Link: http://web.engr.oregonstate.edu/~setyawad/index.html

Project Name: Frozium

Team Members:

- Richard Villagrana
- Darian Wicaksana Setyawan

Project Proposal

a) Overview

The project proposal is a platform that serves as a one stop website that maintains up to date records on the communities of various video game titles. A person can view statistical information such as number of purchases based on years and view if a certain video game title has an active community. It will also harbor information such as player reviews and a synopses of said video game. Each video game has its own review and rating section that is useful for a person to view. In addition the database harbors information on content creators and merchandise of video games. This project is to give numerical values to what video games are popular and how many are actively playing the game. This would help people decide what games are still worth picking up after their time and get a sense if the player base is still alive.

The aim is to remove the need to surf endlessly to find this information and instead be a central hub of sorts similar to a library. There are over 30,000 titles on steam and many more on various platforms, many of those titles are long forgotten but there are resilient communities among them still active. Important information such as trends and number of users is important for multiplier games as they rely on an active server community, the same cannot be said for single player games however knowing that there is a community is helpful information. There are millions of players worldwide and this information is beneficial to users.

The collection method relies on a person to log in and self-report video game titles that they currently play, after a month the website resets and asks if the player still plays those titles removing them active if a user has ceased to play that game. This will keep the data relevant and in an advanced scenario the data is shared to the repository. The second method would require the handler to update the games when new titles are added but this process could be automated.

This solves the issues of having to search if a video game community is still active much less know if servers are still populated if the game is a multiplayer based game. Video game trends change rapidly and knowing if a video game is still played plays a large role in purchases. Alongside the website/database provides a service to have content creators share content relating to video games and allow for businesses to reach their targeted audience for their merchandise such as plushies, toys and collectables.

b) Database Outline

The table “games” , tracks all recorded video game titles. Storing information on genre, release date, Publishers, Developers, Modes, and Title. This is updated by database administrators such as bots inputting new entries or manual entries.

- Game: Videogame title
    - GameID - int, auto inc, not NULL, unique
    - Title - varchar, not NULL
    - PublisherID - int , not NULL
    - DeveloperID - int , not NULL
    - Modes - int, not NULL
    - Release data - date, not NULL
    - Genre - varchar, not NULL

Games share a M:M with Users because games can have multiple users while users can have multiple games

The table “Users:, tracks user reported data on videogame titles that a users play. Information stored here is User name, Game list

- User: Gamers
    - UserId - int - auto inc, not NULL, unique
    - Username - varchar, not NULL unqiue
    - Total games - int, not NULL
    - CC C- int fk
        - Content Creator ID forgien key to Content Creator table

Content Creator, tracks creators, they can choose to specify what game they create content for, tracksl, follower and if it is family friendly.

- Content Creator: Gamers/Creators
    - ID - int auto inc, not NULL , unique
    - CCname - varchar not NULL unique
    - Followers - int
    - Family friendly - varchar

Content Creator shares a M:1 with Users as there can be many content creators to one user but there can only be one user to a content creator handle.

Business tracks information on businesses that cater to gamer demographic through merchandise. Tracks , company name , what game they cater and shop content

- Business : Businesses
    - ID - int auto inc, unique, not NULL, pk
    - Cname - varchar not null
    - GameID - fk
    - ShopID - fk

Business shares a M:1 where business can have multiple shops but those shops can only be related to one shop.

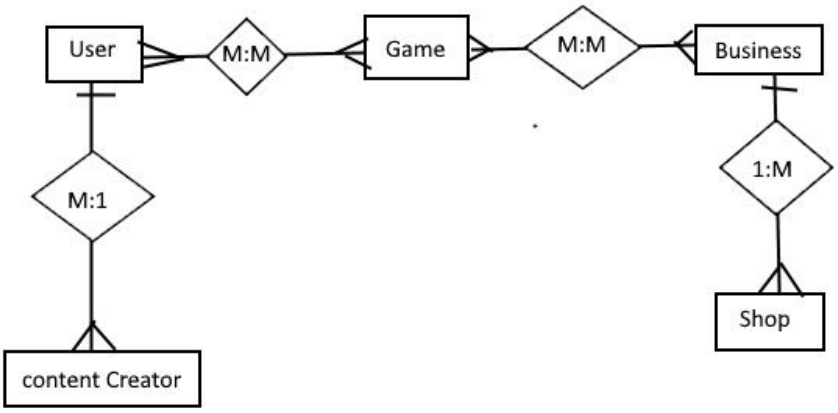
Businesses and Games share M:M as multiple businesses can be related to many games and games can be related to multiple businesses.

Shop entity tracks shop content offered through a business.

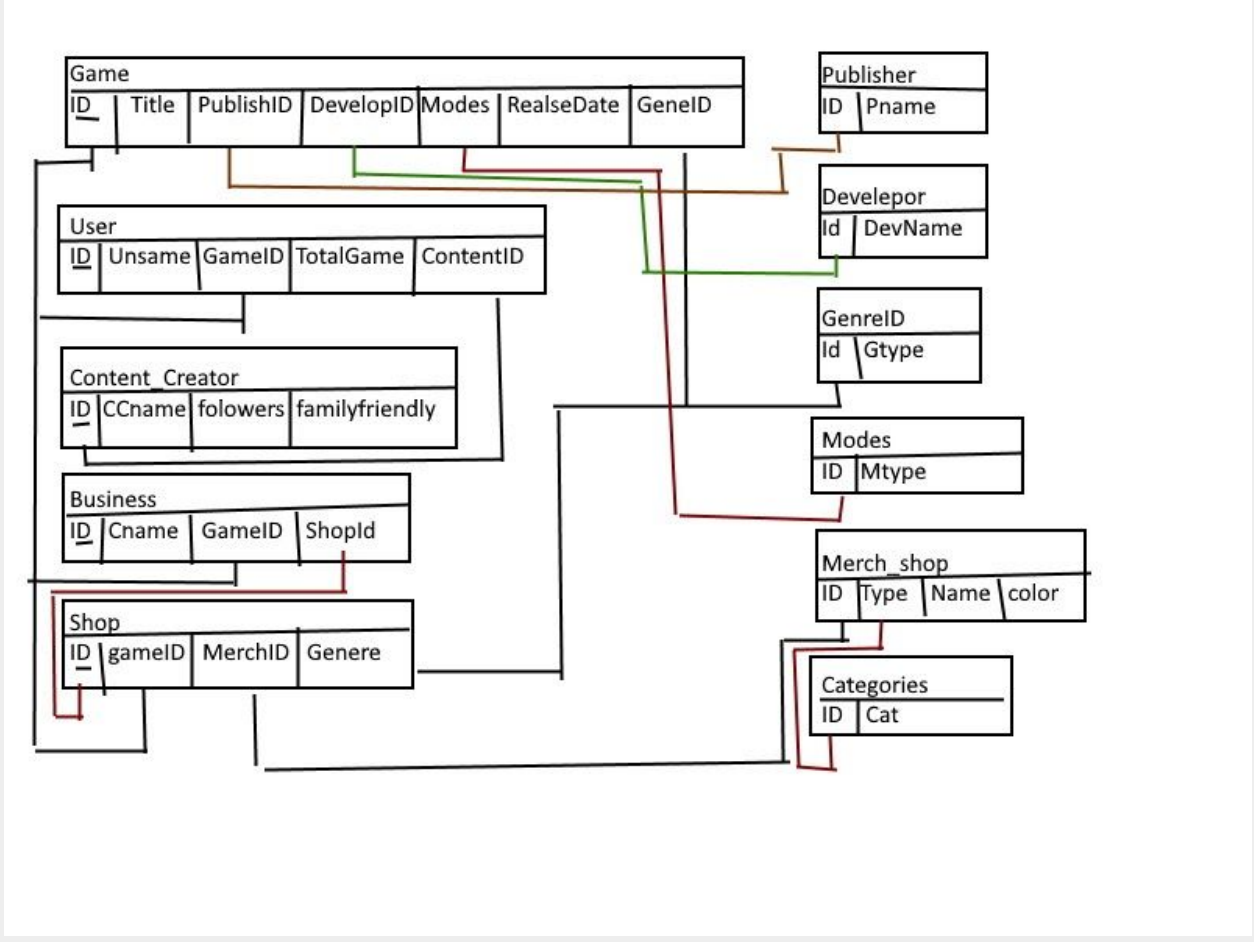
- Shop: Content carried by shop offered though businesses
    - ID - int auto inc unique, not NULL pk
    - gameID - fk
    - MerchID - fk
    - Genre - varchar, not NULL

Shop has M:M with games, due to there can be many shops relating to many to at least one game and games can be related to many to atleast one shop.

c) ER diagram



d) Schema



Feedback Section

Does the overview describe what problem is to be solved by a website with DB back end?

An ambitious proposal that does describe the problem to be solved. But I noticed that the database scale was not mentioned.

Does the overview list specific facts?

Partly, the overview does touch on what could be seen as specific facts, such as cross listing which robots have clear advantages over certain humans and vice-versa, but the specificatins are not complete without numerical data.

Are at least four entities described and does each one represent a single idea to be stored a s a list?

There are five entities, each representing a distinct idea. Each entity seems to be used as a object oriented class rather than a database table though.

For instance, Prisoners's "class" attribute could be a VARCHAR type.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Outline does directly mention entities.

Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

Division of work is clearly stated

Are 1:M relationships correctly formulated?

No, but I understand what they're getting at. For instance, the Armors entity references a 1:M relationship with Model, but the Model entity does not exist.

Is there at least one M:M relationship?

No, now I'm thinking OP accidently submitted the wrong assignment.

Is there consistency in

a) naming between overview and entity/attributes

Yes.

b) entities plural, attributes singular

Yes.

c) use of capitalization for naming?

Yes.

Project was changed to be less of a daunting task and make something more tangible. We changed the project due to the requirements of making a website when our idea was to create a game. Addressed the issues of not having M:M relationships directly stated.

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

It appears that the main page provides a UI that allows one to search a game by name or by time. I am not sure how time fits but by looking at the Games table, there is an attribute for “Date Rel” which I believe is the date the game was released so I believe this field is for this. Once a game has been selected, I believe there should be some page that displays the results of the query. As long as all of the

game stats are provided from all entities, or if the user is able to filter the game types as shown in the Type radio boxes, then I believe the criteria is met.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

The UI shows some input boxes that have a field labeled as Filter which indicates that a user will be allowed to search for a game using a filter. This meets the criteria.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table. If you click on the URL link provided, we can see the backend code on Node, I do not see a route that serves a page for updating the DB.

I am not sure if I see where a user can INSERT or how the website DB is updated to INSERT into the DB. The outline states that the DB will be updated every 7 days however the source of the update is not clear in the UI or description.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

See comment above. Currently, I do not see a way how the INSERT would be accomplished. I think this can be easily implemented if you add another page that allows a user to insert game data.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

A way to DELETE is not obvious at this time. I do not see a way for a user or for the website admin to delete game data. This can be implemented easily by adding a page or another form on the main page to allow a way to delete game data.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

I do not see a page or a UI to make an update. Again, this can be easily implemented on the same page or a separate page using a form with the appropriate fields.

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

It is hard to tell which relationship is is NULLable. The relationship's outline does not say if a relationship is NULLable but my guess would be that a player should be allowed to have no games played (new gamer?).

Do you have any other suggestions for the team to help with their HTML UI?

I think your group is off to a good start since you obviously have a back-end implemented with Node. I think all you really need to do is continue working on the basic functionality of your UI and only focus on making it pretty (if you have time).

Another suggestion would be to make it clear how your website will be updated with the game data. I did not see a page for this to be done manually or if this is done using some other way (via an API?).

Update your DB outline/DB/Schema to show a nullable relationship.

We have taken this feedback very seriously, and made changes accordingly, added a filter for the Date released of the game, utilized the INSERT more explicitly to add video games and their information, and updated DB outline and Schema.

Does the UI utilize a SELECT for every table in the schema?

I was unable to find the UI, or an index.html to refer to for how the UI and the DB interact.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

See previous answer.

Does the UI implement an INSERT for every table in the schema?

See previous answer.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

See previous answer.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?

See previous answer.

Is there at least one UPDATE for any one entity?

See previous answer.

Is at least one relationship NULLable?

I see that in the diagrams of the pdf, yes.

Do you have any other suggestions for the team to help with their HTML UI?

I actually couldn't find the HTML UI? I found one for index, but in general I had trouble finding other parts of this project?

This was a mistake on our part, a wrong link and did not include the correct link when we posted the post. We have updated the link and more features to the website. Also updated the pdf with more information regarding the ER diagram and schema.

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Assuming http://web.engr.oregonstate.edu/~setyawad/CS-340%20Frozium/views/index.handlebars is where I'm supposed to be looking, there is not an explciit SELECT for every table.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

There is at least one SELECT filter, but only for Name, Time, Any, For Fun, and Tournament -- none of the listed tables.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

No insert present at the above specified link

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

No delete present at the above specified link

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

The is no update for any entity or any entity attributes

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

Yes, present in the pdf.

Do you have any other suggestions for the team to help with their HTML UI?

I see that you're templating this with handlebars. On the server, I think you should be able to user `forever` as mentioned earlier in this class to host this node server on your ENGR file space. *Also,* it looks like your `href='/'` on Frozium will take you to the root of the engr file system, which is actually https://engineering.oregonstate.edu/ -- you'll want to get rid of the / or specify the current working directory with ./

We have updated the link, because we gave the wrong link in the pdf before. Updated the website to meet the criteria.