The GameSphere
Videogame store - a databases project

Diego Alejandro Garzón Rodríguez - *20222020009* Daniel Josúe Rozo Rodríguez - *20222020006*
Francisco José de Caldas District University
Databases Foundation - Teacher: Carlos Andres Sierra Virguez

***Abstract*—Given the commercial success that the Steam video game store has had in this last season, the need arises in the company to replicate the database system that said platform has to put it in our own application, the objective is clear, create a database that has the main characteristics of a virtual video game store and that is operable and functional.**

We proposed the design and creation of a simple application with a fairly extensive database, seeking to essentially have the basic functionalities of the platform that the company requested to replicate. The database created has a large repertoire of video games, with detailed information and a fairly clear order that allows you to consult information quickly and easily.

As will be shown below, with this database an application was obtained that allowed the user to have an online video game store in essence, they could know the information about each game, they could see their characteristics and they could also make queries of information about all the games that were within it.

***Index Terms*—Database, digital platform, store, videogames, system**

I. Introduction

ONline video game stores are undoubtedly something totally normal in the current society in which we live, there are different types with a great variety of content and they are undoubtedly very practical for the acquisition of entertainment products today; But where did they come from? As such, until the 1970s, computers began to be used to make any type of purchase. However, it doesn't hurt to give a little context prior to those dates. We could say that it all begins in the 1960s, when EDI or Electronic Data Interchange was created. This creation allowed companies to transmit financial data electronically, such as purchase orders and invoices. The tool improved business transactions and although EDI was initially used to exchange information between companies, it was soon extended to the sale of products online. [7] Already in the 1970s, the first commercial relationships that used a computer as such appeared, although they offered a very limited service. Exactly in 1979, Michael Aldrich invented online shopping, creating the Teleputer, which was a computer entertainment center. This allowed users to purchase products using a computer, but online purchases could only be made through a specific terminal. [2]. In later years, television was also widely used as a way to promote any type of product, but it was not until the 1990s that the Internet became popular and the creation of exclusive portals to sell products, such as Amazon and Ebay, which improved the accessibility of users to discover new brands and make purchases. It was in 1991 that the World Wide Web was created, allowing the Internet to become a tool accessible to everyone, promoting online commerce on a large scale. Already with the Internet in its fury The Global Network Navigator sold the first advertisement on the Internet to a law firm in Silicon Valley; This marked the beginning of online advertising and the creation of an Internet advertising industry. [6]

After that, online commerce only grew and among the things that began to be offered, entertainment and video games were already a point of important need. And this is where the story of Steam, the largest online video game store in the world, begins; in 2002 when Valve, a very successful video game company, decided to create a digital distribution service for its games. [1] The idea behind Steam was to provide a means for players to get updates and patches for their games automatically, which would help combat piracy and cheaters in multiplayer games. The development of the platform began in 2002 and after a successful beta version, it was launched on September 12, 2003. In 2003, since there began to be a boom in multiplayer games, the platform became more famous and hosted many more users. By 2007, Valve had convinced many of the largest and most important PC gaming companies that selling their games on Steam was a good idea. Today, Steam is the largest and most popular video game store in the world, with more than 120 million monthly active users and a catalog of more than 50 thousand games.

Our technology company, which is medium-sized, wants to competently enter the online video game market. To do this, we were asked to replicate the database system that the Steam application brings, and to generate our own application that can be launched as a prototype and that has the most characteristic functionalities of the virtual game store. That is where this project is born; part of the need to generate our own application that allows us to enter the online video game market. To do this, we as engineers dedicated ourselves to analyzing in detail the structure that the application could have, we examined main entities and extracted the most important characteristics of the application; thus generating a database system that maintains the essence of what a virtual video game store is, with its most important characteristics, but at the same time was a realization of our institution. [3] Currently there are undoubtedly many more virtual stores from which we could draw reference, but as engineers, we seek to replicate the best of all of them, the system seeks to be simple, practical and above all accessible to the user, we seek that its practicality

does not only attract the user but keep them with us.

A. 1. METHODS AND MATERIALS

To create the application's database system, in the first instance, the ontology process with 10 steps to get the final draw of the database was used. In it we are in charge of defining the main components of the system, we make each of the entities clear and we describe each of them with all the characteristics that we consider necessary. Clarifying that, throughout the development of the entire project, there were certain changes in this step that allowed better development and a better understanding of the system in general. In this way, the general diagram of the database appears, which is an entity relationship diagram, presented below:

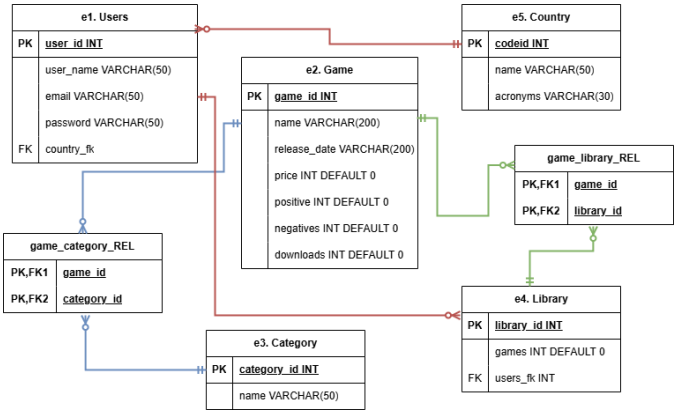

Fig. 1. Entity relationship diagram created in the draw.io tool as a result of the project database design process

The methodology basically focused on generating a replica of the Steam database and from there designing its own system. The features and functionalities that we found useful about the app were:

- A wide catalog of video games with the main characteristics of each game
- That each game have review-type indicators of likes, dislikes or number of downloads, which allow the user to know in a general way the quality of each game in the store
- A personal library for each user where they can see the number of games they have downloaded
- That each user has their own relevant and organized information, such as their ID, their username (to be able to have a profile in the application), their email, their password and their country of origin
- That each game can have or belong to a specific category, so that the user can view the games in a more orderly manner
- That the user can make searches and queries about all the games in the store based on their information

With these clear functionalities and the database system model designed, we began to shape the project. We use a wide variety of tools throughout the development, the main ones include:

- DBeaver CE (Database Management Environment)
- PostgreSQL (Language used for creating and querying the database)
- Python (Programming language used for all the development of the web application external to the database)
- Flask (Web microframework for Python, with which the application's web services could be implemented)
- Additional design tools like Draw.io, Canva and Overleaf

Among the technical decisions that were made throughout the development of the project, it was decided to keep the system as simple as possible. At one point we had a total of ten entities within the database's entity-relationship design, but after certain reviews, we concluded that it was best to leave only the basic functionalities and not include things that might affect the abstraction. general of the system. As an example of this we can cite the "Community" entity that we had in some previous designs, the idea with this entity was to replicate the addition of a community forum that existed in the most recent versions of Steam, where users could interact with others, but Given its characteristics that are more similar to a mini social network than a storage place for information, we chose to eliminate it completely from the system and from there we understood that often less is more.

The database system, as we saw in the final design shown previously, consists of seven related entities, the first of which is "Users", which is responsible for storing all the user information that can be useful in the application. such as your ID (which is unique and is the primary key of the entity), your username, your email address and your password, key data for user identification within the application, in addition to having a foreign key of the "Country" entity that we will describe later, since a user can only have one country of origin, but a country can have many users. We also find the entity "Game" which is responsible for storing all the information related to each and every one of the games in the store, containing game-id (which is the unique identifier id of each game and is the primary key of the entity), name (of the game), release date, price, positive(The number of positive reviews that each game has), negatives(The number of negative reviews that each game has) and downloads(The number of downloads with which each game counts); This entity is possibly the most important given the information it handles and the context of the system. We also have an entity called "Category" that stores each and every category of the games in the store, which only has category-id (primary key and unique identifier of each category) and name (referring to the name of the category) . The "Library" entity is unique and specific to each user since it is responsible for storing the number of games downloaded by the user. On the other hand, we have the "Country" entity that is in charge of storing all the data referring to all the countries in the world, to which a particular user may belong, and which, as we have already seen, is a foreign key of the "Users" entity. ". Finally, there are the two entities that break the many-to-many relationships presented between "Game" and "Category" and between "Game" and "Library"; called "game-category-rel" and "game-library-rel" respectively. These are the entities that we consider most important and that best describe the database system that we want to propose for our application.

On the other hand, each of these entities needs to be

filled with information according to its characteristics, for this we use various data sources that were previously cited in a technical document of the application, but that will be briefly cited below:

- **Kaggle** (Source used to acquire all the necessary information about the games and their respective categories, the information came out in .csv file format)
- **Mockaroo**(Source used to acquire all the necessary user information such as user-names and email addresses, the information came out in .csv file format)
- **GitHub repositories and Datamundial** (Sources used to acquire all the necessary information on the countries of origin of the application users, part of all the information came out in .csv file format and another part, due to technical complications, had to be sorted and entered with the "insert into" command in PostgreSQL)

Each of the .csv files with the respective information was uploaded to the database as an import in a relatively simple way, since DBeaver allows you to organize the csv file that was going to be uploaded by columns, with this you organized and defined the table and columns where the downloaded information was going to go. With the information from all the tables uploaded and with the structure of the database already finished, we proceeded to test using a script all the user queries that we considered important, and all those that worked optimally were left for the user could evidence them. These queries helped us draw up a roadmap for the implementation of the web services of the final application. This application was created with Flask, the previously mentioned tool, and web services were used to perform the queries posed to the database. from the final website and a much more pleasant interface was presented for the end user created with Python.

*B. 2. **RESULTS***

The development process of the proposed database system was undoubtedly a process of continuous improvement, where the adaptability and perseverance of us as developers were tested on several occasions. We previously mentioned all the methodologies, strategies and materials that were of vital importance to carry out the development of the project. Now we want to present the results that we obtained in the final stage, to do so we will show certain screenshots of the final application that demonstrate the functionalities that we seek to replicate from an online video game store. Below are the main screens of the application:

- **Review figures 2, 3 and 4 along with their respective descriptions**

Already within the database management tool, we can find the following:

- **Review figures 6 and 7 along with their respective descriptions**

In these images we can see each of the entities that our final database system of the virtual video game store has, in addition, we can show the number of rows that each one has, so that we can get an idea of the significant amount

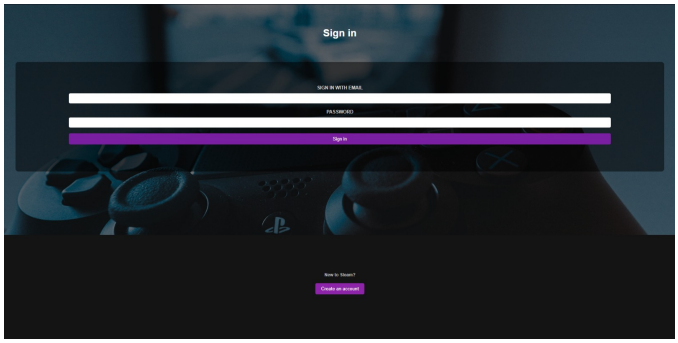

Fig. 2. This is the home screen of the web application, where the user enters their data for validation within the database, if it is found they can enter, if not, they must create their own username and password.

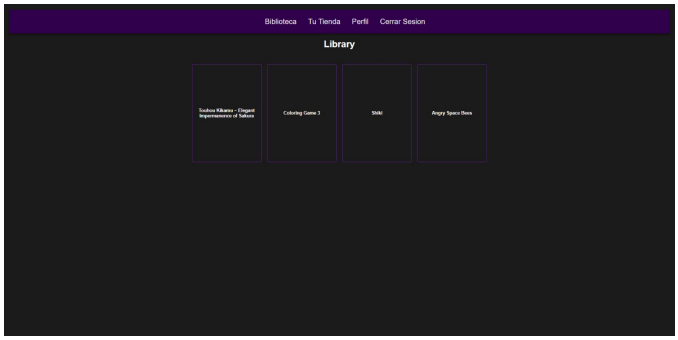

Fig. 3. This is the user's first main screen where they can access their library and view their downloaded games.

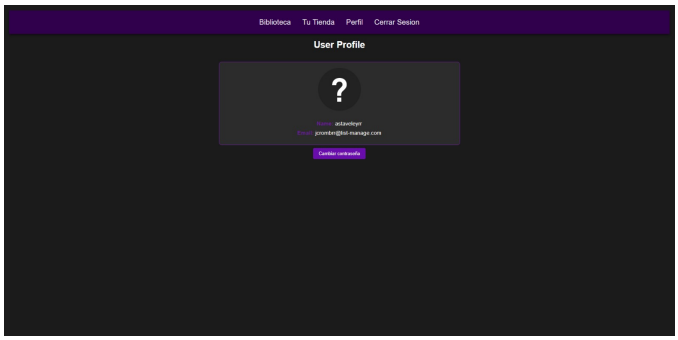

Fig. 4. This is the user's second main screen where they can see their profile data and if they wish, they have the option to log out.

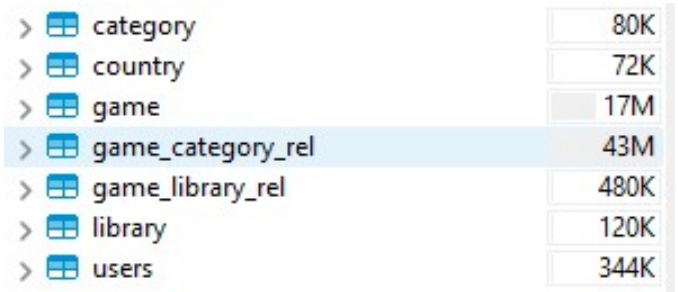

Fig. 5. Database System Entities

Table Name	ID Objeto	Propietario	Tablespace	Cuenta de filas estimada
category	16.956	postgres	pg_default	406
country	16.960	postgres	pg_default	249
game	16.964	postgres	pg_default	60.950
game_category_rel	16.967	postgres	pg_default	775.687
game_library_rel	16.970	postgres	pg_default	5.483
library	16.973	postgres	pg_default	1.000
users	16.978	postgres	pg_default	1.000

Fig. 6. Entities of the database system and their respective number of rows with stored information.

of information that we are handling in this model and we also show that the information of each entity could be entered with complete success. And although we previously presented the entity relationship model of the database made by us in draw.io, we want to show the entity relationship model that the DBeaver CE tool gives us after creating and entering all the necessary information in the database; This is in order to verify that the initially proposed model was actually carried out correctly.

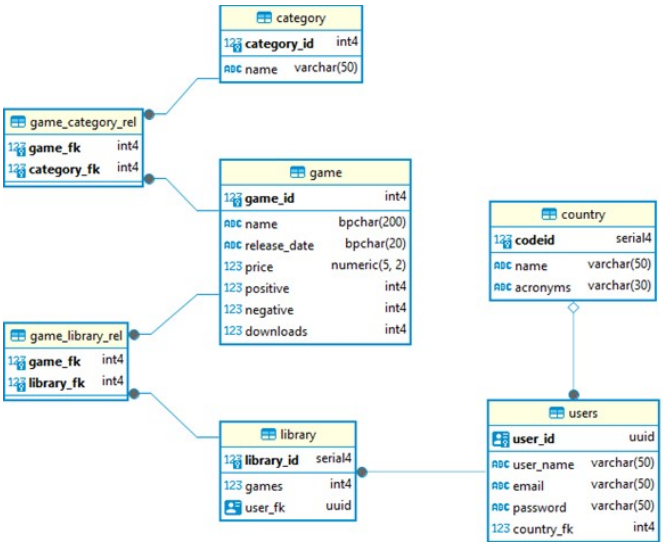

Fig. 7. Entity relationship diagram of the final database automatically generated by the DBeaver CE database management tool.

The database system was tested on several occasions, among the tests carried out there were two that were most significant; The first was the testing of the data itself, where the review was carried out to see if this data that had been acquired really worked and was storable and operable within the database. The test was carried out successfully since, as shown in previous images, all entities were able to house all the information provided to them in their respective columns according to the characteristics of each one. The other most important test within the application was the testing of user queries; It was vital for us that queries could be made with the information entered into the database, since this allowed us to comply with the condition of data operability within the application that had been initially proposed. Numerous queries were made with the final database and it was found that the expected results could be obtained easily and quickly. Information queries to the database are displayed within the application in a separate section, which looks like this:

Fig. 8. Menu of possible queries to make from the web application.

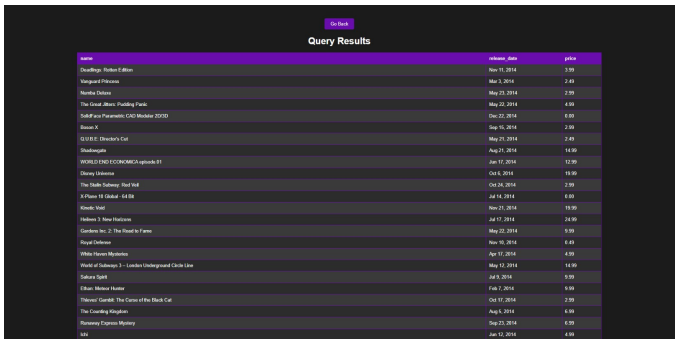

Fig. 9. Results of queries made from the web application.

We have already shown the main screens of the final application and explained the functionalities resulting from them. Next, we show what the data entered in each entity looks like in general, displayed from a query in the DBeaver CE database management tool, clarifying that only a small part is shown as a sample because it would be too complex to show all of them. each of the rows.

- Users (Review figure 10, with its respective description)
- Game (Review figure 11, with its respective description)
- Category (Review figure 12, with its respective description)
- Library (Review figure 13, with its respective description)
- Country (Review figure 14, with its respective description)

Each of the entities, as evidenced in the screenshots, had their respective data review, the great breadth in terms of data volume allowed us to generate precise and useful queries for the end user. The testing philosophy focused on practicality and usefulness; We use unit tests to evaluate the data of each entity in order to see its operation within the proposed operations. We also used integration tests, functional tests and system tests, which had the objective of seeing how all the entities were related as a single system when a particular query was executed.

	user_id	user_name	email	password
1	b7803a1e-3435-4f29-936d-f8c232	ahaistwellu	bolifauntu@ted.com	354OfaWaPemymlr
2	5f104e6b-4106-4c82-9f2c-0323e25	bcornelsv	mdornv@narod.ru	3118163841n
3	d50fce63-7186-4bc1-a5a2-b57edf	tcartmelw	kduplantierw@hexun.com	Rtg9JEjY2MARgI1h
4	29350c6e-e078-4ab7-8d20-2c1788	mpykex	rmccuthaisx@tumblr.com	love4music
5	f91a3ae9-b619-4518-a1cb-a7c024	hpepineauxy	rpatesy@photobucket.com	gadyt897
6	dec80c3b-9fd9-405d-b9c7-b3561	wchampionnetz	cgibbensz@4shared.com	a226joyi
7	0a661f95-3526-4a73-ba70-23d8c6	hmichieli10	dmorad10@ask.com	iqecyka288
8	d7f810a9-e1c1-4fc2-a782-766f2f0	bglavis11	mcopes11@angelfire.com	oraloral93
9	78d8bf5d-a40d-4d8d-afc3-ef7fdb	mwillisch12	gmontez12@icio.us	vav14639832
10	596c7a77-4690-4db3-983c-a1c607	qleming13	mcockrill13@squidoo.com	YzXtV6TUxMwoQKml
11	28eb647b-c525-4a6a-ae22-12b195	dpasley14	mbayless14@toplist.cz	AVYzilyQevot714
12	a28f130f-a0bd-4211-8d36-bc2a35	sfumagalli15	adedenham15@reuters.com	volvo850
13	48b37ab1-6057-4b5e-adb8-018b3	cpittaway16	dwallworth16@state.tx.us	9lvquzz55
14	66b142ce-03ad-4c8f-8517-c0c11d	erankine17	nsobey17@sohu.com	mysekf123
15	13e10af3-f755-4112-b159-669609	bplampin18	ecarlsson18@etsy.com	killadude1
16	15e923ad-8593-455a-acd1-e5cad6	hharrell19	byeowell19@sphinn.com	morior666
17	b9eb975e-5987-4b96-b414-24a13	lhanby1a	obrett1a@desdev.cn	wvhwvo15
18	2f7d2df6-5182-400c-9435-8472be	mbampkin1b	nmiles1b@google.ca	paxton0
19	2e273b49-f26d-4744-a449-03d379	lfacher1c	dpietruszewicz1c@pcworld.c	rfvfrfvf88
20	5454e36f-a7ec-4256-b4f2-4bc4d7	jjanson1d	rmcgee1d@flickr.com	joskes9
21	37944645-bf91-44de-bf38-9e69aa	cgullefant1e	selner1e@netlog.com	turok61258
22	756655dc-dd92-41cf-a5ba-3bed8	tmcquirk1f	bcalifornia1f@ft.com	jubilo5247
23	9875764b-81e5-4d50-9e03-f9a90a	cdeanel6	slinkel6@nps.gov	tcu2ba
24	7971cb45-f4d8-4453-bf3f-97372df	zeilhersen1q	rentwhistle1q@icq.com	jablana123

Fig. 10. Query made from DBeaver CE of the Users class with all its respective information.

	game_id	name	release_date	price	positive	negative	downloads
1	655.310	Train Bandit	Oct 12, 2017	0,99	53	5	20.000
2	1.355.720	Henosis™	Jul 23, 2020	5,99	3	0	20.000
3	1.139.950	Two Weeks in Painland	Feb 3, 2020	0	50	8	20.000
4	1.469.160	Wartune Reborn	Feb 26, 2021	0	87	49	100.000
5	1.659.180	TD Worlds	Jan 9, 2022	10,99	21	7	20.000
6	1.178.150	MazM: Jekyll and Hyde	Apr 2, 2020	14,99	76	6	20.000
7	320.150	Deadlings: Rotten Edition	Nov 11, 2014	3,99	225	45	100.000
8	1.026.420	WARSAW	Oct 2, 2019	23,99	589	212	50.000
9	485.000	Cthulhu Realms	Jul 1, 2016	0	147	58	100.000
10	1.620.060	Clockwork Dungeon	Aug 27, 2021	1,99	5	0	20.000
11	825.930	Royal Battleships	Apr 6, 2018	2,99	36	11	50.000
12	1.454.010	Diary of Lucie	Nov 25, 2020	12,99	100	5	20.000
13	1.330.820	Hunting Unlimited 3	Nov 6, 2020	9,99	17	0	20.000
14	346.560	Hero of the Kingdom II	Feb 20, 2015	7,99	2.046	120	200.000
15	1.362.670	KHIO	Jul 24, 2020	9,99	3	1	20.000
16	290.870	Steam Squad	Jul 28, 2016	17,99	61	29	50.000
17	575.760	Project: R.E.B.O.O.T 2	Jan 27, 2017	2,99	29	28	50.000
18	434.030	Aerofly FS 2 Flight Simulator	Nov 20, 2017	37,49	1.490	408	200.000
19	810.740	Turtle Lu	Mar 22, 2018	2,99	4	10	50.000
20	2.073.470	Kanjozoku Game レーサー	Jul 16, 2022	5,99	392	57	50.000
21	589.250	PowersVR	Oct 10, 2020	9,99	37	22	20.000
22	677.880	Draw Your Game	Aug 10, 2017	4,99	18	50	50.000
23	1.294.780	Pathfinders: Memories	Feb 16, 2021	2,99	38	4	20.000
24	1.342.850	Bright Days in Quarantine	Jul 18, 2020	1,99	24	5	20.000

Fig. 11. Query made from DBeaver CE of the Game class with all its respective information.

	category_id	name
1	1	1980s
2	2	1990's
3	3	2.5D
4	4	2D
5	5	2D Fighter
6	6	2D Platformer
7	7	360 Video
8	8	3D
9	9	3D Fighter
10	10	3D Platformer
11	11	3D Vision
12	12	4 Player Local
13	13	4X
14	14	6DOF
15	15	Abstract
16	16	Action
17	17	Action-Adventur
18	18	Action Roguelike
19	19	Action RPG
20	20	Action RTS
21	21	Addictive
22	22	Adventure
23	23	Agriculture
24	24	Aliens

Fig. 12. Query made from DBeaver CE of the Category class with all its respective information.

Broadly speaking, that is the final web application, where we were able to design, restructure and implement all the

	library_id	games
1	1	0
2	2	0
3	3	0
4	4	0
5	5	0
6	6	0
7	7	0
8	8	0
9	9	0
10	10	0
11	11	0
12	12	0
13	13	0
14	14	0
15	15	0
16	16	0
17	17	0
18	18	0
19	19	0
20	20	0
21	21	0
22	22	0
23	23	0
24	24	0

Fig. 13. Query made from DBeaver CE of the Library class with all its respective information. Here, by default, each user has the number of games set to zero, as the application is used, this number will grow. It was the only column that we did not assign a specific value by default, understanding the functionality and characteristics of the library.

	codeid	name	acronyms
1	4	Afghanistan	AFG
2	248	Åland Islands	ALA
3	8	Albania	ALB
4	12	Algeria	DZA
5	16	American Samoa	ASM
6	20	Andorra	AND
7	24	Angola	AGO
8	660	Anguilla	AIA
9	10	Antarctica	ATA
10	28	Antigua and Barbuda	ATG
11	32	Argentina	ARG
12	51	Armenia	ARM
13	533	Aruba	ABW
14	36	Australia	AUS
15	40	Austria	AUT
16	31	Azerbaijan	AZE
17	44	Bahamas	BHS
18	48	Bahrain	BHR
19	50	Bangladesh	BGD
20	52	Barbados	BRB
21	112	Belarus	BLR
22	56	Belgium	BEL
23	84	Belize	BLZ
24	204	Benin	BEN

Fig. 14. Query made from DBeaver CE of the Country class with all its respective information.

concepts related to a stable and functional database system. Each of the entities that were described in the previous step functions completely normally and contains the necessary data for the system to function optimally as stipulated. The intention of the project is to add value to the company, generating its own system and making said system a starting point for a more sophisticated virtual video game store that in the future can compete with or even surpass Steam. Each entity has its respective backup copy of information in the root folder of the project in order to protect all the data already stored in the database.

II. Conclusion

The success of the application is summarized in that it preserves the basic functionalities and characteristics of what we seek to replicate for the company, a virtual video game store. The work carried out on our part is satisfactory to the extent that the project is useful to the user, we achieved that the database system created had the characteristics of the virtual store that we proposed at the beginning, understanding that less is more, It was achieved that the information carefully stored in each of the entities worked together to achieve a system that not only provides video game information to the user, but also allows them to make certain types of queries in it, such as a virtual store. real. In addition, the user has a system that is easy to use, simple to replicate and, above all, useful for the initially intended purpose. We are sure that this project will add value to the company, both now in the present and much more in the not too distant future.

References

[1] Cabácas, T. (2018). “15 years of Steam, the platform that changed the way we play”. Taken and adapted from: 15 years of Steam, the platform that changed the way we play (muycomputer.com)

[2] GlobalSeller. Rojas. L. (2023). ”What is the history of e-commerce”. Taken and adapted from: The history of e-commerce -GlobalSeller

[3] Kiero. (2022). “12 Characteristics of a Successful Online Store”. Taken and adapted from: 12 characteristics of a successful online store - Kiero Ecommerce Platform: Digital menu — Online Store Without Commissions

[4] Mockaroo. (2022). “Mockaroo”. Taken and adapted from: Mockaroo - Random Data Generator and API Mocking Tool — JSON / CSV / SQL / Excel

[5] RetabloWeb. (2021). “¿What is an Online Store? Characteristics, Elements and Types.”. Taken and adapted from: What is an Online Store? Characteristics, Elements and Types. - RetabloWeb

[6] Rockcontent. Cárdenas. J. (2020). ”¿What is the history of electronic commerce in the world?”. Taken and adpted from: Learn about the history of electronic commerce and how it has evolved until today - Rockcontent

[7] Slideshare. Anaya. J. (2017). ”Timeline on the history and evolution of electronic commerce.”. Taken and adapted from: History and evolution of electronic commerce - Slideshare.