**Centro Universitário UNISATC**

Engenharia de Software 3a fase – Banco de Dados II – Prof.Luciano B. Fernandes

**TRABALHO FINAL COM BASE EM METODOLOGIAS ATIVAS DE APRENDIZAGEM**

**Projeto de banco de dados para um sistema de Streaming**

Nome – GitHub

Davi Chechetto – daviChechetto

Igor Antunes – SantunesIgor

Vinicius Lummertz – Vinicius-Lummertz

Criciúma, 09/06/2024

**URL do projeto no GitHub**

https://github.com/daviChechetto/Projeto-final-BD-II.git

**Modelo ER Físico**

**Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.**

**Dicionário de Dados**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Usuários | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados usuários de cada cliente | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_usuario | | Int | 1 – Sem limite | NOT NULL | X |  | Código de identificador do usuário |
| Email | | Varchar(255) | 3 – 255 | NOT NULL |  |  | Nome do usuário |
| senha\_hash | | Varchar(255) | 1 – 255 | NOT NULL |  |  | Senha do da conta do usuário |
| id\_plano | | Int | 1 – Sem limite | NOT NULL |  | X | Identificador referente ao plano do usuário |
| status | | Varchar(50) | 1 – 50 |  |  |  | Armazena a informação se p usuário está ativo |
| data\_criado | | Timestamp | 4 – 8 |  |  |  | Armazena a data de criação do usuário |
| Data\_modicado | | Timestamp | 4 – 8 |  |  |  | Armazena a data da última modificação da conta |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_usuarios\_id\_plano | |  | X |  | Id\_plano | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Perfis | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados dos perfis pertencentes a cada usuário | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| Id\_perfil | | int | 1 – sem limite | NOT NULL | X |  | Código de identificador do perfil |
| Id\_usuario | | Int | 1 – sem limite | NOT NULL |  | X | Código de identificador do usuário |
| nome | | Varchar(100) | 1 – 100 | NOT NULL |  |  | Nome do perfil |
| avatar\_url | | Varchar(255 | 1 – 255 |  |  |  | URL da foto de perfil |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_perfis\_id\_usuario | |  | X |  | cd\_usuario | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Minha\_Lista | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados da lista de filmes do perfil | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| Id\_minha\_lista | | int | 1 – sem limite | NOT NULL | X |  | Código de identificador da minha lista |
| Id\_perfil | | Int | 1 – sem limite | NOT NULL |  | X | Código de identificador do perfil |
| Id\_conteudo | | Int | 1 – sem limite | NOT NULL |  | X | Código de identificador do conteudo |
| data\_adicao | | Timestamp | 4 – 8 |  |  |  | Armazena a data de adição do filme/série à lista |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_minha\_lista\_idperfil | |  | X |  | Id\_perfil | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Conteudo | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados dos conteúdos oferecidos pelo streaming | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_conteudo | | int | 1 – sem limite | NOT NULL | X |  | Código de identificador do conteúdo |
| titulo | | Varchar(255) | 1 – 255 | NOT NULL |  |  | Título do filme ou série |
| descricao | | Text | 1 – sem limite |  |  |  | Descrição sobre o conteúdo |
| ano\_lancamento | | Int | 1 – sem limite |  |  |  | Ano referente à data de lançamento |
| duracao\_min | | Int | 1 – sem limite |  |  |  | Tempo em minutos da duração do conteúdo |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Episodios | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados sobre os episódios | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| Id\_episodio | | Int | 1 – sem limite | NOT NULL | X |  | Código de identificador do episódio |
| Id\_conteudo | | Int | 1 – sem limite | NOT NULL |  | X | Código de identificador do conteúdo |
| titulo | | Varchar(255) | 1 – 255 | NOT NULL |  |  | Título do episódio |
| numero\_temporada | | Int | 1 – sem limite | NOT NULL |  |  | Número da temporada do episódio |
| numero\_episodio | | Int | 1 – sem limite | NOT NULL |  |  | Número do episódio |
| duracao\_min | | Int | 1 – sem limite |  |  |  | Tempo de duração do episódio |
| video\_url | | Varchar(255) | 1 – 255 |  |  |  | URL do vídeo do episódio |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_episodios\_id\_conteudo | |  | X |  | Id\_conteudo | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Planos | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados de cada plano de assinatura | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_plano | | int | 1 – sem limite | NOT NULL | X |  | Código de identificador do plano |
| nome | | Varchar(100) | 1 – 100 | NOT NULL |  |  | Nome do plano de assinatura |
| preco | | Decimal(10,2) | 1 – 10 | NOT NULL |  |  | Valor do plano |
| qualidade\_max | | Varchar(50) | 1 – 50 |  |  |  | Qualidade máxima de resolução permitida |
| telas\_max | | Int | 1 – sem limite | NOT NULL |  |  | Número de máximo de telas simultâneas |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Historicos | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados dos filmes e séries visto | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| Id\_historico | | int | 1 – sem limite | NOT NULL | X |  | Código de identificador do histórico |
| id\_perfil | | Int | 1 – sem limite | NOT NULL |  | X | Código de identificador do perfil |
| Id\_conteudo | | Int | 1 – sem limite | NOT NULL |  | X | Código de identificador do conteúdo |
| Id\_episodio | | Int | 1 – sem limite |  |  | X | Código de identificador do episódio |
| Progresso\_seg | | Int | 1 – sem limite | NOT NULL |  |  | Tempo onde foi parado o conteúdo |
| Data\_visualizacao | | Timestamp | 1 – sem limite |  |  |  | Data da última vez que foi vizualizado |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_historico\_id\_perfil | |  | X |  | Id\_perfil | | |
| Idx\_historicos\_id\_conteudo | |  | X |  | Id\_conteudo | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Generos | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados dos gêneros dos filmes e séries | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| Id\_genero | | int | 1 – sem limite | NOT NULL | X |  | Código de identificador do gênero |
| Nome | | Varchar(100) | 1 – 100 | NOT NULL |  |  | Nome do gênero |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Conteudo\_Genero | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados dos | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| Id\_conteudo | | int | 1 – sem limite | NOT NULL | X |  | Código de identificador do conteudo |
| Id\_genero | | Int | 1 – sem limite | NOT NULL |  | X | Código de identificador do genero |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_conteudo\_genero\_id\_genero | |  | X |  | Id\_genero | | |

**Script dos comandos DDL para criação do Banco de dados**

CREATE TABLE Planos (

id\_plano SERIAL PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

preco NUMERIC(10, 2) NOT NULL,

qualidade\_max VARCHAR(50),

telas\_max INT NOT NULL

);

CREATE TABLE Usuarios (

id\_usuario SERIAL PRIMARY KEY,

email VARCHAR(255) UNIQUE NOT NULL,

senha\_hash VARCHAR(255) NOT NULL, -- Lembre-se de nunca salvar senhas em texto puro!

id\_plano INT NOT NULL,

status VARCHAR(50) DEFAULT 'active',

data\_criado TIMESTAMP WITH TIME ZONE DEFAULT CURRENT\_TIMESTAMP,

data\_modificado TIMESTAMP WITH TIME ZONE,

CONSTRAINT fk\_plano FOREIGN KEY(id\_plano) REFERENCES Planos(id\_plano)

);

CREATE TABLE Perfis (

id\_perfil SERIAL PRIMARY KEY,

id\_usuario INT NOT NULL,

nome VARCHAR(100) NOT NULL,

avatar\_url VARCHAR(255),

CONSTRAINT fk\_usuario FOREIGN KEY(id\_usuario) REFERENCES Usuarios(id\_usuario) ON DELETE CASCADE -- Se o usuário for deletado, seus perfis também são.

);

CREATE TABLE Generos (

id\_genero SERIAL PRIMARY KEY,

nome VARCHAR(100) UNIQUE NOT NULL

);

CREATE TABLE Conteudos (

id\_conteudo SERIAL PRIMARY KEY,

titulo VARCHAR(255) NOT NULL,

descricao TEXT,

ano\_lancamento INT,

duracao\_min INT, -- Para filmes. Nulo para séries.

tipo\_conteudo content\_type\_enum NOT NULL,

thumbnail\_url VARCHAR(255)

);

CREATE TABLE Episodios (

id\_episodio SERIAL PRIMARY KEY,

id\_conteudo INT NOT NULL, -- A série a que o episódio pertence.

titulo VARCHAR(255) NOT NULL,

numero\_temporada INT NOT NULL,

numero\_episodio INT NOT NULL,

duracao\_min INT,

video\_url VARCHAR(255),

CONSTRAINT fk\_conteudo FOREIGN KEY(id\_conteudo) REFERENCES Conteudos(id\_conteudo) ON DELETE CASCADE,

UNIQUE(id\_conteudo, numero\_temporada, numero\_episodio) -- Garante que não haja episódios duplicados na mesma temporada.

);

CREATE TABLE Conteudo\_Genero (

id\_conteudo INT NOT NULL,

id\_genero INT NOT NULL,

CONSTRAINT fk\_conteudo FOREIGN KEY(id\_conteudo) REFERENCES Conteudos(id\_conteudo) ON DELETE CASCADE,

CONSTRAINT fk\_genero FOREIGN KEY(id\_genero) REFERENCES Generos(id\_genero) ON DELETE RESTRICT, -- Evita apagar um gênero que está em uso.

PRIMARY KEY(id\_conteudo, id\_genero) -- Chave primária composta.

);

CREATE TABLE Minha\_Lista (

id\_minha\_lista SERIAL PRIMARY KEY,

id\_perfil INT NOT NULL,

id\_conteudo INT NOT NULL,

data\_adicao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT\_TIMESTAMP,

CONSTRAINT fk\_perfil FOREIGN KEY(id\_perfil) REFERENCES Perfis(id\_perfil) ON DELETE CASCADE,

CONSTRAINT fk\_conteudo FOREIGN KEY(id\_conteudo) REFERENCES Conteudos(id\_conteudo) ON DELETE CASCADE,

UNIQUE(id\_perfil, id\_conteudo) -- Um perfil não pode ter o mesmo item duas vezes na lista.

);

CREATE TABLE Historicos (

id\_historico SERIAL PRIMARY KEY,

id\_perfil INT NOT NULL,

id\_conteudo INT NOT NULL,

id\_episodio INT, -- Nulo se for um filme.

progresso\_seg INT NOT NULL DEFAULT 0,

data\_visualizacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT\_TIMESTAMP,

CONSTRAINT fk\_perfil FOREIGN KEY(id\_perfil) REFERENCES Perfis(id\_perfil) ON DELETE CASCADE,

CONSTRAINT fk\_conteudo FOREIGN KEY(id\_conteudo) REFERENCES Conteudos(id\_conteudo) ON DELETE CASCADE,

CONSTRAINT fk\_episodio FOREIGN KEY(id\_episodio) REFERENCES Episodios(id\_episodio) ON DELETE CASCADE

);

**Script que popula as tabelas do Banco de dados**

INSERT INTO Planos (nome, preco, qualidade\_max, telas\_max) VALUES

('Básico', 25.90, 'HD', 1),

('Padrão', 39.90, 'Full HD', 2),

('Premium', 55.90, '4K+HDR', 4),

('Estudante', 18.90, 'Full HD', 1),

('Família', 65.90, '4K+HDR', 6),

('Mobile', 14.90, 'SD', 1),

('Anual Básico', 259.90, 'HD', 1),

('Anual Padrão', 399.90, 'Full HD', 2),

('Anual Premium', 559.90, '4K+HDR', 4),

('Teste Grátis', 0.00, 'Full HD', 2);

INSERT INTO Generos (nome) VALUES

('Ação'), ('Comédia'), ('Drama'), ('Ficção Científica'), ('Suspense'),

('Documentário'), ('Terror'), ('Fantasia'), ('Animação'), ('Romance');

INSERT INTO Usuarios (email, senha\_hash, id\_plano) VALUES

('carlos.santos@email.com', '$2b$12$hash.ficticio.para.o.usuario.1', 1),

('fernanda.lima@email.com', '$2b$12$hash.ficticio.para.o.usuario.2', 2),

('roberto.gomes@email.com', '$2b$12$hash.ficticio.para.o.usuario.3', 3),

('lucia.pereira@email.com', '$2b$12$hash.ficticio.para.o.usuario.4', 4),

('marcos.almeida@email.com', '$2b$12$hash.ficticio.para.o.usuario.5', 5),

('vanessa.ribeiro@email.com', '$2b$12$hash.ficticio.para.o.usuario.6', 6),

('paulo.mendes@email.com', '$2b$12$hash.ficticio.para.o.usuario.7', 7),

('juliana.barros@email.com', '$2b$12$hash.ficticio.para.o.usuario.8', 8),

('ricardo.nunes@email.com', '$2b$12$hash.ficticio.para.o.usuario.9', 9),

('beatriz.souza@email.com', '$2b$12$hash.ficticio.para.o.usuario.10', 10);

INSERT INTO Perfis (id\_usuario, nome, avatar\_url) VALUES

(1, 'Carlos', 'https://example.com/avatar/carlos.png'),

(2, 'Fê', 'https://example.com/avatar/fe.png'),

(3, 'Roberto', 'https://example.com/avatar/roberto.png'),

(4, 'Lucia', 'https://example.com/avatar/lucia.png'),

(5, 'Marcos', 'https://example.com/avatar/marcos.png'),

(6, 'Nessa', 'https://example.com/avatar/nessa.png'),

(7, 'Paulo', 'https://example.com/avatar/paulo.png'),

(8, 'Juliana', 'https://example.com/avatar/juliana.png'),

(9, 'Rick', 'https://example.com/avatar/rick.png'),

(10, 'Bia', 'https://example.com/avatar/bia.png');

INSERT INTO Conteudos (titulo, descricao, ano\_lancamento, duracao\_min, tipo\_conteudo, thumbnail\_url) VALUES

('O Poderoso Chefão', 'A saga de uma família mafiosa italiana em Nova York.', 1972, 175, 'movie', 'https://example.com/thumb/godfather.jpg'),

('Pulp Fiction', 'Várias histórias de crime se entrelaçam em Los Angeles.', 1994, 154, 'movie', 'https://example.com/thumb/pulpfiction.jpg'),

('Cidade de Deus', 'A vida de dois jovens em uma favela violenta do Rio de Janeiro.', 2002, 130, 'movie', 'https://example.com/thumb/cidadededeus.jpg'),

('O Menino e a Garça', 'Um jovem garoto se aventura em um mundo mágico após a morte de sua mãe.', 2023, 124, 'movie', 'https://example.com/thumb/boyandheron.jpg'),

('Parasita', 'Uma família pobre se infiltra na vida de uma família rica.', 2019, 132, 'movie', 'https://example.com/thumb/parasite.jpg'),

('Clube da Luta', 'Um homem deprimido forma um clube de luta clandestino.', 1999, 139, 'movie', 'https://example.com/thumb/fightclub.jpg'),

('Breaking Bad', 'Um professor de química com câncer terminal começa a produzir metanfetamina.', 2008, NULL, 'series', 'https://example.com/thumb/breakingbad.jpg'),

('Fleabag', 'Uma jovem tenta lidar com a vida em Londres após uma tragédia.', 2016, NULL, 'series', 'https://example.com/thumb/fleabag.jpg'),

('Arcane', 'As origens de duas campeãs icônicas de League of Legends.', 2021, NULL, 'series', 'https://example.com/thumb/arcane.jpg'),

('The Office', 'Uma sátira sobre o cotidiano de um escritório de uma empresa de papel.', 2005, NULL, 'series', 'https://example.com/thumb/theoffice.jpg');

INSERT INTO Episodios (id\_conteudo, titulo, numero\_temporada, numero\_episodio, duracao\_min) VALUES

(7, 'Pilot', 1, 1, 58), -- Breaking Bad S01E01

(7, 'Cat''s in the Bag...', 1, 2, 48), -- Breaking Bad S01E02

(8, 'Episode 1', 1, 1, 27), -- Fleabag S01E01

(8, 'Episode 2', 1, 2, 26), -- Fleabag S01E02

(9, 'Welcome to the Playground', 1, 1, 41), -- Arcane S01E01

(9, 'Some Mysteries Are Better Left Unsolved', 1, 2, 40), -- Arcane S01E02

(9, 'The Base Violence Necessary for Change', 1, 3, 44), -- Arcane S01E03

(10, 'Pilot', 1, 1, 22), -- The Office S01E01

(10, 'Diversity Day', 1, 2, 23), -- The Office S01E02

(10, 'Health Care', 1, 3, 22); -- The Office S01E03

INSERT INTO Conteudo\_Genero (id\_conteudo, id\_genero) VALUES

(1, 3), -- O Poderoso Chefão -> Drama

(2, 5), -- Pulp Fiction -> Suspense

(3, 3), -- Cidade de Deus -> Drama

(4, 9), -- O Menino e a Garça -> Animação

(4, 8), -- O Menino e a Garça -> Fantasia

(5, 5), -- Parasita -> Suspense

(6, 3), -- Clube da Luta -> Drama

(7, 3), -- Breaking Bad -> Drama

(8, 2), -- Fleabag -> Comédia

(9, 1); -- Arcane -> Ação

INSERT INTO Minha\_Lista (id\_perfil, id\_conteudo) VALUES

(1, 10), -- Carlos adicionou The Office

(1, 9), -- Carlos adicionou Arcane

(2, 1), -- Fê adicionou O Poderoso Chefão

(3, 7), -- Roberto adicionou Breaking Bad

(4, 8), -- Lucia adicionou Fleabag

(5, 5), -- Marcos adicionou Parasita

(6, 4), -- Vanessa adicionou O Menino e a Garça

(7, 3), -- Paulo adicionou Cidade de Deus

(8, 2), -- Juliana adicionou Pulp Fiction

(10, 6); -- Bia adicionou Clube da Luta

INSERT INTO Historicos (id\_perfil, id\_conteudo, id\_episodio, progresso\_seg) VALUES

(1, 10, 8, 1320), -- Carlos assistiu ao ep 8 de The Office (completo)

(1, 10, 9, 600), -- Carlos assistiu 10 min do ep 9 de The Office

(2, 1, NULL, 5400),-- Fê assistiu 90 min de O Poderoso Chefão

(3, 7, 1, 3480), -- Roberto assistiu ao ep 1 de Breaking Bad (completo)

(3, 7, 2, 120), -- Roberto começou a ver o ep 2 de Breaking Bad

(4, 8, 3, 1620), -- Lucia assistiu ao ep 3 de Fleabag (completo)

(5, 5, NULL, 7920),-- Marcos assistiu Parasita (completo)

(8, 2, NULL, 300), -- Juliana começou a ver Pulp Fiction

(9, 9, 5, 2460), -- Rick assistiu ao ep 5 de Arcane (completo)

(10, 6, NULL, 4170);-- Bia assistiu metade de Clube da Luta

**Principais consultas mapeadas baseadas em regras de negócio (mínimo 4)**

-- CONSULTAS DE REGRAS DE NEGÓCIO

-- 1. Listar os 5 conteúdos mais populares (mais adicionados à "Minha Lista").

-- Esta consulta ajuda a entender quais títulos estão gerando mais interesse

-- nos usuários, o que pode guiar decisões de marketing e aquisição de conteúdo.

SELECT

    C.titulo,

    COUNT(ML.id\_conteudo) AS total\_adicoes

FROM

    Minha\_Lista AS ML

JOIN

    Conteudos AS C ON ML.id\_conteudo = C.id\_conteudo

GROUP BY

    C.titulo

ORDER BY

    total\_adicoes DESC

LIMIT 5;

-- 2. Exibir o histórico de visualização de um perfil específico, mostrando o progresso.

-- Essencial para a funcionalidade "Continuar Assistindo". A consulta busca tudo

-- que um perfil já viu, mostrando se é filme ou série e o progresso, ordenando

-- pelo mais recente.

-- (Neste exemplo, usamos o id\_perfil = 1, que pertence ao 'Carlos').

SELECT

    P.nome AS nome\_perfil,

    C.titulo AS titulo\_conteudo,

    -- Concatena os detalhes do episódio, se houver

    COALESCE('Temporada ' || E.numero\_temporada || ', Episódio ' || E.numero\_episodio || ': ' || E.titulo, 'Filme') AS detalhe,

    H.progresso\_seg AS segundos\_assistidos,

    -- Calcula a duração total em segundos para comparar com o progresso

    COALESCE(E.duracao\_min \* 60, C.duracao\_min \* 60) AS duracao\_total\_seg,

    H.data\_visualizacao

FROM

    Historicos AS H

JOIN

    Perfis AS P ON H.id\_perfil = P.id\_perfil

JOIN

    Conteudos AS C ON H.id\_conteudo = C.id\_conteudo

LEFT JOIN

    Episodios AS E ON H.id\_episodio = E.id\_episodio

WHERE

    H.id\_perfil = 1 -- Filtro para um perfil específico

ORDER BY

    H.data\_visualizacao DESC;

-- 3. Identificar usuários que não terminaram de assistir a um conteúdo (filme ou episódio).

-- Esta informação é valiosa para campanhas de reengajamento, como enviar um e-mail

-- ou notificação "Termine de assistir!". A consulta verifica se o progresso é menor

-- que 95% da duração total.

SELECT

    U.email,

    P.nome AS nome\_perfil,

    C.titulo AS titulo\_incompleto

FROM

    Historicos AS H

JOIN

    Perfis AS P ON H.id\_perfil = P.id\_perfil

JOIN

    Usuarios AS U ON P.id\_usuario = U.id\_usuario

JOIN

    Conteudos AS C ON H.id\_conteudo = C.id\_conteudo

LEFT JOIN

    Episodios AS E ON H.id\_episodio = E.id\_episodio

WHERE

    -- O progresso é maior que zero mas menor que 95% do total

    H.progresso\_seg > 0 AND

    H.progresso\_seg < (COALESCE(E.duracao\_min, C.duracao\_min) \* 60 \* 0.95)

GROUP BY -- Agrupado para não enviar múltiplas notificações para o mesmo usuário/conteúdo

    U.email, P.nome, C.titulo;

-- 4. Contar quantos usuários estão em cada plano e calcular a receita mensal recorrente (MRR) por plano.

-- Esta é uma métrica financeira fundamental para qualquer serviço de assinatura.

-- Ajuda a avaliar a popularidade dos planos e a prever a receita.

SELECT

    PL.nome AS nome\_do\_plano,

    COUNT(U.id\_usuario) AS total\_de\_usuarios,

    SUM(PL.preco) AS receita\_mensal\_estimada

FROM

    Usuarios AS U

JOIN

    Planos AS PL ON U.id\_plano = PL.id\_plano

WHERE

    PL.nome NOT LIKE 'Anual%' AND PL.nome != 'Teste Grátis' -- Exclui planos anuais e de teste para um cálculo de MRR mais preciso

GROUP BY

    PL.nome

ORDER BY

    receita\_mensal\_estimada DESC;