Project Step 3

Group 95: Mikasa Hackerman

Team: Eli Mills, Na Kim

Project Title: “Gametyme”

Website URL: https://cs340-gametyme.netlify.app/

Feedback by the peer reviewer

First reviewer:

- *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.

Yes there is a Select for every table except for the GamesPlatforms intersection table, maybe on either the games page or platforms page this could be displayed.

- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
  I myself am not sure about this requirement but I believe that this requirement is covered by the SELECT playthroughs query.
- *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.
  Yes there is an Add form for each entity from the schema, but the intersection table, GamesPlatforms, does not have this implemented in the UI or the DML. Maybe on the platforms page there could be a form to add games to platforms, insert new entries in the intersection 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).

Yes the FK's are accounted for by subqueries in the insert queries. I believe this will produce the desired effect for the requirement. There was not an insert for an M:M

relationship GamesPlatforms. Again this would be for adding entries into the intersection table.

- *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.

Yes DELETES are present for each entity. I could not locate one for a M:M relationship. Maybe the playthroughs table counts for this requirement.

- *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?

Yes UPDATES are present for each entity, and are correctly implemented.

- *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 the relationship between game and genre is set to null, for the genre fk is n the game table. IN the DDL file. This has been done correctly I believe.
- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*

All the tables, and the UI looks great! I especially like how the functionality for users is spread across the sign up, profile pages users. For the purposes of this assignment though I would clearly label these pages so its obvious, that signing up adds a user to the user page, for example. Maybe having these pages as links from the users page could also work, and make logical sense to a grader. The ID's could also be removed from the tables's UI.

Great job!

Second reviewer:

- ***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.**

Your website shows the tables that would be filled with the SELECTs and everything looks accurate!

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

Yes the Users table has a search function

- ***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.**

Yes every table has inserted data, that is correct

- ***Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?*** **In other words, if there is an 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).**

Yes, I can see the MM relationship between Users and Games

- ***Is there at least one DELETE and does at least one DELETE remove things from an 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.**

Yes the delete function is implemented in the playthroughs table

- ***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 majority of the tables have an edit function

- ***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.**

Finish time stamp is nullable in active playthroughs

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

Slapping some CSS in there at some point will help make something good great!

Third reviewer:

- *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.
    - I was not able to find a table/page for GamesPlatforms or related queries.
    - Aside from this, the UI utilizes a SELECT for every table in the Schema. I think including "Existing Sessions" on the same page as "Existing Playthroughs" page is a little confusing - even reading the pdf Draft I'm confused as to the differences between the two.
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
    - I believe the Users entity utilizes a search/filter with a dynamically populated list of properties on the HTML UI. However, I don't think this is implemented in the SQL.
- *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.
    - I was not able to find a table/page for GamesPlatforms or related queries.
    - Yes, every table has an implemented INSERT. There are no "ID" input fields, but my understanding is that's fine as primary keys auto increment.
- *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).
    - I was not able to find a table/page for GamesPlatforms or related queries.
    - Yes, each insert adds the corresponding FK attributes, EXCEPT for the M:M relationship. As far as I can tell there is no M:M intersection table implemented (GamesPlatforms)
- *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.
    - I was not able to find a table/page for GamesPlatforms or related queries.
    - All of the entities have DELETE queries.
- *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?
    - All of the included entities have an UPDATE function.

- *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, at least one relationship is NULLable, for example Companies - Locations
- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*
    - I believe you need to add an entity for GamesPlatforms - I was not able to find one. This also affects other requirements related to M:M relationships.
    - My understanding is that things like a SignUp or Profile are not strictly necessary as this is a website to manage the backend only by an administrator. It may save time/work to focus on other things, but I don't think it detracts from the project.

Fourth reviewer:

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.

The navbar is used for selecting certain tables, and it looks like those tables will be generated via SELECT statements. Users table isn't displayed like the others. I might have misunderstood the assignment, but I believe this is meant to be an admin-facing web app meaning they should be able to access user information.

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

There is a search for Users that looks like it will use SELECT. The search by Game is within the Users section. I don't know if that's a design choice, but personally I might put that in Games.

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.

Yes, there are inserts for every table. These all allow for proper attributes to be included.

Except for users where a table wasn't included.

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).

Yes, the INSERTS included referencing other FK's but uses FK's name attributes instead of ID's which I believe is totally okay to do. Also, auto-increments are not needed. Basically this all looks good to me here.

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.

There are DELETES where appropriate, it looks like it will be able to remove other related items. The many to many is accounted for in Platforms. So I am assuming that deleting an item from here would also delete it from corresponding rows.

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?

There are “edits” on all needed entities. Since the db is normalized to only include FK’s from other tables, any UPDATES will change the needed information for all other tables.

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.

I can only assume that leaving information as empty within add functionality would allow them to be set as NULL. So I’m going to go with yes on this one.

Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.

There are some forms that use “text” that might benefit from select menus instead. Like when adding a new game, maybe make Company and Genre a drop down.

Great work overall though!

Actions based on the feedback

- Based on the feedback that GamesPlatforms was not being implemented in the UI, we added SELECT, INSERT, and UPDATE queries using GamesPlatforms to associate Games with their Platforms, and the appropriate UI elements.
- Based on the suggestions to simplify the Users pages, we removed user-focused forms like sign up/login/user profile page, and replaced with standard add/update/delete controls for Users like all the other entities.
- Based on the feedback to remove ID's from the UI tables, we removed foreign key ID's from tables and replaced them with more human-readable titles using JOIN queries. We did decide to leave the primary key ID's displayed.
- Based on the suggestion to "slap some CSS in there", we added CSS styling to all pages and also added more functionality and responsiveness to the buttons.
- We decided not to separate Playthroughs from Sessions on the Playthroughs page, as was suggested by a peer reviewer. The two entities go hand-in-hand, and we would prefer to keep them in the same place. The intention is to add some dynamic displays/windows to allow the Sessions table to appear/disappear, but this feature will be added later as we further develop our project.
- Based on the feedback to implement filtering logic in our SQL queries and to move the Game search to the Games page, we added an improved filtering search on the Games page. This allows the user to search for a Game based on either Genre, Company, or Platform. The DML file is updated to account for this.
- Based on the suggestion to use dropdowns when adding a new Game, we implemented dropdown menus for selecting Company, Genre, and Platforms on the Create and Edit Game screens. These dropdowns are dynamically populated using SELECT queries.

Upgrades to the draft version

- Added icons and a logo
- Added responsiveness to buttons and toggle options
- Added id and classes to elements and buttons that were missing
- Added multiple attribute to select platform drop down menus
- Added CSS styling to tables, navbar, and pages for a more overall responsiveness look.
- The "DELETE" statements in the DML file now require all attributes to match to help prevent accidental deletion
- Modified DDL.sql file to remove references to specific database for universal import
- Removed a redundant button in adding playthroughs and linked it to redirect to the same page instead of linking to Sessions or a different window
- Hosted website on Netlify to properly load imported styling changes

A) Project and Database Outline

I) Overview:

Gametyme is a database-driven website which provides gamers a platform to track the video games they have played along with their gameplay hours and number of playthroughs. There are over 5 million games in the world, many with their own competitive circles and dedicated fan bases in which tracking the number and duration of playthroughs can be just as important as the game itself!

Gametyme makes it easier for gamers to keep track of the games they play as well as how long each playthrough takes. Gametyme has a large database of 1 million videogames and can support up to 500,000 users. To use our platform, users can simply search for a video game and start a new playthrough. As they make progress on their playthroughs, users can then report their hours by logging new sessions.

For pro members, Gametyme can be used to generate charts, trends, and analytical insights into their performance, which can be especially useful for those in the competitive speedrunning community. Gametyme also aims to foster a supportive gaming community by allowing users to view other users' lists and check out their stats!

II) Database Outline:

a) **Users**: Records the details of Users who have created an account and have logged in games & gameplay hours
   i) user_id: INT, auto_increment, unique, not NULL, PK
   ii) first_name: VARCHAR(45), not NULL
   iii) last_name: VARCHAR(45), not NULL
   iv) username: VARCHAR(45), not NULL
   v) email: VARCHAR(45), not NULL
   vi) **Relationship**:
      1) A 1:M relationship between Users and Playthroughs is implemented with user_id inside of Playthroughs. Users can log multiplePlaythroughs, and each Playthrough can only have one User.

b) **Games**: Records the details of games that users can track hours for or have played
   i) game_id: INT, auto_increment, unique, not NULL, PK
   ii) game_title: VARCHAR(100), not NULL
   iii) game_summary: VARCHAR(1000)
   iv) release_date: DATE
   v) company_id : INT, FK
   vi) genre_id: INT, FK
   vii) **Relationship**:
      1) A M:1 relationship between Games and Genres where genre_id from Genres is implemented as a FK in Games. Genres can have multiple Games, but Games will be limited to one Genre.
      2) A 1:M relationship between Games and Playthroughs is implemented with game_id inside of Playthroughs as a FK.
      3) A M:1 relationship between Games and Companies where company_id from Companies is implemented as a FK in Games. A Company can have produced multiple games, but a Game can only be produced by one Company
      4) A M:M relationship between Games and Platforms which is established through an intersection table named GamesPlatforms. GamesPlatforms will contain both game_id and platform_id as foreign keys. Games are available on multiple Platforms, and many different games can be available on the same Platform.

c) Playthroughs: Created when a User starts a new Playthrough of a Game. Used to track which Games a User has played, when they've started and finished a Game, how many times they've played it, and how many hours they've played it (by summing Session lengths).
   i) playthrough_id: INT, auto_increment, unique, not NULL, PK
   ii) start_timestamp: TIMESTAMP, not NULL
   iii) finish_timestamp: TIMESTAMP
   iv) user_id: INT, FK, not NULL
   v) game_id: INT, FK
   vi) **Relationships:**
      1) A M:1 relationship between Playthroughs and Users where user_id from Users is implemented as a FK in Playthroughs. Users can start multiple Playthroughs, but each Playthrough is only associated with one User.
      2) A M:1 relationship between Playthroughs and Games where game_id from Games is implemented as a FK in Playthroughs.

There can be multiple Playthroughs for the same Game, but only one Game per Playthrough.

3) A 1:M relationship between Playthroughs and Sessions where playthrough_id from Playthroughs is implemented as a FK in Sessions. A Playthrough can be divided across multiple Sessions, but each Session can only be for one Playthrough.

d) **Sessions**: Created whenever a User logs a new session for a particular Playthrough. Records the corresponding Playthrough, the start date and time, and eventually the finish date and time.
   i) session_id: INT, auto_increment, unique, not NULL, PK
   ii) session_start: TIMESTAMP, not NULL
   iii) session_end: TIMESTAMP
   iv) playthrough_id: INT, FK, not NULL
   v) **Relationship**:
      1) A M:1 relationship between Sessions and Playthroughs where playthrough_id from Playthroughs is a FK in Sessions. The M:1 relationship between Sessions and Playthroughs is implemented and established through an intersection table. Playthroughs can be divided into multiple Sessions, but each Session can only belong to one Playthrough.

e) **Genres**: Category entity listing possible genres for Games.
   i) genre_id: INT, auto_increment, unique, not NULL, PK
   ii) genre_name: VARCHAR(45), not NULL
   iii) **Relationship**:
      1) A 1:M relationship between Genres and Games. Games can only be one Genre, but the same Genre can be used for multiple Games.

f) **Companies**: Records the information about each game development studio.
   i) company_id: INT, auto_increment, unique, not NULL, PK
   ii) company_name: VARCHAR(100), not NULL
   iii) location_id: VARCHAR(45), FK
   iv) **Relationship**:
      1) A 1:M relationship between Companies and Games. A Company can produce multiple Games.
      2) A M:1 relationship between Companies and Locations where location_id from Locations is used as a FK in Companies. There

are multiple game companies, but each company has one location for their headquarters.

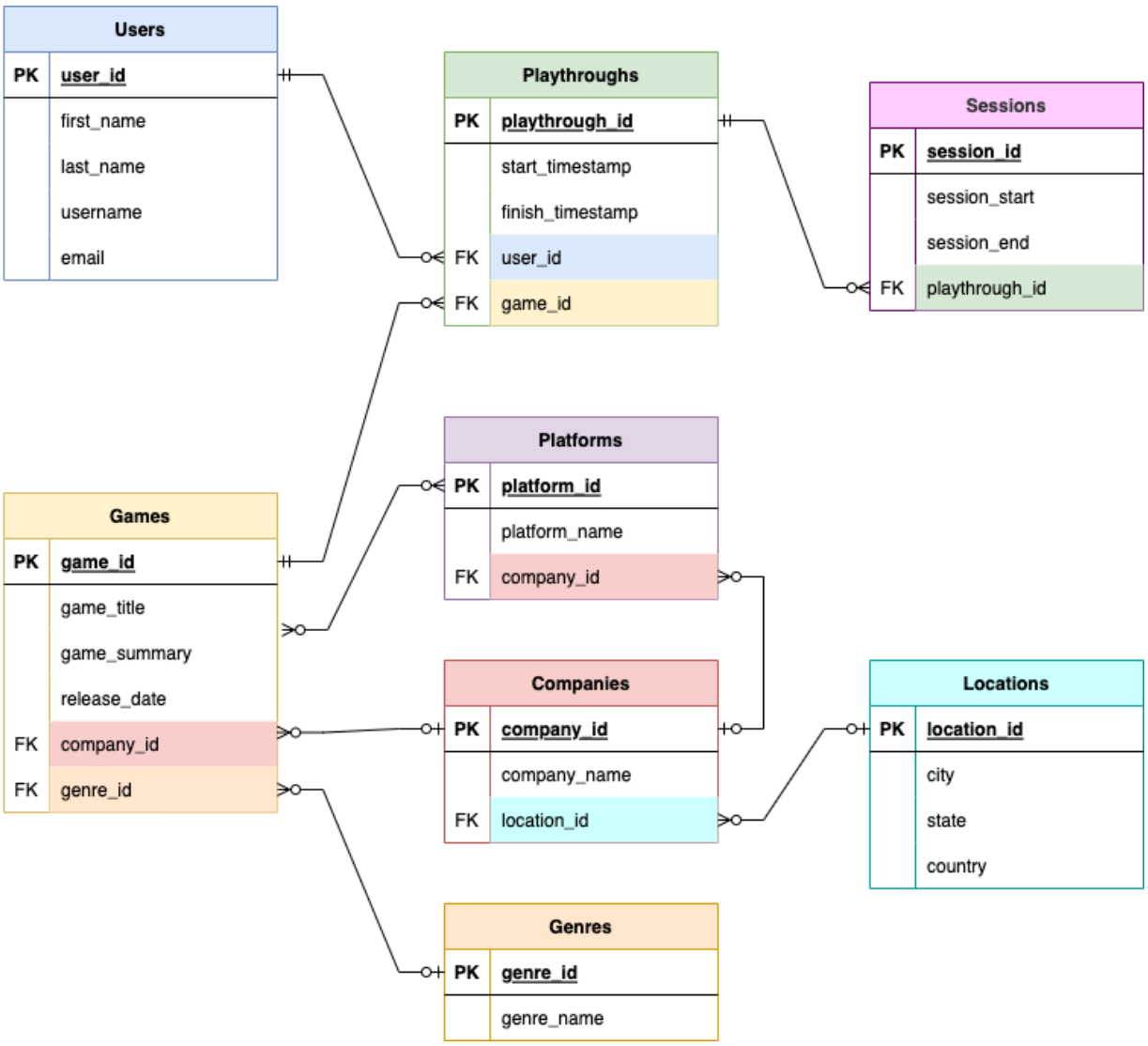
g) **Platforms**: Category entity listing possible gaming platforms.
   i) platform_id: INT, auto_increment, unique, not NULL, PK
   ii) platform_name: VARCHAR(45), not NULL
   iii) company_id: VARCHAR(45), FK
   iv) **Relationship**:
      1) A M:M relationship between Platforms and Games which is established through an intersection table named GamesPlatforms. GamesPlatforms will contain both game_id and platform_id as foreign keys. Games can be on multiple platforms, and the same Platform can be associated with many different Games.
      2) A M:1 relationship between Platforms and Companies, where company_id from Companies is used as a FK in Platforms. A single Company can create many different Platforms, but a given Platform must only belong to a single Company.

h) **Locations**: Category entity listing the locations of game companies.
   i) location_id :INT, not NULL, AUTO INCREMENT, unique, PK
   ii) city: VARCHAR(45), not NULL
   iii) state: VARCHAR(45)
   iv) country : VARCHAR(45), not NULL
   v) **Relationship**:
      1) A 1:M relationship between Locations and Companies where location_id from Locations is used as a FK in Companies. There are multiple game companies, but each company has one location for their headquarters.

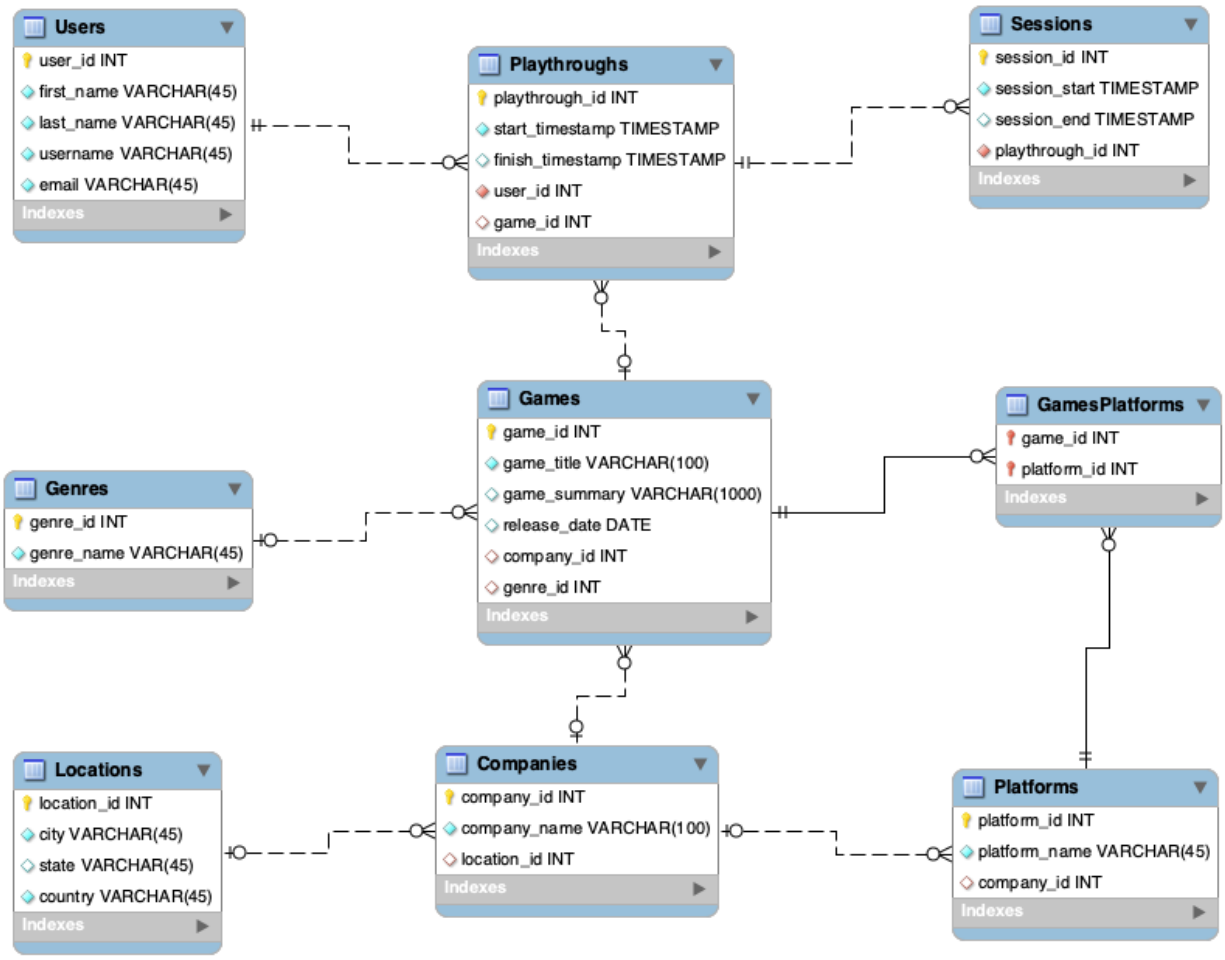
Entity Table Assignments:

- Eli Mills - Genres, Companies, Platforms, Locations
- Na Kim - Users, Games, Playthroughs, Sessions

III) Entity-Relationship Diagram



IV) Schema


Users
user_id INT
first_name VARCHAR(45)
last_name VARCHAR(45)
username VARCHAR(45)
email VARCHAR(45)
Indexes
Playthroughs
playthrough_id INT
start_timestamp TIMESTAMP
finish_timestamp TIMESTAMP
user_id INT
game_id INT
Indexes
Sessions
session_id INT
session_start TIMESTAMP
session_end TIMESTAMP
playthrough_id INT
Indexes
Games
game_id INT
game_title VARCHAR(100)
game_summary VARCHAR(1000)
release_date DATE
company_id INT
genre_id INT
Indexes
GamesPlatforms
game_id INT
platform_id INT
Indexes
Genres
genre_id INT
genre_name VARCHAR(45)
Indexes
Locations
location_id INT
city VARCHAR(45)
state VARCHAR(45)
country VARCHAR(45)
Indexes
Companies
company_id INT
company_name VARCHAR(100)
location_id INT
Indexes
Platforms
platform_id INT
platform_name VARCHAR(45)
company_id INT
Indexes

V) Example Data

Users

user_id	first_name	last_name	username	email
1	Eren	Yeager	Titan	fightme@gmail.com
2	Mikasa	Ackerman	Erennn	ackerman@gmail.com
3	Link	Zelda	Korok	zelda@gmail.com

Games

game_id	game_title	game_summary	release_date	company_id	genre_id
1	Elden Ring	This is an open world dark souls like game where you also die a lot	2022-02-25	1	1
2	The Legend of Zelda: Breath of the Wild	This is an open world Zelda game where you hunt koroks	2017-03-03	2	2
3	Overwatch	This is a multiplayer shooting game where everyone needs healing too much	2016-05-24	3	3

Playthroughs

playthrough_id	start_timestamp	finish_timestamp	user_id	game_id
1	2022-01-04 02:15:01	NULL	1	1
2	2021-11-01 01:30:40	2022-01-12 10:24:10	2	3
3	2020-05-20 04:20:32	2020-12-01 10:02:01	3	2

Sessions

session_id	session_start	session_end	playthrough_id
1	2020-12-11 11:01:03	2020-12-11 21:01:03	3
2	2022-02-01 02:02:13	2022-02-01 03:02:13	2
3	2022-04-11 05:09:23	2022-04-11 10:09:23	1

Genres

genre_id	genre_name
1	action
2	adventure
3	shooter

Companies

company_id	company_name	location_id
3	Blizzard	3
1	FromSoftware	1
5	Microsoft	4
2	Nintendo	2
4	Sony	1

Platforms

platform_id	platform_name	company_id
1	Switch	2
2	PC	5
3	Playstation 4	4
4	Playstation 5	4
5	Xbox One	5
6	Xbox Series X|S	5

Locations

location_id	city	state	country
1	Tokyo	NULL	Japan
2	Kyoto	NULL	Japan
3	Irvine	California	United States
4	Redmond	Washington	United States

GamesPlatforms

game_id	platform_id
1	2
1	3
1	4
1	5
1	6
2	1
3	1
3	2
3	3
3	5

B) Fixes based on Feedback from Previous Steps

Step 1 Final: Feedback from Peer Reviewers

First reviewer

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

Yes, Gametyme is a database-driven website that is used to track the type of video games, hours played, and number of playthroughs. It is a very useful tool for competitive speedrunners who are trying to set fastest records for certain achievements in the game.

Does the overview list specific facts?

Yes it lists that the company, Gametyme has a large database of 1 million games and can support up to 500,000 users.

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

Yes at least four entities exist in the schema including Users Games, Playthrough, Session, Genre, Company, and Platform. Each one represents a single idea to be stored as a list.

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

The outline of the entity does describe the purpose of each list, attribute, datatype and constraints and does describe the relationship between entities. Each entity provides the information that would be of interest. The M:M relationship between Games and Platforms will eventually need to be explained with an intersection table in a 1:M relationship**.** The diagram is well designed and color coded well.

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

The outline clearly indicates which entities will be implemented. It is not shown which team member is primarily assigned to the associated pages.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

The 1:M relationships are correctly formulated. There is at least one M:M relationship. They are between 1: Users and Games and 2: Games and Platform. The intersection table between the User and Games M:M relationship in the diagram is the playthrough entity. However there is no intersection table for the M:M relationship between Games and Platform. It seems that the platform_id in Platforms should have been a FK in Games.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

There is consistency in naming between overview and entity/attributes in the diagram. Entities are plural while attributes mostly in the singular. I suggest changing hours_played to time_played to maintain the singular. The use of capitalization is used for the entities but not for the attributes. However the outline has "Playthrough, Session, Genre, Company and Platform" written as the singular. This should be changed to plural to be consistent with the diagram.

Second reviewer

- Does the overview describe what problem is to be solved by a website with DB back end?
    - Yes, the overview does a really good job of describing what the problem is and how their solution solves it.
- Does the overview list specific facts?
    - Yes, they listed an approximate number of games in the world to help show the potential scope the db could grow to. They also mention what their db is able to handle.
- Are at least four entities described and does each one represent a single idea to be stored as a list?
    - Yes, there are at least 4 entities that all make sense as a single idea.
- Does the outline of entity details describe the purpose of each, list attribute data types and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?
    - Yes, the outline does a good job of describing each entity, its attributes and their data types, as well as the relationships between entities. The relationships are well described and easy to understand why the relationship is necessary.
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
    - The 1:M relationships appear to be correctly formulated, however I did notice that in the description, Games describes a M:M relationship with Users and vice versa, but I do not see the relationship being described in the ERD.
- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - The only issue I see is that the entities described in the overview are singular, and the entities in the ERD are plural, I would just make the overview plural to match your ERD.

By the way this is a super creative idea, and I think people would actually use this!

Great work.

Third reviewer

- Does the overview describe what problem is to be solved by a website with DB back end?
    - Yes, the overview does a good job describing a need for a DB back end solution to track video games that gamers have played and their progress for each game.
- Does the overview list specific facts?
    - Yes, the overview goes into depth in the scale of the fictitious company, with specific numbers. It does a good job of representing and presenting the DB methods a company that can use to provide analytic services for gamers. It clearly explains what the DB is meant to accomplish and what solution it will provide for Gametyme and its community of gamers.
- Are at least four entities described and does each one represent a single idea to be stored as a list?
    - Yes, more than four entities are outlined and each entity has an arguable reason to be outlined as an entity. Each entity relates to other entities and the attributes outlined for each seem relevant.
- Does the outline of entity details describe the purpose of each, list attribute data types and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?
    - Yes, the outline succinctly describes the purpose of each entity in a tidy description. Each attribute is shown and the outline clearly indicates which entities will be implemented. There are entities without many attributes that could potentially include more information. The relationships an entity has are also detailed explicitly for each entity. The outline does not specify which team member will be assigned to each page.
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
    - Yes, the many 1:M relationships are simple to follow and seem to make sense in the outline and diagram. There is one M:M relationship between platform and games between Platform and Games, which makes sense (however, no intersection table in the diagram). There is one M:M relationship between games and users that is not present in the diagram.
- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - Yes, there is consistency in the naming in the diagram, with plural tables/singular ids and a use of capitalization for names of Entities and

lowercase for attributes. The outline has the entity names in the singular, which is not consistent with the diagram.

Step 1 Final: Summary of Feedback from Peer Reviewers

Overall our team project received great feedback.The peer reviewers gave feedback that most of our draft consisted of detailed descriptions, correct attribute data types, and that our project overview described what the problem is to be solved with a DB back end clearly. While we got great feedback, there were a couple of helpful tips and constructive feedback that we received to better improve our project draft.

We received a total of 3 reviewers in which each reviewer brought their own perspectives and feedback. The first feedback we received was that in our database outline, we implemented a M:M relationship between Games and Platforms but did not include an intersection table. The peer reviewer recommended that we would eventually need to explain this M:M relationship with an intersection table in a 1:M relationship. In the draft outline, the reviewer also mentioned that we lacked a section including which team member would be assigned to an entity/page. In addition, the reviewer also suggested that we include platform_id from the entity, Platforms, as a FK in the Games entity. One of the final feedbacks that we got from the first reviewer was that there was an inconsistency in Entities being plural as some entities were singular along with one inconsistency with one attribute being plural instead of singular. Finally, the reviewer suggested we change the Sessions attribute “hours_played” to “time_played” to maintain consistency of attributes remaining singular.

The second reviewer gave similar feedback to the first peer reviewer. The reviewer noticed that there were inconsistencies in entities being plural and that a M:M relationship between Games and Users was not described in the ERD. The third reviewer gave feedback that was also similar to the two reviewers in that there were some inconsistencies with Entity names not being plural, a M:M relationship between platform and games not being depicted in the ERD, no intersection table in ERD for Games and Platforms, and our draft outline not specifying which team member would be assigned to which page. A new addition to the feedback that the third reviewer mentioned was that there were many entities without many attributes that could be improved on with more information.

Step 1 Final: Actions Based on Feedback from Peer Reviewers

- We chose not to act on the feedback from the first reviewer about Games and Platforms not having an intersection table displayed in the ERD. This is due to the assignment description for Project Step 1 Draft stating that for this ERD, it is preferred that we leave intersection tables out.
- Based on the feedback from the first reviewer that we did not include information about which teammate would be assigned to which entity, we added a section to our writeup detailing which entities each team member would be responsible for.
- We chose not to act on the suggestion from the first reviewer to include a FK for Platforms within the Games entity. This is because Platforms and Games have a M:M relationship, so rather than storing one foreign key inside the other, their relationship will be facilitated through an intersection table containing FK's for both Games and Platforms.
- Based on the feedback from the first reviewer that our naming was inconsistent, we corrected the **Database Outline** to list entities as plural nouns to match the ERD.
- Based on the feedback from the first reviewer about changing the attribute "hours_played" to "time_played" from the Sessions Entity to maintain consistency in attributes being singular, we followed the reviewer's advice and switched the attribute to time_played.
- Based on the feedback from the second reviewer about Games and Users not showing a M:M relationship in the ERD, it was realized that it was misleading to refer to this relationship as M:M within the **Database Outline**. This is because, while there is technically a M:M relationship between these two entities, it is probably more accurate to talk about this in the context of the Playthrough entity, which is its own entity beyond a simple intersection table. Because of this, mention of a M:M relationship between Games and Users was omitted from the **Database Outline**.
- Based on the feedback from the third reviewer that some of our entities were lacking in attributes, we decided to add "company_location" to the Companies entity and "platform_make" to the Platforms entity.

Step 1 Final: Other Upgrades to Draft

- Changed type of "time_played" in "Sessions" to DECIMAL(19,2) instead of FLOAT after learning about FLOAT's tendency to lose precision

Step 2 Draft: Summary of Changes based on TA Feedback for Step 1 Final

From the feedback that we received from our TA, our team was recommended to mention the use of foreign keys in our relationship descriptions of Entities as well as giving more details about implementation. We took our TA's feedback and implemented these changes by adding more details to each of our Entity relationship descriptions. We added more details on the use of foreign keys as well as gave more details such as the use of an intersection table to better describe how each Entity is being implemented.

While addressing the points from the feedback received, we also made a few additional changes. We noticed that there was a naming inconsistency between our schema and our draft outline. Thus we changed the attribute company_location to location_id to maintain consistency. In addition, to normalize the tables to 3NF, two additional tables were added. The tables Locations and GamesPlatforms were added to make it easier to reference to the Platforms and Companies tables. Along with these changes, Companies and Platforms Entity relationship descriptions were updated to detail the use of foreign keys and implementation. The attribute location_id was updated to add foreign keys in both Companies and Platforms as well. While working on the schema, we also decided to update the ON DELETE requirements for all foreign keys in the following tables: Playthroughs, Sessions, Games, Companies, and Platforms.

Step 2 Draft: Other Upgrades to Draft

- Changed attribute name in "Companies" from company_location to location_id to maintain naming consistency between the schema and draft outline.
- Normalized tables to 3NF by creating two separate tables, GamesPlatforms and Locations, to easily reference Platforms and Companies entity tables.
- Added more details to Entity relationship descriptions to mention foreign key use and implementation details of the use of an intersection table.
- Through normalization to 3NF, GamesPlatforms and Locations tables were created.Location table was added to the draft outline along with data attributes and relationships. Through this addition, the Companies and Platforms relationships were updated to mention the implementation and foreign key usage details. The attribute location_id from Locations was updated to be a FK in both Companies and Platforms as well.
- Updated ON DELETE requirements for foreign keys in the following tables: Playthroughs, Sessions, Games, Companies, and Platforms.

Step 3 Draft: Feedback from Peer Reviewers for Step 2 Draft

First Reviewer

Hello, Na and Eli! Great project you have. Here is my feedback:

- **Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?**

No, not exactly. While the schema does a great job demonstrating the scope of the project and its relationships between entities, I suggest adding the GamesPlatforms intersection table found there to the outline, now that it is an entity with a unique primary key.

- **Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

No. In the schema and DDL, for the Platform table, company_id is used but in the outline and ERD it is called location_id. I suggest editing or omitting this attribute, as some companies mentioned in the example data do not necessarily have a platform.

- **Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?**

Yes, the schema is clear and readable. Relationship lines are not crossed and the diagram is easy to follow.

- **Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?**

Yes, there is now an intersection table between Games and Platforms with game_id and platform_id as two FKs inside. Again, I suggest adding this table to the outline since it also contains a unique PK called games_platforms_id.

- **Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?**

No, the sample data doesn't suggest any non-normalized issues. All attributes seem to hold on their own and do not depend on each other. I do, however, suggest putting the attributes in Playthroughs and Sessions in the same order for both outline and schema.

- **Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)**

Yes, the SQL file is syntactically correct. The tables and data are successfully added when attempting to verify it through flip.

- **In the SQL, are the data types appropriate considering the description of the attribute in the database outline?**

Yes, the data types are appropriate, logical, and consistent with the description found from the outline, when and where such descriptions are present.

- **In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?**

Yes, the primary and foreign keys are correctly defined and are a 1:1 match with the Schema. Great job including CASCADE operations in the SQL for deleting and updating to reduce data redundancies and anomalies.

- **In the SQL, are relationship tables present when compared to the ERD/Schema?**

Yes, the tables are present, and as mentioned above when discussing the Schema, the SQL file also contains the company_id attribute for the Platforms entity table instead of location_id.

- **In the SQL, is all example data shown in the PDF INSERTED?**

Yes, all example data shown in the PDF is INSERTED in the SQL file. I recommend when inserting to specify the columns you wish to insert to and to use subqueries, so you don't have to hardcode the IDs.

Second Reviewer

- **Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?**
    - The schema follows the physical model laid out by the ERD and database outline except for one thing. location_id is an attribute of Platforms in the database outline but in the schema and ERD company_id takes its place. This should be swapped out.
- **Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**
    - There is consistency aside from the name of the GamesPlatforms intersection table. All attributes use an underscore to deal with spaces but this entity technically uses camel case. Switching the name of the intersection table to follow the naming scheme of the attributes will fix it easily.
- **Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?**
    - Yes, the schema is easy to read and it's straightforward to ascertain all relationships.
- **Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?**
    - Yes, all intersection tables are properly formed.
- **Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?**
    - There are no non-normalization issues. Professor Safonte specified we didn't have to normalize up to 3NF by making a separate entity for Location so it might be an option to leave that out and just go up to 2NF.
- **Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)**
    - Yes, the SQL file is syntactically correct. I was able to source it and build the DB using the provided file without any errors.
- **In the SQL, are the data types appropriate considering the description of the attribute in the database outline?**
    - Yes, all data types seem appropriate given their description in the database outline.
- **In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?**
    - All primary and foreign keys are correct except the GamesPlatforms intersection table. This table has its own primary key, which is unneeded.

You only need the 2 foreign keys here, game_id and platform_id. You can get rid of games_platforms_id. The appropriate CASCADE operations are declared.

- **In the SQL, are relationship tables present when compared to the ERD/Schema?**
    - Yes, the relationship tables are present when compared to the schema.
- **In the SQL, is all example data shown in the PDF INSERTED?**
    - Yes, all example data from the PDF is inserted in the SQL file.

Great work, look forward to seeing the final result!

Third Reviewer

Hi Team,

Cool project idea, please see my review below:

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
    - The schema follows the ER diagram except the GamesPlatforms table that exists in the Schema does not show the same in the ER diagram. I think this may be okay if you were to more clearly show in the ER diagram that it was a M:M relationship between the tables. Otherwise, including the intersection would be needed.
- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - Naming is consistent, ER/Schema having the slight discrepancy mentioned in the point above, all entities are plural, capitalization is used to separate words in the table.
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
    - Schema is easy to read, clearly laid out in the MySQL Workbench tool and separated so that each table is appropriately spaced.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
    - The Intersection table GamesPlatforms contain two FKs and one PK, as required.
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
    - There are no normalization issues that I can see, each table is logically separated and containing only information relevant to its title. No partial/transitive dependencies.
- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)
    - The sql file was loaded into PhPMyAdmin with no issues on my end.
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
    - Data types are appropriately used to match the description and purpose of the entity.

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
    - Cascades are setup with detail, for example, the games table is appropriately setting foreign keys to null so that dependent tables are left with their information. Also updates are set to no action.
- In the SQL, are relationship tables present when compared to the ERD/Schema?
    - Relationships between tables are matching what is shown in the schema, ERD diagram has the same discrepancy mentioned early so not quite in that case.
- In the SQL, is all example data shown in the PDF INSERTED?
    - Example data was populated appropriately on insert statements.

Fourth Reviewer

Hello Na and Eli, this is a really interesting project idea! Looking forward to see how it turns out!

- **Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?**
    - Almost! As others have mentioned the intersection table 'GamesPlatforms' is missing in the ER logical diagram. In addition, based on the outline, the 'Platforms' entity is missing the location_id FK in both the ER logical diagram and the schema.
- **Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**
    - There is some consistency in the naming between all sections of the project, except for the location_id as mentioned earlier. The 'Platforms' entity uses the company_id as a FK in both the ERD and Schema but in the outline, I only see location_id used as an FK. All entities are plural with attributes in the singular.
- **Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?**
    - Yes the schema is clear and readable with no crossed relationship lines.
- **Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?**
    - Yes I understand that the intersection table is clearly 'GamesPlatforms' with two properly declared FKs.
- **Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?**
    - Looking through the sample data, I do not see any non-normalized issues or dependencies.
- **Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)**
    - Yes when I import the SQL file, the file shows up as syntactically correct.
- **In the SQL, are the data types appropriate considering the description of the attribute in the database outline?**
    - Yes the data types are appropriate compared to the description of the attributes in the database outline, except the aforementioned 'Platforms' entity, where 'company_id' is used as an FK instead of 'location_id' which is written in the database outline.

- **In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?**
    - All primary and foreign keys are correctly defined when comparing to the schema. The CASCADE operations are properly declared to help with any anomalies.
- **In the SQL, are relationship tables present when compared to the ERD/Schema?**
    - Yes the relationship tables are present when compared to only the ERD/Schema. I see the M:N entity in the form of the 'GamesPlatforms' entity.
- **In the SQL, is all example data shown in the PDF INSERTED?**
    - Yes, all example data in the SQL file is also shown in the PDF inserted in a very clear manner.

Step 3 Draft: Summary of Feedback from Peer Reviewers for Step 2 Draft

The main issue most reviewers pointed out was to correct the error in the Database Outline, where “location_id” was mistakenly placed in “Platforms” as a foreign key instead of “company_id”. The M:1 relationship listed in “Platforms” is also incorrect. This was likely a copy-paste error and is corrected in this draft.

Another issue several reviewers commented on revolved around the “GamesPlatforms” intersection table. There did not seem to be a consensus on whether to treat this table as an entity. The first reviewer recommended adding it to the database outline section, as they pointed out it should be an entity due to it having a unique primary key in addition to the two foreign keys. However, the second reviewer says that this unique primary key is unnecessary, as each row can be uniquely identified by the two foreign keys, “game_id” and “platform_id”. Conversely, the fourth reviewer took no issue with this, mentioning that this intersection table contains two foreign keys and one primary key as required.

Additionally, the third and fourth reviewer focused on the ER vs Schema diagrams, pointing out that the GamesPlatforms table does not appear in the ER diagram, but does appear in the Schema diagram. The third reviewer says this would be ok if we were to clearly show that there was a M:M relationship between the tables in the ER diagram.

Other suggestions were more clerical. The first reviewer suggested we put the attributes in Playthroughs and Sessions in the same order for both the outline and the Schema. They also recommended using subqueries for the insertion statements in the DDL.sql file, rather than hardcoding foreign key IDs. The second reviewer wrote that the “GamesPlatforms” naming was inconsistent with the rest of the database, since attributes use an underscore between words but “GamesPlatforms” is in camel case. This reviewer also pointed out that the Location table may not be necessary since we did not have to normalize all the way to 3NF.

Step 3 Draft: Actions Based on Feedback from Peer Reviewers for Step 2 Draft

- The database outline section for the “Platforms” entity has been corrected to use “company_id” and describe a M:1 relationship with “Companies”, rather than “Locations”
- The primary key has been removed from the “GamesPlatforms” table, as in this particular case the combination of “game_id” and “platform_id” will uniquely identify each entry
- The ERD will continue not to show the intersection table. This diagram already shows the M:M crow’s feet notation instead. The Schema will continue to show the intersection table.
- “GamesPlatforms” will not be added to the database outline. With the above changes, based on the first reviewer’s feedback “GamesPlatforms” should no longer count as an entity, existing solely as an artifact of implementation in the database.
- The order of attributes listed in the database outline for all entities has been corrected to match the order of the ERD and Schema diagram.
- The DDL.sql file has been updated to use subqueries rather than hard coded IDs. The file was written this way because it was generated as a backup file.
- The naming of “GamesPlatforms” was determined not to be inconsistent based on the reviewer’s reasoning, since “GamesPlatforms” is an entity, not an attribute. All entities are capitalized, this is just the only entity with two words. All attributes use underscores to separate words.
- For now, we will keep the Location table but may consider removing it in the future if we get confirmation that it will be acceptable.

Step 3 Draft: Other Upgrades to Draft

In addition the feedback from peer reviewers mentioned above, we elected to make our own changes to the project as follows:

- Removed the “finish_timestamp” data from “Playthrough” ID 1. This represents an “active Playthrough” as displayed on the website and SELECTed with a query in the DML.sql file.
- Changed “Sessions” attributes from “session_timestamp” and “time_played” to “session_start” and “session_end”. It was decided that these will make it easier to enter data as each will be a simple timestamp, while the total time played can be calculated from subtracting the two timestamps.