Steam: A deconstruction of a virtual video-game store

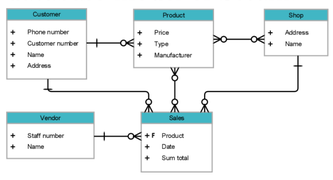
Juan Pablo Borja Espitia, Laura Marcela Santana Sánchez
Universidad distrital Francisco José de Caldas,Bogotá D.C,Colombia
Jpborjae@udistrital.edu.co, lmsantanas@udistrital.edu.co

Introduction

Steam, developed by Valve Corporation, is one of the largest online platforms for buying and selling video games. Valve, founded in 1996 by former Microsoft employees Gabe Newell and Mike Harrington, initially launched Steam as a platform to distribute its own games. Over time, Steam grew into the most influential platform in the gaming industry, serving as a global hub for video game distribution. [1] This project explores the architecture and functionality of Steam's database system. Although Steam began as a modest website built with Java and C++ for downloading Valve games, it has since expanded to publish thousands of games from various developers. The focus of this project is to analyze how Valve designed Steam's database, understanding its key components, and how it serves as a model for developing database systems in the gaming industry. The goal is to deconstruct its structure and explore how its design can inform future projects in the field.

Goal

The project's goal is to apply database foundations, using Steam's system as our follow up objective, starting by the first phase of the design, the database model of Steam, so the project's first part will include the ten steps to design a Database ERM (Entity-Relationship Model) for the first version of the database model of Steam.



Proposed Solution

The Entity Relationship Model (ERM) also known as Entity Relationship Diagram (ERD), is a type of diagram for data modeling, which graphically illustrates the interrelationships of the entities of a database system. Said model, was proposed by the American-Taiwanese theoretical computer scientist, Peter Pin-Shan Chen in 1976. [2] For the Steam ERM Database Model we're gonna use ten steps to design the model:

- Step 1: Define Components
- Step 2: Define entities
- Step 3: Define attributes per entity
- Step 4: Define relationships
- Step 5: Define Relationship types
- Step 6: First ERM Diagram
- Step 7: First Division of Many to Many Relationships
- Step 8: Second ERM Diagram
- Step 9: Get ERM Data Structure
- Step 10: Define Constraint and Properties of Data

Results

Step 1: Define Components

As our first step, the main components that are very relevant into Steam Platform are:

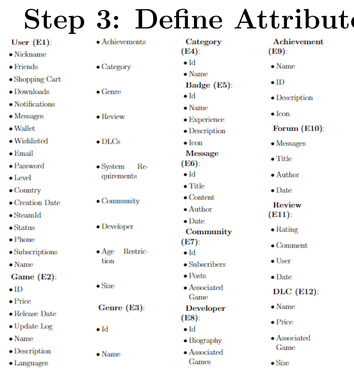
- Games
- Users
- Communities

These components have a strong relationship between themselves, serving as our three pillars into this model.

Step 2: Define Entities

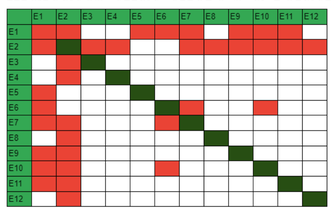
- User - E1
- Game - E2
- Genre - E3
- Category - E4
- Badge - E5
- Message - E6
- Community - E7
- Developer - E8
- Achievement - E9
- Forum - E10
- Review - E11
- DLC (Downloadable content) - E12

Step 3: Define Attributes per Entity

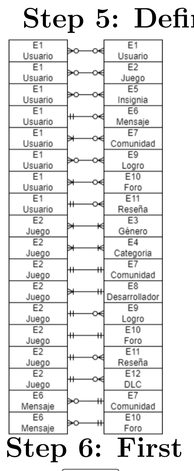


Step 4: Define Relationships

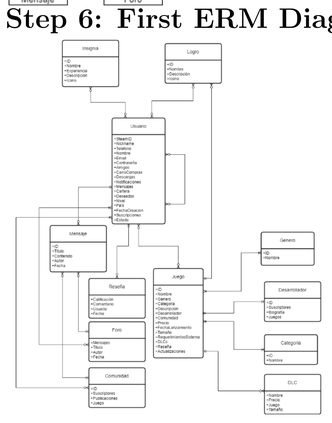
The following table contains all the entities and their possible relationship with each other, the red cell indicates that there's a relationship between those entities.



Step 5: Define Relationships types

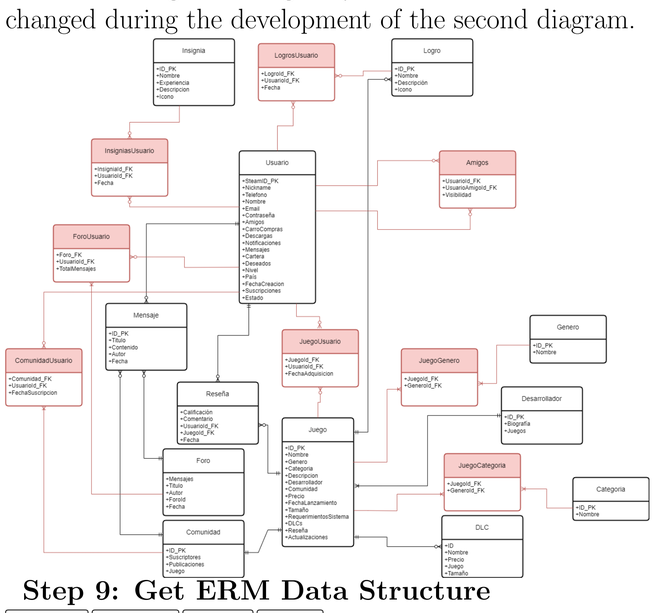


Step 6: First ERM Diagram

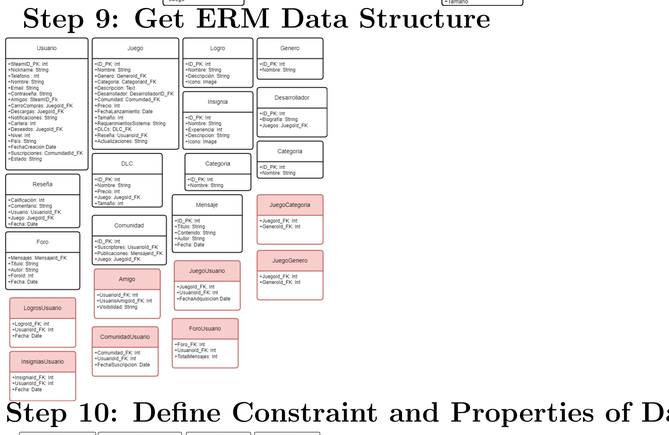


Step 7-8: First Division of Many to Many Relationships and Second ERM Diagram

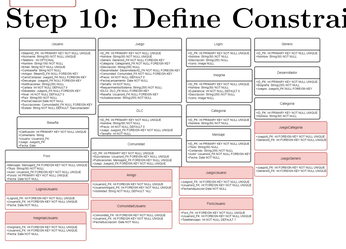
The red colored entities are the separations from many to many relationships, creating new entities each containing the foreign keys of the associated entities previously, and some attributes were slightly changed during the development of the second diagram.



Step 9: Get ERM Data Structure



Step 10: Define Constraint and Properties of Data



Conclusions

- It has been concluded that Steam relies heavily on games and users, meaning that the database model relationships usually aim to the users and the games.
- It was also concluded that the design of this model might be just a lite version of the actual database of steam due to it's complexity and size.

References

[1] Valve Corporation. `Valve Corporation: About Us`. `https://www.valvesoftware.com/es/about`, September 27th, 2021.

[2] SQLearning. `Entity Relationship Model (ERM)`. `https://sqlearning.com/sql-server-introduction/entity-relationship-model/`, July 11th, 2022.

October 4th, 2024