

**Aluno:** Leonardo Heitor Poglia

**Professor:** Martim Dietterle

**Curso:** Engenharia de Software

**Data:** 05/06/2025

**Matéria:** Banco de Dados

## **Exercício Plataforma de Streaming**

## Sumário

<b>Definição do objeto do trabalho</b>	<b>3</b>
Inserção de dados	4
Consultas	4
Ferramentas	5
Entregáveis	5
Forma de entrega	5
<b>Entrega</b>	<b>6</b>
Diagrama E-R	6
DDL	7
DML	12
SQL queries	15

## Definição do objeto do trabalho

Uma plataforma de streaming está informatizando o cadastro de seu acervo, e para isso contratou você para fazer a criação do banco de dados do sistema.

Você irá criar um banco de dados para uma plataforma de streaming cadastrar seu acervo de filmes e séries em um sistema informatizado.

Cabe a você criar a estrutura do banco de dados, suas entidades, atributos, fazer a definição de chaves, views (caso necessário), triggers e procedures em casos em que se aplicam.

Você deverá obedecer aos seguintes critérios:

- O sistema deve permitir o cadastro de atores;
- O sistema deve permitir cadastrar séries;
- O sistema deve permitir cadastrar filmes;
- O sistema deve permitir o cadastro de temporadas (em caso de séries);
- O sistema deve permitir o cadastro de episódios (em caso de séries);
- O sistema deve permitir o cadastro de personagens;
- O sistema deve permitir o cadastro do personagem que o ator interpreta na série/filme;
- O sistema deve permitir o cadastro de diretores;
- O sistema deve permitir o cadastro de diretores que dirigem um episódio (uma série pode ter diferentes diretores em cada episódio);
- O sistema deve permitir o cadastro de diretores que dirigem filmes;
- O sistema deve separar o acervo por faixa etária;
- Os itens do acervo devem fazer parte de um gênero;
- Os filmes e séries devem ter o país de origem em seu cadastro;
- O sistema deve permitir o cadastro de premiações da série;
- O sistema deve permitir o cadastro de premiações de filmes;
- O sistema deve permitir o cadastro de premiações de diretores;
- O sistema deve permitir o cadastro de premiações de atores;
- O sistema deve permitir o cadastro de avaliação por episódios (em caso de séries);
- O sistema deve permitir o cadastro de avaliação por temporadas;
- O sistema deve permitir o cadastro de avaliação por título (a série como um todo), ou do filme;

## Inserção de dados

Você deverá cadastrar dados em todas as tabelas de forma que as consultas a seguir retornem ao menos 3 resultados.

## Consultas

- 1) Liste todas as séries que são europeias, que tenham mais de 3 temporadas e que possuam premiação de atores em pelo menos duas temporadas;
- 2) Liste todas os filmes que tenham sido premiados nas categorias diretor e ator;
- 3) Liste todas as séries que não sejam europeias, que possuam mais atrizes do que atores e que tenham episódios dirigidos por mais de um diretor;
- 4) Liste todas as premiações dos filmes feitos antes de 2003, que sejam do gênero aventura e tenham classificação de faixa etária maior do que 12 anos;
- 5) Liste todos os personagens interpretados por um ator que tenha feito filmes de romance e que nunca tenha sido premiado;
- 6) Liste todos os títulos de episódios em ordem alfabética das séries que tenham mais de 3 temporadas, tenham sido dirigidos por mulheres e tenham classificação livre;
- 7) Liste todos os filmes feitos após 2003 e que tenham atores que tenham participado de séries premiadas.
- 8) Liste todos os atores que participaram de séries antes de 2003 (e não tenham participado de outras séries após esse ano), e tenham feito filmes após 2003 (e não tenham feito filmes antes de 2003). Liste os resultados em ordem inversa de data (mais recente para o mais antigo).
- 9) Liste todos os episódios de séries que tenham no mínimo 4 episódios por temporada com duração maior do que 45 minutos, e que tenham menos de 3 atores nestes episódios (com mais de 45 minutos).
- 10) Liste todos os filmes em resolução 4k que tenham sido dirigidos por um de seus atores.
- 11) Liste todos os filmes que tenham duração de mais de 3 horas, que tenham um ator que interpretou a si mesmo (nome do ator igual ao nome do personagem), que tenham sido lançados antes de 2010 e que tenham recebido alguma premiação (ator, diretor ou filme).
- 12) Liste todas os episódios, os atores e avaliação dos episódios das séries que possuem avaliação média dos episódios da temporada superior a 7.5, que tenham duração média dos episódios da temporada entre 45 e 50 minutos, que sejam do gênero comédia ou ação e que tenham tido mais de um diretor na temporada.
- 13) Liste as avaliações médias das séries e filmes de cada diretor, agrupado por tipo (filme/série).
- 14) Liste os filmes e séries que tenham avaliações de pelo menos dois atores superiores a 8.5, do gênero comédia.
- 15) Crie triggers para que, ao remover um filme ou série do catálogo, todas as premiações deste filme ou série também sejam removidas.
- 16) Crie uma function que retorne o total de premiações de cada filme (ator, diretor e filme);
- 17) Crie uma procedure que inative a exibição de todos os filmes e séries de um determinado país (usado caso exista a proibição de exibição de obras produzidas em algum país específico).

**Ferramentas**

- O trabalho deve ser desenvolvido no Oracle Live SQL (<https://livesql.oracle.com/landing/>)

**Entregáveis**

- 1) Diagrama E-R completo do banco de dados, contendo entidades, chaves primárias, estrangeiras e cardinalidade dos relacionamentos (em formato PDF);
- 2) Script SQL com a criação completa do banco de dados (em formato texto);
- 3) Script SQL com a inserção de dados no banco de dados (em formato texto);
- 4) Script SQL com as queries solicitadas (em formato texto);

**Forma de entrega**

- Link para o repositório GitHub juntamente com o número do commit válido para entrega.
- A entrega será feita por e-mail.



## DDL/DML

```
BEGIN
  FOR t IN (SELECT table_name FROM user_tables) LOOP
    EXECUTE IMMEDIATE 'DROP TABLE ' || t.table_name || ' CASCADE CONSTRAINTS';
  END LOOP;
EXCEPTION WHEN OTHERS THEN NULL; END;
/
```

```
CREATE TABLE pais (
  pais_id    NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  nome       VARCHAR2(100) NOT NULL UNIQUE,
  continente VARCHAR2(50)
);
```

```
CREATE TABLE genero (
  genero_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  nome      VARCHAR2(50) NOT NULL UNIQUE
);
```

```
CREATE TABLE classificacao_idade (
  classificacao_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  rotulo           VARCHAR2(10) NOT NULL UNIQUE,
  idade_minima     NUMBER(2)    NOT NULL
);
```

```
CREATE TABLE ator (
  ator_id    NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  nome       VARCHAR2(100) NOT NULL,
  genero     CHAR(1) CHECK (genero IN ('M', 'F')),
  data_nascimento DATE,
  pais_id    NUMBER REFERENCES pais(pais_id)
);
```

```
CREATE TABLE diretor (
  diretor_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  nome       VARCHAR2(100) NOT NULL,
  genero     CHAR(1) CHECK (genero IN ('M', 'F')),
  data_nascimento DATE,
  pais_id    NUMBER REFERENCES pais(pais_id)
);
```

```
CREATE TABLE filme (
  filme_id    NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  titulo      VARCHAR2(200) NOT NULL,
```

```
ano_lancamento  NUMBER(4),
duracao_min      NUMBER,
resolucao        VARCHAR2(20),
genero_id        NUMBER REFERENCES genero(genero_id),
pais_id          NUMBER REFERENCES pais(pais_id),
classificacao_id NUMBER REFERENCES classificacao_idade(classificacao_id),
esta_ativo       CHAR(1) DEFAULT 'Y' CHECK (esta_ativo IN ('Y','N'))
);
```

```
CREATE TABLE serie (
    serie_id      NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    titulo        VARCHAR2(200) NOT NULL,
    ano_lancamento  NUMBER(4),
    genero_id      NUMBER REFERENCES genero(genero_id),
    pais_id        NUMBER REFERENCES pais(pais_id),
    classificacao_id NUMBER REFERENCES classificacao_idade(classificacao_id),
    esta_ativo     CHAR(1) DEFAULT 'Y' CHECK (esta_ativo IN ('Y','N'))
);
```

```
CREATE TABLE temporada (
    temporada_id  NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    serie_id      NUMBER REFERENCES serie(serie_id),
    numero_temporada NUMBER NOT NULL,
    CONSTRAINT uk_serie_temporada UNIQUE (serie_id, numero_temporada)
);
```

```
CREATE TABLE episodio (
    episodio_id   NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    temporada_id  NUMBER REFERENCES temporada(temporada_id),
    numero_episodio NUMBER NOT NULL,
    titulo        VARCHAR2(200) NOT NULL,
    duracao_min   NUMBER,
    data_lancamento DATE,
    CONSTRAINT uk_episodio UNIQUE (temporada_id, numero_episodio)
);
```

```
CREATE TABLE personagem (
    personagem_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    nome          VARCHAR2(200)
);
```

```
CREATE TABLE ator_filme (
    ator_id       NUMBER REFERENCES ator(ator_id),
    filme_id      NUMBER REFERENCES filme(filme_id),
    personagem_id NUMBER REFERENCES personagem(personagem_id),
    PRIMARY KEY (ator_id, filme_id)
```



);

```
CREATE TABLE ator_episodio (  
    ator_id          NUMBER REFERENCES ator(ator_id),  
    episodio_id      NUMBER REFERENCES episodio(episodio_id),  
    personagem_id    NUMBER REFERENCES personagem(personagem_id),  
    PRIMARY KEY (ator_id, episodio_id)  
);
```

```
CREATE TABLE diretor_filme (  
    diretor_id       NUMBER REFERENCES diretor(diretor_id),  
    filme_id         NUMBER REFERENCES filme(filme_id),  
    PRIMARY KEY (diretor_id, filme_id)  
);
```

```
CREATE TABLE diretor_episodio (  
    diretor_id       NUMBER REFERENCES diretor(diretor_id),  
    episodio_id      NUMBER REFERENCES episodio(episodio_id),  
    PRIMARY KEY (diretor_id, episodio_id)  
);
```

```
CREATE TABLE premiacao (  
    premiacao_id     NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    nome             VARCHAR2(200),  
    ano_premiacao    NUMBER(4),  
    categoria        VARCHAR2(50),  
    ator_id          NUMBER REFERENCES ator(ator_id),  
    diretor_id       NUMBER REFERENCES diretor(diretor_id),  
    filme_id         NUMBER REFERENCES filme(filme_id),  
    serie_id         NUMBER REFERENCES serie(serie_id),  
    temporada_id     NUMBER REFERENCES temporada(temporada_id),  
    episodio_id      NUMBER REFERENCES episodio(episodio_id)  
);
```

```
CREATE TABLE avaliacao (  
    avaliacao_id     NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    nota             NUMBER(3,1) CHECK (nota BETWEEN 0 AND 10),  
    comentarios      VARCHAR2(500),  
    ator_id          NUMBER REFERENCES ator(ator_id),  
    diretor_id       NUMBER REFERENCES diretor(diretor_id),  
    filme_id         NUMBER REFERENCES filme(filme_id),  
    serie_id         NUMBER REFERENCES serie(serie_id),  
    temporada_id     NUMBER REFERENCES temporada(temporada_id),  
    episodio_id      NUMBER REFERENCES episodio(episodio_id)  
);
```

```
CREATE OR REPLACE TRIGGER trg_del_filme_premiacoes
AFTER DELETE ON filme
FOR EACH ROW
BEGIN
    DELETE FROM premiacao WHERE filme_id = :old.filme_id;
END;
/
```

```
CREATE OR REPLACE TRIGGER trg_del_serie_premiacoes
AFTER DELETE ON serie
FOR EACH ROW
BEGIN
    DELETE FROM premiacao WHERE serie_id = :old.serie_id;
END;
/
```

```
CREATE OR REPLACE FUNCTION fn_total_premiacoes_filme (p_filme_id NUMBER)
RETURN NUMBER
IS
    v_total NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_total
    FROM premiacao p
    WHERE p.filme_id      = p_filme_id
       OR p.ator_id       IN (SELECT ator_id FROM ator_filme WHERE filme_id = p_filme_id)
       OR p.diretor_id    IN (SELECT diretor_id FROM diretor_filme WHERE filme_id = p_filme_id);
    RETURN v_total;
END;
/
```

```
CREATE OR REPLACE PROCEDURE pr_inativar_por_pais (p_pais_id NUMBER)
IS
BEGIN
    UPDATE filme SET esta_ativo = 'N' WHERE pais_id = p_pais_id;
    UPDATE serie SET esta_ativo = 'N' WHERE pais_id = p_pais_id;
END;
/
```

```
DELETE FROM premiacao;
DELETE FROM avaliacao;
DELETE FROM diretor_episodio;
DELETE FROM ator_episodio;
DELETE FROM episodio;
DELETE FROM temporada;
DELETE FROM serie;
DELETE FROM diretor_filme;
```

```
DELETE FROM ator_filme;
DELETE FROM filme;
DELETE FROM personagem;
DELETE FROM diretor;
DELETE FROM ator;
DELETE FROM classificacao_idade;
DELETE FROM genero;
DELETE FROM pais;
COMMIT;

DECLARE
    PROCEDURE reiniciar_identidade(p_tabela VARCHAR2, p_coluna VARCHAR2) IS
    BEGIN
        EXECUTE IMMEDIATE 'ALTER TABLE '||p_tabela||' MODIFY '||p_coluna||' GENERATED BY DEFAULT ON
NULL AS IDENTITY (START WITH 1)';
        EXCEPTION WHEN OTHERS THEN NULL;
    END;
BEGIN
    reiniciar_identidade('pais', 'pais_id');
    reiniciar_identidade('genero', 'genero_id');
    reiniciar_identidade('classificacao_idade', 'classificacao_id');
    reiniciar_identidade('ator', 'ator_id');
    reiniciar_identidade('diretor', 'diretor_id');
    reiniciar_identidade('filme', 'filme_id');
    reiniciar_identidade('serie', 'serie_id');
    reiniciar_identidade('personagem', 'personagem_id');
END;
/

INSERT INTO pais (nome, continente) VALUES ('EUA', 'América do Norte');
INSERT INTO pais (nome, continente) VALUES ('Alemanha', 'Europa');
INSERT INTO pais (nome, continente) VALUES ('Reino Unido', 'Europa');
INSERT INTO pais (nome, continente) VALUES ('Brasil', 'América do Sul');
INSERT INTO pais (nome, continente) VALUES ('França', 'Europa');
INSERT INTO pais (nome, continente) VALUES ('Espanha', 'Europa');
INSERT INTO pais (nome, continente) VALUES ('Japão', 'Ásia');

INSERT INTO genero (nome) VALUES ('Ação');
INSERT INTO genero (nome) VALUES ('Comédia');
INSERT INTO genero (nome) VALUES ('Aventura');
INSERT INTO genero (nome) VALUES ('Romance');
INSERT INTO genero (nome) VALUES ('Drama');
INSERT INTO genero (nome) VALUES ('Ficção Científica');

INSERT INTO classificacao_idade (rotulo, idade_minima) VALUES ('L', 0);
INSERT INTO classificacao_idade (rotulo, idade_minima) VALUES ('12', 12);
INSERT INTO classificacao_idade (rotulo, idade_minima) VALUES ('16', 16);
```

```
INSERT INTO classificacao_idade (rotulo, idade_minima) VALUES ('18', 18);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Leo Valente', 'M', DATE  
'1980-05-01', 1);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Sofia Ribeiro', 'F', DATE  
'1985-07-12', 4);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Clara Becker', 'F', DATE  
'1990-03-20', 2);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Arthur Morgan', 'M', DATE  
'1975-11-30', 3);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Matias Schultz', 'M', DATE  
'1982-02-10', 2);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Helena Costa', 'F', DATE  
'1992-09-05', 3);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Kenji Yamamoto', 'M', DATE  
'1988-01-15', 7);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Yumi Hayashi', 'F', DATE  
'1993-08-08', 7);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Rafael Oliveira', 'M', DATE  
'1981-04-22', 4);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Isabela Jimenez', 'F', DATE  
'1987-12-02', 6);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Pierre Dubois', 'M', DATE  
'1979-06-11', 5);
```

```
INSERT INTO ator (nome, genero, data_nascimento, pais_id) VALUES ('Juliette Blanchard', 'F', DATE  
'1991-11-23', 5);
```

```
INSERT INTO diretor (nome, genero, data_nascimento, pais_id) VALUES ('Maximus Powers', 'M', DATE  
'1965-04-22', 1);
```

```
INSERT INTO diretor (nome, genero, data_nascimento, pais_id) VALUES ('Serena Vogel', 'F', DATE  
'1972-10-18', 2);
```

```
INSERT INTO diretor (nome, genero, data_nascimento, pais_id) VALUES ('Beatriz Lima', 'F', DATE  
'1976-06-30', 4);
```

```
INSERT INTO diretor (nome, genero, data_nascimento, pais_id) VALUES ('Victor Reed', 'M', DATE  
'1968-12-09', 3);
```

```
INSERT INTO diretor (nome, genero, data_nascimento, pais_id) VALUES ('Amélie Dubois', 'F', DATE  
'1974-05-05', 5);
```

```
INSERT INTO diretor (nome, genero, data_nascimento, pais_id) VALUES ('Harumi Tanaka', 'F', DATE  
'1970-09-14', 7);
```

```
INSERT INTO diretor (nome, genero, data_nascimento, pais_id) VALUES ('Alejandro Vargas', 'M', DATE  
'1966-02-17', 6);
```

```
INSERT INTO diretor (nome, genero, data_nascimento, pais_id) VALUES ('Samuel Wright', 'M', DATE  
'1963-03-30', 1);
```

```
INSERT INTO personagem (nome) VALUES ('Jonas');
```

```
INSERT INTO personagem (nome) VALUES ('Fantasma, o Cão');
```

```
INSERT INTO personagem (nome) VALUES ('Zeca Pimentel');
```

```
INSERT INTO personagem (nome) VALUES ('Kael');
```

```
INSERT INTO personagem (nome) VALUES ('Explorador Gelo');
INSERT INTO personagem (nome) VALUES ('Dama Escarlata');
INSERT INTO personagem (nome) VALUES ('Cometa Vermelho');
INSERT INTO personagem (nome) VALUES ('Marcus');
INSERT INTO personagem (nome) VALUES ('Tiago');
INSERT INTO personagem (nome) VALUES ('Luc');
```

```
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('Cume da Coragem', 2001, 140, '1080p', 3, 1, 2);
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('A Conexão de Berlim', 1999, 130, '1080p', 1, 2, 3);
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('Rota de Fuga Digital', 2015, 120, '4K', 2, 1, 1);
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('A Longa Jornada', 2009, 190, '1080p', 4, 3, 3);
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('O Segredo do Templo Perdido', 2000, 150, '1080p', 3, 6, 3);
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('Destino Inesperado', 2018, 110, '4K', 2, 4, 1);
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('As Crônicas da Liberdade', 2008, 200, '1080p', 4, 5, 3);
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('Sombras do Império', 1998, 135, '1080p', 3, 5, 3);
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('Protocolo Final', 2021, 115, '4K', 2, 1, 1);
INSERT INTO filme (titulo, ano_lancamento, duracao_min, resolucao, genero_id, pais_id,
classificacao_id)
VALUES ('O Enigma sem Fim', 2005, 185, '1080p', 4, 2, 3);
```

```
INSERT INTO serie (titulo, ano_lancamento, genero_id, pais_id, classificacao_id) VALUES ('Sombras
de Viena', 2010, 1, 2, 3);
INSERT INTO serie (titulo, ano_lancamento, genero_id, pais_id, classificacao_id) VALUES ('Luzes de
Neon', 2011, 5, 2, 3);
INSERT INTO serie (titulo, ano_lancamento, genero_id, pais_id, classificacao_id) VALUES ('Névoa
Sobre o Tâmis', 2012, 1, 3, 3);
INSERT INTO serie (titulo, ano_lancamento, genero_id, pais_id, classificacao_id) VALUES ('Central
do Riso', 2015, 2, 1, 1);
INSERT INTO serie (titulo, ano_lancamento, genero_id, pais_id, classificacao_id) VALUES ('Coqueiro
Abaixo', 2016, 2, 4, 1);
```



```
INSERT INTO serie (titulo, ano_lancamento, genero_id, pais_id, classificacao_id) VALUES ('A Espada  
do Dragão', 2014, 1, 7, 3);  
INSERT INTO serie (titulo, ano_lancamento, genero_id, pais_id, classificacao_id) VALUES ('Crônicas  
de Andrômeda', 1999, 6, 1, 3);  
INSERT INTO serie (titulo, ano_lancamento, genero_id, pais_id, classificacao_id) VALUES ('Cartas de  
um Amor Perdido', 2000, 4, 6, 3);  
INSERT INTO serie (titulo, ano_lancamento, genero_id, pais_id, classificacao_id) VALUES ('Operação  
Trovão', 2002, 1, 5, 3);
```

```
DECLARE  
    v_id_serie NUMBER;  
    v_temporada NUMBER;  
BEGIN  
    FOR v_id_serie IN 1..3 LOOP  
        FOR v_temporada IN 1..5 LOOP  
            INSERT INTO temporada (serie_id, numero_temporada) VALUES (v_id_serie, v_temporada);  
        END LOOP;  
    END LOOP;  
    FOR v_id_serie IN 4..6 LOOP  
        FOR v_temporada IN 1..4 LOOP  
            INSERT INTO temporada (serie_id, numero_temporada) VALUES (v_id_serie, v_temporada);  
        END LOOP;  
    END LOOP;  
    FOR v_id_serie IN 7..9 LOOP  
        INSERT INTO temporada (serie_id, numero_temporada) VALUES (v_id_serie, 1);  
    END LOOP;  
END;  
/  
  
DECLARE  
    v_id_temporada NUMBER;  
    v_num_ep NUMBER;  
BEGIN  
    FOR rec IN (SELECT temporada_id FROM temporada) LOOP  
        v_id_temporada := rec.temporada_id;  
        FOR v_num_ep IN 1..4 LOOP  
            INSERT INTO episodio (temporada_id, numero_episodio, titulo, duracao_min, data_lancamento)  
            VALUES (v_id_temporada, v_num_ep, 'Episodio ' || v_num_ep || 'da Temporada' || v_id_temporada, 46 +  
MOD(v_num_ep,4), DATE '2010-01-01' + v_id_temporada);  
        END LOOP;  
    END LOOP;  
END;  
/  
  
INSERT INTO ator_filme VALUES (11,10,10);  
INSERT INTO ator_filme VALUES (4,4,8);  
INSERT INTO ator_filme VALUES (2,3,3);  
INSERT INTO ator_filme VALUES (9,7,9);
```

```
INSERT INTO ator_filme VALUES (6,6,3);
INSERT INTO ator_filme VALUES (8,9,3);
INSERT INTO ator_filme VALUES (5,9,3);
INSERT INTO ator_filme VALUES (1,1,1);
INSERT INTO ator_filme VALUES (10,9,3);
INSERT INTO ator_filme VALUES (2,9,3);
INSERT INTO ator_filme VALUES (3,5,5);
```

```
INSERT INTO ator_episodio VALUES (2,(SELECT episodio_id FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=4 AND t.numero_temporada=1 AND
e.numero_episodio=1),3);
INSERT INTO ator_episodio VALUES (6,(SELECT episodio_id FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=4 AND t.numero_temporada=1 AND
e.numero_episodio=1),3);
INSERT INTO ator_episodio VALUES (2,(SELECT episodio_id FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=5 AND t.numero_temporada=1 AND
e.numero_episodio=1),3);
INSERT INTO ator_episodio VALUES (6,(SELECT episodio_id FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=5 AND t.numero_temporada=1 AND
e.numero_episodio=1),3);
INSERT INTO ator_episodio VALUES (8,(SELECT episodio_id FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=6 AND t.numero_temporada=1 AND
e.numero_episodio=1),4);
INSERT INTO ator_episodio VALUES (5,(SELECT episodio_id FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=7 AND e.numero_episodio=1),1);
INSERT INTO ator_episodio VALUES (10,(SELECT episodio_id FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=8 AND e.numero_episodio=1),1);
INSERT INTO ator_episodio VALUES (11,(SELECT episodio_id FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=9 AND e.numero_episodio=1),1);
```

```
INSERT INTO diretor_filme VALUES (1,1);
INSERT INTO diretor_filme VALUES (2,2);
INSERT INTO diretor_filme VALUES (2,3);
INSERT INTO diretor_filme VALUES (4,4);
INSERT INTO diretor_filme VALUES (5,5);
INSERT INTO diretor_filme VALUES (6,6);
INSERT INTO diretor_filme VALUES (5,7);
INSERT INTO diretor_filme VALUES (5,8);
INSERT INTO diretor_filme VALUES (8,9);
INSERT INTO diretor_filme VALUES (8,10);
```

```
DECLARE
```

```
    v_id_ep NUMBER;
```

```
BEGIN
```

```
    SELECT e.episodio_id INTO v_id_ep FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=4 AND t.numero_temporada=1 AND e.numero_episodio=1;
    INSERT INTO diretor_episodio VALUES (2,v_id_ep);
    INSERT INTO diretor_episodio VALUES (3,v_id_ep);
```

```

SELECT e.episodio_id INTO v_id_ep FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=4 AND t.numero_temporada=3 AND e.numero_episodio=1;
INSERT INTO diretor_episodio VALUES (2,v_id_ep);
INSERT INTO diretor_episodio VALUES (3,v_id_ep);
SELECT e.episodio_id INTO v_id_ep FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=5 AND t.numero_temporada=1 AND e.numero_episodio=1;
INSERT INTO diretor_episodio VALUES (3,v_id_ep);
INSERT INTO diretor_episodio VALUES (2,v_id_ep);
SELECT e.episodio_id INTO v_id_ep FROM episodio e JOIN temporada t ON
t.temporada_id=e.temporada_id WHERE t.serie_id=6 AND t.numero_temporada=1 AND e.numero_episodio=1;
INSERT INTO diretor_episodio VALUES (6,v_id_ep);
INSERT INTO diretor_episodio VALUES (4,v_id_ep);
END;
/

```

```

INSERT INTO premiacao (nome, ano_premiacao, categoria, ator_id, serie_id, temporada_id)
SELECT 'Prêmio de Ator', 2012, 'Ator', 1, 1, temporada_id
FROM temporada
WHERE serie_id = 1 AND numero_temporada IN (1,2,3,4);

```

```

INSERT INTO premiacao (nome, ano_premiacao, categoria, ator_id, serie_id, temporada_id)
SELECT 'Prêmio de Ator', 2013, 'Ator', 3, 2, temporada_id
FROM temporada
WHERE serie_id = 2 AND numero_temporada IN (1,2,3,4);

```

```

INSERT INTO premiacao (nome, ano_premiacao, categoria, ator_id, serie_id, temporada_id)
SELECT 'Prêmio de Ator', 2014, 'Ator', 6, 3, temporada_id
FROM temporada
WHERE serie_id = 3 AND numero_temporada IN (1,2,3,4);

```

```

INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Melhor Diretor', 2002,
'Diretor', 1);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Melhor Ator', 2002,
'Ator', 1);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Melhor Diretor', 2000,
'Diretor', 2);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Melhor Ator', 2000,
'Ator', 2);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Melhor Diretor', 2001,
'Diretor', 5);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Melhor Ator', 2001,
'Ator', 5);
INSERT INTO premiacao (nome, ano_premiacao, categoria, serie_id) VALUES ('Melhor Série', 2017,
'Série', 4);
INSERT INTO premiacao (nome, ano_premiacao, categoria, serie_id) VALUES ('Melhor Série', 2018,
'Série', 5);

```





```
INSERT INTO premiacao (nome, ano_premiacao, categoria, serie_id) VALUES ('Melhor Série', 2019, 'Série', 6);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Prêmio Aventura', 2002, 'Filme', 1);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Prêmio Aventura', 2002, 'Filme', 5);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Prêmio Aventura', 1999, 'Filme', 8);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Prêmio de Longa-Metragem', 2009, 'Filme', 4);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Prêmio de Longa-Metragem', 2008, 'Filme', 7);
INSERT INTO premiacao (nome, ano_premiacao, categoria, filme_id) VALUES ('Prêmio de Longa-Metragem', 2005, 'Filme', 10);
```

```
INSERT INTO avaliacao (nota, ator_id, filme_id) VALUES (9.0, 2, 3);
INSERT INTO avaliacao (nota, ator_id, filme_id) VALUES (8.8, 6, 3);
INSERT INTO avaliacao (nota, ator_id, filme_id) VALUES (9.2, 2, 6);
INSERT INTO avaliacao (nota, ator_id, filme_id) VALUES (9.0, 6, 6);
INSERT INTO avaliacao (nota, ator_id, filme_id) VALUES (9.1, 2, 9);
INSERT INTO avaliacao (nota, ator_id, filme_id) VALUES (9.3, 10, 9);
```

```
INSERT INTO avaliacao (nota, ator_id, episodio_id)
SELECT 9.0, 2, e.episodio_id FROM episodio e JOIN temporada t ON t.temporada_id=e.temporada_id
WHERE t.serie_id=4 AND t.numero_temporada=1 AND e.numero_episodio=1;
INSERT INTO avaliacao (nota, ator_id, episodio_id)
SELECT 8.9, 6, e.episodio_id FROM episodio e JOIN temporada t ON t.temporada_id=e.temporada_id
WHERE t.serie_id=4 AND t.numero_temporada=1 AND e.numero_episodio=1;
INSERT INTO avaliacao (nota, ator_id, episodio_id)
SELECT 9.2, 2, e.episodio_id FROM episodio e JOIN temporada t ON t.temporada_id=e.temporada_id
WHERE t.serie_id=5 AND t.numero_temporada=1 AND e.numero_episodio=1;
INSERT INTO avaliacao (nota, ator_id, episodio_id)
SELECT 9.1, 6, e.episodio_id FROM episodio e JOIN temporada t ON t.temporada_id=e.temporada_id
WHERE t.serie_id=5 AND t.numero_temporada=1 AND e.numero_episodio=1;
```

```
INSERT INTO avaliacao (nota, episodio_id)
SELECT 8.0, episodio_id FROM episodio e JOIN temporada t ON t.temporada_id=e.temporada_id WHERE
t.serie_id=4 AND t.numero_temporada=1;
```

```
INSERT INTO avaliacao (nota, episodio_id)
SELECT 8.2, e.episodio_id FROM episodio e JOIN temporada t ON t.temporada_id=e.temporada_id WHERE
t.serie_id=6 AND t.numero_temporada=1 AND e.numero_episodio=1;
```

```
CREATE OR REPLACE VIEW v_series_eur_ator_premiacoes AS
SELECT s.serie_id,
       s.titulo
```

```
FROM serie s
JOIN pais p ON p.pais_id = s.pais_id AND p.continente = 'Europa'
JOIN temporada t ON t.serie_id = s.serie_id
JOIN premiacao pr ON pr.serie_id = s.serie_id AND pr.categoria = 'Ator' AND pr.temporada_id =
t.temporada_id
GROUP BY s.serie_id, s.titulo
HAVING COUNT(DISTINCT t.temporada_id) > 3
      AND COUNT(DISTINCT pr.temporada_id) >= 2;

CREATE OR REPLACE VIEW v_filmes_dir_e_ator_premiacao AS
SELECT f.*
FROM filme f
WHERE EXISTS (SELECT 1 FROM premiacao WHERE filme_id = f.filme_id AND categoria = 'Diretor')
      AND EXISTS (SELECT 1 FROM premiacao WHERE filme_id = f.filme_id AND categoria = 'Ator');

CREATE OR REPLACE VIEW v_series_ao_eu_atrizes_maioria AS
SELECT s.serie_id,
      s.titulo
FROM serie s
JOIN pais p ON p.pais_id = s.pais_id AND p.continente <> 'Europa'
JOIN temporada t ON t.serie_id = s.serie_id
JOIN episodio e ON e.temporada_id = t.temporada_id
JOIN ator_episodio ae ON ae.episodio_id = e.episodio_id
JOIN diretor_episodio de ON de.episodio_id = e.episodio_id
GROUP BY s.serie_id, s.titulo
HAVING SUM(CASE WHEN ae.ator_id IN (SELECT ator_id FROM ator WHERE genero = 'F') THEN 1 ELSE 0 END)
>
      SUM(CASE WHEN ae.ator_id IN (SELECT ator_id FROM ator WHERE genero = 'M') THEN 1 ELSE 0 END)
      AND COUNT(DISTINCT de.diretor_id) > 1;

CREATE OR REPLACE VIEW v_episodios_mulheres_livre_gt3temporada AS
SELECT e.*
FROM episodio e
JOIN temporada t ON t.temporada_id = e.temporada_id
JOIN serie s ON s.serie_id = t.serie_id
JOIN classificacao_idade r ON r.classificacao_id = s.classificacao_id AND r.idade_minima = 0
WHERE EXISTS (SELECT 1 FROM temporada t2 WHERE t2.serie_id = s.serie_id HAVING COUNT(*) > 3)
      AND EXISTS (SELECT 1 FROM diretor_episodio de JOIN diretor d ON d.diretor_id = de.diretor_id AND
d.genero = 'F' WHERE de.episodio_id = e.episodio_id);

CREATE OR REPLACE VIEW v_filmes_pos2003_elenco_series_premiadas AS
SELECT DISTINCT f.*
FROM filme f
JOIN ator_filme af ON af.filme_id = f.filme_id
JOIN ator_episodio ae ON ae.ator_id = af.ator_id
JOIN episodio ep ON ep.episodio_id = ae.episodio_id
JOIN temporada t ON t.temporada_id = ep.temporada_id
```

```
JOIN serie          s  ON s.serie_id = t.serie_id
JOIN premiacao      pr ON pr.serie_id = s.serie_id AND pr.categoria = 'Série'
WHERE f.ano_lancamento > 2003;
```

```
CREATE OR REPLACE VIEW v_ator_seriesantigas_filmesnovos AS
SELECT DISTINCT a.ator_id, a.nome
FROM ator a
JOIN ator_filme    af ON af.ator_id = a.ator_id
JOIN filme         f  ON f.filme_id = af.filme_id AND f.ano_lancamento > 2003
JOIN ator_episodio ae ON ae.ator_id = a.ator_id
JOIN episodio      ep ON ep.episodio_id = ae.episodio_id
JOIN temporada     t  ON t.temporada_id = ep.temporada_id
JOIN serie         s  ON s.serie_id = t.serie_id AND s.ano_lancamento < 2003
WHERE NOT EXISTS (
    SELECT 1 FROM ator_episodio ae2
    JOIN episodio ep2 ON ep2.episodio_id = ae2.episodio_id
    JOIN temporada t2 ON t2.temporada_id = ep2.temporada_id
    JOIN serie s2 ON s2.serie_id = t2.serie_id
    WHERE ae2.ator_id = a.ator_id AND s2.ano_lancamento >= 2003)
AND NOT EXISTS (
    SELECT 1 FROM ator_filme af2
    JOIN filme f2 ON f2.filme_id = af2.filme_id
    WHERE af2.ator_id = a.ator_id AND f2.ano_lancamento <= 2003);
```

```
CREATE OR REPLACE VIEW v_diretor_media_avaliacao AS
SELECT d.diretor_id,
       d.nome,
       'Filme' AS tipo_trabalho,
       AVG(av.nota) AS media_avaliacao
FROM diretor d
JOIN diretor_filme df ON df.diretor_id = d.diretor_id
JOIN filme         f  ON f.filme_id = df.filme_id
JOIN avaliacao     av ON av.filme_id = f.filme_id
GROUP BY d.diretor_id, d.nome
UNION ALL
SELECT d.diretor_id,
       d.nome,
       'Série' AS tipo_trabalho,
       AVG(av.nota) AS media_avaliacao
FROM diretor d
JOIN diretor_episodio de ON de.diretor_id = d.diretor_id
JOIN episodio         e  ON e.episodio_id = de.episodio_id
JOIN avaliacao        av ON av.episodio_id = e.episodio_id
GROUP BY d.diretor_id, d.nome;
```

```
CREATE OR REPLACE VIEW v_titulos_comedia_multi_ator_alta_avaliacao AS
SELECT titulo, 'Filme' AS tipo
FROM (
```

```
SELECT f.titulo, COUNT(*) cnt
FROM filme f
JOIN genero g ON g.genero_id = f.genero_id AND g.nome = 'Comédia'
JOIN ator_filme af ON af.filme_id = f.filme_id
JOIN avaliacao av ON av.ator_id = af.ator_id AND av.filme_id = f.filme_id AND av.nota > 8.5
GROUP BY f.titulo
HAVING COUNT(*) >= 2)

UNION ALL

SELECT titulo, 'Série' AS tipo
FROM (
    SELECT s.titulo, COUNT(DISTINCT ae.ator_id) cnt
    FROM serie s
    JOIN genero g ON g.genero_id = s.genero_id AND g.nome = 'Comédia'
    JOIN temporada t ON t.serie_id = s.serie_id
    JOIN episodio e ON e.temporada_id = t.temporada_id
    JOIN ator_episodio ae ON ae.episodio_id = e.episodio_id
    JOIN avaliacao av ON av.ator_id = ae.ator_id AND av.episodio_id = e.episodio_id AND av.nota > 8.5
    GROUP BY s.titulo
    HAVING COUNT(DISTINCT ae.ator_id) >= 2);

COMMIT;
```

## Queries

1)

```
SELECT titulo
FROM v_series_eur_ator_premiacoes;
```

2)

```
SELECT titulo
FROM v_filmes_dir_e_ator_premiacao;
```

3)

```
SELECT titulo
FROM v_series_nao_eu_atrizes_maioria;
```

4)

```
SELECT p.*
FROM premiacao p
JOIN filme f ON f.filme_id = p.filme_id
JOIN genero g ON g.genero_id = f.genero_id AND g.nome = 'Aventura'
JOIN classificacao_idade ci ON ci.classificacao_id = f.classificacao_id AND ci.idade_minima > 12
WHERE f.ano_lancamento < 2003;
```

5)

```
SELECT DISTINCT p.nome AS nome_personagem
FROM personagem p
JOIN ator_filme af ON af.personagem_id = p.personagem_id
JOIN filme f ON f.filme_id = af.filme_id
JOIN genero g ON g.genero_id = f.genero_id AND g.nome = 'Romance'
JOIN ator a ON a.ator_id = af.ator_id
WHERE NOT EXISTS (SELECT 1 FROM premiacao WHERE ator_id = a.ator_id);
```

6)

```
SELECT titulo
FROM v_episodios_mulheres_livre_gt3temporada
ORDER BY titulo;
```

7)

```
SELECT titulo
FROM v_filmes_pos2003_elenco_series_premiadas;
```

8)

```
SELECT a.nome,  
       f.ano_lancamento AS ano_filme  
FROM   v_ator_seriesantigas_filmesnovos a  
JOIN   ator_filme af ON af.ator_id = a.ator_id  
JOIN   filme f      ON f.filme_id = af.filme_id AND f.ano_lancamento > 2003  
ORDER BY f.ano_lancamento DESC;
```

9)

```
SELECT e.titulo  
FROM   episodio e  
JOIN   temporada t ON t.temporada_id = e.temporada_id  
JOIN   serie s     ON s.serie_id = t.serie_id  
WHERE  e.duracao_min > 45  
GROUP BY e.episodio_id, e.titulo, t.temporada_id  
HAVING COUNT(*) >= 1  
       AND (SELECT COUNT(*) FROM episodio e2 WHERE e2.temporada_id = t.temporada_id AND e2.duracao_min  
> 45) >= 4  
       AND (SELECT COUNT(DISTINCT ator_id) FROM ator_episodio ae WHERE ae.episodio_id = e.episodio_id)  
< 3;
```

10)

```
SELECT f.titulo  
FROM   filme f  
JOIN   diretor_filme df ON df.filme_id = f.filme_id  
JOIN   ator_filme af ON af.filme_id = f.filme_id AND af.ator_id = df.diretor_id  
WHERE  f.resolucao = '4K';
```

11)

```
SELECT DISTINCT f.titulo  
FROM   filme f  
JOIN   ator_filme af ON af.filme_id = f.filme_id  
JOIN   ator a      ON a.ator_id = af.ator_id AND a.nome = (SELECT nome FROM personagem WHERE  
personagem_id = af.personagem_id)  
WHERE  f.duracao_min > 180  
       AND f.ano_lancamento < 2010  
       AND EXISTS (SELECT 1 FROM premiacao WHERE filme_id = f.filme_id OR ator_id = a.ator_id OR  
diretor_id IN (SELECT diretor_id FROM diretor_filme WHERE filme_id = f.filme_id));
```

12)

```
SELECT e.titulo AS titulo_episodio,  
       a.nome   AS nome_ator,  
       av.nota  
FROM   episodio e  
JOIN   avaliacao av      ON av.episodio_id = e.episodio_id  
JOIN   ator_episodio ae  ON ae.episodio_id = e.episodio_id
```

```
JOIN ator a ON a.ator_id = ae.ator_id
JOIN temporada t ON t.temporada_id = e.temporada_id
JOIN serie s ON s.serie_id = t.serie_id
JOIN genero g ON g.genero_id = s.genero_id AND g.nome IN ('Comédia', 'Ação')
WHERE (SELECT AVG(av2.nota) FROM avaliacao av2 JOIN episodio e2 ON e2.episodio_id =
av2.episodio_id WHERE e2.temporada_id = t.temporada_id) > 7.5
AND (SELECT AVG(duracao_min) FROM episodio e3 WHERE e3.temporada_id = t.temporada_id) BETWEEN 45
AND 50
AND (SELECT COUNT(DISTINCT diretor_id) FROM diretor_episodio de WHERE de.episodio_id IN (SELECT
episodio_id FROM episodio WHERE temporada_id = t.temporada_id)) > 1;
```

13)

```
SELECT nome, tipo_trabalho, media_avaliacao
FROM v_diretor_media_avaliacao;
```

14)

```
SELECT titulo, tipo
FROM v_titulos_comedia_multi_ator_alta_avaliacao;
```