

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

Relatório de Banco de Dados: Parte 2.1

Projeto Lógico e Implementação PostgreSQL (Steam)

Alunos: Rafael Gomes Oliveira Santos
Alícia Vitória Sousa Santos

Professor: André Britto

São Cristóvão
2026

1 Especificação e Descrição do Problema

O sistema desenvolvido visa gerir uma plataforma de distribuição digital baseada no ecossistema Steam. As principais regras de negócio implementadas são:

- **Gestão de Identidade e Finanças:** Cada usuário possui um identificador único, e-mail exclusivo e uma carteira vinculada (relacionamento 1:1) com precisão decimal para saldos.
- **Especialização de Produtos:** Utiliza-se uma hierarquia onde **Produto** é a generalização, especializada em **Jogos**, **Software** e **DLC** através de tabelas distintas com integridade referencial.
- **Normalização de Atributos:** Atributos multivvalorados (Idiomas) e compostos (Requisitos) foram normalizados para atender à 1FN.
- **Entidades Fracas:** As conquistas são dependentes existencialmente dos jogos, possuindo identificação composta.

2 Projeto Conceitual e Mapeamento Relacional

O projeto conceitual seguiu a notação de Heuser. O mapeamento para o modelo relacional priorizou a manutenção das restrições de integridade através de chaves estrangeiras com ações de **CASCADE** e **RESTRICT** em pontos estratégicos.

3 Implementação no PostgreSQL (Amazon RDS)

A implementação técnica no serviço gerenciado da AWS focou nas propriedades **ACID**:

- **Domínios e Tipos:** Uso de **SERIAL** para autoincremento, **INTERVAL** para durações (tempo jogado) e **BYTEA** para dados binários.
- **Isolamento via Schemas:** Criação do schema **steam** para organizar o namespace do projeto.
- **Restrições Semânticas:** Implementação de cláusulas **CHECK** para validar intervalos de votos e tipos de requisitos.

4 Script SQL

```
1 --  
2 -- PROJETO LOGICO: ECOSSISTEMA STEAM  
3 -- Disciplina: Banco de Dados I  
4 --  
5  
6 CREATE SCHEMA IF NOT EXISTS steam;  
7 SET search_path TO steam;  
8  
9 -- Limpeza para reinicializacao do ambiente de testes  
10 DROP TABLE IF EXISTS Jogos_has_Genero , DLC , Software , RequisitoSistema ,  
    Conquista ,  
    Nota_Fiscal , Compra , Biblioteca , Avaliacao , Publicadora , Desenvolvedora ,  
    Jogos , Produto , Genero , Usuario , Carteira CASCADE ;  
11  
12  
13  
14 --  
15 -- NUCLEO: USUARIOS E FINANCEIRO  
16 --  
17  
18 CREATE TABLE IF NOT EXISTS Carteira (  
19     id_carteira SERIAL PRIMARY KEY ,  
20     Moeda VARCHAR(16) NOT NULL ,  
21     saldo_atual NUMERIC(10, 2) NOT NULL DEFAULT 0.00  
22 );  
23  
24 CREATE TABLE IF NOT EXISTS Usuario (  
25     id_usuario SERIAL PRIMARY KEY ,  
26     Nome_Completo VARCHAR(100) NOT NULL ,  
27     email VARCHAR(255) NOT NULL UNIQUE ,  
28     senha VARCHAR(32) NOT NULL ,  
29     nivel INT NOT NULL DEFAULT 1 ,  
30     id_carteira INT NOT NULL ,  
31     CONSTRAINT fk_Usuario_Carteira FOREIGN KEY (id_carteira)  
32         REFERENCES Carteira (id_carteira) ON DELETE CASCADE ON UPDATE  
            CASCADE  
33 );  
34  
35 --
```

```

36 -- CATALOGO E CATEGORIZACAO
37 --
38
39 CREATE TABLE IF NOT EXISTS Genero (
40     id_genero SERIAL PRIMARY KEY,
41     nome VARCHAR(255) NOT NULL UNIQUE
42 );
43
44 CREATE TABLE IF NOT EXISTS Produto (
45     id_produto SERIAL PRIMARY KEY,
46     Titulo VARCHAR(100) NOT NULL,
47     preco NUMERIC(10, 2) NOT NULL DEFAULT 0.00,
48     Descricao TEXT NOT NULL,
49     Data_lancamento DATE NOT NULL
50 );
51
52 --
53 -- ESPECIALIZACOES (PRODUTO)
54 --
55
56 CREATE TABLE IF NOT EXISTS Jogos (
57     id_produto INT PRIMARY KEY,
58     Suporte_Controle BOOLEAN NOT NULL DEFAULT FALSE,
59     Faixa_etaria VARCHAR(45) NOT NULL,
60     idiomas VARCHAR(45)[] NOT NULL,
61     CONSTRAINT fk_Jogos_Produto FOREIGN KEY (id_produto)
62         REFERENCES Produto (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
63 );
64
65 CREATE TABLE IF NOT EXISTS Software (
66     id_produto INT PRIMARY KEY,
67     versao_software VARCHAR(32) NOT NULL,
68     tipo_licenca VARCHAR(45) NOT NULL,
69     CONSTRAINT fk_Software_Produto FOREIGN KEY (id_produto)
70         REFERENCES Produto (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
71 );
72
73 CREATE TABLE IF NOT EXISTS DLC (
74     id_produto INT PRIMARY KEY,

```

```

75    requer_jogo_base BOOLEAN NOT NULL DEFAULT TRUE,
76    tamanho_download VARCHAR(45),
77    CONSTRAINT fk_DLC_Produto FOREIGN KEY (id_produto)
78        REFERENCES Produto (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
79 );
80
81 --
82 -- AGENTES DO MERCADO (CHAVES NATURAIS)
83 --
84
85 CREATE TABLE IF NOT EXISTS Desenvolvedora (
86     Nome_empresa VARCHAR(255) PRIMARY KEY,
87     pais_sede VARCHAR(45) NOT NULL,
88     website VARCHAR(100), -- Null-able
89     id_jogo INT NOT NULL,
90     CONSTRAINT fk_Desenvolvedora_Jogos FOREIGN KEY (id_jogo)
91         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
92 );
93
94 CREATE TABLE IF NOT EXISTS Publicadora (
95     Nome_Publicadora VARCHAR(255) PRIMARY KEY,
96     contato VARCHAR(100) NOT NULL,
97     id_produto INT NOT NULL,
98     CONSTRAINT fk_Publicadora_Produto FOREIGN KEY (id_produto)
99         REFERENCES Produto (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
100 );
101
102 --
103 -- INTERACAO E AUDITORIA
104 --
105
106 -- PK composta pelas FKs (Identifica a avaliacao unica usuario-jogo)
107 CREATE TABLE IF NOT EXISTS Avaliacao (
108     id_usuario INT NOT NULL,
109     id_jogo INT NOT NULL,
110     voto SMALLINT CHECK (voto >= 0 AND voto <= 5),
111     Texto_analise TEXT NULL,
112     data_postagem TIMESTAMP NULL,
113     PRIMARY KEY (id_usuario, id_jogo),

```

```

114     CONSTRAINT fk_Avaliacao_Usuario FOREIGN KEY (id_usuario)
115         REFERENCES Usuario (id_usuario) ON DELETE RESTRICT ON UPDATE CASCADE
116     ,
117     CONSTRAINT fk_Avaliacao_Jogos FOREIGN KEY (id_jogo)
118         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
119     );
120
121 -- tempo_jogado como TIMESTAMP e Null-able
122 CREATE TABLE IF NOT EXISTS Biblioteca (
123     id_usuario INT PRIMARY KEY,
124     tempo_jogado TIMESTAMP NULL,
125     status_instalacao BOOLEAN DEFAULT FALSE,
126     CONSTRAINT fk_Biblioteca_Usuario FOREIGN KEY (id_usuario)
127         REFERENCES Usuario (id_usuario) ON DELETE CASCADE ON UPDATE CASCADE
128 );
129
130 CREATE TABLE IF NOT EXISTS Compra (
131     id_compra SERIAL PRIMARY KEY,
132     status VARCHAR(20) NOT NULL,
133     metodo_pagamento VARCHAR(32) NOT NULL,
134     valor_pago NUMERIC(10, 2) NOT NULL,
135     data_compra TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
136     id_produto INT NOT NULL,
137     id_usuario INT NOT NULL,
138     CONSTRAINT fk_Compra_Produto FOREIGN KEY (id_produto)
139         REFERENCES Produto (id_produto) ON DELETE RESTRICT ON UPDATE CASCADE
140     ,
141     CONSTRAINT fk_Compra_Usuario FOREIGN KEY (id_usuario)
142         REFERENCES Usuario (id_usuario) ON DELETE RESTRICT ON UPDATE CASCADE
143 );
144
145 CREATE TABLE IF NOT EXISTS Nota_Fiscal (
146     id_nota SERIAL PRIMARY KEY,
147     numero_serie INT NOT NULL,
148     valor_total NUMERIC(10, 2) NOT NULL,
149     id_compra INT NOT NULL UNIQUE,
150     CONSTRAINT fk_Nota_Fiscal_Compra FOREIGN KEY (id_compra)
151         REFERENCES Compra (id_compra) ON DELETE CASCADE ON UPDATE CASCADE
152 );
153
154

```

-- DETALHES TECNICOS E PROGRESSAO

--

```

155
156 CREATE TABLE IF NOT EXISTS Jogos_has_Genero (
157     id_jogo INT NOT NULL ,
158     id_genero INT NOT NULL ,
159     PRIMARY KEY (id_jogo, id_genero) ,
160     CONSTRAINT fk_Jogos_Genero_Jogo FOREIGN KEY (id_jogo)
161         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE ,
162     CONSTRAINT fk_Jogos_Genero_Gen FOREIGN KEY (id_genero)
163         REFERENCES Genero (id_genero) ON DELETE RESTRICT ON UPDATE CASCADE
164 );
165
166 -- Campos Titulo, descricao e imagem_icone definidos como NOT NULL (NN)
167 CREATE TABLE IF NOT EXISTS Conquista (
168     id_conquista INT NOT NULL ,
169     id_jogo INT NOT NULL ,
170     Titulo VARCHAR(100) NOT NULL ,
171     descricao TEXT NOT NULL ,
172     imagem_icone BYTEA NOT NULL ,
173     PRIMARY KEY (id_conquista, id_jogo) ,
174     CONSTRAINT fk_Conquista_Jogos FOREIGN KEY (id_jogo)
175         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
176 );
177
178 CREATE TABLE IF NOT EXISTS RequisitoSistema (
179     id_requisito SERIAL PRIMARY KEY ,
180     CPU VARCHAR(100) ,
181     GPU VARCHAR(100) ,
182     Memoria_ram VARCHAR(45) ,
183     armazenamento_disco VARCHAR(45) ,
184     sistema_operacional VARCHAR(45) NOT NULL ,
185     tipo VARCHAR(45) NOT NULL CHECK (tipo IN ('Minimo', 'Recomendado')) ,
186     id_jogo INT NOT NULL ,
187     CONSTRAINT fk_Requisito_Jogos FOREIGN KEY (id_jogo)
188         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
189 );
190
191 --
-----+
-----+
192 -- PERFORMANCE E ACESSO
193 --
-----+
-----+
194
195 CREATE INDEX IF NOT EXISTS idx_usuario_email ON Usuario(email);
196 CREATE INDEX IF NOT EXISTS idx_compra_data ON Compra(data_compra);
197

```

```

198 DO $$  

199 BEGIN  

200   IF NOT EXISTS (SELECT FROM pg_catalog.pg_roles WHERE rolname = '  

      professor') THEN  

201     CREATE USER professor WITH PASSWORD 'professor';  

202   END IF;  

203 END $$;  

204  

205 GRANT USAGE ON SCHEMA steam TO professor;  

206 GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA steam TO professor;  

207 GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA steam TO professor;  

208 ALTER USER professor SET search_path TO steam, public;

```

5 Instância Amazon RDS

O banco de dados encontra-se ativo e disponível para testes docentes. O Security Group foi configurado para permitir tráfego de entrada na porta 5432.

- **Endpoint:** steamdb.cximu3gskw3g.us-east-1.rds.amazonaws.com
- **Credenciais:** Usuário professor | Senha professor.