



Pedro Lemos Mariano - 6968

Thiago Matheus de Oliveira Costa - 8101

Pedro Lucas de Oliveira Costa - 8127

*Projeto de lab. Banco de dados*

**ALGARFLIX**

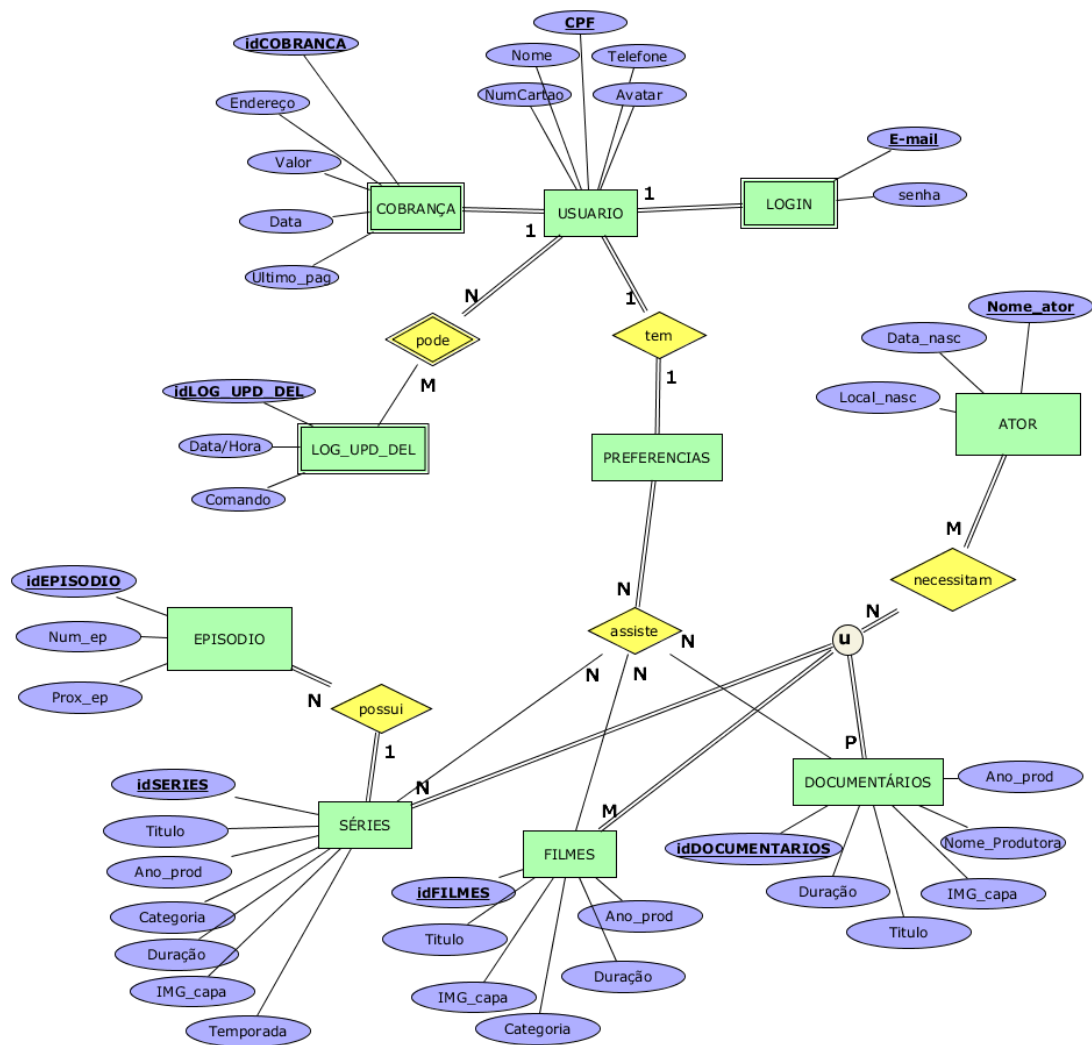
Rio Paranaíba – MG

**2024**

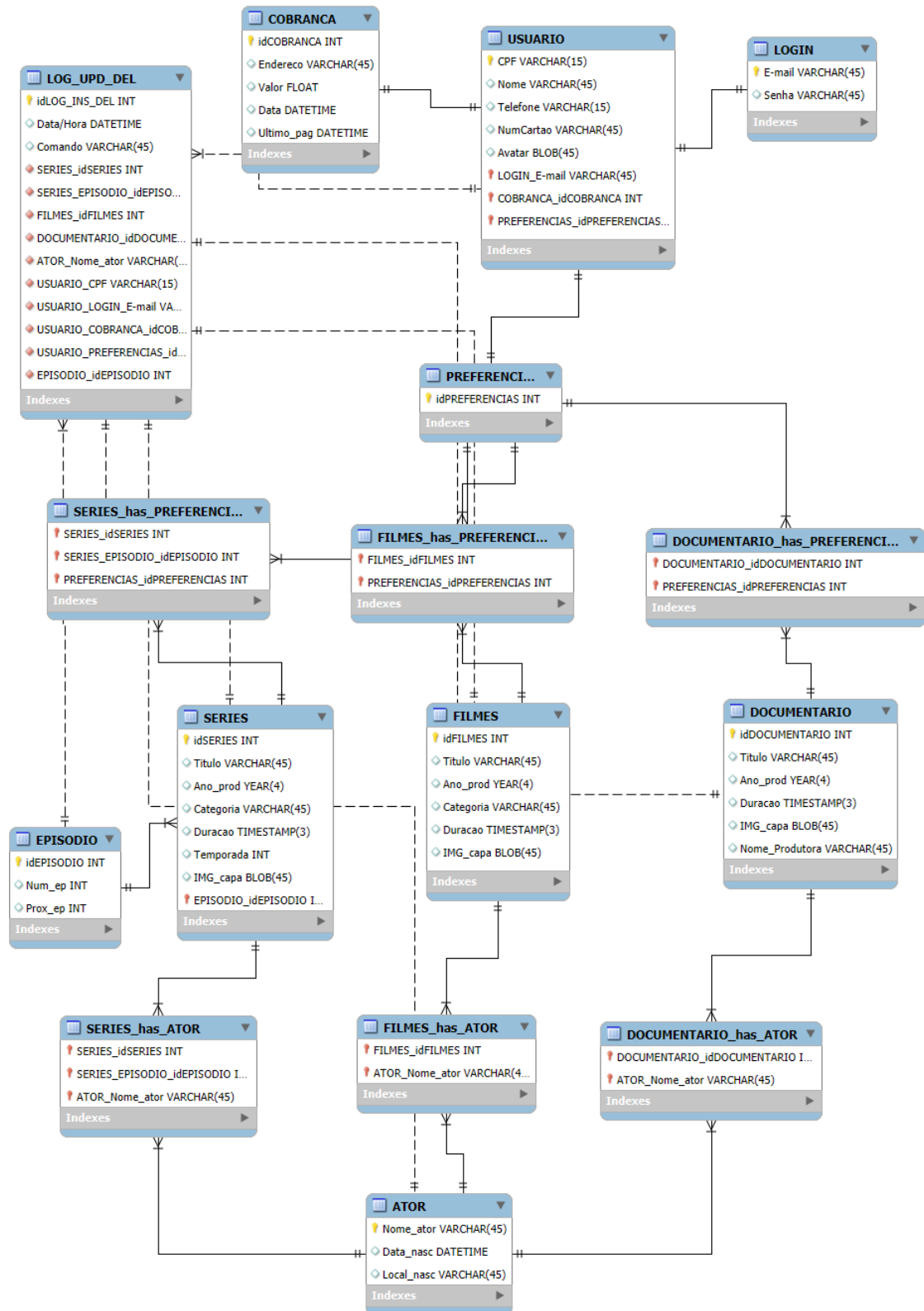
# Sumário

1. Modelo Entidade Relacionamento	3
2. Modelo Relacional	4
3. Código SQL de criação do banco de dados	5
4. Código SQL de inserção de registros no banco de dados	9
5. Código SQL de criação das <i>rules</i> do banco de dados	12
6. Código SQL de criação das <i>stored procedures</i> do banco de dados	15

## 1. MODELO ENTIDADE RELACIONAMENTO:



## 2. MODELO RELACIONAL:



### 3. CÓDIGO SQL DE CRIAÇÃO DO BANCO DE DADOS:

```
1. CREATE TABLE LOGIN (  
2.     email VARCHAR(45) NOT NULL,  
3.     senha VARCHAR(45),  
4.     PRIMARY KEY (email)  
5. );  
6.  
7. CREATE TABLE COBRANCA (  
8.     idcobranca SERIAL PRIMARY KEY,  
9.     endereco VARCHAR(45),  
10.    valor REAL,  
11.    data TIMESTAMP,  
12.    ultimo_pag TIMESTAMP  
13. );  
14.  
15. CREATE TABLE PREFERENCIAS (  
16.     idpreferencias SERIAL PRIMARY KEY  
17. );  
18.  
19. CREATE TABLE EPISODIO (  
20.     idepisodio SERIAL PRIMARY KEY,  
21.     num_ep INTEGER,  
22.     prox_ep INTEGER  
23. );  
24.  
25. CREATE TABLE SERIES (  
26.     idseries SERIAL PRIMARY KEY,  
27.     titulo VARCHAR(45),  
28.     ano_prod INTEGER,  
29.     categoria VARCHAR(45),  
30.     duracao VARCHAR(10),  
31.     temporada INTEGER,  
32.     img_capa BLOB,  
33.     episodio_idepisodio INTEGER NOT NULL,  
34.     FOREIGN KEY (episodio_idepisodio) REFERENCES EPISODIO  
    (idepisodio)  
35. );  
36.  
37. CREATE TABLE FILMES (  
38.     idfilmes SERIAL PRIMARY KEY,  
39.     titulo VARCHAR(45),  
40.     ano_prod INTEGER,  
41.     categoria VARCHAR(45),  
42.     duracao VARCHAR(10),  
43.     img_capa BLOB  
44. );  
45.
```

```
46. CREATE TABLE DOCUMENTARIO (  
47.     iddocumentario SERIAL PRIMARY KEY,  
48.     titulo VARCHAR(45),  
49.     ano_prod INTEGER,  
50.     duracao VARCHAR(10),  
51.     img_capa BLOB,  
52.     nome_produtores VARCHAR(45)  
53. );  
54.  
55. CREATE TABLE ATOR (  
56.     nome_ator VARCHAR(45) PRIMARY KEY,  
57.     data_nasc DATE,  
58.     local_nasc VARCHAR(45)  
59. );  
60.  
61. CREATE TABLE LOG_UPD_DEL (  
62.     idlog_ins_del SERIAL PRIMARY KEY,  
63.     data_hora TIMESTAMP,  
64.     comando VARCHAR(45),  
65.     series_idseries INTEGER NOT NULL,  
66.     series_episodio_idepisodio INTEGER NOT NULL,  
67.     filmes_idfilmes INTEGER NOT NULL,  
68.     documentario_iddocumentario INTEGER NOT NULL,  
69.     ator_nome_ator VARCHAR(45) NOT NULL,  
70.     USUARIO_CPF VARCHAR(15) NOT NULL,  
71.     USUARIO_LOGIN_EMAIL VARCHAR(45) NOT NULL,  
72.     USUARIO_COBRANCA_IDCOBRANCA INTEGER NOT NULL,  
73.     USUARIO_PREFERENCIAS_IDPREFERENCIAS INTEGER NOT NULL,  
74.     EPISODIO_IDEPISODIO INTEGER NOT NULL,  
75.     FOREIGN KEY (series_idseries) REFERENCES SERIES (idseries),  
76.     FOREIGN KEY (series_episodio_idepisodio) REFERENCES  
        EPISODIO (idepisodio),  
77.     FOREIGN KEY (filmes_idfilmes) REFERENCES FILMES (idfilmes),  
78.     FOREIGN KEY (documentario_iddocumentario) REFERENCES  
        DOCUMENTARIO (iddocumentario),  
79.     FOREIGN KEY (ator_nome_ator) REFERENCES ATOR (nome_ator),  
80.     FOREIGN KEY (USUARIO_CPF, USUARIO_LOGIN_EMAIL,  
        USUARIO_COBRANCA_IDCOBRANCA,  
        USUARIO_PREFERENCIAS_IDPREFERENCIAS) REFERENCES USUARIO (cpf,  
        login_email, cobranca_idcobranca,  
        preferencias_idpreferencias),  
81.     FOREIGN KEY (EPISODIO_IDEPISODIO) REFERENCES EPISODIO  
        (idepisodio)  
82. );  
83.  
84. CREATE TABLE USUARIO (  
85.     cpf VARCHAR(15) NOT NULL,  
86.     nome VARCHAR(45),  
87.     telefone VARCHAR(15),
```

```
88. numcartao VARCHAR(45),
89. avatar BLOB,
90. login_email VARCHAR(45) NOT NULL,
91. cobranca_idcobranca INTEGER NOT NULL,
92. preferencias_idpreferencias INTEGER NOT NULL,
93. PRIMARY KEY (cpf, login_email, cobranca_idcobranca,
    preferencias_idpreferencias),
94. FOREIGN KEY (login_email) REFERENCES LOGIN (email),
95. FOREIGN KEY (cobranca_idcobranca) REFERENCES COBRANCA
    (idcobranca),
96. FOREIGN KEY (preferencias_idpreferencias) REFERENCES
    PREFERENCIAS (idpreferencias),
97. );
98.
99. CREATE TABLE FILMES_has_ATOM (
100. filmes_idfilmes INTEGER NOT NULL,
101. ator_nome_ator VARCHAR(45) NOT NULL,
102. PRIMARY KEY (filmes_idfilmes, ator_nome_ator),
103. FOREIGN KEY (filmes_idfilmes) REFERENCES FILMES (idfilmes),
104. FOREIGN KEY (ator_nome_ator) REFERENCES ATOR (nome_ator)
105. );
106.
107. CREATE TABLE DOCUMENTARIO_has_ATOM (
108. documentario_iddocumentario INTEGER NOT NULL,
109. ator_nome_ator VARCHAR(45) NOT NULL,
110. PRIMARY KEY (documentario_iddocumentario, ator_nome_ator),
111. FOREIGN KEY (documentario_iddocumentario) REFERENCES
    DOCUMENTARIO (iddocumentario),
112. FOREIGN KEY (ator_nome_ator) REFERENCES ATOR (nome_ator)
113. );
114.
115. CREATE TABLE SERIES_has_ATOM (
116. series_idseries INTEGER NOT NULL,
117. series_episodio_idepisodio INTEGER NOT NULL,
118. ator_nome_ator VARCHAR(45) NOT NULL,
119. PRIMARY KEY (series_idseries, series_episodio_idepisodio,
    ator_nome_ator),
120. FOREIGN KEY (series_idseries) REFERENCES SERIES (idseries),
121. FOREIGN KEY (series_episodio_idepisodio) REFERENCES
    EPISODIO (idepisodio),
122. FOREIGN KEY (ator_nome_ator) REFERENCES ATOR (nome_ator)
123. );
124.
125. CREATE TABLE SERIES_has_PREFERENCIAS (
126. series_idseries INTEGER NOT NULL,
127. series_episodio_idepisodio INTEGER NOT NULL,
128. preferencias_idpreferencias INTEGER NOT NULL,
129. PRIMARY KEY (series_idseries, series_episodio_idepisodio,
    preferencias_idpreferencias),
```

```
130. FOREIGN KEY (series_idseries) REFERENCES SERIES (idseries),
131. FOREIGN KEY (series_episodio_idepisodio) REFERENCES
    EPISODIO (idepisodio),
132. FOREIGN KEY (preferencias_idpreferencias) REFERENCES
    PREFERENCIAS (idpreferencias)
133.);
134.
135. CREATE TABLE FILMES_has_PREFERENCIAS (
136.     filmes_idfilmes INTEGER NOT NULL,
137.     preferencias_idpreferencias INTEGER NOT NULL,
138.     PRIMARY KEY (filmes_idfilmes, preferencias_idpreferencias),
139.     FOREIGN KEY (filmes_idfilmes) REFERENCES FILMES (idfilmes),
140.     FOREIGN KEY (preferencias_idpreferencias) REFERENCES
        PREFERENCIAS (idpreferencias)
141.);
142.
143. CREATE TABLE DOCUMENTARIO_has_PREFERENCIAS (
144.     documentario_iddocumentario INTEGER NOT NULL,
145.     preferencias_idpreferencias INTEGER NOT NULL,
146.     PRIMARY KEY (documentario_iddocumentario,
        preferencias_idpreferencias),
147.     FOREIGN KEY (documentario_iddocumentario) REFERENCES
        DOCUMENTARIO (iddocumentario),
148.     FOREIGN KEY (preferencias_idpreferencias) REFERENCES
        PREFERENCIAS (idpreferencias)
149.);
```



#### 4. CÓDIGO SQL DE INSERÇÃO DE REGISTROS NO BANCO DE DADOS:

```
1. INSERT INTO LOGIN (email, senha) VALUES
2. ('alice.smith@example.com', 'alice2024'),
3. ('bob.johnson@example.com', 'bob2024'),
4. ('carol.davis@example.com', 'carol2024');
5.
6. INSERT INTO COBRANCA (endereco, valor, data, ultimo_pag)
   VALUES
7. ('Av. Paulista, 100', 250.00, '2024-04-10 12:00:00', '2024-
   04-20 12:00:00'),
8. ('Rua dos Três Irmãos, 456', 300.00, '2024-05-15 14:00:00',
   '2024-05-25 14:00:00'),
9. ('Praça da Sé, 789', 150.00, '2024-06-20 16:00:00', '2024-06-
   30 16:00:00');
10.
11.     INSERT INTO PREFERENCIAS DEFAULT VALUES;
12.     INSERT INTO PREFERENCIAS DEFAULT VALUES;
13.     INSERT INTO PREFERENCIAS DEFAULT VALUES;
14.
15.     INSERT INTO EPISODIO (num_ep, prox_ep) VALUES
16.     (10, 2),
17.     (20, 3),
18.     (3, NULL);
19.
20.     INSERT INTO SERIES (titulo, ano_prod, categoria,
   duracao, temporada, img_capa, episodio_id, episodio) VALUES
21.     ('Breaking Bad', 2008, 'Crime', '50:00:00', 5, NULL,
   1),
22.     ('Stranger Things', 2016, 'Sci-Fi', '55:00:00', 4,
   NULL, 2),
23.     ('The Crown', 2016, 'Drama', '60:00:00', 6, NULL, 3);
24.
25.     INSERT INTO FILMES (titulo, ano_prod, categoria,
   duracao, img_capa) VALUES
26.     ('Inception', 2010, 'Sci-Fi', '02:28:00', NULL),
27.     ('The Godfather', 1972, 'Crime', '02:55:00', NULL),
28.     ('Parasite', 2019, 'Thriller', '02:12:00', NULL);
29.
30.     INSERT INTO DOCUMENTARIO (titulo, ano_prod, duracao,
   img_capa, nome_produtores) VALUES
31.     ('Planet Earth', 2006, '08:00:00', NULL, 'BBC'),
32.     ('The Social Dilemma', 2020, '01:34:00', NULL,
   'Netflix'),
```

```

33.      ('13th', 2016, '01:40:00', NULL, 'Netflix');
34.
35.      INSERT INTO ATOR (nome_ator, data_nasc, local_nasc)
      VALUES
36.      ('Bryan Cranston', '1956-03-07', 'Los Angeles, CA'),
37.      ('Millie Bobby Brown', '2004-02-19', 'Madrid, Spain'),
38.      ('Olivia Colman', '1974-01-30', 'Norwich, England');
39.
40.      INSERT INTO USUARIO (cpf, nome, telefone, numcartao,
      avatar, login_email, cobranca_idcobranca,
      preferencias_idpreferencias) VALUES
41.      ('11122233344', 'Alice Smith', '555-1234', '1234 5678
      9876 5432', NULL, 'alice.smith@example.com', 1, 1),
42.      ('22233344455', 'Bob Johnson', '555-5678', '2345 6789
      8765 4321', NULL, 'bob.johnson@example.com', 2, 2),
43.      ('33344455566', 'Carol Davis', '555-8765', '3456 7890
      7654 3210', NULL, 'carol.davis@example.com', 3, 3);
44.
45.      INSERT INTO FILMES_has_ATOR (filmes_idfilmes,
      ator_nome_ator) VALUES
46.      (1, 'Leonardo DiCaprio'),
47.      (2, 'Marlon Brando'),
48.      (3, 'Song Kang-ho');
49.
50.      INSERT INTO DOCUMENTARIO_has_ATOR
      (documentario_iddocumentario, ator_nome_ator) VALUES
51.      (1, 'David Attenborough'),
52.      (2, 'Tristan Harris'),
53.      (3, 'Ava DuVernay');
54.
55.      INSERT INTO SERIES_has_ATOR (series_idseries,
      series_episodio_idepisodio, ator_nome_ator) VALUES
56.      (1, 1, 'Bryan Cranston'),
57.      (2, 1, 'Millie Bobby Brown'),
58.      (3, 1, 'Olivia Colman');
59.
60.      INSERT INTO SERIES_has_PREFERENCIAS (series_idseries,
      series_episodio_idepisodio, preferencias_idpreferencias)
      VALUES
61.      (1, 1, 1),
62.      (2, 1, 2),
63.      (3, 1, 3);
64.
65.
66.

```

```
67.      INSERT INTO FILMES_has_PREFERENCIAS (filmes_idfilmes,
        preferencias_idpreferencias) VALUES
68.      (1, 1),
69.      (2, 2),
70.      (3, 3);
71.
72.      INSERT INTO DOCUMENTARIO_has_PREFERENCIAS
        (documentario_iddocumentario, preferencias_idpreferencias)
        VALUES
73.      (1, 1),
74.      (2, 2),
75.      (3, 3);
76.
77.      INSERT INTO USUARIO(cpf, avatar)
78.      VALUES(11122233344, lo_import('E:foto_alice.JPG'));
79.
80.      INSERT INTO USUARIO(cpf, avatar)
81.      VALUES(22233344455, lo_import('E:foto_bob.JPG'));
82.
83.      INSERT INTO USUARIO(cpf, avatar)
84.      VALUES(33344455566, lo_import('E:foto_carol.JPG'));
85.
```

## 5. CÓDIGO SQL DE CRIAÇÃO DAS RULES DO BANCO DE DADOS:

```
1. -----1
2. CREATE RULE delete_avatar_rule AS
3. ON DELETE TO USUARIO
4. DO ALSO
5. UPDATE USUARIO SET avatar = NULL WHERE cpf = OLD.cpf;
6.
7. ----- 2
8. CREATE RULE delete_filme_img_capa_rule AS
9. ON DELETE TO FILMES
10. DO ALSO
11. UPDATE FILMES SET img_capa = NULL WHERE idfilmes =
    OLD.idfilmes;
12.
13. CREATE RULE delete_documentario_img_capa_rule AS
14. ON DELETE TO DOCUMENTARIO
15. DO ALSO
16. UPDATE DOCUMENTARIO SET img_capa = NULL WHERE
    iddocumentario = OLD.iddocumentario;
17.
18. CREATE RULE delete_series_img_capa_rule AS
19. ON DELETE TO SERIES
20. DO ALSO
21. UPDATE SERIES SET img_capa = NULL WHERE idseries =
    OLD.idseries;
22.
23. -----3
24.
25. ALTER TABLE FILMES
26. ADD COLUMN avaliacao REAL;
27.
28. ALTER TABLE SERIES
29. ADD COLUMN avaliacao REAL;
30.
31. ALTER TABLE SERIES
32. ADD COLUMN avaliacao REAL;
33.
34. ALTER TABLE DOCUMENTARIO
35. ADD COLUMN avaliacao REAL;
36.
37. -----LOG
38.
39. CREATE OR REPLACE RULE log_update_series AS
```

```
40.         ON UPDATE TO SERIES
41.         DO ALSO INSERT INTO LOG_UPD_DEL (data_hora, comando,
        series_idseries, series_episodio_idepisodio, USUARIO_CPF,
        USUARIO_LOGIN_EMAIL, USUARIO_COBRANCA_IDCOBRANCA,
        USUARIO_PREFERENCIAS_IDPREFERENCIAS, EPISODIO_IDEPISODIO)
42.         VALUES (CURRENT_TIMESTAMP, 'UPDATE', OLD.idseries,
        OLD.episodio_idepisodio, NULL, NULL, NULL, NULL, NULL);
43.
44.         CREATE OR REPLACE RULE log_delete_series AS
45.         ON DELETE TO SERIES
46.         DO ALSO INSERT INTO LOG_UPD_DEL (data_hora, comando,
        series_idseries, series_episodio_idepisodio, USUARIO_CPF,
        USUARIO_LOGIN_EMAIL, USUARIO_COBRANCA_IDCOBRANCA,
        USUARIO_PREFERENCIAS_IDPREFERENCIAS, EPISODIO_IDEPISODIO)
47.         VALUES (CURRENT_TIMESTAMP, 'DELETE', OLD.idseries,
        OLD.episodio_idepisodio, NULL, NULL, NULL, NULL, NULL);
48.
49.         -----
50.
51.         CREATE OR REPLACE RULE log_update_filmes AS
52.         ON UPDATE TO FILMES
53.         DO ALSO INSERT INTO LOG_UPD_DEL (data_hora, comando,
        filmes_idfilmes, series_idseries, series_episodio_idepisodio,
        documentario_iddocumentario, ator_nome_ator, USUARIO_CPF,
        USUARIO_LOGIN_EMAIL, USUARIO_COBRANCA_IDCOBRANCA,
        USUARIO_PREFERENCIAS_IDPREFERENCIAS, EPISODIO_IDEPISODIO)
54.         VALUES (CURRENT_TIMESTAMP, 'UPDATE', OLD.idfilmes,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);
55.
56.         CREATE OR REPLACE RULE log_delete_filmes AS
57.         ON DELETE TO FILMES
58.         DO ALSO INSERT INTO LOG_UPD_DEL (data_hora, comando,
        filmes_idfilmes, series_idseries, series_episodio_idepisodio,
        documentario_iddocumentario, ator_nome_ator, USUARIO_CPF,
        USUARIO_LOGIN_EMAIL, USUARIO_COBRANCA_IDCOBRANCA,
        USUARIO_PREFERENCIAS_IDPREFERENCIAS, EPISODIO_IDEPISODIO)
59.         VALUES (CURRENT_TIMESTAMP, 'DELETE', OLD.idfilmes,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);
60.
61.         -----
62.
63.         CREATE OR REPLACE RULE log_update_documentario AS
64.         ON UPDATE TO DOCUMENTARIO
65.         DO ALSO INSERT INTO LOG_UPD_DEL (data_hora, comando,
        documentario_iddocumentario, series_idseries,
        series_episodio_idepisodio, filmes_idfilmes, ator_nome_ator,
```

```

USUARIO_CPF, USUARIO_LOGIN_EMAIL,
USUARIO_COBRANCA_IDCOBRANCA,
USUARIO_PREFERENCIAS_IDPREFERENCIAS, EPISODIO_IDEPISODIO)
66.     VALUES (CURRENT_TIMESTAMP, 'UPDATE',
        OLD.iddocumentario, NULL, NULL, NULL, NULL, NULL, NULL,
        NULL, NULL);
67.
68.     CREATE OR REPLACE RULE log_delete_documentario AS
69.     ON DELETE TO DOCUMENTARIO
70.     DO ALSO INSERT INTO LOG_UPD_DEL (data_hora, comando,
        documentario_iddocumentario, series_idseries,
        series_episodio_idepisodio, filmes_idfilmes, ator_nome_ator,
        USUARIO_CPF, USUARIO_LOGIN_EMAIL,
        USUARIO_COBRANCA_IDCOBRANCA,
        USUARIO_PREFERENCIAS_IDPREFERENCIAS, EPISODIO_IDEPISODIO)
71.     VALUES (CURRENT_TIMESTAMP, 'DELETE',
        OLD.iddocumentario, NULL, NULL, NULL, NULL, NULL, NULL,
        NULL, NULL);
72.
73.     -----
74.
75.     CREATE OR REPLACE RULE log_update_ator AS
76.     ON UPDATE TO ATOR
77.     DO ALSO INSERT INTO LOG_UPD_DEL (data_hora, comando,
        ator_nome_ator, series_idseries, series_episodio_idepisodio,
        filmes_idfilmes, documentario_iddocumentario, USUARIO_CPF,
        USUARIO_LOGIN_EMAIL, USUARIO_COBRANCA_IDCOBRANCA,
        USUARIO_PREFERENCIAS_IDPREFERENCIAS, EPISODIO_IDEPISODIO)
78.     VALUES (CURRENT_TIMESTAMP, 'UPDATE', OLD.nome_ator,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);
79.
80.     CREATE OR REPLACE RULE log_delete_ator AS
81.     ON DELETE TO ATOR
82.     DO ALSO INSERT INTO LOG_UPD_DEL (data_hora, comando,
        ator_nome_ator, series_idseries, series_episodio_idepisodio,
        filmes_idfilmes, documentario_iddocumentario, USUARIO_CPF,
        USUARIO_LOGIN_EMAIL, USUARIO_COBRANCA_IDCOBRANCA,
        USUARIO_PREFERENCIAS_IDPREFERENCIAS, EPISODIO_IDEPISODIO)
83.     VALUES (CURRENT_TIMESTAMP, 'DELETE', OLD.nome_ator,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);

```

## 6. CÓDIGO SQL DE CRIAÇÃO DAS STORED PROCEDURES (FUNCTIONS) DO BANCO DE DADOS:

```
1. -----4
2. CREATE OR REPLACE FUNCTION avaliar_video(
3.     p_cpf VARCHAR(15),
4.     p_tipo_video VARCHAR(20),
5.     p_id_video INTEGER,
6.     p_avaliacao REAL
7. ) RETURNS VOID AS $$
8. BEGIN
9.     IF p_avaliacao < 0 OR p_avaliacao > 10 THEN
10.         RAISE EXCEPTION 'A nota de avaliação deve estar
    entre 0 e 10';
11.     END IF;
12.
13.     IF p_tipo_video = 'FILME' THEN
14.         UPDATE FILMES
15.         SET avaliacao = p_avaliacao
16.         WHERE idfilmes = p_id_video;
17.
18.     ELSIF p_tipo_video = 'SERIE' THEN
19.         UPDATE SERIES
20.         SET avaliacao = p_avaliacao
21.         WHERE idseries = p_id_video;
22.
23.     ELSIF p_tipo_video = 'DOCUMENTARIO' THEN
24.         UPDATE DOCUMENTARIO
25.         SET avaliacao = p_avaliacao
26.         WHERE iddocumentario = p_id_video;
27.
28.     ELSE
29.         RAISE EXCEPTION 'Tipo de vídeo desconhecido:
    %', p_tipo_video;
30.     END IF;
31.
32.     RAISE NOTICE 'Avaliação atribuída com sucesso.';
33. END;
34.
35. -----5
36.
37. CREATE OR REPLACE FUNCTION
    buscar_videos_por_ator(p_nome_ator VARCHAR)
38. RETURNS TABLE(tipo_video VARCHAR, titulo VARCHAR,
    ano_prod INTEGER, categoria VARCHAR) AS $$
```

```

39.      BEGIN
40.          RETURN QUERY
41.          SELECT 'Filme' AS tipo_video, f.titulo, f.ano_prod,
              f.categoria
42.          FROM FILMES f
43.          INNER JOIN FILMES_has_ATOM fa ON f.idfilmes =
              fa.filmes_idfilmes
44.          WHERE fa.ator_nome_ator = p_nome_ator
45.
46.          UNION
47.
48.          SELECT 'Serie' AS tipo_video, s.titulo, s.ano_prod,
              s.categoria
49.          FROM SERIES s
50.          INNER JOIN SERIES_has_ATOM sa ON s.idseries =
              sa.series_idseries
51.          WHERE sa.ator_nome_ator = p_nome_ator
52.
53.          UNION
54.
55.          SELECT 'Documentario' AS tipo_video, d.titulo,
              d.ano_prod, NULL AS categoria
56.          FROM DOCUMENTARIO d
57.          INNER JOIN DOCUMENTARIO_has_ATOM da ON
              d.iddocumentario = da.documentario_iddocumentario
58.          WHERE da.ator_nome_ator = p_nome_ator;
59.      END;
60.
61.      -----6
62.
63.      CREATE OR REPLACE FUNCTION
        buscar_videos_por_titulo_inicial(p_titulo_inicial VARCHAR)
64.      RETURNS TABLE(tipo_video VARCHAR, titulo VARCHAR,
        ano_prod INTEGER, categoria VARCHAR) AS $$
65.      BEGIN
66.          RETURN QUERY
67.          SELECT 'Filme' AS tipo_video, titulo, ano_prod,
              categoria
68.          FROM FILMES
69.          WHERE titulo ILIKE p_titulo_inicial || '%'
70.
71.          UNION
72.
73.          SELECT 'Serie' AS tipo_video, titulo, ano_prod,
              categoria

```



```

74.         FROM SERIES
75.         WHERE titulo ILIKE p_titulo_inicial || '%'
76.
77.         UNION
78.
79.         SELECT 'Documentario' AS tipo_video, titulo,
      ano_prod, NULL AS categoria
80.         FROM DOCUMENTARIO
81.         WHERE titulo ILIKE p_titulo_inicial || '%';
82.     END;
83.
84. -----7
85. CREATE OR REPLACE FUNCTION get_videos_by_type(
86.     p_tipo_video VARCHAR(15)
87. )
88. RETURNS TABLE (
89.     tipo_video VARCHAR(15),
90.     titulo VARCHAR(45),
91.     ano_prod INTEGER,
92.     categoria VARCHAR(45)
93. ) AS $$
94. BEGIN
95.     IF p_tipo_video = 'Filme' THEN
96.         RETURN QUERY
97.         SELECT 'Filme' AS tipo_video, f.titulo,
      f.ano_prod, f.categoria
98.         FROM FILMES f;
99.
100.     ELSIF p_tipo_video = 'Serie' THEN
101.         RETURN QUERY
102.         SELECT 'Serie' AS tipo_video, s.titulo,
      s.ano_prod, s.categoria
103.         FROM SERIES s;
104.
105.     ELSIF p_tipo_video = 'Documentario' THEN
106.         RETURN QUERY
107.         SELECT 'Documentario' AS tipo_video, d.titulo,
      d.ano_prod, d.categoria
108.         FROM DOCUMENTARIO d;
109.
110.     ELSE
111.         RAISE EXCEPTION 'Tipo de vídeo inválido: %. Use
      "Filme", "Serie" ou "Documentario".', p_tipo_video;
112.     END IF;
113. END;

```

```
114.
115. -----8
116.
117. CREATE OR REPLACE FUNCTION
    buscar_filmes_series_por_categoria(p_categoria VARCHAR)
118. RETURNS TABLE(tipo_video VARCHAR, titulo VARCHAR,
    ano_prod INTEGER, avaliacao REAL) AS $$
119. BEGIN
120.     RETURN QUERY
121.     SELECT 'Filme' AS tipo_video, titulo, ano_prod,
    avaliacao
122.     FROM FILMES
123.     WHERE categoria = p_categoria
124.     ORDER BY avaliacao DESC
125.
126. UNION
127.
128.     SELECT 'Serie' AS tipo_video, titulo, ano_prod,
    avaliacao
129.     FROM SERIES
130.     WHERE categoria = p_categoria
131.     ORDER BY avaliacao DESC;
132. END;
```