

INFO6210 ASSIGNMENT 2 : REPORT

Database Domain: Game

The entire Database has been built as a repository for the Online Games with entities and attributes revolving around the game.

Entities:

The entities used in the our database design are as follows

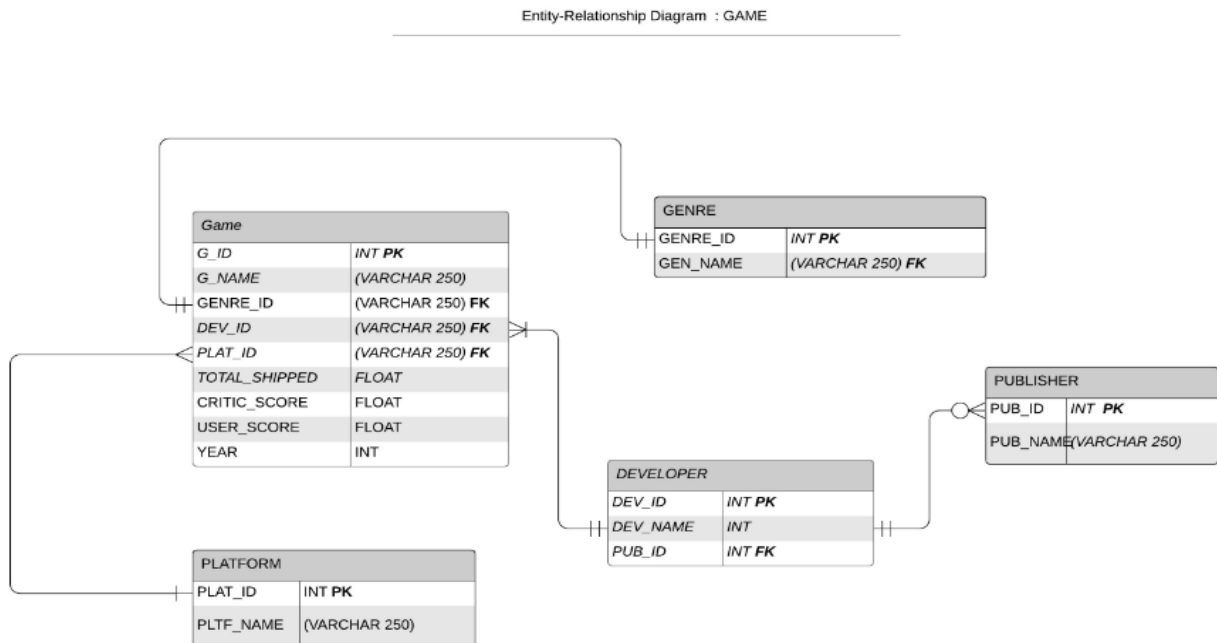
Game: A entity which holds the game details

Genre: A entity that stores the type of genre each game can be associated to.

Developer: A entity that tells information about the developer of the game on which we like to gather the information.

Publisher: A entity that holds information on the each publisher of the game.

Platform: An entity about the platform of the game.



ER-DIAGRAM



Questions:

1. What are the ranges, data types and format of all of the attributes in your entities?

Game Entity:

G_ID : An unique id that is used to identify each game, which is of datatype INT. This shall be considered as the primary key for the Game Table.

G_NAME: an attribute that tells the name of each game. This attribute shall be of Varchar datatype since we can expect different versions of the same game released as a sequel.

GENRE_ID: an attribute that hold the key to the genre type. This shall be of the same datatype in genre entity.

DEV_ID: an attribute that hold the key to the developer of the game. This shall be of the same datatype in developer entity.

PLAT_ID: an attribute that hold the key to which platform the game is. This shall be of the same datatype in platform entity.

TOTAL_SHIPPED: an attribute that talks about the total number of shipment made for that game. This shall be of float datatype as we represent the shipments in millions and we want to restrict it to small range of numbers.

CRITIC_SCORE: an attribute that talks about the rating of the game from a critic and shall be of float datatype since the scores are expected to be in decimal values.

USER_SCORE: an attribute that talks about the rating of the game from a user and shall be of float datatype since the scores are expected to be in decimal values.

YEAR: an attribute that tells about the year of release of the game.

Genre Entity:

GENRE_ID: an unique id that is used to identify each genre type for the game entity, which is of datatype INT.

GEN_NAME: an attribute that tells the genre name and is of varchar datatype.

Developer Entity:

DEV_ID: an unique id that is used to identify each developer for the game entity, which is of datatype INT.

DEV_NAME: an attribute that tells the developer's name and is of varchar datatype.

PUB_ID: an attribute that tells the id of the publisher to which the developer is related to and is the same datatype of that of the Publisher entity.

Publisher Entity:

PUB_ID: an unique id that is used to identify each publisher for the developer entity, which is of datatype INT.

PUB_NAME: an attribute that tells the publisher's name and is of varchar datatype.

Platform Entity:

PLAT_ID: an unique id that is used to identify each platform the game entity can be of, which is of datatype INT.

PLTF_NAME: an attribute that tells the platform name and is of varchar datatype.

2. When should you use an entity versus attribute?

An entity is the table name for each we want to gather the details about and each entity can hold multiple attributes which are the columns associated within a table. For example, Platform is an entity which talks about the platform for the games and holds the attributes like **PLAT_ID** and **PLTF_NAME** which discuss about unique id for a platform and the corresponding name.

3. When should you use an entity or relationship, and placement of attributes?

If an attributes has a relationship specific to one particular entity then that becomes an attribute if the same can be used to link to another entity say like primary -foreign key reference then that becomes an relationship, eg. **PUB_ID** in **Developer** table.

4. How did you choose your keys? Which are unique?

For the tables we have built an ID column has been added to each of the table which holds the primary key constraint. This constraint ensures that each entry for that ID column must be unique and should be not null value.

5. Did you model hierarchies using the “ISA” design element? Why or why not?

No ISA design element is used, as most of our entities have one-to-many or one-to-one relationship between each other.

6. Were there design alternatives? What are their tradeoffs: entity vs. attribute, entity vs. relationship, binary vs. ternary relationships?

There are multiple options with which we can develop an ER depending upon the requirement and depth of analysis. For instance we can build a primary key using combinations of **Game Name** and **Platform_ID** attributes, but we have used a **G_ID** of INT type for the primary key within the **GAME** table.

7. Where are you going to find real-world data to populate your model?

We have created the API IDs for collecting details from twitter about the currently trending games based on tweets.

AUDIT

AUDIT VALIDITY/ACCURACY:

We say data is accurate only when it is neat and with no null or junk values. Here we used query table to check if the data holds any null values.

AUDIT COMPLETENESS:

We have limited the data amount for the initial entry into the tables. Furthermore records shall be pulled in depending upon the requirements

AUDIT CONSISTENCY/UNIFORMITY:

The conceptualized table and the table using the python code has the same schema followed.

AUTHORS:
VIGNESH, SUJITH, ASHWIN
02/21/2020