

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  
Alicia 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 multivalorados (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 foi estruturado seguindo a metodologia de projeto de banco de dados, partindo de uma abstração de alto nível até a implementação física.

### 2.1 Modelo Conceitual (DER)

O Diagrama Entidade-Relacionamento (DER) utiliza a notação de Heuser para descrever a semântica do ecossistema. Destaca-se a especialização total do **Produto**, onde a cardinalidade  $(0, N)$  em **Conquistas** e **Gêneros** reflete a opcionalidade desses dados para diferentes títulos. O atributo *Idiomas* é tratado como multivalorado, prevendo sua futura implementação otimizada.



### 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, `TIMESTAMP` para registros temporais e `NUMERIC` para precisão financeira.
- **Extensões Objeto-Relacionais:** Implementação do atributo *Idiomas* como `VARCHAR[]`, permitindo o armazenamento de múltiplos valores em uma única coluna sem violar o desempenho em consultas de leitura.
- **Ações Referenciais:** Uso estratégico de `ON DELETE RESTRICT` em tabelas de *Compra* e *Avaliação* para preservar a integridade histórica e financeira.

## 4 Script SQL

```
1  --
2  --=====
3  -- PROJETO LOGICO: ECOSSISTEMA STEAM
4  -- Disciplina: Banco de Dados I
5  --=====
6
7  CREATE SCHEMA IF NOT EXISTS steam;
8  SET search_path TO steam;
9
10 -- Limpeza para reinicializacao do ambiente de testes
11 DROP TABLE IF EXISTS Jogos_has_Genero, DLC, Software, RequisitoSistema,
12     Conquista,
13     Nota_Fiscal, Compra, Biblioteca, Biblioteca_has_Produto, Avaliacao,
14     Publicadora, Desenvolvedora,
15     Jogos, Produto, Genero, Usuario, Carteira CASCADE;
16
17
18 --=====
19
20 -- NUCLEO: USUARIOS E FINANCEIRO
21 --=====
22
23
24 CREATE TABLE IF NOT EXISTS Carteira (
25     id_carteira SERIAL PRIMARY KEY,
26     Moeda VARCHAR(16) NOT NULL,
27     saldo_atual NUMERIC(10, 2) NOT NULL DEFAULT 0.00
28 );
29
30 CREATE TABLE IF NOT EXISTS Usuario (
31     id_usuario SERIAL PRIMARY KEY,
32     Nome_Completo VARCHAR(100) NOT NULL,
33     email VARCHAR(255) NOT NULL UNIQUE,
34     senha VARCHAR(32) NOT NULL,
35     nivel INT NOT NULL DEFAULT 1,
36     id_carteira INT NOT NULL,
37     CONSTRAINT fk_Usuario_Carteira FOREIGN KEY (id_carteira)
38         REFERENCES Carteira (id_carteira) ON DELETE CASCADE ON UPDATE
39         CASCADE
40 );
```

```

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_Controlo 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     -----
83
84 -- AGENTES DO MERCADO (CHAVES NATURAIS)
85     -----
86
87
88
89
90
91
92
93
94 CREATE TABLE IF NOT EXISTS Desenvolvedora (
95     Nome_empresa VARCHAR(255) PRIMARY KEY,
96     pais_sede VARCHAR(45) NOT NULL,
97     website VARCHAR(100),
98     id_jogo INT NOT NULL,
99     CONSTRAINT fk_Developedora_Jogos FOREIGN KEY (id_jogo)
100         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
101 );
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

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

```



```

157 REFERENCES Compra (id_compra) ON DELETE CASCADE ON UPDATE CASCADE
158 );
159
160 --
161 -- DETALHES TECNICOS E PROGRESSAO
162 --
163
164 CREATE TABLE IF NOT EXISTS Jogos_has_Genero (
165     id_jogo INT NOT NULL,
166     id_genero INT NOT NULL,
167     PRIMARY KEY (id_jogo, id_genero),
168     CONSTRAINT fk_Jogos_Genero_Jogo FOREIGN KEY (id_jogo)
169         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE,
170     CONSTRAINT fk_Jogos_Genero_Gen FOREIGN KEY (id_genero)
171         REFERENCES Genero (id_genero) ON DELETE RESTRICT ON UPDATE CASCADE
172 );
173
174 CREATE TABLE IF NOT EXISTS Conquista (
175     id_conquista INT NOT NULL,
176     id_jogo INT NOT NULL,
177     Titulo VARCHAR(100) NOT NULL,
178     descricao TEXT NOT NULL,
179     imagem_icone BYTEA NOT NULL,
180     PRIMARY KEY (id_conquista, id_jogo),
181     CONSTRAINT fk_Conquista_Jogos FOREIGN KEY (id_jogo)
182         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
183 );
184
185 CREATE TABLE IF NOT EXISTS RequisitoSistema (
186     id_requisito SERIAL PRIMARY KEY,
187     CPU VARCHAR(100),
188     GPU VARCHAR(100),
189     Memoria_ram VARCHAR(45),
190     armazenamento_disco VARCHAR(45),
191     sistema_operacional VARCHAR(45) NOT NULL,
192     tipo VARCHAR(45) NOT NULL CHECK (tipo IN ('Minimo', 'Recomendado')),
193     id_jogo INT NOT NULL,
194     CONSTRAINT fk_Requisito_Jogos FOREIGN KEY (id_jogo)
195         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
196 );
197
198 --

```

```

199 -- PERFORMANCE E ACESSO
200 --
201
202 CREATE INDEX IF NOT EXISTS idx_usuario_email ON Usuario(email);
203 CREATE INDEX IF NOT EXISTS idx_compra_data ON Compra(data_compra);
204
205 DO $$
206 BEGIN
207     IF NOT EXISTS (SELECT FROM pg_catalog.pg_roles WHERE rolname = '
208         professor') THEN
209         CREATE USER professor WITH PASSWORD 'professor';
210     END IF;
211 END $$;
212
213 GRANT USAGE ON SCHEMA steam TO professor;
214 GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA steam TO professor;
215 GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA steam TO professor;
216 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.