

Universidade Federal de Sergipe  
Centro de Ciências Exatas e Tecnologia  
Departamento de Computação

# Relatório de Banco de Dados: Fase 2.1

Projeto Lógico e Implementação Relacional (Steam)

**Alunos:** Rafael Gomes Oliveira Santos  
Alicia Vitória Sousa Santos

**Professor:** André Britto

São Cristóvão  
2026

# Sumário

<b>1</b>	<b>Especificação e Descrição do Problema</b>	<b>2</b>
<b>2</b>	<b>Projeto Conceitual (DER)</b>	<b>2</b>
<b>3</b>	<b>Projeto Lógico Relacional</b>	<b>3</b>
3.1	Estratégias de Mapeamento . . . . .	3
<b>4</b>	<b>Implementação SQL (PostgreSQL)</b>	<b>4</b>
<b>5</b>	<b>Acesso ao Ambiente (AWS RDS)</b>	<b>11</b>

# 1 Especificação e Descrição do Problema

O sistema desenvolvido visa gerir uma plataforma de distribuição digital de jogos e softwares, baseada no ecossistema da Steam. As principais regras de negócio modeladas para o ambiente relacional são:

- **Gestão de Identidade e Finanças:** Cada usuário possui um identificador único e uma carteira digital vinculada (relacionamento 1:1), permitindo transações financeiras com precisão decimal.
- **Especialização de Produtos:** Implementou-se uma hierarquia onde **Produto** atua como entidade generalista, especializada em **Jogos**, **Software** e **DLC**. Essa estrutura garante que atributos comuns (título, preço) sejam centralizados, enquanto atributos específicos (requisitos de sistema, versão) sejam segregados.
- **Normalização:** Atributos complexos foram tratados para respeitar as formas normais. O atributo multivalorado *idiomas* foi otimizado utilizando o recurso de arrays nativos do PostgreSQL, garantindo desempenho sem violar a integridade semântica.
- **Dependências e Agregações:** Entidades como **Conquista** e **RequisitoSistema** são tratadas como entidades fracas, dependendo existencialmente do Jogo. O relacionamento de compra inclui a agregação **Item\_compra**, registrando o histórico de preço pago no momento da transação.

## 2 Projeto Conceitual (DER)

O Diagrama Entidade-Relacionamento (DER) apresentado abaixo utiliza a notação de Heuser para descrever a semântica do banco de dados.



2. **Relacionamentos N:M:** Foram decompostos em tabelas associativas (Jogos\_has\_Genero, Biblioteca, Item\_compra), contendo chaves compostas.
3. **Integridade:** Definiu-se restrições de UNIQUE para e-mails e nomes de gênero, além de CHECK constraints para validação de votos (0-5) e tipos de requisitos.

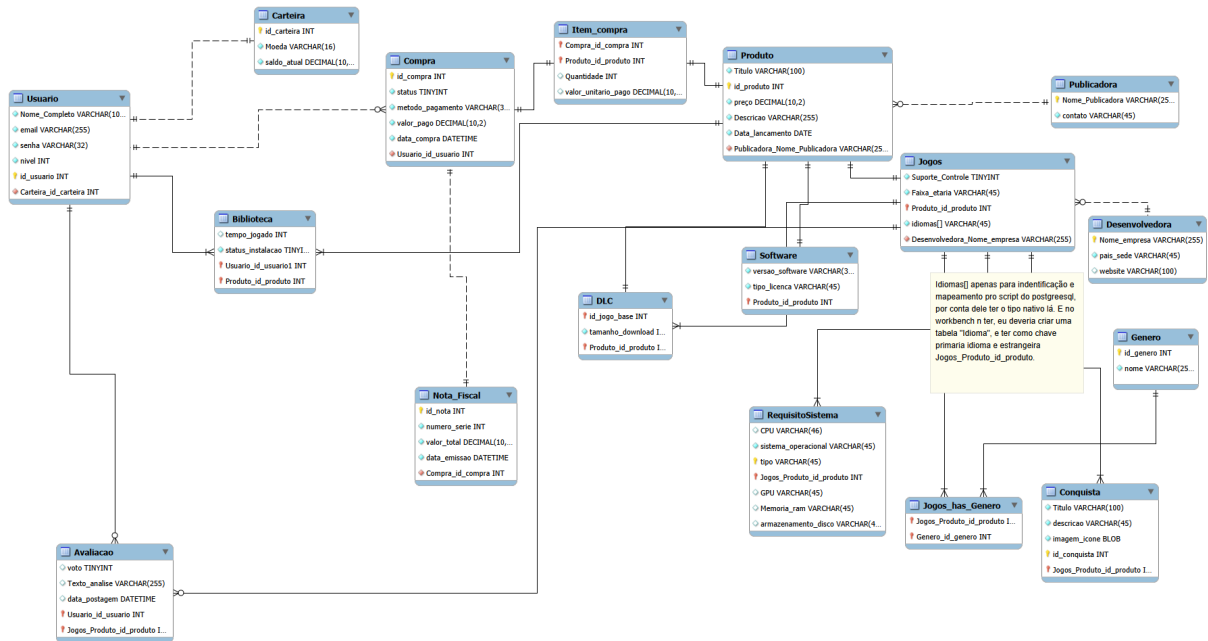


Figura 2: Modelo Lógico Relacional (Workbench)

## 4 Implementação SQL (PostgreSQL)

O banco de dados foi implementado no SGBD PostgreSQL hospedado na AWS RDS. Abaixo, o script DDL utilizado para a criação das estruturas, contemplando tipos de dados (SERIAL, NUMERIC, BYTEA), restrições de integridade e indexação.

```

1  -- PostgreSQL Script - Convers o do MySQL Workbench
2  -- Modelo: Steam Database      Vers o: 1.0
3  -- PostgreSQL Forward Engineering
4
5  -----
6  -- Schema steam
7  -----
8  CREATE SCHEMA IF NOT EXISTS steam;
9  SET search_path TO steam;
10
11 -----
12 -- Table steam.Carteira

```

```

13 -----
14 CREATE TABLE IF NOT EXISTS Carteira (
15     id_carteira SERIAL PRIMARY KEY,
16     Moeda VARCHAR(16) NOT NULL,
17     saldo_atual NUMERIC(10,2) NOT NULL
18 );
19
20 -----
21 -- Table steam.Usuario
22 -----
23 CREATE TABLE IF NOT EXISTS Usuario (
24     Nome_Completo VARCHAR(100) NOT NULL,
25     email VARCHAR(255) NOT NULL UNIQUE,
26     senha VARCHAR(32) NOT NULL,
27     nivel INT NOT NULL,
28     id_usuario SERIAL PRIMARY KEY,
29     Carteira_id_carteira INT NOT NULL UNIQUE,
30     CONSTRAINT fk_Usuario_Carteira
31         FOREIGN KEY (Carteira_id_carteira)
32         REFERENCES Carteira (id_carteira)
33         ON DELETE NO ACTION
34         ON UPDATE NO ACTION
35 );
36
37 CREATE INDEX idx_Usuario_Carteira ON Usuario(Carteira_id_carteira);
38
39 -----
40 -- Table steam.Genero
41 -----
42 CREATE TABLE IF NOT EXISTS Genero (
43     id_genero SERIAL PRIMARY KEY,
44     nome VARCHAR(255) NOT NULL UNIQUE
45 );
46
47 -----
48 -- Table steam.Desenvolvedora
49 -----
50 CREATE TABLE IF NOT EXISTS Desenvolvedora (
51     Nome_empresa VARCHAR(255) PRIMARY KEY,
52     pais_sede VARCHAR(45) NOT NULL,
53     website VARCHAR(100) NULL
54 );
55
56 -----
57 -- Table steam.Publicadora
58 -----
59 CREATE TABLE IF NOT EXISTS Publicadora (

```

```

60 Nome_Publicadora VARCHAR(255) PRIMARY KEY,
61 contato VARCHAR(45) NOT NULL
62 );
63
64 -----
65 -- Table steam.Produto
66 -----
67 CREATE TABLE IF NOT EXISTS Produto (
68     Titulo VARCHAR(100) NOT NULL,
69     id_produto SERIAL PRIMARY KEY,
70     preco NUMERIC(10,2) NOT NULL,
71     Descricao VARCHAR(255) NOT NULL,
72     Data_lancamento DATE NOT NULL,
73     Publicadora_Nome_Publicadora VARCHAR(255) NOT NULL,
74     CONSTRAINT fk_Produto_Publicadora1
75         FOREIGN KEY (Publicadora_Nome_Publicadora)
76         REFERENCES Publicadora (Nome_Publicadora)
77         ON DELETE NO ACTION
78         ON UPDATE NO ACTION
79 );
80
81 CREATE INDEX idx_Produto_Publicadora ON Produto(
82     Publicadora_Nome_Publicadora);
83
84 -----
85 -- Table steam.Jogos
86 -----
87 CREATE TABLE IF NOT EXISTS Jogos (
88     Suporte_Controle BOOLEAN NOT NULL,
89     Faixa_etaria VARCHAR(45) NOT NULL,
90     Produto_id_produto INT PRIMARY KEY,
91     idiomas VARCHAR(45)[] NOT NULL,
92     Desenvolvedora_Nome_empresa VARCHAR(255) NOT NULL,
93     CONSTRAINT fk_Jogos_Produto1
94         FOREIGN KEY (Produto_id_produto)
95         REFERENCES Produto (id_produto)
96         ON DELETE NO ACTION
97         ON UPDATE NO ACTION,
98     CONSTRAINT fk_Jogos_Desenvolvedora1
99         FOREIGN KEY (Desenvolvedora_Nome_empresa)
100         REFERENCES Desenvolvedora (Nome_empresa)
101         ON DELETE NO ACTION
102         ON UPDATE NO ACTION
103 );
104 CREATE INDEX idx_Jogos_Desenvolvedora ON Jogos(
105     Desenvolvedora_Nome_empresa);

```

```

105
106 -----
107 -- Table steam.Avaliacao
108 -----
109 CREATE TABLE IF NOT EXISTS Avaliacao (
110     voto SMALLINT NULL,
111     Texto_analise VARCHAR(255) NULL,
112     data_postagem TIMESTAMP NULL,
113     Usuario_id_usuario INT NOT NULL,
114     Jogos_Produto_id_produto INT NOT NULL,
115     PRIMARY KEY (Jogos_Produto_id_produto, Usuario_id_usuario),
116     CONSTRAINT fk_Avaliacao_Usuario1
117         FOREIGN KEY (Usuario_id_usuario)
118         REFERENCES Usuario (id_usuario)
119         ON DELETE NO ACTION
120         ON UPDATE NO ACTION,
121     CONSTRAINT fk_Avaliacao_Jogos1
122         FOREIGN KEY (Jogos_Produto_id_produto)
123         REFERENCES Jogos (Produto_id_produto)
124         ON DELETE NO ACTION
125         ON UPDATE NO ACTION
126 );
127
128 CREATE INDEX idx_Avaliacao_Usuario ON Avaliacao(Usuario_id_usuario);
129 CREATE INDEX idx_Avaliacao_Jogos ON Avaliacao(Jogos_Produto_id_produto);
130
131 -----
132 -- Table steam.Biblioteca
133 -----
134 CREATE TABLE IF NOT EXISTS Biblioteca (
135     tempo_jogado INT NULL,
136     status_instalacao BOOLEAN NOT NULL,
137     Usuario_id_usuario1 INT NOT NULL,
138     Produto_id_produto INT NOT NULL,
139     PRIMARY KEY (Usuario_id_usuario1, Produto_id_produto),
140     CONSTRAINT fk_Biblioteca_Usuario1
141         FOREIGN KEY (Usuario_id_usuario1)
142         REFERENCES Usuario (id_usuario)
143         ON DELETE NO ACTION
144         ON UPDATE NO ACTION,
145     CONSTRAINT fk_Biblioteca_Produto1
146         FOREIGN KEY (Produto_id_produto)
147         REFERENCES Produto (id_produto)
148         ON DELETE NO ACTION
149         ON UPDATE NO ACTION
150 );
151

```



```

152 CREATE INDEX idx_Biblioteca_Usuario ON Biblioteca(Usuario_id_usuario1);
153 CREATE INDEX idx_Biblioteca_Produto ON Biblioteca(Produto_id_produto);
154
155 -----
156 -- Table steam.Compra
157 -----
158 CREATE TABLE IF NOT EXISTS Compra (
159     id_compra SERIAL PRIMARY KEY,
160     status SMALLINT NOT NULL,
161     metodo_pagamento VARCHAR(32) NOT NULL,
162     valor_pago NUMERIC(10,2) NOT NULL,
163     data_compra TIMESTAMP NOT NULL,
164     Usuario_id_usuario INT NOT NULL,
165     CONSTRAINT fk_Compra_Usuario1
166         FOREIGN KEY (Usuario_id_usuario)
167         REFERENCES Usuario (id_usuario)
168         ON DELETE NO ACTION
169         ON UPDATE NO ACTION
170 );
171
172 CREATE INDEX idx_Compra_Usuario ON Compra(Usuario_id_usuario);
173
174 -----
175 -- Table steam.Nota_Fiscal
176 -----
177 CREATE TABLE IF NOT EXISTS Nota_Fiscal (
178     id_nota SERIAL PRIMARY KEY,
179     numeroSerie INT NOT NULL,
180     valor_total NUMERIC(10,2) NOT NULL,
181     data_emissao TIMESTAMP NOT NULL,
182     Compra_id_compra INT NOT NULL,
183     CONSTRAINT fk_Nota_Fiscal_Compra1
184         FOREIGN KEY (Compra_id_compra)
185         REFERENCES Compra (id_compra)
186         ON DELETE NO ACTION
187         ON UPDATE NO ACTION
188 );
189
190 CREATE INDEX idx_Nota_Fiscal_Compra ON Nota_Fiscal(Compra_id_compra);
191
192 -----
193 -- Table steam.Conquista
194 -----
195 CREATE TABLE IF NOT EXISTS Conquista (
196     Titulo VARCHAR(100) NOT NULL,
197     descricao VARCHAR(45) NOT NULL,
198     imagem_icone BYTEA NOT NULL,

```

```

199     id_conquista SERIAL,
200     Jogos_Produto_id_produto INT NOT NULL,
201     PRIMARY KEY (id_conquista, Jogos_Produto_id_produto),
202     CONSTRAINT fk_Conquista_Jogos1
203         FOREIGN KEY (Jogos_Produto_id_produto)
204         REFERENCES Jogos (Produto_id_produto)
205         ON DELETE NO ACTION
206         ON UPDATE NO ACTION
207 );
208
209 CREATE INDEX idx_Conquista_Jogos ON Conquista(Jogos_Produto_id_produto);
210
211 -----
212 -- Table steam.RequisitoSistema
213 -----
214 CREATE TABLE IF NOT EXISTS RequisitoSistema (
215     CPU VARCHAR(46) NULL,
216     sistema_operacional VARCHAR(45) NOT NULL,
217     tipo VARCHAR(45) NOT NULL,
218     Jogos_Produto_id_produto INT NOT NULL,
219     GPU VARCHAR(45) NULL,
220     Memoria_ram VARCHAR(45) NULL,
221     armazenamento_disco VARCHAR(45) NULL,
222     PRIMARY KEY (Jogos_Produto_id_produto, tipo),
223     CONSTRAINT fk_RequisitoSistema_Jogos1
224         FOREIGN KEY (Jogos_Produto_id_produto)
225         REFERENCES Jogos (Produto_id_produto)
226         ON DELETE NO ACTION
227         ON UPDATE NO ACTION
228 );
229
230 CREATE INDEX idx_RequisitoSistema_Jogos ON RequisitoSistema(
231     Jogos_Produto_id_produto);
232
233 -----
234 -- Table steam.Software
235 -----
236 CREATE TABLE IF NOT EXISTS Software (
237     versao_software VARCHAR(32) NOT NULL,
238     tipo_licenca VARCHAR(45) NOT NULL,
239     Produto_id_produto INT PRIMARY KEY,
240     CONSTRAINT fk_Software_Produto1
241         FOREIGN KEY (Produto_id_produto)
242         REFERENCES Produto (id_produto)
243         ON DELETE NO ACTION
244         ON UPDATE NO ACTION
245 );

```

```

245
246 CREATE INDEX idx_Software_Produto ON Software(Produto_id_produto);
247
248 -----
249 -- Table steam.DLC
250 -----
251 CREATE TABLE IF NOT EXISTS DLC (
252     id_jogo_base INT NOT NULL,
253     tamanho_download INT NOT NULL,
254     Produto_id_produto INT NOT NULL,
255     PRIMARY KEY (Produto_id_produto, id_jogo_base),
256     CONSTRAINT fk_DLC_Produto1
257         FOREIGN KEY (Produto_id_produto)
258         REFERENCES Produto (id_produto)
259         ON DELETE NO ACTION
260         ON UPDATE NO ACTION,
261     CONSTRAINT fk_DLC_Jogos1
262         FOREIGN KEY (id_jogo_base)
263         REFERENCES Jogos (Produto_id_produto)
264         ON DELETE NO ACTION
265         ON UPDATE NO ACTION
266 );
267
268 CREATE INDEX idx_DLC_Jogos ON DLC(id_jogo_base);
269
270 -----
271 -- Table steam.Jogos_has_Genero
272 -----
273 CREATE TABLE IF NOT EXISTS Jogos_has_Genero (
274     Jogos_Produto_id_produto INT NOT NULL,
275     Genero_id_genero INT NOT NULL,
276     PRIMARY KEY (Genero_id_genero, Jogos_Produto_id_produto),
277     CONSTRAINT fk_Jogos_has_Genero_Jogos1
278         FOREIGN KEY (Jogos_Produto_id_produto)
279         REFERENCES Jogos (Produto_id_produto)
280         ON DELETE NO ACTION
281         ON UPDATE NO ACTION,
282     CONSTRAINT fk_Jogos_has_Genero_Genero1
283         FOREIGN KEY (Genero_id_genero)
284         REFERENCES Genero (id_genero)
285         ON DELETE NO ACTION
286         ON UPDATE NO ACTION
287 );
288
289 CREATE INDEX idx_Jogos_has_Genero_Genero ON Jogos_has_Genero(
    Genero_id_genero);

```

```

290 CREATE INDEX idx_Jogos_has_Genero_Jogos ON Jogos_has_Genero(
      Jogos_Produto_id_produto);
291
292 -----
293 -- Table steam.Item_compra
294 -----
295 CREATE TABLE IF NOT EXISTS Item_compra (
296     Compra_id_compra INT NOT NULL,
297     Produto_id_produto INT NOT NULL,
298     Quantidade INT NULL,
299     valor_unitario_pago NUMERIC(10,2) NULL,
300     PRIMARY KEY (Produto_id_produto, Compra_id_compra),
301     CONSTRAINT fk_Item_compra_Compra1
302         FOREIGN KEY (Compra_id_compra)
303         REFERENCES Compra (id_compra)
304         ON DELETE NO ACTION
305         ON UPDATE NO ACTION,
306     CONSTRAINT fk_Item_compra_Produto1
307         FOREIGN KEY (Produto_id_produto)
308         REFERENCES Produto (id_produto)
309         ON DELETE NO ACTION
310         ON UPDATE NO ACTION
311 );
312
313 CREATE INDEX idx_Item_compra_Compra ON Item_compra(Compra_id_compra);
314
315 -----
316 -- Criar usu rio professor
317 -----
318 DROP USER IF EXISTS professor;
319 CREATE USER professor WITH PASSWORD 'professor';
320 GRANT ALL PRIVILEGES ON SCHEMA steam TO professor;
321 GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA steam TO professor;
322 GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA steam TO professor;

```

## 5 Acesso ao Ambiente (AWS RDS)

O banco de dados encontra-se ativo e disponível para avaliação. O *Security Group* foi configurado para permitir conexões externas na porta padrão.

- **Endpoint:** steamdb.cximu3gskw3g.us-east-1.rds.amazonaws.com
- **Porta:** 5432
- **Database:** steam

- **Usuário:** professor
- **Senha:** professor