Group 106 - Ellie Anna Evans, Jianbo Ning

Project step 5

Website URL: http://web.engr.oregonstate.edu/~evansdr/cs340project/public/index.html

Project Name：Game sales website

## Fixes based on Feedback from Step 4

We modified our ddl.sql and dmq.sql code.
1. We fixed most of the errors syntactically.
2. We fixed the description of the data type.
3. There were also many errors in foreign keys, which we have also fixed.
4. We fixed most of the query syntax errors.
5. Improved some queries.

## Feedback by the peer reviewer

**Nicholas's suggestions:**

Once you add everything above, you should also add a link to the other pages, not just the home page.

Our action: We add a link to the other pages.

**Navi's suggestions:**

U I think you may want to consider adding a navigation bar to make it easier to navigate at the pages on the website instead of having a back to main page link. It can be done via HTML. Look forward to seeing the site.

Our action: We added a navigation bar to make it easier to navigate.

**Thanh Thao's suggestions:**

Implement a navigation bar so that any page can be reached from any other page. As it stands, there is only a link back to the main page from any other page.

Our action: We added a navigation bar to make it easier to navigate.

**Ting Ting's suggestions:**

On Games.html, the "Search for a game" is a little ambiguous about search by what attribute of Games. An instruction like "Search by game_id" or "game_id" before the input field would be more clear and specific to guide the user's input. If a user is new to the database system, it will be easy for him/her to follow what to put in that input box.

Add a "Delete" on Games.html for deleting a game or add a "Delete" on Genres.html for deleting a genre. Since the relationship of Games to Genres is M:M, deleting a game could meet the criteria "at least one DELETE removes things from a M:M relationship". But I think when a game is deleted, the relevant data with the same game_id in the intersection table "Genre-Games" should also be deleted. Otherwise there will be some Genre-Games with non-exist game_id after this game is deleted. The cases are the same with deleting Genres. Show your NULLable relationships Customers to Reviews or Games to Reviews on HTML UI.

Our action: We added delete to the game page.

## Updates to Draft Version

We made several updates to the draft, including adding most of the CSS, adding a navbar, and more structure to the HTML. We are not yet complete with the website, some CSS formatting still needs to be implemented for the table and the inside of the form boxes.

## Fixes based on Feedback from Step 2:

**Feedback by the peer reviewer** :

**Steven's feedback:**

1. I am wondering if Sales and Games should be M:M. You of course can define it however you choose, but with your definition of Games to Sales being 1:M doesn't that mean that a given game title can be part of many Sales (which makes sense), but that a given sale can only contain a single game? Again, that may have been your intention, but more likely it seems like you would be a customer to be able to select multiple games in a

given sale transaction. If that is the case, you would have a M:M relationship and need a composite table, like you built between Genres and Games.

**Actions based on the feedback:** We decided not to take action, because this can be defined according to our own wishes. We insist on using our will.

**Andrew's feedback:**

1. My suggestion for one way you could improve this section would be adding more numerical facts, like how many customers you expect to be able to handle, and the amount of new games released per year that you add to your products.
2. I did notice that your price attribute and sales_price attribute for sales were the int data type. In my opinion, these attributes should probably be a decimal data type to allow for prices such as $69.99, as the int data type only allows for whole numbers.
3. You should change the Games-Genres table to Game_Genres to get rid of the special character "-".

1.**Actions based on the feedback:** We decided not to take action because we didn't think we needed to add more numerical facts.

2.**Actions based on the feedback:** We accept this suggestion and make a change, change int to float

3.**Actions based on the feedback:** We accept this suggestion and make a change, we change the Games-Genres table to Game_Genres.

## Project Outline and Database Outline - Final Version:

## a) Overview

We are the top video game website with annual sales of $2 billion, selling a wide variety of games, from both AAA developers and indie developers alike. With so many games, and so many users, it is incredibly important that we keep all that data secure and organized. Between sensitive user data like payment and login info, as well as our massive library of games, a database solves both organizational and security issues. It allows us to keep track of all our user's information, easily pull it up, and safely store it away from prying eyes. It also allows us to organize and keep track of our growing library of games, as well as allowing us to swiftly deliver those games to the customer. Efficiency, organization and security are our top priorities, and a database lets us accomplish all of those goals.

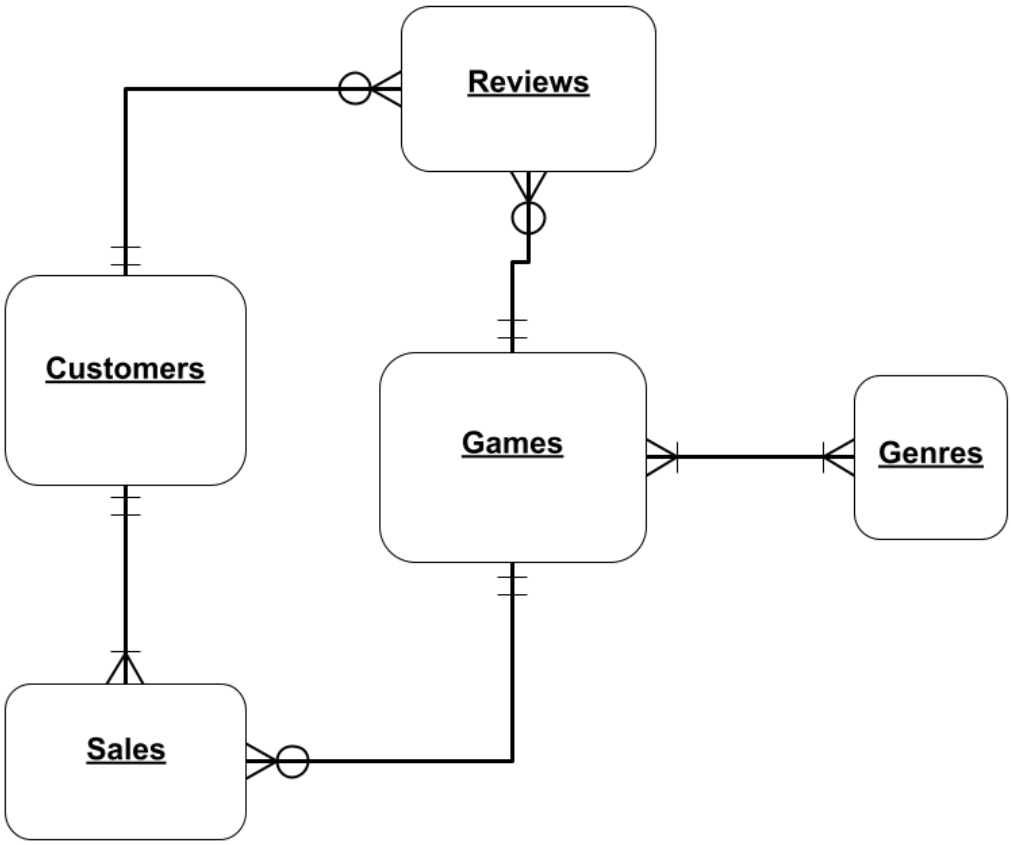
## b) Database Outline, in Words

- **Customers**: Records details about customers who make purchases from our website
    - customer_id: int, auto_increment, unique, not NULL, PK
    - email: varchar, not NULL
    - password: varchar, not NULL, within 20 characters
    - first_name: varchar, not NULL
    - last_name: varchar, not NULL
    - address: varchar, not NULL
    - Relationship: Has a one to many relationship with Sales, implemented through a foreign key in Sales (customer_id)
    - Relationship: Has a one to many relationship with Reviews, implemented with a foreign key in Reviews (cusomer_id)
- **Reviews**: Contains the information about customer reviews for a game

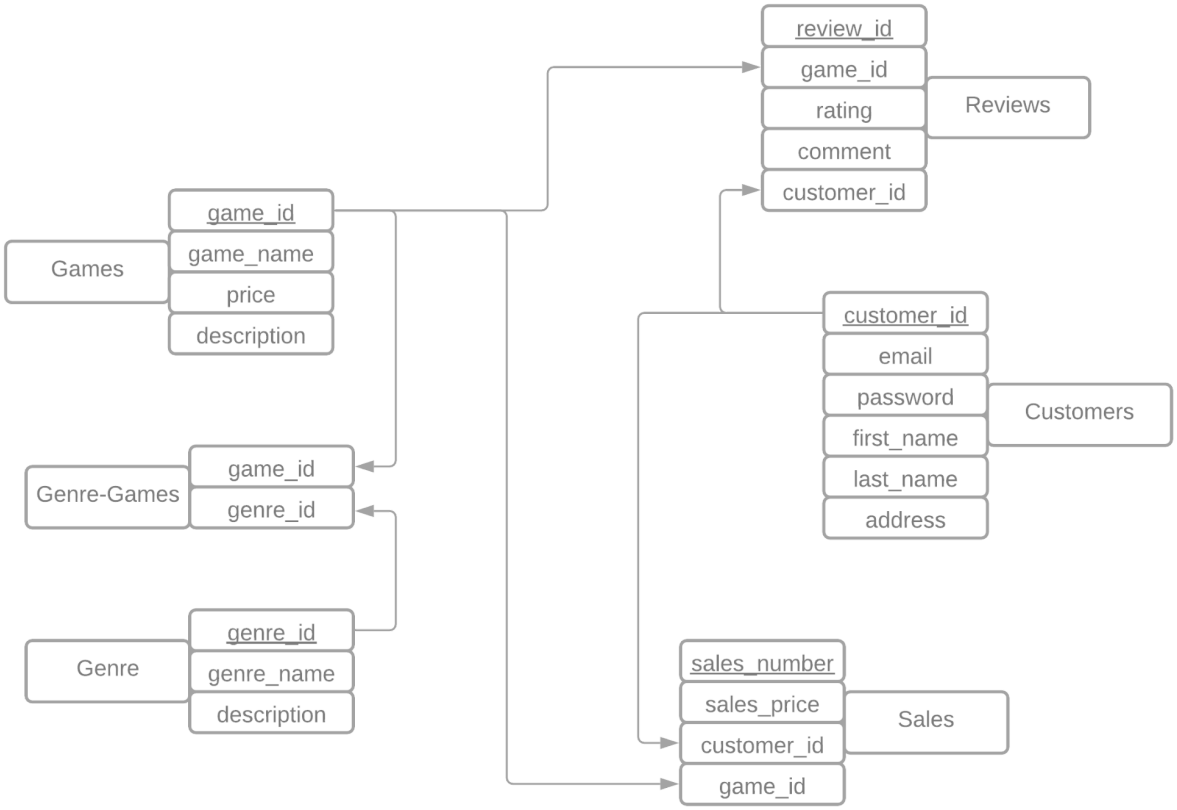
    - review_id:  int, auto_increment, unique, not NULL, PK
    - game_id: int, not NULL, foreign key from Games
    - rating: int, not NULL, range 0-5
    - comment: varchar, not NULL
    - customer_id: int, not NULL, foreign key from Customers
    - Relationship: Reviews on the game is a many-to-one relationship, implemented through a foreign key (game_id) Why is it important because Reviews determine the feedback of a game.
- **Games**: Contains information about the games we sell
    - game_id:  int, auto_increment, unique, not NULL, PK
    - game_name: varchar, not NULL
    - price: float, not NULL
    - description: varchar, not NULL
    - Relationship : There is a one-to-many relationship between games and sales. Sales determine the popularity of the game.
    - Relationship : The relationship between Genre and Games is many-to-many relationship, and is implemented through a table pairing game_id and genre_id
- **Genres**: Contains a list of genres that a game can be
    - genre_id:  int, auto_increment, unique, not NULL, PK
    - genre_name: varchar, not NULL
    - description: varchar, not NULL
    - Relationship: The relationship between Genre and Games is many-to-many relationship, and is implemented through a table pairing game_id and genre_id

- **Genres_Games**: Describes the relationships between Games and Genre
    - game_id: int, foreign key from Games
    - genre_id: int, foreign key from Genre
- **Sales**: Contains the records of all the sales we make
    - sales_number: int, auto_increment, unique, not NULL, PK
    - sales_price: float, not NULL
    - customer_id: int, foreign key from Customer
    - game_id: int, foreign key from Games
    - Relationship: Sales and games have a many-to-one relationship, implemented through a foreign key from Games (game_id)

## c) Entity_Relationship Diagram:


Reviews
Customers
Games
Genres
Sales

## d) Schema:var


review_id
game_id
rating
comment
customer_id
Reviews
game_id
game_name
price
description
Games
customer_id
email
password
first_name
last_name
address
Customers
game_id
genre_id
Genre-Games
genre_id
genre_name
description
Genre
sales_number
sales_price
customer_id
game_id
Sales