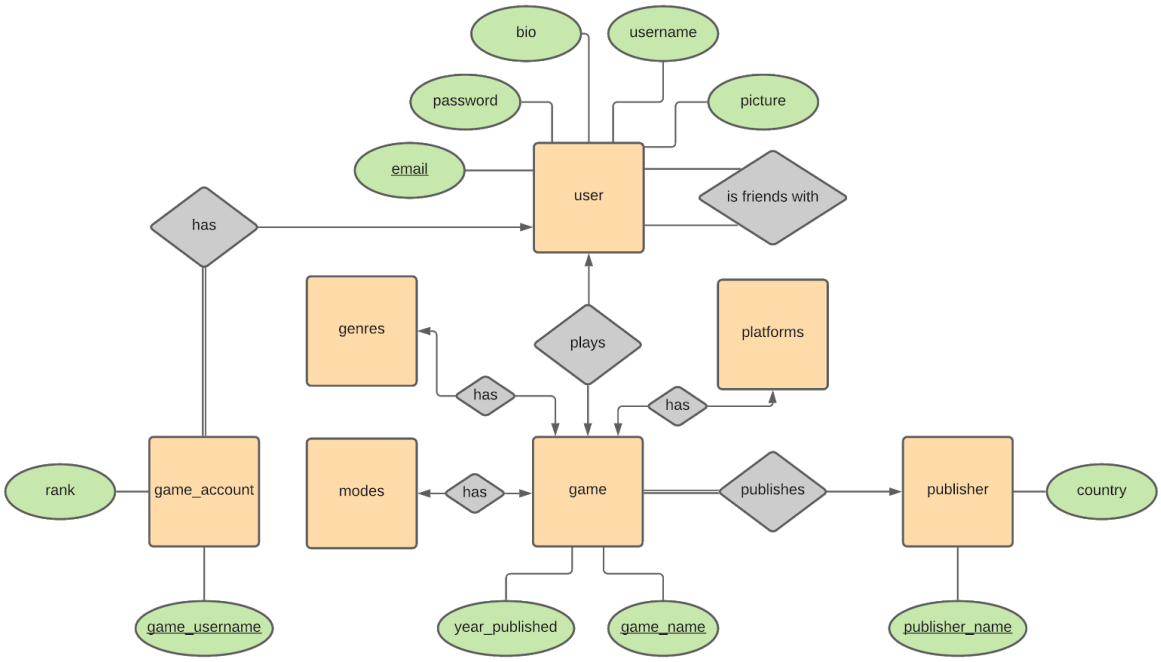
Final Database Project

Anita Ho (ahh2re), Liem Budzien (lb7me), Cassie Quach (cq5yc), Timothy Han (txh7es)

1. Database design
    - Final E-R diagram



    - Final Tables

        users(email, password, bio, username, picture, first_name, last_name)
        game(game_name, year_published, description)
        publisher(publisher_name, country)
        game_account(game_username, rank, game_name, username)
        is_friends_with(email_1, email_2)
        plays(email, game_name)
        publishes(game_name, publisher_name)
        favorites(username, game_name)
        has_genres(game_name, genre)
        has_modes(game_name, modes)
        has_platforms(game_name, platform)

2. Database programming
    - Specify where you host your database: Amazon RDS (Relational Database Service)
    - Specify where you host your app: Amazon EC2

- In order to deploy and run our project, we started an Linux AMI EC2 and an RDS instance on Amazon AWS. Then, we set up security groups around the RDS and EC2, only allowing HTTP code from the EC2 to access the RDS. Next, we placed an Apache server on the EC2, installed phpMyAdmin and proceeded to write our project in a git repo on the EC2. To access the app, you can go to http://ec2-54-91-203-237.compute-1.amazonaws.com/.
- Advanced SQL commands are used to log important changes to our database information. It is useful because it allows us to keep track of changes in the database and make adjustments accordingly. We also use advanced SQL commands to update multiple tables when end users add games to their library.

3. Describe the database security at the database level
    - Specify whether the security is set for developers or end users

        The security is set for developers.
    - Discuss how you set up security at the database level (access control)

        Security was set up on the database level by limiting what kind of traffic is allowed to access the database. Thus, we can limit the traffic to only trusted sources. For debugging purposes, we allowed the use of phpMyAdmin, but access to that service is also protected by username and password.
    - Submit the SQL commands you use to limited / set privileges such as

        ```
        GRANT admin-access ON users TO tutorial_user
        REVOKE admin-access ON users FROM lb7me
        ```

4. Describe the database security at the application level
    - Discuss how database security at the application level is incorporated in your project.

        We ensure that you must be logged in, in order to change your user information. You cannot change other user's information. We also used prepared/parameterized statements where text boxes are exposed to users to prevent sql injections. In addition, where standard queries are used, the text from forms are escaped and converted to plain strings in order to prevent any unwanted code from being run. We check on each page for the user logged in and do not allow them to change things, such as add games, if they are not logged in. We use session variables to determine the log-in status of a user, and if a user is not logged in they are unable to access any page that can edit database information. In other words, they have 'view-only' access.

- Submit code snippet(s) to illustrate how security aspect is implemented and to support your discussion.

```php
$username = $_GET["username"];
$usr = mysql_escape_string($username);
$password = $_GET["pwd"];
$pwd = mysql_escape_string($password);
```

```php
<?php
        $stmt = mysqli_stmt_init($connection);
        if(isset($_POST['submit'])) {
           $firstName = $_POST['firstName'];
           $lastName = $_POST['lastName'];
           $bio = $_POST['bio'];
           $query = "UPDATE users SET firstName=?, lastName=?, bio=?  WHERE email= '" . $email . "'";
           mysqli_stmt_prepare($stmt, $query);
           mysqli_stmt_bind_param($stmt, "sss", $firstName, $lastName, $bio);
           mysqli_stmt_execute($stmt);
           mysqli_stmt_close($stmt);
           echo 'Successfully saved! Please refresh';
        }
?>
```

```php
<?php session_start();
        if(isset($_SESSION['email']))
        {
    ?>

```

```php
<?php
            }
        else{
            header('Location: /loginPage/login.php');
        }
        ?>
```