID` = 'Update' WHERE
(`MeetingID` = '104') and (`RequirmentID` = 'Test');
```

#Deletion   RequirmentsCovered Table

```
DELETE FROM `Scoutsdb1`.`RequirementsCovered` WHERE (`MeetingID` = '104') and
(`RequirmentID` = 'Update');
```

#Left Join attendance and Members

```
SELECT FirstName, LastName, MeetingID, Attended
FROM Scoutsdb1.Attendance
Left Join Members
ON Attendance.MemberID=Members.MemberID;
```

Aggregate Function Example

```
SELECT MeetingID, Count(Attended)
FROM Scoutsdb1.Attendance
Left Join Members
ON Attendance.MemberID=Members.MemberID
Where Attended= 'Yes'
Group by MeetingID;
```

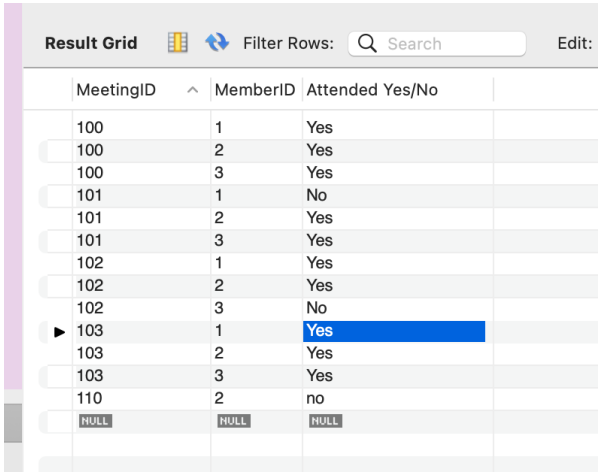
####Aggregate Function

```
Select BadgeName, Count(RequirementID) as Number_of_Requirments
From Badges
Left Join BadgeRequirements
ON Badges.BadgeID=BadgeRequirements.BadgeID
Group By BadgeName;
```

####### Two Left Join

```
Select MeetingDesc, RequirementDesc
From Meeting
Left join RequirementsCovered
On Meeting.MeetingID=RequirementsCovered.MeetingID
Left Join BadgeRequirements
ON RequirementsCovered.RequirmentID=BadgeRequirements.RequirementID;
```

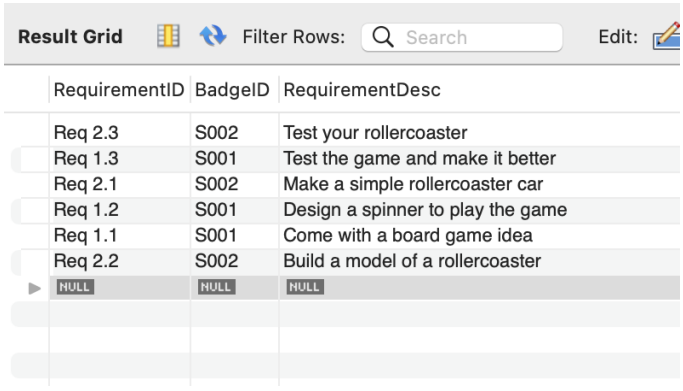
Attendance Table



Result Grid Filter Rows: Search Edit:

MeetingID	MemberID	Attended Yes/No
100	1	Yes
100	2	Yes
100	3	Yes
101	1	No
101	2	Yes
101	3	Yes
102	1	Yes
102	2	Yes
102	3	No
103	1	Yes
103	2	Yes
103	3	Yes
110	2	no
NULL	NULL	NULL

Badge Requirements table



Result Grid Filter Rows: Search Edit:

RequirementID	BadgeID	RequirementDesc
Req 2.3	S002	Test your rollercoaster
Req 1.3	S001	Test the game and make it better
Req 2.1	S002	Make a simple rollercoaster car
Req 1.2	S001	Design a spinner to play the game
Req 1.1	S001	Come with a board game idea
Req 2.2	S002	Build a model of a rollercoaster
NULL	NULL	NULL

Badges Table

Result Grid Filter Rows: Search Edit:

BadgeID	BadgeName	BadgeDesc	BadgeType
S001	BoardGame	Board Games Design	Mechanical Engineering
S002	RollerCoaster	RollerCoaster Design	Mechanical Engineering
NULL	NULL	NULL	NULL

Badges Earned Table

Result Grid Filter Rows:

MemberID	BadgeID	Earned
1	S001	No
1	S002	Yes
2	S001	Yes
2	S002	Yes
3	S001	Yes
3	S002	No
NULL	NULL	NULL

Meetings Table

MeetingID	MeetingName	MeetingDesc	Date	Duration(min)	Location
100	Badge 1.1	Starting the first badge	2022-01-01	60	Scout Leader House
101	Badge 1.2	Finishing first badge	2022-01-15	70	Scout Leader House
102	Badge 2.1	Starting second badge	2022-01-30	60	Scout Leader House
103	Badge 2.2	Finishing second badge	2022-02-14	55	Scout Leader House
NULL	NULL	NULL	NULL	NULL	NULL

Members Table

MemberID	FirstName	LastName	DOB
1	Atticus	Finch	2010-02-14
2	Moana	Waialiki	2010-12-06
3	Lilo	Stitch	2008-09-17
NULL	NULL	NULL	NULL

Requirements Covered

MeetingID	RequirmentID
100	Req 1.1
100	Req 1.2
101	Req 1.3
102	Req 2.1
103	Req 2.2
103	Req 2.3
NULL	NULL

Conclusion:

The focus of the database entailed documenting and tracking the girl scout badges earned by the troop members. The database consists of 7 relations which are named : Attendance, Badge Requirements, Badges, Badges Earned, Meetings, Members, Requirements Covered. It is set up for the data to be edited and/or updated when new scouts are added to the roster or are no longer with the girl scout troop. It allows for the troop leader to log meetings regarding the badges to be earned. The schema was successfully tested to perform insertion,updating, and deletion functions as well as joining tables as described.

References:

https://www.quickdatabasediagrams.com/

https://creately.com/