|  |  |  |  |
| --- | --- | --- | --- |
| Desenho de personagem de desenhos animados com texto preto sobre fundo branco  Descrição gerada automaticamente com confiança média | Ano Letivo:  20251 | Turno:  Noturno | Local:  A definir |
| Curso:  ENGENHARIA DA COMPUTAÇÃO | | Turma:  3ª Fase |
| Disciplina:  BANCO DE DADOS II | | Professor(a):  JORGE LUIZ DA SILVA | |

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

**Projeto de banco de dados para um Sistema de Assinatura para academias**

Vinícius dos Santos Nascimento - https://github.com/ViniciusDSN

Brendon Córdova Silveira - https://github.com/Lightkiin

Vitor Muneretto Tinelli - https://github.com/VitorTinelli

Lucas de Bitencourt Frasson - https://github.com/herudegan

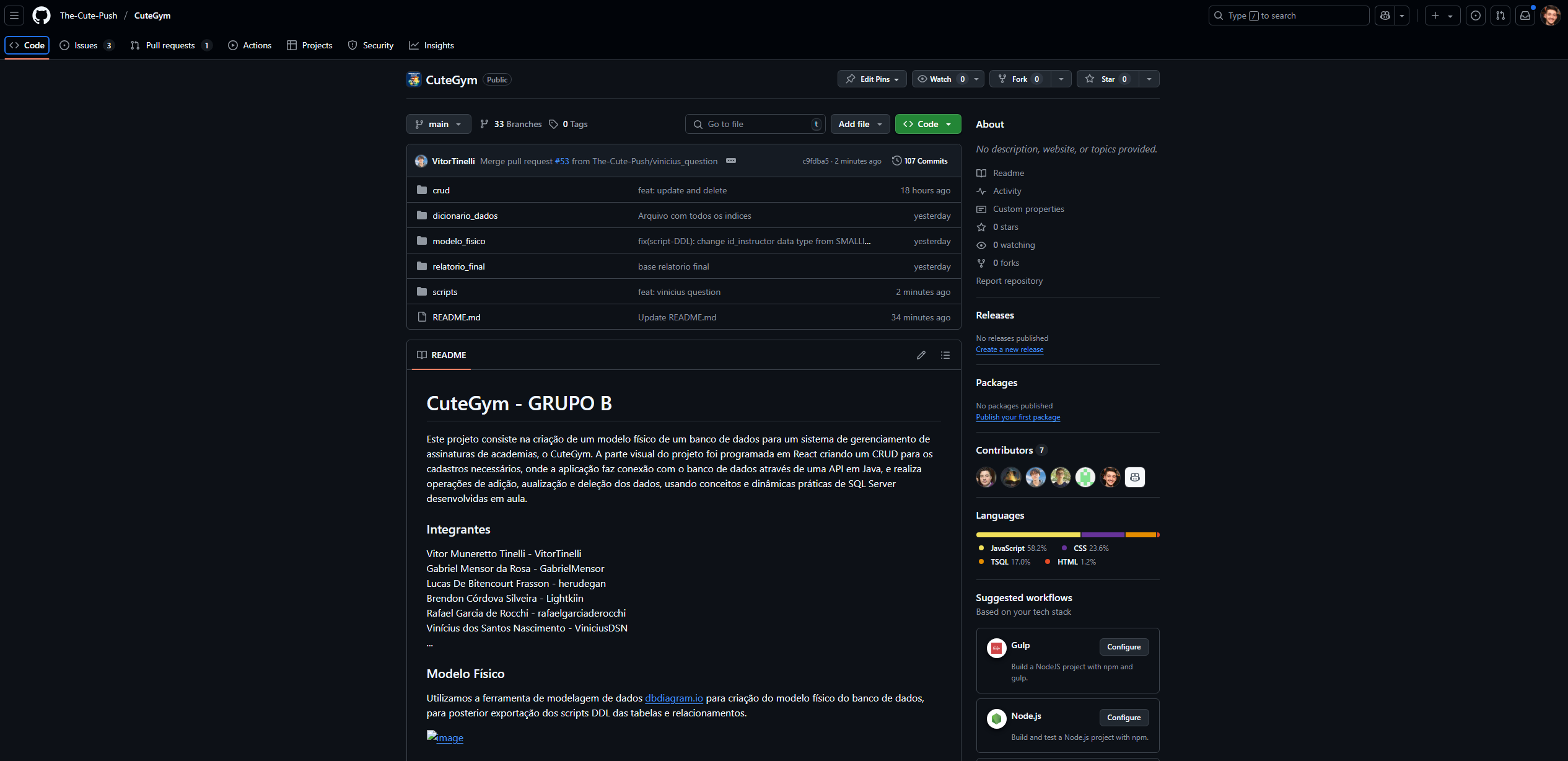
Rafael Garcia de Rocchi - https://github.com/rafaelgarciaderocchi

Gabriel Mensor da Rosa - https://github.com/GabrielMensor

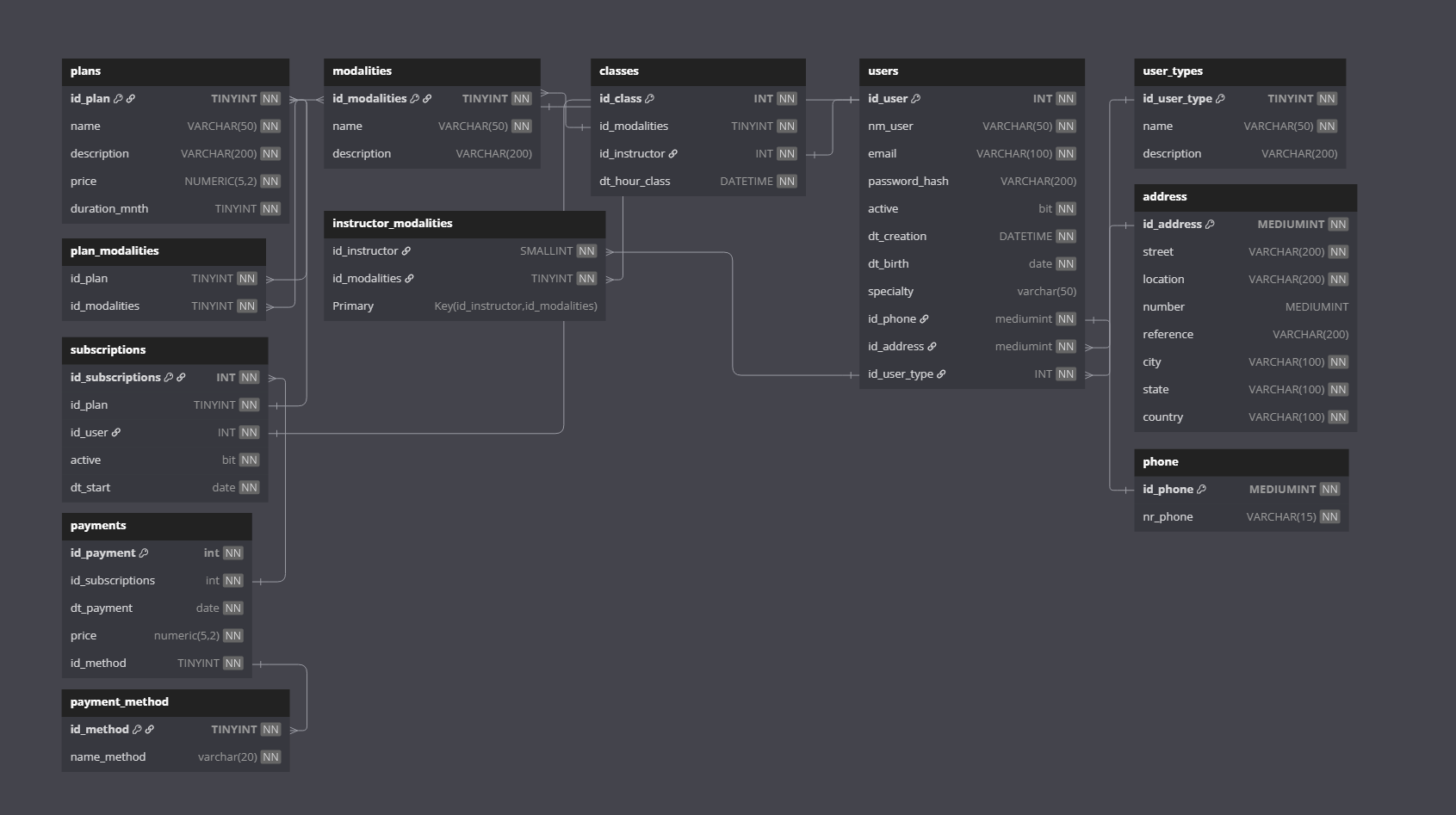
Criciúma, 23/06/2025

**URL do projeto no GitHub**

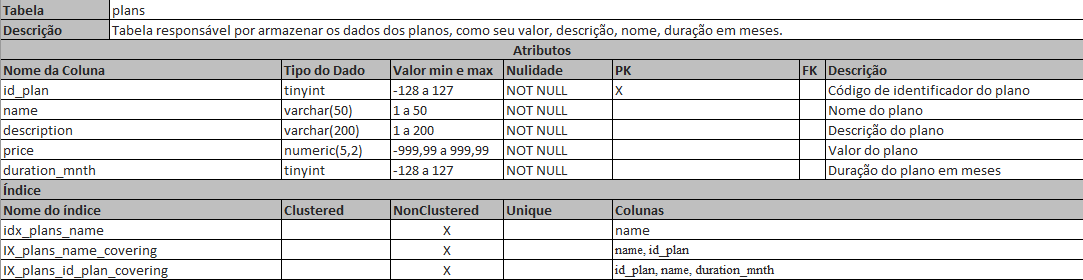
[https://github.com/The-Cute-Push/CuteGym](%20https://github.com/The-Cute-Push/CuteGym)

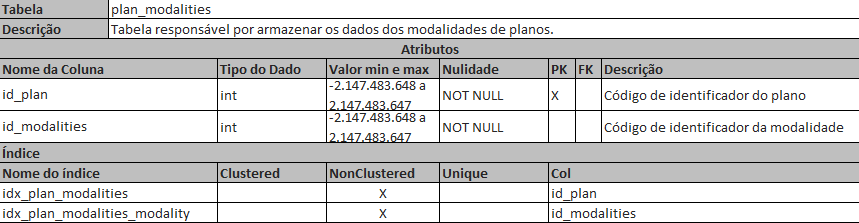
****

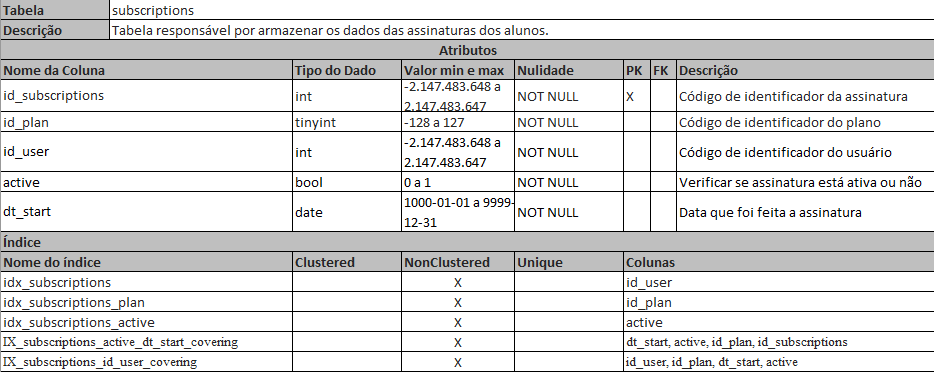
**Modelo ER Físico**

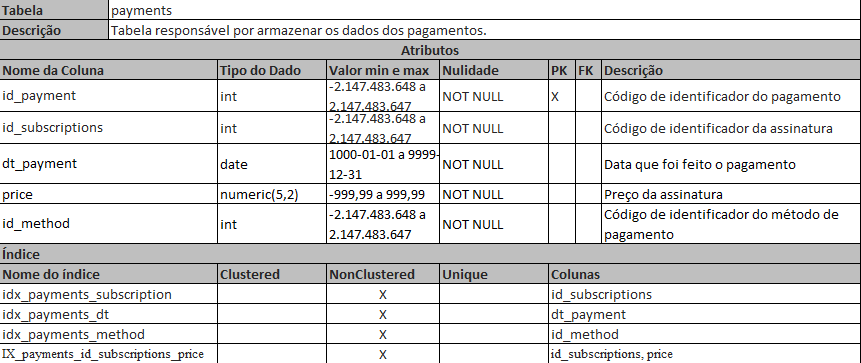
****

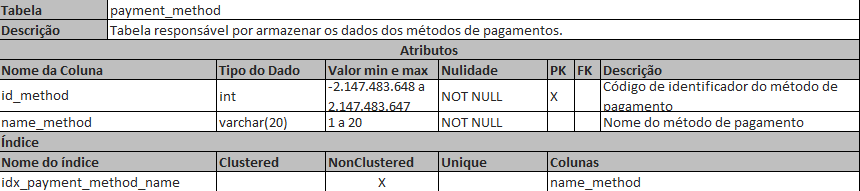
**Dicionário de Dados**

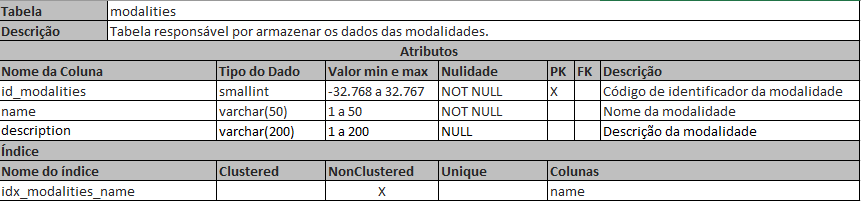


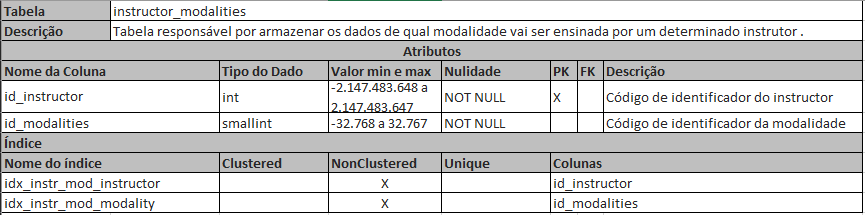


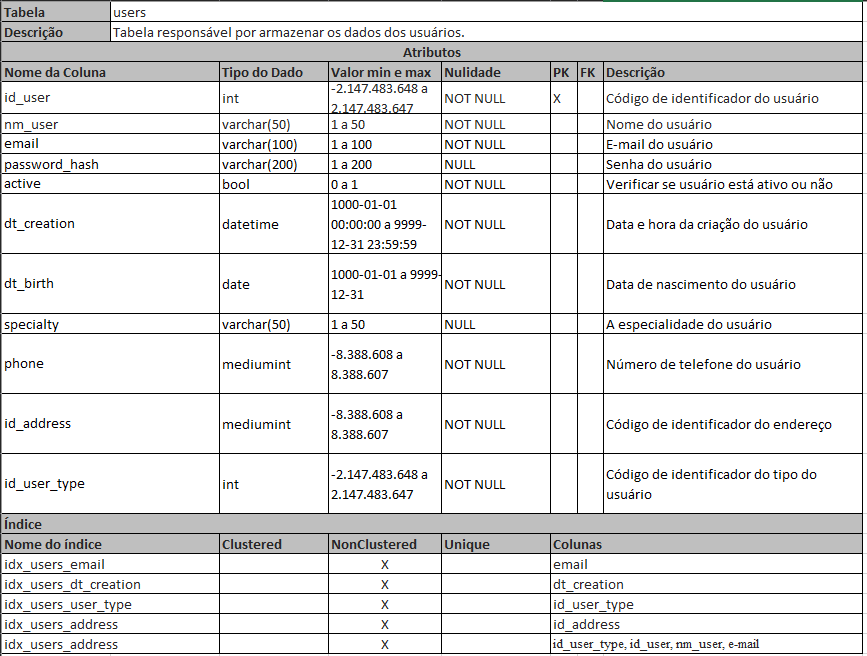
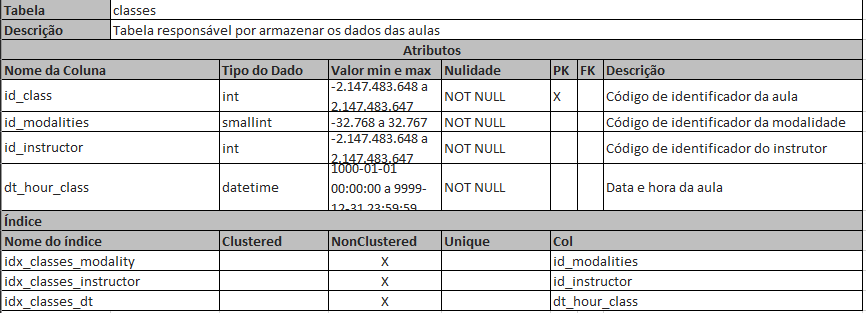


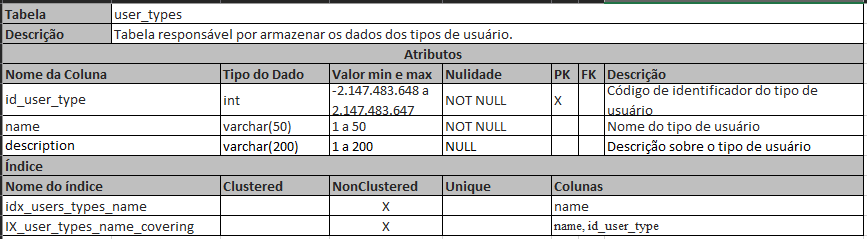


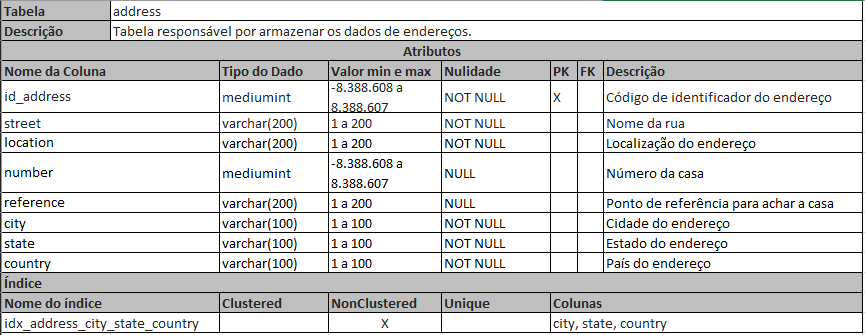


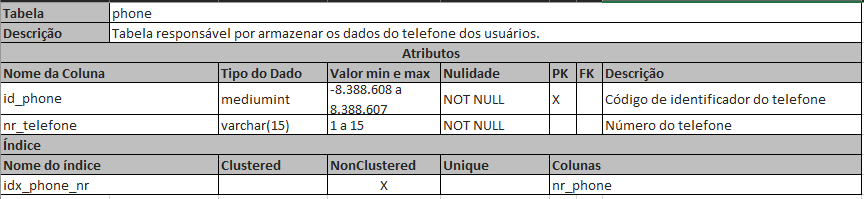












**Script dos comandos DDL para criação do Banco de dados (mínimo 8 tabelas)**

IF OBJECT\_ID('address', 'U') IS NULL

CREATE TABLE [address] (

[id\_address] INT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_address] > 0),

[street] VARCHAR(200) NOT NULL,

[location] VARCHAR(200) NOT NULL,

[number] INT CHECK ([number] >= 0),

[reference] VARCHAR(200),

[city] VARCHAR(100) NOT NULL,

[state] VARCHAR(100) NOT NULL,

[country] VARCHAR(100) NOT NULL

);

IF OBJECT\_ID('phone', 'U') IS NULL

CREATE TABLE [phone] (

[id\_phone] INT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_phone] > 0),

[nr\_phone] VARCHAR(15) NOT NULL

);

IF OBJECT\_ID('user\_types', 'U') IS NULL

CREATE TABLE [user\_types] (

[id\_user\_type] TINYINT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_user\_type] > 0),

[name] VARCHAR(50) NOT NULL,

[description] VARCHAR(200)

);

IF OBJECT\_ID('users', 'U') IS NULL

CREATE TABLE [users] (

[id\_user] INT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_user] > 0),

[nm\_user] VARCHAR(50) NOT NULL,

[email] VARCHAR(100) NOT NULL,

[password\_hash] VARCHAR(200),

[active] BIT NOT NULL,

[dt\_creation] DATETIME NOT NULL,

[dt\_birth] DATE NOT NULL,

[specialty] VARCHAR(50),

[id\_phone] INT NOT NULL CHECK ([id\_phone] > 0),

[id\_address] INT NOT NULL CHECK ([id\_address] > 0),

[id\_user\_type] TINYINT NOT NULL CHECK ([id\_user\_type] > 0),

CONSTRAINT FK\_Users\_Phone FOREIGN KEY ([id\_phone]) REFERENCES [phone] ([id\_phone]),

CONSTRAINT FK\_Users\_Address FOREIGN KEY ([id\_address]) REFERENCES [address] ([id\_address]),

CONSTRAINT FK\_Users\_UserTypes FOREIGN KEY ([id\_user\_type]) REFERENCES [user\_types] ([id\_user\_type])

);

IF OBJECT\_ID('modalities', 'U') IS NULL

CREATE TABLE [modalities] (

[id\_modalities] TINYINT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_modalities] > 0),

[name] VARCHAR(50) NOT NULL,

[description] VARCHAR(200)

);

IF OBJECT\_ID('instructor\_modalities', 'U') IS NULL

CREATE TABLE [instructor\_modalities] (

[id\_instructor] INT NOT NULL CHECK ([id\_instructor] > 0),

[id\_modalities] TINYINT NOT NULL CHECK ([id\_modalities] > 0),

PRIMARY KEY (id\_instructor, id\_modalities),

CONSTRAINT FK\_InstructorModalities\_Users FOREIGN KEY ([id\_instructor]) REFERENCES [users] ([id\_user]),

CONSTRAINT FK\_InstructorModalities\_Modalities FOREIGN KEY ([id\_modalities]) REFERENCES [modalities] ([id\_modalities])

);

IF OBJECT\_ID('plans', 'U') IS NULL

CREATE TABLE [plans] (

[id\_plan] TINYINT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_plan] > 0),

[name] VARCHAR(50) NOT NULL,

[description] VARCHAR(200) NOT NULL,

[price] NUMERIC(5,2) NOT NULL CHECK ([price] >= 0),

[duration\_mnth] TINYINT NOT NULL CHECK ([duration\_mnth] > 0)

);

IF OBJECT\_ID('plan\_modalities', 'U') IS NULL

CREATE TABLE [plan\_modalities] (

[id\_plan] TINYINT NOT NULL CHECK ([id\_plan] > 0),

[id\_modalities] TINYINT NOT NULL CHECK ([id\_modalities] > 0),

PRIMARY KEY (id\_plan, id\_modalities),

CONSTRAINT FK\_PlanModalities\_Plans FOREIGN KEY ([id\_plan]) REFERENCES [plans] ([id\_plan]),

CONSTRAINT FK\_PlanModalities\_Modalities FOREIGN KEY ([id\_modalities]) REFERENCES [modalities] ([id\_modalities])

);

IF OBJECT\_ID('subscriptions', 'U') IS NULL

CREATE TABLE [subscriptions] (

[id\_subscriptions] INT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_subscriptions] > 0),

[id\_plan] TINYINT NOT NULL CHECK ([id\_plan] > 0),

[id\_user] INT NOT NULL CHECK ([id\_user] > 0),

[active] BIT NOT NULL,

[dt\_start] DATE NOT NULL,

[dt\_end] DATE NULL,

CONSTRAINT FK\_Subscriptions\_Plans FOREIGN KEY ([id\_plan]) REFERENCES [plans] ([id\_plan]),

CONSTRAINT FK\_Subscriptions\_Users FOREIGN KEY ([id\_user]) REFERENCES [users] ([id\_user])

);

IF OBJECT\_ID('payment\_method', 'U') IS NULL

CREATE TABLE [payment\_method] (

[id\_method] TINYINT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_method] > 0),

[name\_method] VARCHAR(20) NOT NULL

);

IF OBJECT\_ID('payments', 'U') IS NULL

CREATE TABLE [payments] (

[id\_payment] INT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_payment] > 0),

[id\_subscriptions] INT NOT NULL CHECK ([id\_subscriptions] > 0),

[dt\_payment] DATE NOT NULL,

[price] NUMERIC(5,2) NOT NULL CHECK ([price] >= 0),

[id\_method] TINYINT NOT NULL CHECK ([id\_method] > 0),

CONSTRAINT FK\_Payments\_Subscriptions FOREIGN KEY ([id\_subscriptions]) REFERENCES [subscriptions] ([id\_subscriptions]),

CONSTRAINT FK\_Payments\_PaymentMethod FOREIGN KEY ([id\_method]) REFERENCES [payment\_method] ([id\_method])

);

IF OBJECT\_ID('classes', 'U') IS NULL

CREATE TABLE [classes] (

[id\_class] INT PRIMARY KEY IDENTITY(1,1) CHECK ([id\_class] > 0),

[id\_modalities] TINYINT NOT NULL CHECK ([id\_modalities] > 0),

[id\_instructor] INT NOT NULL CHECK ([id\_instructor] > 0),

[dt\_hour\_class] DATETIME NOT NULL,

CONSTRAINT FK\_Classes\_Modalities FOREIGN KEY ([id\_modalities]) REFERENCES [modalities] ([id\_modalities]),

CONSTRAINT FK\_Classes\_Instructor FOREIGN KEY ([id\_instructor]) REFERENCES [users] ([id\_user])

);

**Script que popula as tabelas do Banco de dados (mínimo 10 itens principais)**

INSERT INTO [address] ([street], [location], [number], [reference], [city], [state], [country])

VALUES

('Rua das Flores', 'Centro', 101, 'Próximo ao Parque da Cidade', 'São Paulo', 'SP', 'Brasil'),

('Avenida Paulista', 'Bela Vista', 202, 'Ao lado da Biblioteca Municipal', 'São Paulo', 'SP', 'Brasil'),

('Rua Augusta', 'Consolação', 303, 'Em frente ao Shopping', 'São Paulo', 'SP', 'Brasil'),

('Rua XV de Novembro', 'Centro', 404, 'Próximo ao Museu', 'Curitiba', 'PR', 'Brasil'),

('Avenida Atlântica', 'Copacabana', 505, 'Perto da Prefeitura', 'Rio de Janeiro', 'RJ', 'Brasil'),

('Rua do Ouvidor', 'Centro', 606, 'Próximo à Escola', 'Rio de Janeiro', 'RJ', 'Brasil'),

('Avenida Boa Viagem', 'Boa Viagem', 707, 'Perto do Estádio', 'Recife', 'PE', 'Brasil'),

('Rua das Laranjeiras', 'Funcionários', 808, 'Atrás do Hospital', 'Belo Horizonte', 'MG', 'Brasil'),

('Avenida Beira Mar', 'Meireles', 909, 'Próximo ao Rio', 'Fortaleza', 'CE', 'Brasil'),

('Rua da Praia', 'Centro Histórico', 111, 'Perto do Mercado', 'Porto Alegre', 'RS', 'Brasil'),

('Rua dos Bandeirantes', 'Vila Madalena', 123, 'Próximo ao Metrô', 'São Paulo', 'SP', 'Brasil'),

('Avenida Brasil', 'Zona Norte', 456, 'Ao lado do Shopping Norte', 'Rio de Janeiro', 'RJ', 'Brasil'),

('Rua Marechal Deodoro', 'Centro', 789, 'Em frente à Praça', 'Curitiba', 'PR', 'Brasil'),

('Avenida Goiás', 'Setor Central', 321, 'Próximo ao Teatro', 'Goiânia', 'GO', 'Brasil'),

('Rua Halfeld', 'Centro', 654, 'Perto da Catedral', 'Juiz de Fora', 'MG', 'Brasil'),

('Avenida Presidente Vargas', 'Centro', 987, 'Ao lado do Mercado Central', 'Fortaleza', 'CE', 'Brasil'),

('Rua José Bonifácio', 'Centro Histórico', 147, 'Próximo ao Museu', 'São Luís', 'MA', 'Brasil'),

('Avenida Sete de Setembro', 'Centro', 258, 'Em frente ao Pelourinho', 'Salvador', 'BA', 'Brasil'),

('Rua Barão do Rio Branco', 'Centro', 369, 'Próximo à Rodoviária', 'Campo Grande', 'MS', 'Brasil'),

('Avenida Afonso Pena', 'Centro', 741, 'Ao lado do Parque Municipal', 'Belo Horizonte', 'MG', 'Brasil'),

('Rua das Flores', 'Centro', 101, 'Próximo ao Parque da Cidade', 'São Paulo', 'SP', 'Brasil'),

('Avenida Paulista', 'Bela Vista', 202, 'Ao lado da Biblioteca Municipal', 'São Paulo', 'SP', 'Brasil'),

('Rua Augusta', 'Consolação', 303, 'Em frente ao Shopping', 'São Paulo', 'SP', 'Brasil'),

('Rua XV de Novembro', 'Centro', 404, 'Próximo ao Museu', 'Curitiba', 'PR', 'Brasil'),

('Avenida Atlântica', 'Copacabana', 505, 'Perto da Prefeitura', 'Rio de Janeiro', 'RJ', 'Brasil'),

('Rua do Ouvidor', 'Centro', 606, 'Próximo à Escola', 'Rio de Janeiro', 'RJ', 'Brasil'),

('Avenida Boa Viagem', 'Boa Viagem', 707, 'Perto do Estádio', 'Recife', 'PE', 'Brasil'),

('Rua das Laranjeiras', 'Funcionários', 808, 'Atrás do Hospital', 'Belo Horizonte', 'MG', 'Brasil'),

('Avenida Beira Mar', 'Meireles', 909, 'Próximo ao Rio', 'Fortaleza', 'CE', 'Brasil'),

('Rua da Praia', 'Centro Histórico', 111, 'Perto do Mercado', 'Porto Alegre', 'RS', 'Brasil'),

('Rua dos Bandeirantes', 'Vila Madalena', 123, 'Próximo ao Metrô', 'São Paulo', 'SP', 'Brasil'),

('Avenida Brasil', 'Zona Norte', 456, 'Ao lado do Shopping Norte', 'Rio de Janeiro', 'RJ', 'Brasil'),

('Rua Marechal Deodoro', 'Centro', 789, 'Em frente à Praça', 'Curitiba', 'PR', 'Brasil'),

('Avenida Goiás', 'Setor Central', 321, 'Próximo ao Teatro', 'Goiânia', 'GO', 'Brasil'),

('Rua Halfeld', 'Centro', 654, 'Perto da Catedral', 'Juiz de Fora', 'MG', 'Brasil'),

('Avenida Presidente Vargas', 'Centro', 987, 'Ao lado do Mercado Central', 'Fortaleza', 'CE', 'Brasil'),

('Rua José Bonifácio', 'Centro Histórico', 147, 'Próximo ao Museu', 'São Luís', 'MA', 'Brasil'),

('Avenida Sete de Setembro', 'Centro', 258, 'Em frente ao Pelourinho', 'Salvador', 'BA', 'Brasil'),

('Rua Barão do Rio Branco', 'Centro', 369, 'Próximo à Rodoviária', 'Campo Grande', 'MS', 'Brasil'),

('Avenida Afonso Pena', 'Centro', 741, 'Ao lado do Parque Municipal', 'Belo Horizonte', 'MG', 'Brasil'),

('Rua das Flores', 'Centro', 101, 'Próximo ao Parque da Cidade', 'São Paulo', 'SP', 'Brasil'),

('Avenida Paulista', 'Bela Vista', 202, 'Ao lado da Biblioteca Municipal', 'São Paulo', 'SP', 'Brasil'),

('Rua Augusta', 'Consolação', 303, 'Em frente ao Shopping', 'São Paulo', 'SP', 'Brasil'),

('Rua XV de Novembro', 'Centro', 404, 'Próximo ao Museu', 'Curitiba', 'PR', 'Brasil'),

('Avenida Atlântica', 'Copacabana', 505, 'Perto da Prefeitura', 'Rio de Janeiro', 'RJ', 'Brasil'),

('Rua do Ouvidor', 'Centro', 606, 'Próximo à Escola', 'Rio de Janeiro', 'RJ', 'Brasil'),

('Avenida Boa Viagem', 'Boa Viagem', 707, 'Perto do Estádio', 'Recife', 'PE', 'Brasil'),

('Rua das Laranjeiras', 'Funcionários', 808, 'Atrás do Hospital', 'Belo Horizonte', 'MG', 'Brasil'),

('Avenida Beira Mar', 'Meireles', 909, 'Próximo ao Rio', 'Fortaleza', 'CE', 'Brasil'),

('Rua da Praia', 'Centro Histórico', 111, 'Perto do Mercado', 'Porto Alegre', 'RS', 'Brasil'),

('Rua dos Bandeirantes', 'Vila Madalena', 123, 'Próximo ao Metrô', 'São Paulo', 'SP', 'Brasil'),

('Avenida Brasil', 'Zona Norte', 456, 'Ao lado do Shopping Norte', 'Rio de Janeiro', 'RJ', 'Brasil'),

('Rua Marechal Deodoro', 'Centro', 789, 'Em frente à Praça', 'Curitiba', 'PR', 'Brasil'),

('Avenida Goiás', 'Setor Central', 321, 'Próximo ao Teatro', 'Goiânia', 'GO', 'Brasil'),

('Rua Halfeld', 'Centro', 654, 'Perto da Catedral', 'Juiz de Fora', 'MG', 'Brasil'),

('Avenida Presidente Vargas', 'Centro', 987, 'Ao lado do Mercado Central', 'Fortaleza', 'CE', 'Brasil'),

('Rua José Bonifácio', 'Centro Histórico', 147, 'Próximo ao Museu', 'São Luís', 'MA', 'Brasil'),

('Avenida Sete de Setembro', 'Centro', 258, 'Em frente ao Pelourinho', 'Salvador', 'BA', 'Brasil'),

('Rua Barão do Rio Branco', 'Centro', 369, 'Próximo à Rodoviária', 'Campo Grande', 'MS', 'Brasil'),

('Avenida Afonso Pena', 'Centro', 741, 'Ao lado do Parque Municipal', 'Belo Horizonte', 'MG', 'Brasil');

INSERT INTO [phone] ([nr\_phone])

VALUES

('(11) 99123-4567'),

('(11) 98234-5678'),

('(21) 97345-6789'),

('(21) 96456-7890'),

('(31) 95567-8901'),

('(41) 94678-9012'),

('(51) 93789-0123'),

('(81) 92890-1234'),

('(85) 91901-2345'),

('(47) 90012-3456'),

('(11) 99123-4567'),

('(11) 98234-5678'),

('(21) 97345-6789'),

('(21) 96456-7890'),

('(31) 95567-8901'),

('(41) 94678-9012'),

('(51) 93789-0123'),

('(81) 92890-1234'),

('(85) 91901-2345'),

('(47) 90012-3456'),

('(11) 99123-4567'),

('(11) 98234-5678'),

('(21) 97345-6789'),

('(21) 96456-7890'),

('(31) 95567-8901'),

('(41) 94678-9012'),

('(51) 93789-0123'),

('(81) 92890-1234'),

('(85) 91901-2345'),

('(47) 90012-3456');

INSERT INTO [user\_types] ([name], [description])

VALUES

('Administrador', 'Administrador com acesso completo'),

('Instrutor', 'Instrutor com acesso limitado'),

('Membro', 'Membro regular com acesso básico'),

('Recepcionista', 'Responsável pela recepção e agendamentos'),

('Gerente', 'Supervisiona as operações'),

('Personal Trainer', 'Responsável por treinamento personalizado'),

('Nutricionista', 'Fornece orientação dietética'),

('Faxineiro', 'Responsável pela limpeza'),

('Técnico', 'Manutenção dos equipamentos da academia'),

('Marketing', 'Responsável pelas tarefas promocionais');

INSERT INTO [users] ([nm\_user], [email], [password\_hash], [active], [dt\_creation], [dt\_birth], [specialty], [id\_phone], [id\_address], [id\_user\_type])

VALUES

('João Silva', 'joao.silva@email.com', 'hash1', 1, '2023-01-01', '1990-01-01', 'Yoga', 1, 1, 2),

('Maria Santos', 'maria.santos@email.com', 'hash2', 1, '2023-01-02', '1985-02-02', 'Pilates', 2, 2, 2),

('Ana Oliveira', 'ana.oliveira@email.com', 'hash3', 1, '2023-01-03', '1995-03-03', 'Cardio', 3, 3, 3),

('Carlos Ferreira', 'carlos.ferreira@email.com', 'hash4', 1, '2023-01-04', '1988-04-04', 'Musculação', 4, 4, 2),

('Fernanda Costa', 'fernanda.costa@email.com', 'hash5', 1, '2023-01-05', '1992-05-05', 'Alongamento', 5, 5, 3),

('Ricardo Souza', 'ricardo.souza@email.com', 'hash6', 1, '2023-01-06', '1983-06-06', 'Crossfit', 6, 6, 2),

('Patrícia Lima', 'patricia.lima@email.com', 'hash7', 1, '2023-01-07', '1991-07-07', 'HIIT', 7, 7, 3),

('Bruno Alves', 'bruno.alves@email.com', 'hash8', 1, '2023-01-08', '1989-08-08', 'Spinning', 8, 8, 2),

('Juliana Pereira', 'juliana.pereira@email.com', 'hash9', 1, '2023-01-09', '1993-09-09', 'Zumba', 9, 9, 3),

('Rodrigo Martins', 'rodrigo.martins@email.com', 'hash10', 1, '2023-01-10', '1987-10-10', 'Boxe', 10, 10, 2),

('João Silva', 'joao.silva@email.com', 'hash1', 1, '2023-01-01', '1990-01-01', 'Yoga', 11, 11, 3),

('Maria Santos', 'maria.santos@email.com', 'hash2', 1, '2023-01-02', '1985-02-02', 'Pilates', 12, 12, 3),

('Ana Oliveira', 'ana.oliveira@email.com', 'hash3', 1, '2023-01-03', '1995-03-03', 'Cardio', 13, 13, 3),

('Carlos Ferreira', 'carlos.ferreira@email.com', 'hash4', 1, '2023-01-04', '1988-04-04', 'Musculação', 14, 14, 3),

('Fernanda Costa', 'fernanda.costa@email.com', 'hash5', 1, '2023-01-05', '1992-05-05', 'Alongamento', 15, 15, 3),

('Ricardo Souza', 'ricardo.souza@email.com', 'hash6', 1, '2023-01-06', '1983-06-06', 'Crossfit', 16, 16, 3),

('Patrícia Lima', 'patricia.lima@email.com', 'hash7', 1, '2023-01-07', '1991-07-07', 'HIIT', 17, 17, 3),

('Bruno Alves', 'bruno.alves@email.com', 'hash8', 1, '2023-01-08', '1989-08-08', 'Spinning', 18, 18, 3),

('Juliana Pereira', 'juliana.pereira@email.com', 'hash9', 1, '2023-01-09', '1993-09-09', 'Zumba', 19, 19, 3),

('Rodrigo Martins', 'rodrigo.martins@email.com', 'hash10', 0, '2023-01-10', '1987-10-10', 'Boxe', 20, 20, 3),

('João Silva', 'joao.silva@email.com', 'hash1', 0, '2023-01-01', '1990-01-01', 'Yoga', 21, 21, 3),

('Maria Santos', 'maria.santos@email.com', 'hash2', 0, '2023-01-02', '1985-02-02', 'Pilates', 22, 22, 3),

('Ana Oliveira', 'ana.oliveira@email.com', 'hash3', 0, '2023-01-03', '1995-03-03', 'Cardio', 23, 23, 3),

('Carlos Ferreira', 'carlos.ferreira@email.com', 'hash4', 0, '2023-01-04', '1988-04-04', 'Musculação', 24, 24, 3),

('Fernanda Costa', 'fernanda.costa@email.com', 'hash5', 0, '2023-01-05', '1992-05-05', 'Alongamento', 25, 25, 3),

('Ricardo Souza', 'ricardo.souza@email.com', 'hash6', 0, '2023-01-06', '1983-06-06', 'Crossfit', 26, 26, 3),

('Patrícia Lima', 'patricia.lima@email.com', 'hash7', 0, '2023-01-07', '1991-07-07', 'HIIT', 27, 27, 3),

('Bruno Alves', 'bruno.alves@email.com', 'hash8', 0, '2023-01-08', '1989-08-08', 'Spinning', 28, 28, 3),

('Juliana Pereira', 'juliana.pereira@email.com', 'hash9', 0, '2023-01-09', '1993-09-09', 'Zumba', 29, 29, 3),

('Rodrigo Martins', 'rodrigo.martins@email.com', 'hash10', 0, '2023-01-10', '1987-10-10', 'Boxe', 30, 30, 3);

INSERT INTO [modalities] ([name], [description])

VALUES

('Yoga', 'Prática de disciplinas físicas, mentais e espirituais'),

('Pilates', 'Sistema de condicionamento físico desenvolvido no início do século XX'),

('Cardio', 'Exercícios que aumentam a frequência cardíaca e melhoram a saúde cardiovascular'),

('Crossfit', 'Treinamento funcional de alta intensidade'),

('HIIT', 'Treinamento intervalado de alta intensidade'),

('Zumba', 'Treino aeróbico baseado em dança'),

('Spinning', 'Treinos de ciclismo estacionário'),

('Musculação', 'Exercícios para construção muscular'),

('Boxe', 'Esporte de combate e exercício'),

('Alongamento', 'Exercícios de flexibilidade e relaxamento muscular');

INSERT INTO [plans] ([name], [description], [price], [duration\_mnth])

VALUES

('Plano Básico', 'Acesso às instalações da academia e aulas em grupo', 89.90, 1),

('Plano Premium', 'Acesso às instalações, aulas em grupo e personal trainer', 149.90, 1),

('Plano Anual', 'Acesso às instalações e aulas em grupo por um ano', 899.90, 12),

('Plano Estudante', 'Plano com desconto para estudantes', 59.90, 1),

('Plano Família', 'Acesso compartilhado para até 4 membros da família', 299.90, 1),

('Plano Idoso', 'Tarifas especiais para idosos', 69.90, 1),

('Plano Final de Semana', 'Acesso à academia apenas nos fins de semana', 49.90, 1),

('Plano VIP', 'Acesso completo com benefícios exclusivos', 299.90, 1),

('Plano Experimental', 'Teste gratuito de 7 dias', 0.00, 1),

('Plano Corporativo', 'Plano com desconto para empresas', 599.90, 12);

INSERT INTO [instructor\_modalities] ([id\_instructor], [id\_modalities])

VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10);

INSERT INTO [plan\_modalities] ([id\_plan], [id\_modalities])

VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10);

INSERT INTO [subscriptions] ([id\_plan], [id\_user], [active], [dt\_start])

VALUES

(1, 1, 1, '2023-01-01'),

(2, 2, 1, '2024-01-02'),

(3, 3, 1, '2023-06-03'),

(4, 4, 1, '2022-01-04'),

(5, 5, 1, '2021-06-05'),

(6, 6, 1, '2025-06-06'),

(7, 7, 1, '2022-01-07'),

(8, 8, 1, '2021-01-08'),

(9, 9, 1, '2020-06-09'),

(10, 10, 1, '2023-01-10'),

(1, 11, 1, '2023-06-01'),

(2, 12, 1, '2024-06-02'),

(3, 13, 1, '2024-01-03'),

(4, 14, 1, '2025-01-04'),

(5, 15, 1, '2021-06-05'),

(6, 16, 1, '2022-01-06'),

(7, 17, 1, '2022-01-07'),

(8, 18, 1, '2020-06-08'),

(9, 19, 1, '2023-01-09');

INSERT INTO [subscriptions] ([id\_plan], [id\_user], [active], [dt\_start], [dt\_end])

VALUES

(10, 20, 1, '2023-06-10', '2024-06-10'),

(1, 21, 1, '2024-01-01', '2024-03-01'),

(2, 22, 1, '2021-01-02', '2021-05-02'),

(3, 23, 1, '2021-01-03', '2022-07-03'),

(4, 24, 1, '2022-01-04', '2023-08-04'),

(5, 25, 1, '2023-06-05', '2024-09-05'),

(6, 26, 1, '2024-01-06', '2025-05-06'),

(7, 27, 1, '2025-01-07', '2025-02-07'),

(8, 28, 1, '2023-01-08', '2023-04-08'),

(9, 29, 1, '2023-06-09', '2024-02-09'),

(10, 30, 1, '2025-01-10', '2025-09-10');

INSERT INTO [payment\_method] ([name\_method])

VALUES

('Cartão de Crédito'),

('Cartão de Débito'),

('PIX'),

('Boleto Bancário'),

('Dinheiro'),

('Transf. Bancária'),

('Criptomoeda'),

('Google Pay'),

('Apple Pay'),

('Vale-presente');

INSERT INTO [payments] ([id\_subscriptions], [dt\_payment], [price], [id\_method])

VALUES

(1, '2023-01-01', 89.90, 1),

(2, '2024-01-02', 149.90, 2),

(3, '2023-01-03', 899.90, 3),

(4, '2022-01-04', 59.90, 4),

(5, '2021-01-05', 299.90, 5),

(6, '2025-01-06', 69.90, 6),

(7, '2022-01-07', 49.90, 7),

(8, '2021-01-08', 299.90, 8),

(9, '2020-01-09', 0.00, 9),

(10, '2023-01-10', 599.90, 10),

(11, '2023-01-01', 89.90, 1),

(12, '2024-01-02', 149.90, 2),

(13, '2024-01-03', 899.90, 3),

(14, '2025-01-04', 59.90, 4),

(15, '2021-01-05', 299.90, 5),

(16, '2022-01-06', 69.90, 6),

(17, '2022-01-07', 49.90, 7),

(18, '2020-01-08', 299.90, 8),

(19, '2023-01-09', 0.00, 9),

(20, '2023-01-10', 599.90, 10),

(21, '2024-01-01', 89.90, 1),

(22, '2021-01-02', 149.90, 2),

(23, '2021-01-03', 899.90, 3),

(24, '2022-01-04', 59.90, 4),

(25, '2023-01-05', 299.90, 5),

(26, '2024-01-06', 69.90, 6),

(27, '2025-01-07', 49.90, 7),

(28, '2023-01-08', 299.90, 8),

(29, '2023-01-09', 0.00, 9),

(30, '2025-01-10', 599.90, 10);

INSERT INTO [classes] ([id\_modalities], [id\_instructor], [dt\_hour\_class])

VALUES

(1, 1, '2023-01-01 10:00:00'),

(2, 2, '2023-01-02 11:00:00'),

(3, 3, '2023-01-03 12:00:00'),

(4, 4, '2023-01-04 13:00:00'),

(5, 5, '2023-01-05 14:00:00'),

(6, 6, '2023-01-06 15:00:00'),

(7, 7, '2023-01-07 16:00:00'),

(8, 8, '2023-01-08 17:00:00'),

(9, 9, '2023-01-09 18:00:00'),

(10, 10, '2023-01-10 19:00:00');

**Principais consultas mapeadas baseadas em regras de negócio (mínimo 6)**

1. **Qual(is) carro(s) possui(em) mais sinistros cadastrados?**

Neste cenário, podemos interpretar "carros" como os "usuários" e "sinistros" (eventos) como os "pagamentos" que eles realizaram. A pergunta se torna: "Qual usuário realizou mais pagamentos?".

**Pergunta Adaptada:** Qual usuário possui mais pagamentos registrados?

SQL

SELECT

u.nm\_user,

COUNT(p.id\_payment) AS TotalDePagamentos

FROM

payments p

JOIN

subscriptions s ON p.id\_subscriptions = s.id\_subscription

JOIN

users u ON s.id\_user = u.id\_user

GROUP BY

u.nm\_user

ORDER BY

TotalDePagamentos DESC;

**2. Quantos clientes cadastrados possuem letra “a” no meio e não no final?**

A lógica aqui é a mesma, apenas trocamos a tabela cliente por users e a coluna nome por nm\_user.

**Pergunta Adaptada:** Quantos usuários cadastrados possuem a letra "a" no nome, mas não como última letra?

SQL

-- Para listar os nomes que correspondem ao critério:

SELECT

nm\_user

FROM

users

WHERE

nm\_user LIKE '%a%' AND nm\_user NOT LIKE '%a';

-- Para contar a quantidade total, como solicitado na pergunta:

SELECT

COUNT(\*) AS TotalDeUsuarios

FROM

users

WHERE

nm\_user LIKE '%a%' AND nm\_user NOT LIKE '%a';

**3. Quantos clientes não possuem apólice?**

Aqui, "clientes" são users e "apólices" são subscriptions. Usamos a mesma abordagem com LEFT JOIN para encontrar usuários que não têm uma assinatura correspondente.

**Pergunta Adaptada:** Quantos usuários não possuem uma assinatura (subscription)?

SQL

-- Para listar os usuários sem assinatura:

SELECT

u.nm\_user

FROM

users u

LEFT JOIN

subscriptions s ON u.id\_user = s.id\_user

WHERE

s.id\_subscription IS NULL;

-- Para contar a quantidade total:

SELECT

COUNT(u.id\_user) AS UsuariosSemAssinatura

FROM

users u

LEFT JOIN

subscriptions s ON u.id\_user = s.id\_user

WHERE

s.id\_subscription IS NULL;

*Nos dados de exemplo fornecidos, todos os usuários possuem uma assinatura, então o resultado desta consulta seria 0.*

**4. Quais carros não possuem apólice?**

A pergunta original é muito parecida com a anterior. Para torná-la mais útil no contexto da academia, podemos adaptá-la para perguntar: "Quais modalidades de exercício não estão incluídas em nenhum plano?". Isso nos ajuda a identificar serviços que não estão sendo oferecidos aos clientes.

**Pergunta Adaptada:** Quais modalidades não estão incluídas em nenhum plano?

SQL

-- Método 1: Usando LEFT JOIN (geralmente mais performático)

SELECT

m.name

FROM

modalities m

LEFT JOIN

plan\_modalities pm ON m.id\_modalities = pm.id\_modalities

WHERE

pm.id\_plan IS NULL;

-- Método 2: Usando NOT IN (alternativa que também funciona)

SELECT

name

FROM

modalities

WHERE

id\_modalities NOT IN (SELECT DISTINCT id\_modalities FROM plan\_modalities);