

Universidade Federal de São Paulo

Trabalho Final de Banco de Dados

Banco de Dados – Netflix

Nome: Victor Akio Domoto

150546 – victor.domoto@unifesp.br

Nome: Yuri dos Santos Araújo

143669 – yuri.araujo@unifesp.br

SGBD: MySQL.

Domínio escolhido: banco de dados da Netflix.

Entidades: usuário, plano, formaPagamento, cartaoCreditoDebito, cartaoPrepago, planoPagto, perfil, conteúdo, gênero, cenasMomentos, conteudoGeneroCenasMomentos, ator, personagem.

Mapeamento:

- Usuário (codUsuario, nome, sobrenome, email, senha, #codPlano);
- Plano (codPlano, nome, preco, descricao);
- FormaPagamento (codFormPag, nome);
 - CartaoCreditoDebito(#codFormPag, codCartaoCredDebito, nome, sobrenome, numCartao, dataValidade, codigoCVV);
 - CartaoPrePago (#codFormPag, codCartaoPrePago, PIN);
- PlanoPagto (#codPlano, #codFormPag);
- Perfil (codPerfil, nome, classificacaoEtaria, iniciaEpisodioAutomaticamente, reproduzAutomaticamentePrevias);
- Conteúdo (codConteudo, nome, criacao, classificacao, relevancia, tipoConteudo);
 - Serie (#codConteudo, quantTemporadas);
 - Filme (#codConteudo, tempoDuracao);
- Gênero (codGenero, nome, descricao);
- Cenas-momentos (codCenaMomento, nome, descricao);
- ConteudoGeneroCenasMomentos(#codConteudo, #codGenero, #codCenaMomento);
- Ator (codAtor, nome, sobrenome);
- Personagem (codPersonagem, nome, sobrenome, #codConteudo, #codAtor).

Diagrama de Entidade e Relacionamento:

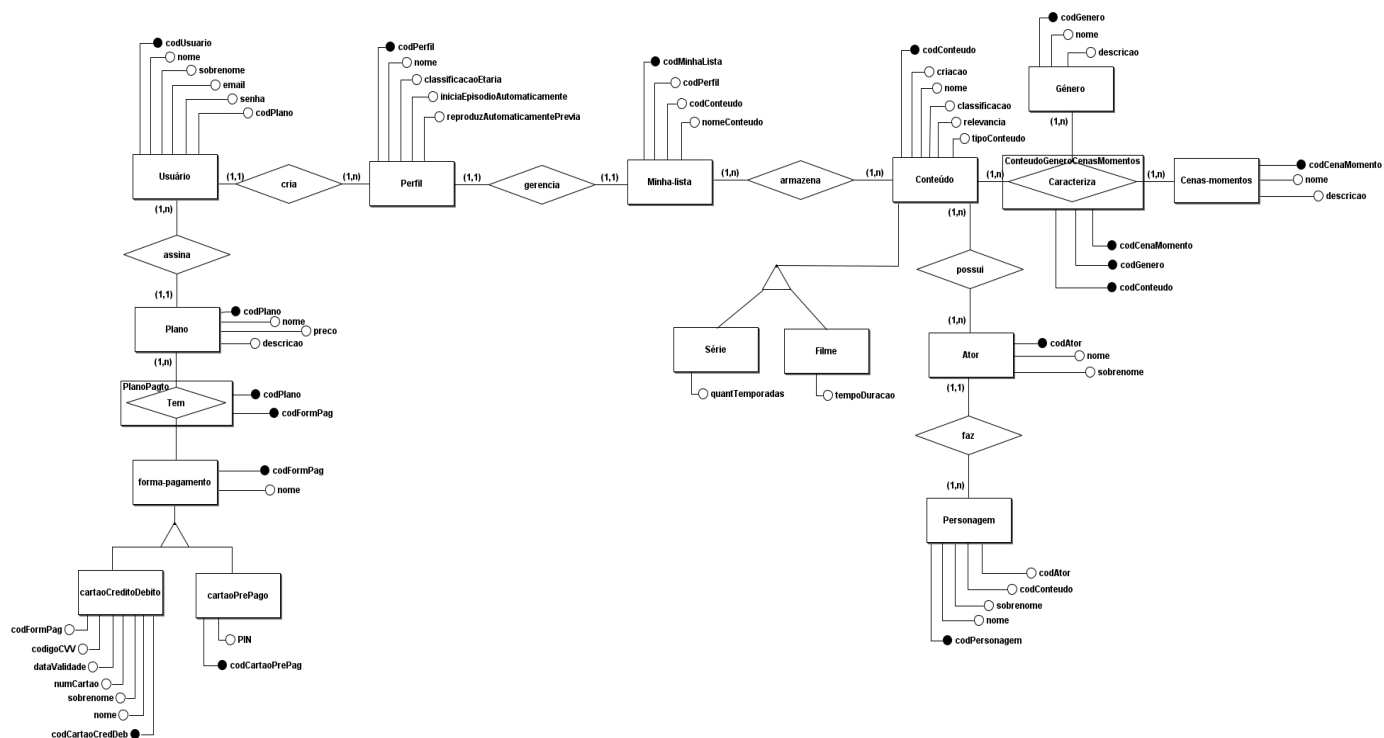


Figura 1 – DER NETFLIX

DDL e DML da implementação:

CREATE DATABASE NETFLIX;

USE NETFLIX;

#Plano (codPlano, nome, preco, qualidade, resolucao, disponibilidade);

CREATE TABLE Plano(codPlano integer, nome varchar(20), preco float, qualidade varchar(20), resolucao varchar(20), disponibilidade boolean,

PRIMARY KEY(codPlano)

) Engine=iNNODB;

#Forma-pagamento (codFormPag, tipo);

CREATE TABLE formaPagamento(codFormPag integer, tipo varchar(20),

PRIMARY KEY(codFormPag)

) Engine=iNNODB;

#Usuário (codUsuario, nome, sobrenome, email, senha, #codPlano, #codFormaPagamento);

CREATE TABLE Usuario(codUsuario integer, nome varchar(20), sobrenome varchar(20), email varchar(40), senha varchar(30), codPlano integer,

PRIMARY KEY(codUsuario),

FOREIGN KEY(codPlano) REFERENCES Plano(codPlano)

) Engine=iNNODB;

#CartaoCreditoDebito(#codFormPag, codCartaoCredDebito, nome, sobrenome, numCartao,dataValidade, codigoCVV);

CREATE TABLE CartaoCreditoDebito(codFormPag integer, codCartaoCredDebito integer, nome varchar(20), sobrenome varchar(20), numCartao varchar(20), dataValidade date, codigoCVV integer,

PRIMARY KEY(codCartaoCredDebito),

FOREIGN KEY (codFormPag) REFERENCES formaPagamento(codFormPag)

) Engine=iNNODB;

#CartaoPrePago (#codFormPag, codCartaoPrePago, PIN);

CREATE TABLE CartaoPrePago(codFormPag integer, codCartaoPrePago integer, PIN varchar(30),

PRIMARY KEY(codCartaoPrePago),

FOREIGN KEY (codFormPag) REFERENCES formaPagamento(codFormPag)

) Engine=iNNODB;

#Perfil (codPerfil, nome, classificacaoEtaria, iniciaEpisodioAutomaticamente, reproduzAutomaticamentePrevias);

CREATE TABLE Perfil(codPerfil integer, nome varchar(20), classEtaria integer, iniEpAuto boolean,

PRIMARY KEY(codPerfil)

) Engine=iNNODB;

#Conteúdo (codConteudo, nome, criacao, classificacao, relevancia, tipoConteudo);

CREATE TABLE Conteudo(codConteudo integer, nome varchar(30), criacao date, classEtaria integer, relevancia integer,

PRIMARY KEY(codConteudo)

) Engine=INNODB;

#Gênero (codGenero, nome, descricao);

CREATE TABLE Genero(codGenero integer, nome varchar(20), descricao varchar(100),

PRIMARY KEY(codGenero)

) Engine=INNODB;

#Cenas-momentos (codCenasMomentos, nome, descricao);

CREATE TABLE cenasMomentos(codCenasMomentos integer, nome varchar(20), descricao varchar(100),

PRIMARY KEY(codCenasMomentos)

) Engine=INNODB;

#Serie (#codConteudo, quantTemporadas);

CREATE TABLE Serie(codConteudo integer, quantTemporadas integer,

PRIMARY KEY(codConteudo),

FOREIGN KEY (codConteudo) REFERENCES Conteudo(codConteudo)

) Engine=INNODB;

#Filme (#codConteudo, tempoDuracao);

CREATE TABLE Filme(codConteudo integer, tempoDuracao time,

PRIMARY KEY(codConteudo),

FOREIGN KEY (codConteudo) REFERENCES Conteudo(codConteudo)

) Engine=INNODB;

#Ator (codAtor, nome, sobrenome);

CREATE TABLE Ator(codAtor integer, nome varchar(20), sobrenome varchar(20),

PRIMARY KEY(codAtor)

) Engine=INNODB;

#Personagem (codPersonagem, nome, sobrenome, #codConteudo, #codAtor);

```
CREATE TABLE Personagem(codPersonagem integer, nome varchar(20), sobrenome  
varchar(20), codConteudo integer, codAtor integer,
```

```
PRIMARY KEY (codPersonagem),
```

```
FOREIGN KEY (codConteudo) REFERENCES Conteudo(codConteudo),
```

```
FOREIGN KEY (codAtor) REFERENCES Ator(codAtor)
```

```
) Engine=INNODB;
```

#PlanoPagto (#codPlano, #codFormPag);

```
CREATE TABLE PlanoPagto(codPlano integer, codFormPag integer,
```

```
PRIMARY KEY(codPlano,codFormPag),
```

```
FOREIGN KEY(codPlano) REFERENCES Plano(codPlano),
```

```
FOREIGN KEY(codFormPag) REFERENCES formaPagamento(codFormPag)
```

```
) Engine=INNODB;
```

#ConteudoGeneroCenasMomentos (#codConteudo, #codGenero, #codCenasMomentos);

```
CREATE TABLE conteudoGeneroCenasMomentos(codConteudo integer, codGenero integer,  
codCenasMomentos integer,
```

```
PRIMARY KEY(codConteudo, codGenero, codCenasMomentos),
```

```
FOREIGN KEY(codConteudo) REFERENCES Conteudo(codConteudo),
```

```
FOREIGN KEY(codGenero) REFERENCES Genero(codGenero),
```

```
FOREIGN KEY(codCenasMomentos) REFERENCES cenasMomentos(codCenasMomentos)
```

```
) Engine=INNODB;
```

Inserções:

#Inserindo os planos

```
INSERT INTO Plano VALUES (1, 'Básico', '25.90', 'Boa', '480p', True);
```

```
INSERT INTO Plano VALUES (2, 'Padrão', '39.90', 'Melhor', '1080p', True);
```

```
INSERT INTO Plano VALUES (3, 'Premium', '55.90', 'Superior ', '4K+HDR', True);
```

#Inserindo as formas de pagamento

```
INSERT INTO formaPagamento VALUES (1, 'CartaoCreditoDebito');
```

```
INSERT INTO formaPagamento VALUES (2, 'CartaoPrePago');
```

#Inserindo os usuários

```
INSERT INTO Usuario VALUES(1, 'Yuri', 'Araujo', 'yuri.araujo@unifesp.br', '1234', '3');
```

```
INSERT INTO Usuario VALUES(2, 'Victor', 'Akio', 'victor.domoto@unifesp.br', '4321', '2');
```

```
INSERT INTO Usuario VALUES(3, 'Henrique', 'Andrade', 'Henrique.Andrade@unifesp.br', '2424', '1');
```

```
INSERT INTO Usuario VALUES(4, 'Ana', 'Santos', 'ana.santos@unifesp.br', '5469', '2');
```

```
INSERT INTO Usuario VALUES(5, 'Maria', 'Silva', 'maria.silva@unifesp.br', '7896', '3');
```

#Inserindo cartões

```
INSERT INTO CartaoCreditoDebito VALUES(1, 1, 'Yuri', 'Araujo', '0000 1111 2222 3333', '2026-02-09', 199);
```

```
INSERT INTO CartaoCreditoDebito VALUES(1, 2, 'Victor', 'Domoto', '1111 1111 2222 3333', '2026-03-10', 299);
```

```
INSERT INTO CartaoCreditoDebito VALUES(1, 3, 'Henrique', 'Andrade', '2222 1111 2222 3333', '2026-04-11', 399);
```

```
INSERT INTO CartaoCreditoDebito VALUES(1, 4, 'Ana', 'Santos', '3333 1111 4444 7777', '2025-07-12', 156);
```

```
INSERT INTO CartaoCreditoDebito VALUES(1, 5, 'Maria', 'Silva', '4444 7896 1254 6532', '2022-11-12', 986);
```

```
INSERT INTO CartaoCreditoDebito VALUES(2, 6, 'João', 'Pedro', '5555 6589 8759 9832', '2029-10-02', 986);
```

#Inserindo perfis

```
INSERT INTO Perfil VALUES (1, 'Yuri', 18, false);
```

```
INSERT INTO Perfil VALUES (2, 'Victor', 18, false);
```

```
INSERT INTO Perfil VALUES (3, 'Julio', 14, true);
```

```
INSERT INTO Perfil VALUES (4, 'Ana', 16, false);
```

```
INSERT INTO Perfil VALUES (5, 'Maria', 10, true);
```

#Inserindo conteúdo

```
INSERT INTO Conteudo VALUES (1, 'The Witcher', '2019-12-20', 18, 99);
```

INSERT INTO Conteudo VALUES (2, 'Peaky Blinders', '2013-09-12', 18, 94);

INSERT INTO Conteudo VALUES (3, 'Arcane', '2021-09-06', 14, 100);

INSERT INTO Conteudo VALUES (4, 'Breaking Bad', '2008-01-20', 18, 98);

INSERT INTO Conteudo VALUES (5, 'Suits', '2011-07-23', 14, 98);

#Inserindo atores

INSERT INTO Ator VALUES (1, 'Tom', 'Hanks');

INSERT INTO Ator VALUES (2, 'Will', 'Smith');

INSERT INTO Ator VALUES (3, 'Meryl', 'Streep');

INSERT INTO Ator VALUES (4, 'Tom', 'Crise');

INSERT INTO Ator VALUES (5, 'Camila', 'Queiroz');

INSERT INTO Ator VALUES (6, 'Henry', 'Cavill');

INSERT INTO Ator VALUES (7, 'Cillian', 'Murphy');

INSERT INTO Ator VALUES (8, 'Ella', 'Purnell');

INSERT INTO Ator VALUES (9, 'Bryan', 'Cranston');

INSERT INTO Ator VALUES (10, 'Gabriel', 'Macht');

#Inserindo cartões pré-pago

INSERT INTO CartaoPrePago VALUES (2, 1, 'a123bc');

INSERT INTO CartaoPrePago VALUES (2, 2, '5a6s5e');

INSERT INTO CartaoPrePago VALUES (2, 3, '9a8e7r');

INSERT INTO CartaoPrePago VALUES (2, 4, '2a5w62');

INSERT INTO CartaoPrePago VALUES (2, 5, '85a2e1');

#Inserindo cenas e momentos

INSERT INTO cenasMomentos VALUES (1, 'Violência', 'Contém cenas com violencia');

INSERT INTO cenasMomentos VALUES (2, 'Empolgantes', 'Contém cenas empolgantes');

INSERT INTO cenasMomentos VALUES (3, 'Irreverentes', 'Contém cenas irreverentes');

INSERT INTO cenasMomentos VALUES (4, 'Suspense no ar', 'Contém cenas com suspense');

INSERT INTO cenasMomentos VALUES (5, 'Humor seco', 'Contém cenas com humor seco');

#Inserindo gêneros

```
INSERT INTO Genero VALUES (1, 'Anime', 'contém animação');  
INSERT INTO Genero VALUES (2, 'Ação', 'contém ação');  
INSERT INTO Genero VALUES (3, 'Drama', 'contém drama');  
INSERT INTO Genero VALUES (4, 'Comédia', 'contém comédia');  
INSERT INTO Genero VALUES (5, 'Aventura', 'contém aventura');
```

#Inserindo conteúdo-gênero-cenasMomentos

```
INSERT INTO conteudoGeneroCenasMomentos VALUES (1, 5, 2);  
INSERT INTO conteudoGeneroCenasMomentos VALUES (2, 3, 1);  
INSERT INTO conteudoGeneroCenasMomentos VALUES (3, 4, 3);  
INSERT INTO conteudoGeneroCenasMomentos VALUES (4, 2, 2);  
INSERT INTO conteudoGeneroCenasMomentos VALUES (5, 1, 4);
```

#Inserindo conteúdos

```
INSERT INTO Conteudo VALUES (6, 'Ilha do Medo', '2010-05-16', 16, 92);  
INSERT INTO Conteudo VALUES (7, 'El Camino', '2019-08-13', 16, 90);  
INSERT INTO Conteudo VALUES (8, 'Arcane', '2021-09-06', 14, 100);  
INSERT INTO Conteudo VALUES (9, 'Scarface', '1983-02-23', 18, 89);  
INSERT INTO Conteudo VALUES (10, 'Fuja', '2021-04-07', 16, 91);
```

#Inserindo Filmes

```
INSERT INTO Filme VALUES(6,'02:56:00');  
INSERT INTO Filme VALUES(7,'01:40:50');  
INSERT INTO Filme VALUES(8,'00:59:45');  
INSERT INTO Filme VALUES(9,'03:01:52');  
INSERT INTO Filme VALUES(10,'01:25:48');
```


#Inserindo personagens

```
INSERT INTO Personagem VALUES (1,'Geralt', 'Rivia', 1, 6);  
INSERT INTO Personagem VALUES (2,'Tommy', 'Shelby', 2, 7);  
INSERT INTO Personagem VALUES (3,'Jinx', 'Powder', 3, 8);  
INSERT INTO Personagem VALUES (4,'Walter', 'White', 4, 9);  
INSERT INTO Personagem VALUES (5,'Harvey', 'Specter', 5, 10);
```

#Inserindo planoPagamento

```
INSERT INTO PlanoPagto VALUES (1, 1);  
INSERT INTO PlanoPagto VALUES (2, 1);  
INSERT INTO PlanoPagto VALUES (2, 2);  
INSERT INTO PlanoPagto VALUES (3, 2);  
INSERT INTO PlanoPagto VALUES (3, 1);
```

#Inserindo séries

```
INSERT INTO Serie VALUES (1, 2);  
INSERT INTO Serie VALUES (2, 4);  
INSERT INTO Serie VALUES (3, 1);  
INSERT INTO Serie VALUES (4, 5);  
INSERT INTO Serie VALUES (5, 5);
```

Consultando tabelas:

#1. Selecionando os conteúdos no qual a relevância é maior ou igual a 99

```
SELECT Conteudo.nome  
FROM Conteudo, conteudoGeneroCenasMomentos  
WHERE Conteudo.codConteudo=conteudoGeneroCenasMomentos.codConteudo AND  
Conteudo.relevancia>=99;
```

```
mysql> #Consultas nas tabelas
mysql> #1. Seleccionando os conteúdos no qual a relevância é maior ou igual a 99
mysql> SELECT Conteudo.nome
-> FROM Conteudo, conteudoGeneroCenasMomentos
-> WHERE Conteudo.codConteudo=conteudoGeneroCenasMomentos.codConteudo AND Conteudo.relevancia>=99;
+-----+
| nome |
+-----+
| The Witcher |
| Arcane |
+-----+
2 rows in set (0,00 sec)
```

Figura 2 – selecionando conteúdo por relevância

#2. Seleccionando os conteúdos no qual o filme foi criado após 2020

SELECT Conteudo.nome

FROM Conteudo, conteudoGeneroCenasMomentos

WHERE Conteudo.codConteudo=conteudoGeneroCenasMomentos.codConteudo AND
Conteudo.criacao>'2020-01-01';

```
mysql> #2. Seleccionando os conteúdos no qual o filme foi criado após 2020
mysql> SELECT Conteudo.nome
-> FROM Conteudo, conteudoGeneroCenasMomentos
-> WHERE Conteudo.codConteudo=conteudoGeneroCenasMomentos.codConteudo AND Conteudo.criacao>'2020-01-01';
+-----+
| nome |
+-----+
| Arcane |
+-----+
1 row in set (0,00 sec)
```

Figura 3 – selecionando conteúdo por data

#3. Seleccionado o conteúdo no qual CenasMomentos sejam 'Empolgantes'

SELECT Conteudo.nome

FROM Conteudo, conteudoGeneroCenasMomentos, cenasMomentos

WHERE Conteudo.codConteudo=conteudoGeneroCenasMomentos.codConteudo AND
cenasMomentos.nome='Empolgantes' AND
cenasMomentos.codCenasMomentos=conteudoGeneroCenasMomentos.codCenasMomentos;

```
mysql> #3. Seleccionado o conteúdo no qual CenasMomentos sejam 'Empolgantes'
mysql> SELECT Conteudo.nome
-> FROM Conteudo, conteudoGeneroCenasMomentos, cenasMomentos
-> WHERE Conteudo.codConteudo=conteudoGeneroCenasMomentos.codConteudo AND cenasMomentos.nome='Empolgantes' A
ND cenasMomentos.codCenasMomentos=conteudoGeneroCenasMomentos.codCenasMomentos;
+-----+
| nome |
+-----+
| The Witcher |
| Breaking Bad |
+-----+
2 rows in set (0,00 sec)
```

Figura 4 – selecionando conteúdo por cenas

#4. Seleccionado as pessoas cuja forma de pagamento é cartão pré pago

SELECT CartaoCreditoDebito.nome, CartaoCreditoDebito.sobrenome

FROM CartaoCreditoDebito, formaPagamento

WHERE formaPagamento.codFormPag=2 AND CartaoCreditoDebito.codFormPag =
formaPagamento.codFormPag;

```
mysql> #4. Seleccionado as pessoas cuja forma de pagamento é cartao pre pago
mysql> SELECT CartaoCreditoDebito.nome, CartaoCreditoDebito.sobrenome
-> FROM CartaoCreditoDebito, formaPagamento
-> WHERE formaPagamento.codFormPag=2 AND CartaoCreditoDebito.codFormPag = formaPagamento.codFormPag;
+-----+
| nome | sobrenome |
+-----+
| João | Pedro    |
+-----+
1 row in set (0,00 sec)
```

Figura 5 – selecionando pessoas por forma de pagamento

#5. Seleccionando os planos cujo valor é inferior a R\$ 26.00

SELECT Plano.nome

FROM Plano, PlanoPagto

WHERE Plano.preco<=26 AND Plano.codPlano=PlanoPagto.codPlano;

```
mysql> #5. Seleccionando os planos cujo valor é inferior a R$ 26.00
mysql> SELECT Plano.nome
-> FROM Plano, PlanoPagto
-> WHERE Plano.preco<=26 AND Plano.codPlano=PlanoPagto.codPlano;
+-----+
| nome |
+-----+
| Básico |
+-----+
1 row in set (0,00 sec)
```

Figura 6 – selecionando planos por valor

Consultando com JOIN LEFT:

#1. Seleccionando todos e qualquer conteudo que tiver

SELECT Conteudo.nome, Conteudo.codConteudo

FROM Conteudo

LEFT JOIN Filme

ON Conteudo.codConteudo = Filme.codConteudo;

```
mysql> #Consultando com JOIN LEFT
mysql> #1. Selecionando todos conteudos e qualquer conteudo que tiver
mysql> SELECT Conteudo.nome, Conteudo.codConteudo
-> FROM Conteudo
-> LEFT JOIN Filme
-> ON Conteudo.codConteudo = Filme.codConteudo;
```

nome	codConteudo
The Witcher	1
Peaky Blinders	2
Arcane	3
Breaking Bad	4
Suits	5
Ilha do Medo	6
El Camino	7
Arcane	8
Scarface	9
Fuja	10

10 rows in set (0,00 sec)

Figura 7 – selecionando conteúdos com LEF JOIN

#2. Selecionando os registros da tabela esquerda e os correspondentes da tabela direita

SELECT * FROM CartaoCreditoDebito

AS C LEFT JOIN formaPagamento

AS F ON C.codFormPag = F.codFormPag;

```
mysql> #2. Selecionando os registros da tabela esquerda e os correspondentes da tabela direita
mysql> SELECT * FROM CartaoCreditoDebito
-> AS C LEFT JOIN formaPagamento
-> AS F ON C.codFormPag = F.codFormPag;
```

codFormPag	codCartaoCredDebito	nome	sobrenome	numCartao	dataValidade	codigoCVV	codFormPag	tipo
1	1	Yuri	Araujo	0000 1111 2222 3333	2026-02-09	199	1	CartaoCreditoDebito
1	2	Victor	Domoto	1111 1111 2222 3333	2026-03-10	299	1	CartaoCreditoDebito
1	3	Henrique	Andrade	2222 1111 2222 3333	2026-04-11	399	1	CartaoCreditoDebito
1	4	Ana	Santos	3333 1111 4444 7777	2025-07-12	156	1	CartaoCreditoDebito
1	5	Maria	Silva	4444 7896 1254 6532	2022-11-12	986	1	CartaoCreditoDebito
2	6	João	Pedro	5555 6589 8759 9832	2029-10-02	986	2	CartaoPrePago

6 rows in set (0,00 sec)

Figura 8 – selecionando tabelas com LEFT JOIN

Consultas com os operadores (AVG, SUM) usando ORDER BY, HAVING e GROUP BY:

#1. Selecionando os filmes em ordem de tempo de duracao

SELECT Conteudo.nome, Filme.tempoDuracao

FROM Conteudo, Filme

WHERE Conteudo.classEtaria>0 AND Conteudo.codConteudo = Filme.codConteudo

ORDER BY Filme.tempoDuracao;

```
mysql> #Consultas com os operadores (AVG, SUM) usando ORDER BY, HAVING e GROUP BY:
mysql> #1. Seleccionando os filmes em ordem de tempo de duracao
mysql> SELECT Conteudo.nome, Filme.tempoDuracao
-> FROM Conteudo, Filme
-> WHERE Conteudo.classEtaria>0 AND Conteudo.codConteudo = Filme.codConteudo
-> ORDER BY Filme.tempoDuracao;
+-----+-----+
| nome      | tempoDuracao |
+-----+-----+
| Arcane     | 00:59:45     |
| Fuja      | 01:25:48     |
| El Camino  | 01:40:50     |
| Ilha do Medo | 02:56:00     |
| Scarface   | 03:01:52     |
+-----+-----+
5 rows in set (0,00 sec)
```

Figura 9 – seleccionando filmes em ordem de tempo

#2. Somando a idade de todos perfis

SELECT SUM(Perfil.classEtaria)

FROM Perfil

WHERE Perfil.codPerfil!=0;

```
mysql> #2. Somando a idade de todos perfis
mysql> SELECT SUM(Perfil.classEtaria)
-> FROM Perfil
-> WHERE Perfil.codPerfil!=0;
+-----+
| SUM(Perfil.classEtaria) |
+-----+
| 76 |
+-----+
1 row in set (0,00 sec)
```

Figura 10 – seleccionando a soma da idade de todos perfis

#3. Seleccionado os nomes e quantidade de temporada das series em ordem decrescente da quantidade de temporadas

SELECT Conteudo.nome, Serie.quantTemporadas

FROM Conteudo, Serie

WHERE Conteudo.codConteudo!=0 AND Conteudo.codConteudo = Serie.codConteudo

ORDER BY Serie.quantTemporadas DESC;

```
mysql> #3. Selecionado os nomes e quantidade de temporada das series em ordem decrescente da quantidade de temporadas
mysql> SELECT Conteudo.nome, Serie.quantTemporadas
      -> FROM Conteudo, Serie
      -> WHERE Conteudo.codConteudo!=0 AND Conteudo.codConteudo = Serie.codConteudo
      -> ORDER BY Serie.quantTemporadas DESC;
+-----+
| nome      | quantTemporadas |
+-----+
| Breaking Bad | 5 |
| Suits      | 5 |
| Peaky Blinders | 4 |
| The Witcher | 2 |
| Arcane     | 1 |
+-----+
5 rows in set (0,01 sec)
```

Figura 11 – selecionando os nomes e quantidade de temporadas das séries

#4. Selecionando o preço dos planos em ordem decrescente

SELECT (preco)

FROM Plano

WHERE preco>0

ORDER BY preco DESC;

```
mysql> #4. Selecionando o preço dos planos em ordem decrescente
mysql> SELECT (preco)
      -> FROM Plano
      -> WHERE preco>0
      -> ORDER BY preco DESC;
+-----+
| preco |
+-----+
| 55.9 |
| 39.9 |
| 25.9 |
+-----+
3 rows in set (0,00 sec)
```

Figura 12 – selecionando os preços dos planos em ordem decrescente

#5 Selecionando os personagens em ordem

SELECT Personagem.nome

FROM Personagem, Ator

WHERE Personagem.codAtor = Ator.codAtor

GROUP BY Personagem.nome

ORDER BY Personagem.nome;

```
mysql> #5 Selecionando os personagens em ordem
mysql> SELECT Personagem.nome
-> FROM Personagem, Ator
-> WHERE Personagem.codAtor = Ator.codAtor
-> GROUP BY Personagem.nome
-> ORDER BY Personagem.nome;
+-----+
| nome  |
+-----+
| Geralt |
| Harvey |
| Jinx  |
| Tommy |
| Walter |
+-----+
5 rows in set (0,00 sec)
```

Figura 13 – selecionando os personagens em ordem alfabética