

FACULDADE ÁREA1
CIÊNCIA DA COMPUTAÇÃO

KAIQUE DIAS PEREIRA
BRUNO SOUSA SILVA
CAUÃ GUILHERME DOS SANTOS
PAULO HENRIQUE LEAL DOS SANTOS
YURI DE JESUS FERNANDES MENDES
DENILSON SANTOS DE JESUS
PEDRO SARMENTO RANGEL SANTOS

PROJETO DE BANCO DE DADOS BASEADO EM UM SERVIÇO DE STREAMING

SALVADOR/BA
2023

KAIQUE DIAS PEREIRA
BRUNO SOUSA SILVA
CAUÃ GUILHERME DOS SANTOS
PAULO HENRIQUE LEAL DOS SANTOS
YURI DE JESUS FERNANDES MENDES
DENILSON SANTOS DE JESUS
PEDRO SARMENTO RANGEL SANTOS

PROJETO DE BANCO DE DADOS BASEADO EM UM SERVIÇO DE STREAMING

Projeto de Banco de Dados baseado em
um Serviço de Streaming apresentado ao
Curso de Ciência da Computação da
Faculdade Área1

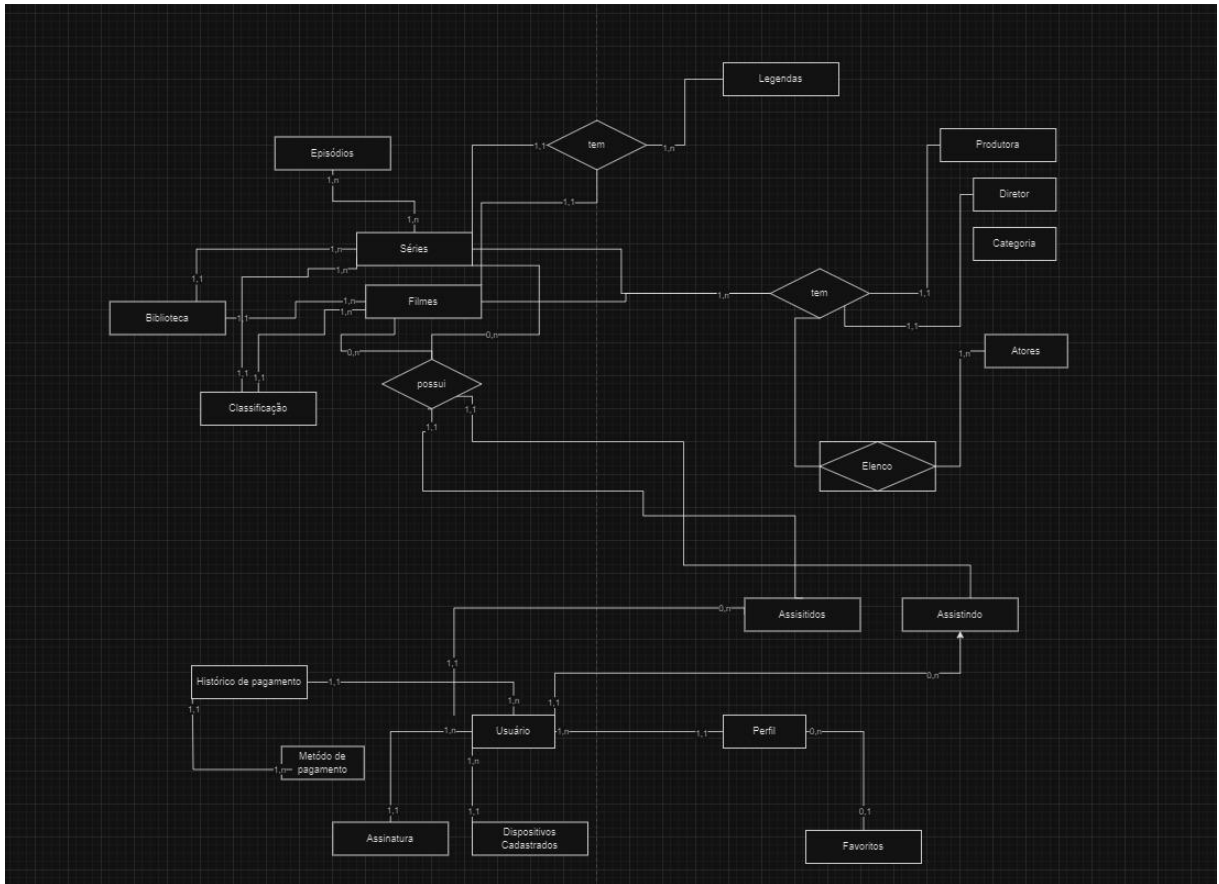
SALVADOR/BA

2023

1. Introdução

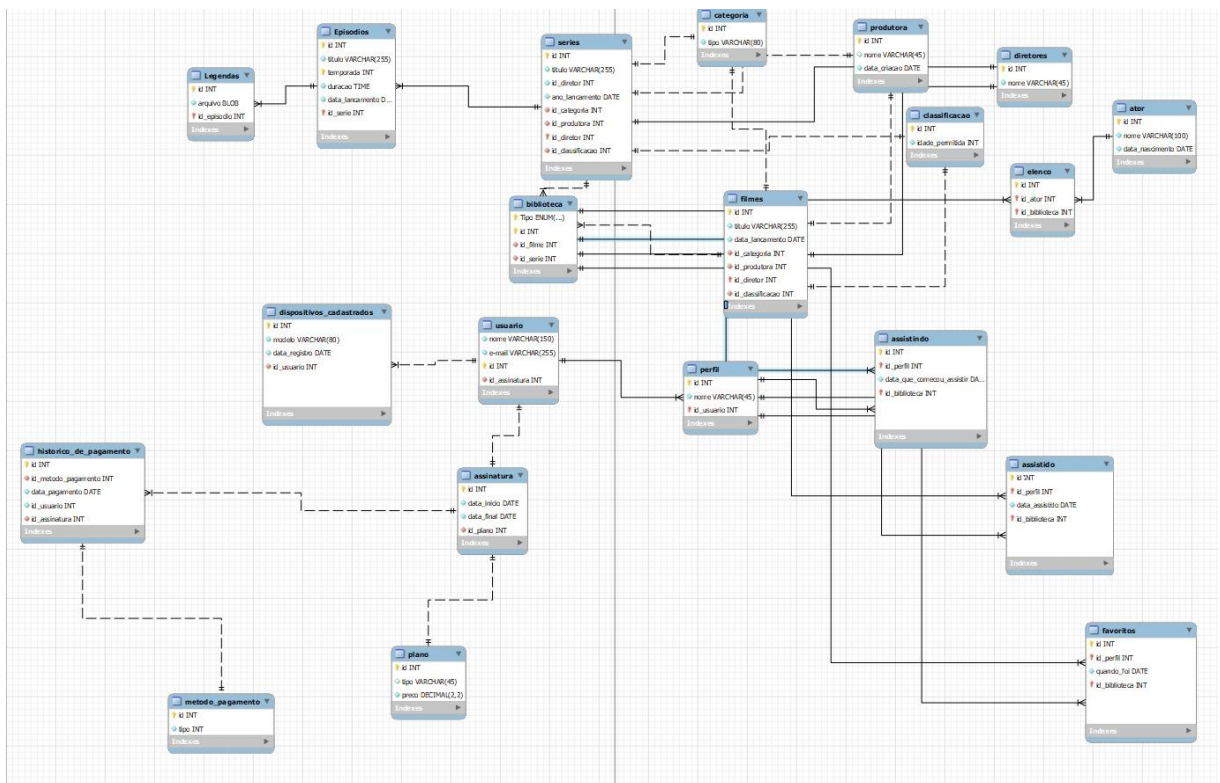
Nosso objetivo é representar de forma conceitual, lógica e física um banco de dados de um serviço de streaming de filmes e séries, prezando o aprendizado do fluxo lógico de como é mantido e desenvolvido um sistema tão robusto e complexo de um serviço de streaming, onde nós esperamos aprender técnicas de consulta, otimização e eficiência dos dados. Para melhor produtividade do nosso grupo, nós nos dividimos em pares para realizar as questões solicitadas do trabalho e manter o nível de qualidade esperado. Focando em entender o funcionamento de um serviço de streaming e identificando quem são as entidades participantes do sistema e identificando seus atributos, tendo a compreensão de como cada uma desempenha seu papel no sistema.

2. Modelo Conceitual



O Modelo de Entidade-Relacionamento (MER) apresentado abaixo descreve a estrutura de dados e os relacionamentos que compõem o Wyden-flix. Este sistema foi pensado desde o início para ser algo voltado ao produto e não a empresa de tal forma a que o usuário é totalmente livre assinar o nosso serviço, para ter diversos perfis e acessá-los em vários dispositivos assim tendo toda liberdade e conforto para assistir seus filmes/series do seu diretor ou produtora preferidos e até mesmo criar sua própria biblioteca de conteúdos favoritos.

3. Modelo Lógico



Usuário: É figura principal para o funcionamento autenticação e personalização do sistema, o mesmo possui como atributos uma id única para identificação, endereço de e-mail, a id da assinatura para identificar se é usuário pagante ou não e um nome.

Assinatura: É o principal método de validação de acesso ao aplicativo por parte do usuário, ela se relaciona também diretamente ao histórico de pagamento e o plano escolhido pelo usuário tem como atributos uma id para identificação da assinatura uma data de início e término.

Dispositivos cadastrados: É a forma pela qual é possível ver a quantidade de aparelhos que estão vinculados a uma assinatura.

Perfil: Onde é possível ver as informações do usuário, como nome e a identificação dele.

Biblioteca: É onde serão armazenados os filmes assistidos, favoritos e em reprodução.

Filmes: Onde será exibido as opções a serem assistidas contendo categorias,elenco, classificação indicativa e diretor.

Séries: Onde será exibido as opções a serem assistidas contendo categorias,elenco, classificação indicativa e diretor.

4. Modelo físico

-- MySQL Script generated by MySQL Workbench

-- Wed Nov 8 21:36:11 2023

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_
DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBST
ITUTION';

-- -----

-- Schema wydenflix

-- -----

-- -----

-- Schema wydenflix

-- -----

CREATE SCHEMA IF NOT EXISTS `wydenflix` DEFAULT CHARACTER SET utf8 ;
USE `wydenflix` ;

-- -----

-- Table `wydenflix`.`plano`

-- -----

CREATE TABLE IF NOT EXISTS `plano` (
 `id` INT NOT NULL,
 `tipo` VARCHAR(45) NULL UNIQUE,
 `preco` DECIMAL(5,2) NOT NULL,
 PRIMARY KEY (`id`)
)

```
ENGINE = InnoDB;
```

```
-- -----
```

```
-- Table `wydenflix`.`assinatura`
```

```
-- -----
```

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`assinatura` (  
    `id` INT NOT NULL,  
    `data_inicio` DATE NOT NULL,  
    `data_final` DATE NOT NULL,  
    `id_plano` INT NOT NULL,  
    PRIMARY KEY (`id`),  
    CONSTRAINT `fk_assinatura_plano1`  
    FOREIGN KEY (`id_plano`)  
    REFERENCES `wydenflix`.`plano` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----
```

```
-- Table `wydenflix`.`conta`
```

```
-- -----
```

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`conta` (  
    `id` INT NOT NULL,  
    `email` VARCHAR(255) NOT NULL UNIQUE,  
    `senha` VARCHAR(20) NOT NULL,  
    PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-- -----
```

```
-- Table `wydenflix`.`usuario`
```

```
-- -----
```

```

CREATE TABLE IF NOT EXISTS `wydenflix`.`usuario` (
  `nome` VARCHAR(150) NOT NULL,
  `email` VARCHAR(255) NOT NULL UNIQUE,
  `id` INT NOT NULL,
  `id_assinatura` INT NOT NULL,
  `conta_id` INT NOT NULL,
  PRIMARY KEY (`id`, `conta_id`),
  CONSTRAINT `fk_tb_usuario_Assinatura1`
    FOREIGN KEY (`id_assinatura`)
      REFERENCES `wydenflix`.`assinatura` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_usuario_conta1`
    FOREIGN KEY (`email`)
      REFERENCES `wydenflix`.`conta` (`email`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `wydenflix`.`perfil`
-----

```

```

CREATE TABLE IF NOT EXISTS `wydenflix`.`perfil` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  `id_usuario` INT NOT NULL,
  PRIMARY KEY (`id`, `id_usuario`),
  CONSTRAINT `fk_tb_perfil_tb_usuario1`
    FOREIGN KEY (`id_usuario`)
      REFERENCES `wydenflix`.`usuario` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```



```

-----
-- Table `wydenflix`.`dispositivos_cadastrados`
-----

CREATE TABLE IF NOT EXISTS `wydenflix`.`dispositivos_cadastrados` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `modelo` VARCHAR(80) NOT NULL,
  `data_registro` DATE NOT NULL,
  `id_usuario` INT NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `fk_Dispositivos_Cadastrados_tb_usuario1`
  FOREIGN KEY (`id_usuario`)
  REFERENCES `wydenflix`.`usuario` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `wydenflix`.`metodo_pagamento`
-----

CREATE TABLE IF NOT EXISTS `wydenflix`.`metodo_pagamento` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `tipo` INT NOT NULL UNIQUE,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

```

```

-----
-- Table `wydenflix`.`categoria`
-----

CREATE TABLE IF NOT EXISTS `wydenflix`.`categoria` (
  `id` INT NOT NULL AUTO_INCREMENT,

```

```
`tipo` VARCHAR(80) NOT NULL UNIQUE,  
PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `wydenflix`.`produtora`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`produtora` (  
  `id` INT NOT NULL,  
  `nome` VARCHAR(45) NULL,  
  `data_criacao` DATE NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `wydenflix`.`diretores`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`diretores` (  
  `id` INT NOT NULL,  
  `nome` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `wydenflix`.`classificacao`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`classificacao` (  
  `id` INT NOT NULL,  
  `idade_permitida` INT NOT NULL UNIQUE,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

-- Table `wydenflix`.`filmes`

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`filmes` (  
  `id` INT NOT NULL,  
  `titulo` VARCHAR(255) NOT NULL,  
  `data_lancamento` DATE NOT NULL,  
  `id_categoria` INT NOT NULL,  
  `id_produtores` INT NOT NULL,  
  `id_diretor` INT NOT NULL,  
  `id_classificacao` INT NOT NULL,  
  PRIMARY KEY (`id`, `id_diretor`),  
  CONSTRAINT `fk_filmes_Categoria1`  
    FOREIGN KEY (`id_categoria`)  
    REFERENCES `wydenflix`.`categoria` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_filmes_produtores1`  
    FOREIGN KEY (`id_produtores`)  
    REFERENCES `wydenflix`.`produtores` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_filmes_diretores1`  
    FOREIGN KEY (`id_diretor`)  
    REFERENCES `wydenflix`.`diretores` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_filmes_classificacao1`  
    FOREIGN KEY (`id_classificacao`)  
    REFERENCES `wydenflix`.`classificacao` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

ENGINE = InnoDB;

```
-----  
-- Table `wydenflix`.`series`  
-----  
  
CREATE TABLE IF NOT EXISTS `wydenflix`.`series` (  
  `id` INT NOT NULL,  
  `titulo` VARCHAR(255) NOT NULL,  
  `ano_lancamento` DATE NOT NULL,  
  `id_categoria` INT NOT NULL,  
  `id_produto` INT NOT NULL,  
  `id_diretor` INT NOT NULL,  
  `id_classificacao` INT NOT NULL,  
  PRIMARY KEY (`id`, `id_diretor`),  
  CONSTRAINT `fk_Series_Categoria1`  
    FOREIGN KEY (`id_categoria`)  
      REFERENCES `wydenflix`.`categoria` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_series_produto1`  
    FOREIGN KEY (`id_produto`)  
      REFERENCES `wydenflix`.`produto` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_series_diretores1`  
    FOREIGN KEY (`id_diretor`)  
      REFERENCES `wydenflix`.`diretores` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_series_classificacao1`  
    FOREIGN KEY (`id_classificacao`)  
      REFERENCES `wydenflix`.`classificacao` (`id`)  
      ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `wydenflix`.`biblioteca`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`biblioteca` (
  `Tipo` ENUM("SERIE", "FILME") NOT NULL,
  `id` INT NOT NULL AUTO_INCREMENT,
  `id_filme` INT NULL UNIQUE,
  `id_serie` INT NULL UNIQUE,
  PRIMARY KEY (`id`, `Tipo`),
  CONSTRAINT `fk_Biblioteca_Filmes1`
    FOREIGN KEY (`id_filme`)
    REFERENCES `wydenflix`.`filmes` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Biblioteca_Series1`
    FOREIGN KEY (`id_serie`)
    REFERENCES `wydenflix`.`series` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `wydenflix`.`assistindo`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`assistindo` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `id_perfil` INT NOT NULL,
  `data_que_comecou_assistir` DATE NOT NULL,
  `id_biblioteca` INT NOT NULL,
```

```
PRIMARY KEY (`id`, `id_biblioteca`, `id_perfil`),  
CONSTRAINT `fk_Assistindo_tb_perfil1`  
FOREIGN KEY (`id_perfil`)  
REFERENCES `wydenflix`.`perfil` (`id`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT `fk_Assistindo_Biblioteca1`  
FOREIGN KEY (`id_biblioteca`)  
REFERENCES `wydenflix`.`biblioteca` (`id`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `wydenflix`.`Episodios`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`Episodios` (  
  `id` INT NOT NULL,  
  `titulo` VARCHAR(255) NOT NULL,  
  `temporada` INT NOT NULL,  
  `duracao` TIME NOT NULL,  
  `data_lancamento` DATE NOT NULL,  
  `id_serie` INT NOT NULL,  
  PRIMARY KEY (`id`, `id_serie`, `temporada`),  
  CONSTRAINT `fk_Episodios_Series1`  
  FOREIGN KEY (`id_serie`)  
  REFERENCES `wydenflix`.`series` (`id`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `wydenflix`.`ator`

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`ator` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `data_nascimento` DATE NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

-- Table `wydenflix`.`elenco`

```
CREATE TABLE IF NOT EXISTS `wydenflix`.`elenco` (  
  `id` INT NOT NULL,  
  `id_ator` INT NOT NULL,  
  `id_biblioteca` INT NOT NULL,  
  PRIMARY KEY (`id`, `id_ator`, `id_biblioteca`),  
  CONSTRAINT `fk_Elenco_Ator1`  
    FOREIGN KEY (`id_ator`)  
      REFERENCES `wydenflix`.`ator` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Elenco_Biblioteca1`  
    FOREIGN KEY (`id_biblioteca`)  
      REFERENCES `wydenflix`.`biblioteca` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `wydenflix`.`historico_de_pagamento`

```

CREATE TABLE IF NOT EXISTS `wydenflix`.`historico_de_pagamento` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `id_metodo_pagamento` INT NOT NULL,
  `data_pagamento` DATE NOT NULL,
  `id_usuario` INT NOT NULL,
  `idassinatura` INT NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `fk_Historico de Pagamento_Metodo de Pagamento1`
    FOREIGN KEY (`id_metodo_pagamento`)
    REFERENCES `wydenflix`.`metodo_pagamento` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Historico de Pagamento_Assinatura1`
    FOREIGN KEY (`idassinatura`)
    REFERENCES `wydenflix`.`assinatura` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

-- -----

-- Table `wydenflix`.`Legendas`

-- -----

```

CREATE TABLE IF NOT EXISTS `wydenflix`.`Legendas` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `arquivo` BLOB NOT NULL UNIQUE,
  `id_episodio` INT NOT NULL,
  PRIMARY KEY (`id`, `id_episodio`),
  CONSTRAINT `fk_Legendas_Episodios1`
    FOREIGN KEY (`id_episodio`)
    REFERENCES `wydenflix`.`Episodios` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```



```

-----
-- Table `wydenflix`.`assistido`
-----

CREATE TABLE IF NOT EXISTS `wydenflix`.`assistido` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `id_perfil` INT NOT NULL,
  `data_assistido` DATE NOT NULL,
  `id_biblioteca` INT NOT NULL,
  PRIMARY KEY (`id`, `id_perfil`, `id_biblioteca`),
  CONSTRAINT `fk_Assistido_tb_perfil1`
    FOREIGN KEY (`id_perfil`)
      REFERENCES `wydenflix`.`perfil` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Assistido_Biblioteca1`
    FOREIGN KEY (`id_biblioteca`)
      REFERENCES `wydenflix`.`biblioteca` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `wydenflix`.`favoritos`
-----

CREATE TABLE IF NOT EXISTS `wydenflix`.`favoritos` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `id_perfil` INT NOT NULL,
  `quando_foi` DATE NOT NULL,
  `id_biblioteca` INT NOT NULL,
  PRIMARY KEY (`id`, `id_perfil`, `id_biblioteca`),
  CONSTRAINT `fk_Favoritos_tb_perfil1`

```

```

FOREIGN KEY (`id_perfil`)
REFERENCES `wydenflix`.`perfil` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Favoritos_Biblioteca1`
FOREIGN KEY (`id_biblioteca`)
REFERENCES `wydenflix`.`biblioteca` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

ALTER TABLE metodo_pagamento
MODIFY COLUMN tipo VARCHAR(100) NOT NULL;

```

```

ALTER TABLE elenco
MODIFY COLUMN id INT AUTO_INCREMENT;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

5

```

INSERT INTO wydenflix.conta (id, email, senha) VALUES
(1, 'usuario1@example.com', 'senha123'),
(2, 'usuario2@example.com', 'senha456'),
(3, 'usuario3@example.com', 'senha789'),
(4, 'usuario4@example.com', 'senha123'),
(5, 'usuario5@example.com', 'senha789');

```

id	email	senha
1	usuario1@example.com	senha123
2	usuario2@example.com	senha456
3	usuario3@example.com	senha789
4	usuario4@example.com	senha123

```

INSERT INTO wydenflix.metodo_pagamento (tipo) VALUES ('Cartão de Crédito');
INSERT INTO wydenflix.metodo_pagamento (tipo) VALUES ('Cartão de Débito');
INSERT INTO wydenflix.metodo_pagamento (tipo) VALUES ('PIX');
INSERT INTO wydenflix.metodo_pagamento (tipo) VALUES ('Boleto');
INSERT INTO wydenflix.metodo_pagamento (tipo) VALUES ('TED');

```

id	tipo
6	Boleto
3	Cartão de Crédito
4	Cartão de Débito
5	PIX
7	TED
NULL	NULL

```

INSERT INTO wydenflix.plano (id, tipo, preco) VALUES
(1, 'Plano Básico', 9.99),
(2, 'Plano Padrão', 14.99),
(3, 'Plano Premium', 19.99),
(4, 'Plano Promocional', 17.99),
(5, 'Plano Gratuito', 0);

```

id	tipo	preco
1	Plano Básico	9.99
2	Plano Padrão	14.99
3	Plano Premium	19.99
4	Plano Promocional	17.99
5	Plano Gratuito	0.00

```

INSERT INTO wydenflix.assinatura (id, data_inicio, data_final, id_plano) VALUES
(1, '2023-01-01', '2023-12-31', 1),
(2, '2023-01-15', '2023-12-31', 2),
(3, '2023-02-01', '2023-12-31', 3),
(4, '2023-02-01', '2023-12-31', 5),
(5, '2023-02-01', '2023-12-31', 3);

```

id	data_inicio	data_final	id_plano
1	2023-01-01	2023-12-31	1
2	2023-01-15	2023-12-31	2
3	2023-02-01	2023-12-31	3
4	2023-02-01	2023-12-31	5
5	2023-02-01	2023-12-31	3

```
INSERT INTO wydenflix.usuario (nome, email, id, idassinatura, conta_id) VALUES
('João Silva', 'usuario1@example.com', 1, 1, 1),
('Maria Souza', 'usuario2@example.com', 2, 2, 2),
('Carlos Oliveira', 'usuario3@example.com', 3, 3, 3),
('Emilia faiz', 'usuario4@example.com', 4, 1, 4),
('Tanjiro Yakumata', 'usuario5@example.com', 5, 5, 5);
```

João Silva	usuario1@example.com	1	1	1
Maria Souza	usuario2@example.com	2	2	2
Carlos Oliveira	usuario3@example.com	3	3	3
Emilia faiz	usuario4@example.com	4	1	4
Tanjiro Yakumata	usuario5@example.com	5	5	5

```
INSERT INTO `wydenflix`.`historico_de_pagamento` (`id_metodo_pagamento`,
`data_pagamento`, `id_usuario`, `idassinatura`) VALUES
(6, '2023-10-15', 1, 1),
(7, '2023-11-20', 2, 2),
(7, '2023-12-25', 3, 3),
(6, '2024-01-10', 1, 1),
(6, '2024-02-12', 2, 2),
(3, '2024-03-18', 3, 3),
(3, '2024-04-05', 1, 1),
(3, '2024-05-22', 2, 2),
(4, '2024-06-30', 3, 3),
(4, '2024-07-15', 1, 1),
(4, '2024-08-20', 2, 2),
(5, '2024-09-25', 3, 3),
(5, '2024-10-10', 1, 1),
(5, '2024-11-12', 2, 2),
(7, '2024-12-18', 3, 3),
(7, '2025-01-05', 1, 1),
```

(7, '2025-02-22', 2, 2),
 (3, '2025-03-30', 3, 3),
 (3, '2025-04-15', 1, 1),
 (3, '2025-05-20', 2, 2);

id	id_metodo_pagamento	data_pagamento	id_usuario	id_assinatura
21	6	2023-10-15	1	1
22	7	2023-11-20	2	2
23	7	2023-12-25	3	3
24	6	2024-01-10	1	1
25	6	2024-02-12	2	2
26	3	2024-03-18	3	3
27	3	2024-04-05	1	1
28	3	2024-05-22	2	2
29	4	2024-06-30	3	3
30	4	2024-07-15	1	1
31	4	2024-08-20	2	2
32	5	2024-09-25	3	3
33	5	2024-10-10	1	1
34	5	2024-11-12	2	2
35	7	2024-12-18	3	3
36	7	2025-01-05	1	1
37	7	2025-02-22	2	2
38	3	2025-03-30	3	3
39	3	2025-04-15	1	1
40	3	2025-05-20	2	2

INSERT INTO wydenflix.dispositivos_cadastrados (id, modelo, data_registro, id_usuario) VALUES

(1, 'Smart TV Samsung', '2023-01-02', 1),
 (2, 'Tablet Apple', '2023-01-20', 2),
 (3, 'Console Xbox', '2023-02-10', 3),
 (4, 'PS\$', '2023-02-11', 3),
 (5, 'Computador AlienWare', '2023-08-10', 1);

id	modelo	data_registro	id_usuario
1	Smart TV Samsung	2023-01-02	1
2	Tablet Apple	2023-01-20	2
3	Console Xbox	2023-02-10	3
4	PS\$	2023-02-11	3
5	Computador AlienWare	2023-08-10	1

```
INSERT INTO wydenflix.perfil (nome, id_usuario) VALUES
('Perfil Principal', 1),
('Perfil Infantil', 1),
('Perfil Principal', 2),
('Perfil de Convidado', 3),
('Misca Musca', 3);
```

id	nome	id_usuario
1	Perfil Principal	1
2	Perfil Infantil	1
3	Perfil Principal	2
4	Perfil de Convidado	3
5	Misca Musca	3

```
INSERT INTO wydenflix.diretores (id, nome) VALUES (1, 'Christopher Nolan');
INSERT INTO wydenflix.diretores (id, nome) VALUES (2, 'Quentin Tarantino');
INSERT INTO wydenflix.diretores (id, nome) VALUES (3, 'Steven Spielberg');
INSERT INTO wydenflix.diretores (id, nome) VALUES (4, 'Denzel Washigton');
INSERT INTO wydenflix.diretores (id, nome) VALUES (5, 'Morgan Freeman');
```

id	nome
1	Christopher Nolan
2	Quentin Tarantino
3	Steven Spielberg
4	Denzel Washigton
5	Morgan Freeman

```
INSERT INTO wydenflix.classificacao (id, idade_permitida) VALUES
(1, 6),
(2, 10),
(3, 14),
(4, 16),
(5, 18);
```

id	idade_permitida
1	6
2	10
3	14
4	16
5	18

```
INSERT INTO wydenflix.categoria (tipo) VALUES ('Ação');
```

```

INSERT INTO wydenflix.categoria (tipo) VALUES ('Comédia');
INSERT INTO wydenflix.categoria (tipo) VALUES ('Drama');
INSERT INTO wydenflix.categoria (tipo) VALUES ('Suspense');
INSERT INTO wydenflix.categoria (tipo) VALUES ('Romance');

```

id	tipo
1	Ação
2	Comédia
3	Drama
5	Romance
4	Suspense

```

INSERT INTO wydenflix.produtora (id, nome, data_criacao) VALUES (1, 'Warner Bros', '1923-04-04');
INSERT INTO wydenflix.produtora (id, nome, data_criacao) VALUES (2, 'Paramount Pictures', '1912-05-08');
INSERT INTO wydenflix.produtora (id, nome, data_criacao) VALUES (3, 'The Walt Disney Company', '1923-10-16');
INSERT INTO wydenflix.produtora (id, nome, data_criacao) VALUES (4, '21 Century FOX', '1919-12-16');
INSERT INTO wydenflix.produtora (id, nome, data_criacao) VALUES (5, 'Netflix Production', '2005-08-27');

```

id	nome	data_criacao
1	Warner Bros	1923-04-04
2	Paramount Pictures	1912-05-08
3	The Walt Disney Company	1923-10-16
4	21 Century FOX	1919-12-16
5	Netflix Production	2005-08-27

```

INSERT INTO wydenflix.series (id, titulo, ano_lancamento, id_categoria, id_produtora, id_diretor, id_classificacao) VALUES
(2, 'Stranger Things', '2016-07-15', 1, 1, 2, 2),
(3, 'The Crown', '2016-11-04', 1, 2, 3, 1),
(4, 'Black Mirror', '2011-12-04', 2, 1, 4, 3),
(5, 'Narcos', '2015-08-28', 1, 3, 5, 4);

```

```
INSERT INTO wydenflix.series (id, titulo, ano_lancamento, id_categoria,
id_produto, id_diretor, id_classificacao) VALUES
(6, 'The Witcher', '2019-12-20', 1, 1, 4, 3),
(7, 'Money Heist', '2017-05-02', 2, 2, 4, 4),
(8, 'The Mandalorian', '2019-11-12', 1, 3, 2, 2),
(9, 'Breaking Bad', '2008-01-20', 1, 1, 2, 5);
```

```
INSERT INTO wydenflix.series (id, titulo, ano_lancamento, id_categoria,
id_produto, id_diretor, id_classificacao) VALUES
(10, 'The Office', '2005-03-24', 2, 2, 4, 2),
(11, 'Stranger Things 2', '2017-10-27', 1, 1, 2, 2),
(12, 'Westworld', '2016-10-02', 1, 3, 1, 4),
(13, 'Friends', '1994-09-22', 2, 2, 5, 2);
```

id	titulo	ano_lancamento	id_categoria	id_produto	id_diretor	id_classificacao
2	Stranger Things	2016-07-15	1	1	2	2
3	The Crown	2016-11-04	1	2	3	1
4	Black Mirror	2011-12-04	2	1	4	3
5	Narcos	2015-08-28	1	3	5	4
6	The Witcher	2019-12-20	1	1	4	3
7	Money Heist	2017-05-02	2	2	4	4
8	The Mandalorian	2019-11-12	1	3	2	2
9	Breaking Bad	2008-01-20	1	1	2	5
10	The Office	2005-03-24	2	2	4	2
11	Stranger Things 2	2017-10-27	1	1	2	2
12	Westworld	2016-10-02	1	3	1	4
13	Friends	1994-09-22	2	2	5	2

```
INSERT INTO wydenflix.Episodios (id, titulo, temporada, duracao, data_lancamento,
id_serie) VALUES
(1, 'Chapter One: The Vanishing of Will Byers', 1, '00:50:00', '2016-07-15', 2),
(2, 'Chapter Two: The Weirdo on Maple Street', 1, '00:45:00', '2016-07-15', 2),
(3, 'Chapter One: Wolferton Splash', 1, '00:55:00', '2016-11-04', 3),
(4, 'Chapter Two: Hyde Park Corner', 1, '01:00:00', '2016-11-04', 3),
(5, 'The National Anthem', 1, '00:44:00', '2011-12-04', 4);
```



```
INSERT INTO wydenflix.Episodios (id, titulo, temporada, duracao, data_lancamento, id_serie) VALUES
```

```
(6, 'Fifteen Million Merits', 1, '00:49:00', '2011-12-04', 4),
(7, 'Nosedive', 3, '00:58:00', '2016-10-21', 4),
(8, 'USS Callister', 4, '00:51:00', '2017-12-29', 4),
(9, 'The Original', 1, '00:58:00', '2016-10-02', 12);
```

```
INSERT INTO wydenflix.Episodios (id, titulo, temporada, duracao, data_lancamento, id_serie) VALUES
```

```
(10, 'Chapter One: The Mandalorian', 1, '00:40:00', '2019-11-12', 8),
(11, 'Breaking Bad Pilot', 1, '00:58:00', '2008-01-20', 9),
(12, 'Dinner Party', 4, '00:22:00', '2009-04-10', 10),
(13, 'The One Where Everybody Finds Out', 5, '00:22:00', '1999-05-11', 13);
```

id	titulo	temporada	duracao	data_lancamento	id_serie
1	Chapter One: The Vanishing of Will Byers	1	00:50:00	2016-07-15	2
2	Chapter Two: The Weirdo on Maple Street	1	00:45:00	2016-07-15	2
3	Chapter One: Wolferton Splash	1	00:55:00	2016-11-04	3
4	Chapter Two: Hyde Park Corner	1	01:00:00	2016-11-04	3
5	The National Anthem	1	00:44:00	2011-12-04	4
6	Fifteen Million Merits	1	00:49:00	2011-12-04	4
7	Nosedive	3	00:58:00	2016-10-21	4
8	USS Callister	4	00:51:00	2017-12-29	4
9	The Original	1	00:58:00	2016-10-02	12
10	Chapter One: The Mandalorian	1	00:40:00	2019-11-12	8
11	Breaking Bad Pilot	1	00:58:00	2008-01-20	9
12	Dinner Party	4	00:22:00	2009-04-10	10
13	The One Where Everybody Finds Out	5	00:22:00	1999-05-11	13

```
INSERT INTO wydenflix.Legendas (arquivo, id_episodio) VALUES
```

```
('Legenda_Episodio1.srt', 1),
('Legenda_Episodio2.srt', 2),
('Legenda_Episodio3.srt', 3),
('Legenda_Episodio4.srt', 4),
('Legenda_Episodio5.srt', 5);
```

```
INSERT INTO wydenflix.Legendas (arquivo, id_episodio) VALUES
('Legenda_Episodio6.srt', 6),
('Legenda_Episodio7.srt', 7),
('Legenda_Episodio8.srt', 8),
('Legenda_Episodio9.srt', 9),
('Legenda_Episodio10.srt', 10);
```

```
INSERT INTO wydenflix.Legendas (arquivo, id_episodio) VALUES
('Legenda_Episodio11.srt', 11),
('Legenda_Episodio12.srt', 12),
('Legenda_Episodio13.srt', 13),
('Legenda_Episodio14.srt', 12),
('Legenda_Episodio15.srt', 11);
```

id	arquivo	id_episodio
1	BLOB	1
2	BLOB	2
3	BLOB	3
4	BLOB	4
5	BLOB	5
6	BLOB	6
7	BLOB	7
8	BLOB	8
9	BLOB	9
10	BLOB	10
16	BLOB	11
17	BLOB	12
18	BLOB	13
19	BLOB	12
20	BLOB	11

```
INSERT INTO filmes(id, titulo, data_lancamento, id_categoria, id_produtores,
id_diretor, id_classificacao) VALUES (1, "A Cor Púrpura", "1985-12-18",3, 1,3,3);
INSERT INTO filmes(id, titulo, data_lancamento, id_categoria, id_produtores,
id_diretor, id_classificacao) VALUES (2, "A.I. Artificial Intelligence", "1985-07-29",3,
1,3,3);
```

```
INSERT INTO filmes(id, titulo, data_lancamento, id_categoria, id_produtores, id_diretor, id_classificacao) VALUES (3, "Kill Bill: Volume 1", "2003-10-10", 1, 1, 2, 5);
```

```
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,  
`id_produtores`, `id_diretor`, `id_classificacao`)  
VALUES (4, 'Schindler\'s List', '1993-12-15', 2, 2, 2, 3);
```

-- Filme 5

```
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,  
`id_produtores`, `id_diretor`, `id_classificacao`)  
VALUES (5, 'Saving Private Ryan', '1998-07-24', 2, 2, 2, 4);
```

```
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,  
`id_produtores`, `id_diretor`, `id_classificacao`)  
VALUES (6, 'The Shawshank Redemption', '1994-09-23', 2, 3, 3, 3);
```

```
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,  
`id_produtores`, `id_diretor`, `id_classificacao`)  
VALUES (7, 'The Godfather', '1972-03-24', 2, 3, 3, 4);
```

```
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,  
`id_produtores`, `id_diretor`, `id_classificacao`)  
VALUES (8, 'The Dark Knight', '2008-07-18', 3, 4, 4, 5);
```

```
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,  
`id_produtores`, `id_diretor`, `id_classificacao`)  
VALUES (9, 'Inception', '2010-07-16', 3, 4, 4, 5);
```

```
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,
`id_produtoa`, `id_diretor`, `id_classificacao`)
VALUES (10, 'Pulp Fiction', '1994-05-12', 3, 5, 5, 4);
```

```
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,
`id_produtoa`, `id_diretor`, `id_classificacao`)
VALUES (11, 'Forrest Gump', '1994-07-06', 3, 5, 5, 3);
```

```
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,
`id_produtoa`, `id_diretor`, `id_classificacao`)
VALUES (12, 'The Matrix', '1999-03-31', 4, 5, 2, 5);
INSERT INTO `wydenflix`.`filmes` (`id`, `titulo`, `data_lancamento`, `id_categoria`,
`id_produtoa`, `id_diretor`, `id_classificacao`)
VALUES (13, 'The Lord of the Rings: The Fellowship of the Ring', '2001-12-19', 4, 5,
1, 3);
```

id	titulo	data_lancamento	id_categoria	id_produtoa	id_diretor	id_classificacao
1	A Cor Púrpura	1985-12-18	3	1	3	3
2	A.I. Artificial Intelligence	1985-07-29	3	1	3	3
3	Kill Bill: Volume 1	2003-10-10	1	1	2	5
4	Schindler's List	1993-12-15	2	2	2	3
5	Saving Private Ryan	1998-07-24	2	2	2	4
6	The Shawshank Redemption	1994-09-23	2	3	3	3
7	The Godfather	1972-03-24	2	3	3	4
8	The Dark Knight	2008-07-18	3	4	4	5
9	Inception	2010-07-16	3	4	4	5
10	Pulp Fiction	1994-05-12	3	5	5	4
11	Forrest Gump	1994-07-06	3	5	5	3
12	The Matrix	1999-03-31	4	5	2	5
13	The Lord of the Rings: The ...	2001-12-19	4	5	1	3

```
INSERT INTO wydenflix.biblioteca (Tipo, id_filme, id_serie) VALUES
('FILME', 1, NULL),
('FILME', 2, NULL),
('FILME', 3, NULL),
```

```
('FILME', 4, NULL),  
('FILME', 5, NULL);
```

```
INSERT INTO wydenflix.biblioteca (Tipo, id_filme, id_serie) VALUES  
('SERIE', NULL, 12),  
('SERIE', NULL, 2),  
('SERIE', NULL, 3),  
('SERIE', NULL, 4),  
('SERIE', NULL, 5);
```

```
INSERT INTO wydenflix.biblioteca (Tipo, id_filme, id_serie) VALUES  
('FILME', 6, NULL),  
('FILME', 7, NULL),  
('FILME', 8, NULL),  
('SERIE', NULL, 6),  
('SERIE', NULL, 7),  
('SERIE', NULL, 8);
```

Tipo	id	id_filme	id_serie
FILME	1	1	NULL
FILME	2	2	NULL
FILME	3	3	NULL
FILME	4	4	NULL
FILME	5	5	NULL
SERIE	11	NULL	12
SERIE	12	NULL	2
SERIE	13	NULL	3
SERIE	14	NULL	4
SERIE	15	NULL	5
FILME	16	6	NULL
FILME	17	7	NULL
FILME	18	8	NULL
SERIE	19	NULL	6
SERIE	20	NULL	7
SERIE	21	NULL	8

```
INSERT INTO wydenflix.favoritos (id_perfil, quando_foi, id_biblioteca) VALUES
(1, '2023-11-08', 1),
(1, '2023-11-08', 3),
(2, '2023-11-08', 2),
(2, '2023-11-08', 4),
(3, '2023-11-08', 5),
(3, '2023-11-08', 12);
```

```
INSERT INTO wydenflix.favoritos (id_perfil, quando_foi, id_biblioteca) VALUES
(1, '2023-11-09', 14),
(2, '2023-11-09', 14),
(3, '2023-11-09', 14),
(4, '2023-11-09', 11),
(4, '2023-11-09', 13);
```

```
INSERT INTO wydenflix.favoritos (id_perfil, quando_foi, id_biblioteca) VALUES
(4, '2023-11-08', 2),
(5, '2023-11-08', 1),
(5, '2023-11-08', 3),
(3, '2023-11-08', 4),
(3, '2023-11-08', 15);
```

id	id_perfil	quando_foi	id_biblioteca
7	1	2023-11-08	1
8	1	2023-11-08	3
9	2	2023-11-08	2
10	2	2023-11-08	4
11	3	2023-11-08	5
12	3	2023-11-08	12
13	1	2023-11-09	14
14	2	2023-11-09	2023-11-09
15	3	2023-11-09	14
16	4	2023-11-09	11
17	4	2023-11-09	13
23	4	2023-11-08	2
24	5	2023-11-08	1
25	5	2023-11-08	3
26	3	2023-11-08	4
27	3	2023-11-08	15
28	1	2023-11-09	14
29	2	2023-11-09	14
30	3	2023-11-09	14
31	4	2023-11-09	11

```
INSERT INTO wydenflix.assistido (id_perfil, data_assistido, id_biblioteca) VALUES
(1, '2023-11-08', 1),
(1, '2023-11-08', 3),
(2, '2023-11-08', 2),
(2, '2023-11-08', 4),
(3, '2023-11-08', 5),
(3, '2023-11-08', 1);
```

```
INSERT INTO wydenflix.assistido (id_perfil, data_assistido, id_biblioteca) VALUES
(4, '2023-11-08', 2),
(5, '2023-11-08', 1),
(5, '2023-11-08', 3),
(3, '2023-11-08', 4),
(2, '2023-11-08', 14);
```

12	4	2023-11-08	2
13	5	2023-11-08	1
14	5	2023-11-08	3
15	3	2023-11-08	4
16	2	2023-11-08	14
17	1	2023-11-08	1
18	1	2023-11-08	3
19	2	2023-11-08	2
20	2	2023-11-08	4
21	3	2023-11-08	5
22	3	2023-11-08	1

```
INSERT INTO wydenflix.assistindo (id_perfil, data_que_comecou_assistir,
id_biblioteca) VALUES
(1, '2023-11-08', 1),
(1, '2023-11-08', 3),
(2, '2023-11-08', 2),
(2, '2023-11-08', 4),
(3, '2023-11-08', 5),
(3, '2023-11-08', 12);
```

```
INSERT INTO wydenflix.assistindo (id_perfil, data_que_comecou_assistir,
id_biblioteca) VALUES
(1, '2023-11-09', 15),
(2, '2023-11-09', 13),
(3, '2023-11-09', 13),
(4, '2023-11-09', 13),
(4, '2023-11-09', 13);
```

```
INSERT INTO wydenflix.assistindo (id_perfil, data_que_comecou_assistir,
id_biblioteca) VALUES
(4, '2023-11-08', 2),
(5, '2023-11-08', 1),
```



```
(5, '2023-11-08', 3),
(2, '2023-11-08', 4),
(1, '2023-11-08', 12);
```

id	id_perfil	data_que_comecou_assistir	id_biblioteca
1	1	2023-11-08	1
2	1	2023-11-08	3
3	2	2023-11-08	2
4	2	2023-11-08	4
5	3	2023-11-08	5
6	3	2023-11-08	12
7	1	2023-11-09	15
8	2	2023-11-09	13
9	3	2023-11-09	13
10	4	2023-11-09	13
11	4	2023-11-09	13
17	4	2023-11-08	2
18	5	2023-11-08	1

-- Inserir atores fictícios

```
INSERT INTO wydenflix.ator (nome, data_nascimento) VALUES
('Tom Hanks', '1956-07-09'),
('Scarlett Johansson', '1984-11-22'),
('Leonardo DiCaprio', '1974-11-11'),
('Jennifer Lawrence', '1990-08-15'),
('Chris Hemsworth', '1983-08-11');
```

-- Inserir mais atores fictícios

```
INSERT INTO wydenflix.ator (nome, data_nascimento) VALUES
('Gal Gadot', '1985-04-30'),
('Robert Downey Jr.', '1965-04-04'),
('Emma Stone', '1988-11-06'),
('Chris Evans', '1981-06-13'),
('Margot Robbie', '1990-07-02');
```

-- Inserir atores adicionais

```
INSERT INTO wydenflix.ator (nome, data_nascimento) VALUES
```

('Dwayne Johnson', '1972-05-02'),
 ('Angelina Jolie', '1975-06-04'),
 ('Brad Pitt', '1963-12-18'),
 ('Will Smith', '1968-09-25'),
 ('Natalie Portman', '1981-06-09');

id	nome	data_nascimento
1	Tom Hanks	1956-07-09
2	Scarlett Johansson	1984-11-22
3	Leonardo DiCaprio	1974-11-11
4	Jennifer Lawrence	1990-08-15
5	Chris Hemsworth	1983-08-11
6	Gal Gadot	1985-04-30
7	Robert Downey Jr.	1965-04-04
8	Emma Stone	1988-11-06
9	Chris Evans	1981-06-13
10	Margot Robbie	1990-07-02
11	Dwayne Johnson	1972-05-02
12	Angelina Jolie	1975-06-04
13	Brad Pitt	1963-12-18
14	Will Smith	1968-09-25
15	Natalie Portman	1981-06-09

INSERT INTO wydenflix.elenco (id_ator, id_biblioteca) VALUES

(1, 1),
 (2, 2),
 (3, 3),
 (4, 4),
 (5, 5),
 (6, 12),
 (7, 14),
 (8, 13),
 (9, 14),
 (10, 14),
 (11, 12),
 (12, 12);

```
INSERT INTO wydenflix.elenco (id_ator, id_biblioteca) VALUES
```

```
(13, 1),  
(14, 2),  
(15, 3),  
(12, 4),  
(15, 5),  
(8, 11),  
(7, 12),  
(3, 15),  
(2, 15),  
(2, 15),  
(3, 11),  
(4, 12);
```

```
INSERT INTO wydenflix.elenco (id_ator, id_biblioteca) VALUES
```

```
(5, 1),  
(6, 2),  
(7, 3),  
(8, 4),  
(9, 5),  
(4, 15),  
(1, 14),  
(2, 14),  
(3, 14),  
(4, 14),  
(5, 15),  
(6, 12);
```

id	id_ator	id_biblioteca
25	1	1
26	2	2
27	3	3
28	4	4
29	5	5
30	6	12
31	7	14
32	8	13
33	9	14
34	10	14
35	11	12
36	12	12
61	13	1
62	14	2
63	15	3
64	12	4
65	15	5
66	8	11
67	7	12
68	3	15
69	2	15

6.1

Usuário	Select	Insert	Update	Delete
Adm	SIM	SIM	SIM	SIM
Usuário Pago	SIM	SIM	SIM	SIM
Produtora	SIM	SIM	NÃO	NÃO
Usuário não pagante	NÃO	NÃO	NÃO	NÃO
Consultor	SIM	NÃO	NÃO	NÃO

6.2

```
CREATE USER 'usuario pago'@'localhost' IDENTIFIED VIA mysql_native_password
USING '***';GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'usuario
pago'@'localhost' REQUIRE NONE WITH MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0
```

```
MAX_USER_CONNECTIONS 0;GRANT ALL PRIVILEGES ON `wydenflix`. * TO 'usuario pago'@'localhost';
```

```
CREATE USER 'usuario nao pago'@'localhost' IDENTIFIED VIA mysql_native_password USING '****';GRANT USAGE ON *.* TO 'usuario nao pago'@'localhost' REQUIRE NONE WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;GRANT ALL PRIVILEGES ON `wydenflix`. * TO 'usuario nao pago'@'localhost';
```

```
REVOKE ALL PRIVILEGES ON `wydenflix`. * FROM 'usuario nao pago'@'localhost';GRANT USAGE ON `wydenflix`. * TO 'usuario nao pago'@'localhost';
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO `usuario pago`@`localhost`;
```

```
CREATE USER 'produtora'@'localhost' IDENTIFIED VIA mysql_native_password USING '****';GRANT SELECT, INSERT ON *.* TO 'produtora'@'localhost' REQUIRE NONE WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
```

```
CREATE USER 'consultor'@'localhost' IDENTIFIED VIA mysql_native_password USING '****';GRANT SELECT ON *.* TO 'consultor'@'localhost' REQUIRE NONE WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
```

7.1

```
SELECT nome, email FROM usuario;
```

nome	email
João Silva	usuario1@example.com
Maria Souza	usuario2@example.com
Carlos Oliveira	usuario3@example.com
Emilia faiz	usuario4@example.com
Tanjiro Yakumata	usuario5@example.com

SELECT * FROM filmes;

id	titulo	data_lancamento	id_categoria	id_produto	id_diretor	id_classificacao
1	A Cor Púrpura	1985-12-18	3	1	3	3
2	A.I. Artificial Intelligence	1985-07-29	3	1	3	3
3	Kill Bill: Volume 1	2003-10-10	1	1	2	5
4	Schindler's List	1993-12-15	2	2	2	3
5	Saving Private Ryan	1998-07-24	2	2	2	4
6	The Shawshank Redemption	1994-09-23	2	3	3	3
7	The Godfather	1972-03-24	2	3	3	4
8	The Dark Knight	2008-07-18	3	4	4	5
9	Inception	2010-07-16	3	4	4	5
10	Pulp Fiction	1994-05-12	3	5	5	4
11	Forrest Gump	1994-07-06	3	5	5	3
12	The Matrix	1999-03-31	4	5	2	5
13	The Lord of the Rings: The ...	2001-12-19	4	5	1	3

SELECT tipo, preco FROM plano;

tipo	preco
Plano Básico	9.99
Plano Padrão	14.99
Plano Premium	19.99
Plano Promocional	17.99
Plano Gratuito	0.00

7.2

SELECT * FROM filmes

WHERE data_lancamento BETWEEN "2000-01-01" AND "2010-01-01";

id	titulo	data_lancamento	id_categoria	id_produto	id_diretor	id_classificacao
3	Kill Bill: Volume 1	2003-10-10	1	1	2	5
8	The Dark Knight	2008-07-18	3	4	4	5
13	The Lord of the Rings: The Fellowship of the Ring	2001-12-19	4	5	1	3

SELECT * FROM episódios

WHERE temporada = 1 and data_lancamento >= "2015-01-01";

id	titulo	temporada	duracao	data_lancamento	id_serie
1	Chapter One: The Vanishing of Will Byers	1	00:50:00	2016-07-15	2
2	Chapter Two: The Weirdo on Maple Street	1	00:45:00	2016-07-15	2
3	Chapter One: Wolferton Splash	1	00:55:00	2016-11-04	3
4	Chapter Two: Hyde Park Corner	1	01:00:00	2016-11-04	3
9	The Original	1	00:58:00	2016-10-02	12
10	Chapter One: The Mandalorian	1	00:40:00	2019-11-12	8

SELECT * FROM historico_de_pagamento

WHERE id >= 10 and id <= 100;

id	id_metodo_pagamento	data_pagamento	id_usuario	id_assinatura
21	6	2023-10-15	1	1
22	7	2023-11-20	2	2
23	7	2023-12-25	3	3
24	6	2024-01-10	1	1
25	6	2024-02-12	2	2
26	3	2024-03-18	3	3
27	3	2024-04-05	1	1
28	3	2024-05-22	2	2
29	4	2024-06-30	3	3
30	4	2024-07-15	1	1
31	4	2024-08-20	2	2
32	5	2024-09-25	3	3
33	5	2024-10-10	1	1

7.3

SELECT * FROM filmes, series;

id	titulo	data_lancamento	id_categoria	id_produto	id_diretor	id_classificacao	id	titulo	ano_lancamento	id_categoria	id_produto	i
1	A Cor Púrpura	1985-12-18	3	1	3	3	2	Stranger Things	2016-07-15	1	1	2
1	A Cor Púrpura	1985-12-18	3	1	3	3	3	The Crown	2016-11-04	1	2	3
1	A Cor Púrpura	1985-12-18	3	1	3	3	4	Black Mirror	2011-12-04	2	1	4
1	A Cor Púrpura	1985-12-18	3	1	3	3	5	Narcos	2015-08-28	1	3	5
1	A Cor Púrpura	1985-12-18	3	1	3	3	6	The Witcher	2019-12-20	1	1	4
1	A Cor Púrpura	1985-12-18	3	1	3	3	7	Money Heist	2017-05-02	2	2	4
1	A Cor Púrpura	1985-12-18	3	1	3	3	8	The Mandalorian	2019-11-12	1	3	2
1	A Cor Púrpura	1985-12-18	3	1	3	3	9	Breaking Bad	2008-01-20	1	1	2
1	A Cor Púrpura	1985-12-18	3	1	3	3	10	The Office	2005-03-24	2	2	4
1	A Cor Púrpura	1985-12-18	3	1	3	3	11	Stranger Things 2	2017-10-27	1	1	2
1	A Cor Púrpura	1985-12-18	3	1	3	3	12	Westworld	2016-10-02	1	3	1
1	A Cor Púrpura	1985-12-18	3	1	3	3	13	Friends	1994-09-22	2	2	5

SELECT * FROM favoritos, assistido, assistindo;

id	id_perfil	quando_foi	id_biblioteca	id	id_perfil	data_assistido	id_biblioteca	id	id_perfil	data_que_comecou_assistir	id_biblioteca
7	1	2023-11-08	1	12	4	2023-11-08	2	1	1	2023-11-08	1
7	1	2023-11-08	1	13	5	2023-11-08	1	1	1	2023-11-08	1
7	1	2023-11-08	1	14	5	2023-11-08	3	1	1	2023-11-08	1
7	1	2023-11-08	1	15	3	2023-11-08	4	1	1	2023-11-08	1
7	1	2023-11-08	1	16	2	2023-11-08	14	1	1	2023-11-08	1
7	1	2023-11-08	1	17	1	2023-11-08	1	1	1	2023-11-08	1
7	1	2023-11-08	1	18	1	2023-11-08	3	1	1	2023-11-08	1
7	1	2023-11-08	1	19	2	2023-11-08	2	1	1	2023-11-08	1
7	1	2023-11-08	1	20	2	2023-11-08	4	1	1	2023-11-08	1
7	1	2023-11-08	1	21	3	2023-11-08	5	1	1	2023-11-08	1
7	1	2023-11-08	1	22	3	2023-11-08	1	1	1	2023-11-08	1
7	1	2023-11-08	1	12	4	2023-11-08	2	2	1	2023-11-08	3
7	1	2023-11-08	1	13	5	2023-11-08	1	2	1	2023-11-08	3

SELECT * FROM conta, perfil;

id	email	senha	id	nome	id_usuario
1	usuario1@example.com	senha123	1	Perfil Principal	1
2	usuario2@example.com	senha456	1	Perfil Principal	1
3	usuario3@example.com	senha789	1	Perfil Principal	1
4	usuario4@example.com	senha123	1	Perfil Principal	1
5	usuario5@example.com	senha789	1	Perfil Principal	1
1	usuario1@example.com	senha123	2	Perfil Infantil	1
2	usuario2@example.com	senha456	2	Perfil Infantil	1
3	usuario3@example.com	senha789	2	Perfil Infantil	1
4	usuario4@example.com	senha123	2	Perfil Infantil	1
5	usuario5@example.com	senha789	2	Perfil Infantil	1
1	usuario1@example.com	senha123	3	Perfil Principal	2
2	usuario2@example.com	usuario1@example.com		Perfil Principal	2
3	usuario3@example.com	senha789	3	Perfil Principal	2

SELECT nome, email, tipo

FROM usuario

INNER JOIN assinatura on assinatura.id = usuario.id_assinatura

INNER JOIN plano ON plano.id = assinatura.id_plano;

nome	email	tipo
João Silva	usuario1@example.com	Plano Básico
Maria Souza	usuario2@example.com	Plano Padrão
Carlos Oliveira	usuario3@example.com	Plano Premium
Emilia faiz	usuario4@example.com	Plano Básico
Tanjiro Yakumata	usuario5@example.com	Plano Premium

SELECT titulo, data_lancamento, diretores.nome AS nome_diretor, classificacao.idade_permitida, categoria.tipo, produtora.nome FROM biblioteca

INNER JOIN filmes ON filmes.id = id_filme

INNER JOIN categoria ON categoria.id = filmes.id_categoria

INNER JOIN produtora ON produtora.id = filmes.id_produtores

INNER JOIN classificacao ON classificacao.id = filmes.id_classificacao

INNER JOIN diretores ON diretores.id = filmes.id_diretor

WHERE id_serie IS NULL;

titulo	data_lancamento	nome_diretor	idade_permitida	tipo	nome
A Cor Púrpura	1985-12-18	Steven Spielberg	14	Drama	Warner Bros
A.I. Artificial Intelligence	1985-07-29	Steven Spielberg	14	Drama	Warner Bros
Kill Bill: Volume 1	2003-10-10	Quentin Tarantino	18	Ação	Warner Bros
Schindler's List	1993-12-15	Quentin Tarantino	14	Comédia	Paramount Pictures
Saving Private Ryan	1998-07-24	Quentin Tarantino	16	Comédia	Paramount Pictures
The Shawshank Redemption	1994-09-23	Steven Spielberg	14	Comédia	The Walt Disney Company
The Godfather	1972-03-24	Steven Spielberg	16	Comédia	The Walt Disney Company
The Dark Knight	2008-07-18	Denzel Washigton	18	Drama	21 Century FOX

SELECT series.titulo, series.ano_lancamento, ator.nome FROM biblioteca

INNER JOIN series ON series.id = biblioteca.id_serie

INNER JOIN elenco ON elenco.id_biblioteca = biblioteca.id

INNER JOIN ator on ator.id = elenco.id_ator

WHERE biblioteca.id_filme IS NULL;

titulo	ano_lancamento	nome
Westworld	2016-10-02	Emma Stone
Westworld	2016-10-02	Leonardo DiCaprio
Stranger Things	2016-07-15	Gal Gadot
Stranger Things	2016-07-15	Dwayne Johnson
Stranger Things	2016-07-15	Angelina Jolie
Stranger Things	2016-07-15	Robert Downey Jr.
Stranger Things	2016-07-15	Jennifer Lawrence
Stranger Things	2016-07-15	Gal Gadot
The Crown	2016-11-04	Emma Stone
Black Mirror	2011-12-04	Emma Stone Jr.
Black Mirror	2011-12-04	Chris Evans
Black Mirror	2011-12-04	Margot Robbie
Black Mirror	2011-12-04	Tom Hanks

8. Normalização

A normalização no modelo de dados do WydenFlix desempenha um papel essencial na estruturação eficiente das informações armazenadas no banco de dados, seguindo princípios fundamentais para garantir a consistência e integridade dos dados.

Ao analisar a arquitetura das tabelas, percebemos uma cuidadosa aplicação dos conceitos de normalização. Cada tabela representa uma entidade única, como Categoria, Diretores, Produtora, Método de Pagamento, Plano, Assinatura, Conta, Usuário, Perfil, Dispositivos Cadastrados, Filmes, Classificação, Séries, Episódios, Ator, Legendas, Biblioteca, Favoritos, Assistido, Assistindo, Elenco e Histórico de Pagamento. A minimização de redundâncias é evidente na estrutura do banco de

dados. Por exemplo, as informações relacionadas aos usuários são armazenadas na tabela Usuário, evitando duplicações desnecessárias em outras partes do sistema. Isso não apenas economiza espaço de armazenamento, mas também simplifica a manutenção e atualização dos dados. Os relacionamentos entre as tabelas são estabelecidos através de chaves estrangeiras, criando uma rede de conexões entre as diferentes entidades. Um exemplo notável é a relação entre `Assinatura` e `Plano`, onde a chave estrangeira na tabela Assinatura referencia o id correspondente na tabela `Plano`. Essa abordagem cria uma integração coesa entre as diferentes partes do sistema, permitindo consultas complexas e garantindo a consistência dos dados. Em resumo, a normalização no WydenFlix não é apenas uma prática técnica, mas uma estratégia cuidadosa para garantir que a estrutura de dados seja eficiente, livre de redundâncias desnecessárias e capaz de manter a integridade dos dados em todo o sistema. Essa abordagem sólida contribui para a confiabilidade e desempenho do serviço de streaming, proporcionando uma experiência consistente e confiável aos usuários.

9. Conclusão

Após a análise e desenvolvimento detalhado do banco de dados para o serviço de streaming de filmes e séries proposto, é possível concluir que atingimos com sucesso os objetivos definidos no início do processo. Nosso foco estava em representar de forma conceitual, lógica e física a estrutura do banco de dados, proporcionando uma compreensão profunda do funcionamento de um sistema robusto e complexo de streaming. Ao longo do trabalho, exploramos a modelagem de dados, abrangendo entidades como usuários, assinaturas, filmes, séries, planos, entre outras. A divisão em pares facilitou a abordagem de cada aspecto de maneira mais eficiente, garantindo consistência e qualidade em cada etapa do desenvolvimento. A compreensão do fluxo lógico do sistema se revelou fundamental, especialmente ao identificar as entidades participantes e seus atributos. A aplicação dos princípios de normalização permitiu a criação de um banco de dados eficiente, minimizando redundâncias e estabelecendo relações sólidas entre as diferentes partes do sistema. Além disso, a experiência proporcionou insights valiosos sobre técnicas de consulta, otimização e eficiência dos dados. A complexidade do serviço de streaming exigiu uma abordagem estruturada e organizada, refletindo-se na clareza da representação do banco de

dados. Em última análise, este projeto não apenas contribuiu para a nossa compreensão prática de design de banco de dados, mas também ofereceu uma visão aprofundada do funcionamento interno de um serviço de streaming. Estamos confiantes de que as habilidades adquiridas neste processo serão aplicáveis em contextos mais amplos, preparando-nos para enfrentar desafios similares no futuro.