**CPSC 304 Project Cover Page**

Milestone #: 2

Date: October 24’Th, 2021

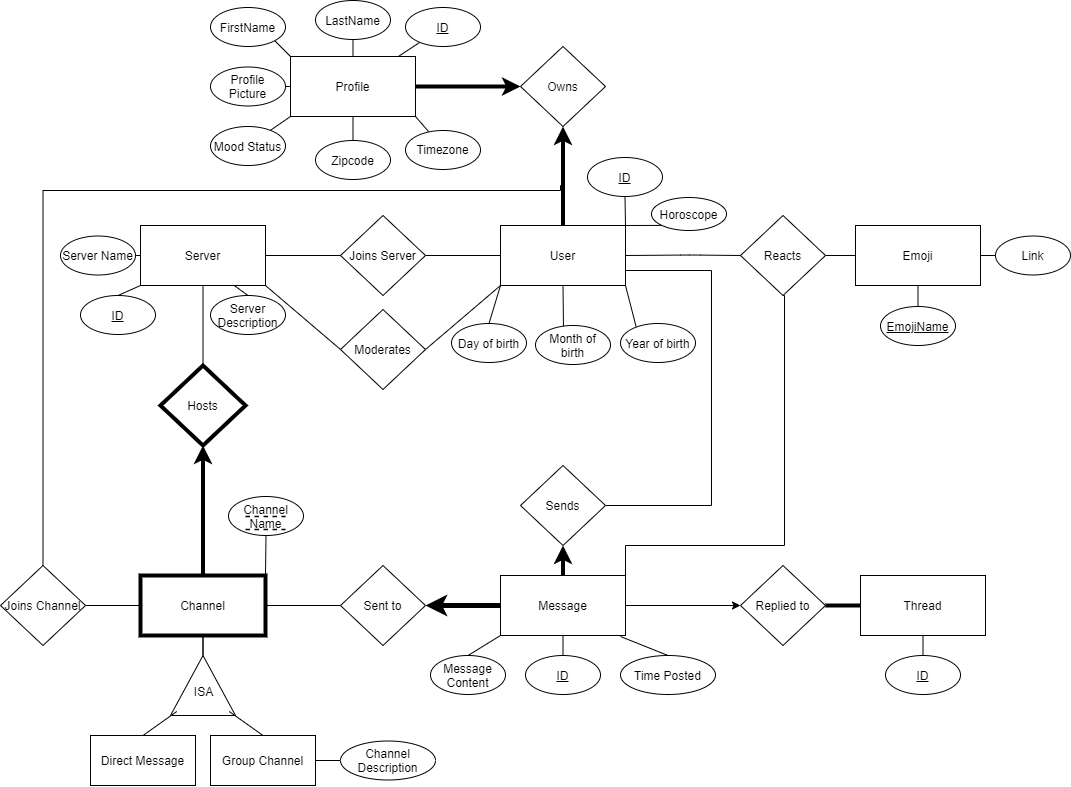
Group Number: 69

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Student Number** | **CS Alias (Userid)** | **Preferred E-mail Address** |
| Anton Grier | 53096566 | b8e2b | anty.grier@gmail.com |
| Kim Dao | 11177847 | f1m2b | kdaokdao@gmail.com |
| Ahnaf Muqset Haque | 68464171 | a4n2b | ahnafh321@yahoo.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 the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

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

1. **ER Diagram**



|  |  |
| --- | --- |
| **Change** | **Reason** |
| The “Has” relationship between Server and Channel is changed to “Hosts” | To improve descriptiveness |
| Group is changed to GroupChannel | To be more precise |
| The “Part of” relationship between Thread and Message is changed to “Replies”. | To improve descriptiveness |
| The ISA User relationship between Admin and Regular, is removed. The “Moderates” relationship between Admin and Server is moved to be between User and Server. | This removes two redundant relations, Admin and Regular. “Moderates” is changed to be between some User and Server, without a total participation requirement. |
| The “Adds” relationship between User and Emoji is removed. | We don’t want to implement this functionality anymore. |
| Removed relationship between Emoji and Server. | Decided to have the same emojis for all servers, so this will reduce redundancy, otherwise every server would have to map to all the same emojis. |
| Added total participation between Message and Channel. | Message cannot exist without a channel. |
| Added Zipcode and Timezone attributes to Profile. | Needed to add a functional dependency to decompose over |
| Added birth day, month, year and horoscope attributes to User. | Needed to add a functional dependency to decompose over |
| Have Profile reference User, instead of the other way around. | This is to ensure that any User will not be deleted, due to the ON DELETE NO ACTION taking place in Profile, referencing User |

1. Relational Schema

|  |  |
| --- | --- |
| Server | (ID: integer, ServerName: string, description: string) |
| Channel | (ChannelName: string, **ServerID**: integer) |
| GroupChannel | (**ChannelName**: string, **ServerID**: integer, Description: string) |
| DirectMessage | (**ChannelName**: string, **ServerID**: integer) |
| User | (ID: integer) |
| Profile | (ID: integer, **UserID**: integer, FirstName: string, LastName: string, ProfilePicture: string, MoodStatus: string) |
| Message | (ID: integer, **UserID**: integer, **ServerID**: integer, **ChannelName**: string, **ThreadID**: integer, MessageContent: string, TimePosted: integer) |
| Thread | (ID: integer) |
| JoinsServer | (**UserID**: integer, **ServerID**: integer) |
| JoinsChannel | (**UserID**: integer, **ServerID**: integer, **ChannelName**: string) |
| Moderates | (**UserID**: integer, **ServerID**: integer) |
| Emoji | (EmojiName: string, Link: string) |
| Reacts | (**UserID**: integer, **MessageID**: integer, **EmojiName**: integer) |

1. Functional Dependencies

|  |  |
| --- | --- |
| **Table** | **Functional Dependencies** |
| Users |  |
| Profile |  |
| Server |  |
| Message |  |
| Emoji |  |
| GroupChannel |  |
| DirectMessage | No functional dependencies, as Primary key is the only attribute |
| Channel |
| Thread |
| Reacts |
| Moderates |
| JoinsServer |
| JoinsChannel |

1. Normalization

We will be decomposing using BCNF. The only relations that break BCNF are **User** and **Profile**.

**User Decomposition:**

The first Functional Dependency is in BCNF, as *ID* is a superkey. The second functional dependency is not in BCNF, because the left-hand side is not a superkey. We will decompose based on the second FD.

|  |  |
| --- | --- |
| *User (ID, birthDay, birthMonth, birthYear, horoscope)* | |
|  | |
| ***User*** *(ID, birthDay, birthMonth, birthYear)* | ***Horoscopes*** *(birthDay, birthMonth, horoscope)* |

**Profile Decomposition:**

The first Functional Dependency is in BCNF, as *ID* is a superkey. The second functional dependency is not in BCNF, because the left-hand side is not a superkey. We will decompose based on the second FD.

|  |  |
| --- | --- |
| *Profile (ID,* ***UserID****, firstName, lastName, profilePicture, zipcode, timezone)* | |
|  | |
| ***Profile*** *(ID,* ***UserID****, firstName, lastName, profilePicture, zipcode)* | ***Location*** *(zipcode, timezone)* |

1. SQL DDL

<https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_a4n2b_b8e2b_f1m2b/blob/master/milestones/m2/ddls.sql>

1. Relation Instances

<https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_a4n2b_b8e2b_f1m2b/blob/master/milestones/m2/tuples.sql>

1. Queries

**Insert Operations:**

**Insert**: add user profiles to Profile table

**Insert**: add users to User table

**Insert**: add emojis to Emoji table

**Delete Operation:**

**Delete:** delete message from Message table with specified ID/timestamp/text

**Delete:** delete emoji from Emoji table with specified name

**Update Operation:**

**Update:** update user horoscope with new value from User table where user ID matches the ID of the user for whom we want to update the horoscope

**Update:** update server name with new value from Server table where server ID matches the ID of the server for which we want to update the name

**Update:** update emoji link (changing how the emoji looks) with new value where the emoji name matches the name specified

**Update:** update profile picture with new value from Profile table where profile ID matches the ID of the profile for which we want to update profile picture

**Update:** update zipcode with new value from Profile table where profile ID matches the ID of the profile for which we want to update zipcode

**Selection Operation:**

**Selection**: select all (\*) from Emoji table

**Selection**: select all (\*) from Profile table

**Selection**: select all (\*) from Server table

**Selection**: select all (\*) from Moderates table

**Selection**: select all (\*) from Message table where it contains a specified timestamp

**Projection Operation:** (Haven’t learned it yet, however, the following is our best attempt for understanding how projection works)

**Projection**: select MessageContent, ChannelName from Message table where it contains a specified Timestamp

**Projection**: select FirstName, ProfilePicture and Zipcode from Profile where it contains a specified Timezone

**Join:** (Haven’t learned it yet)

**Aggregation with Group by:** (Haven’t learned it yet)

**Aggregation with Having:** (Haven’t learned it yet)

**Nested Aggregation with Group by:** (Haven’t learned it yet)

**Division:** (Haven’t learned it yet)