**Centro Universitário UNISATC**

Engenharia de Software 3a fase – Banco de Dados II – Prof.Jorge Luiz da Silva

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

**Projeto de banco de dados para um sistema de Gestão de Estacionamento**

Alex Farias De Abreu Nabo - @Alex-Farias

Icaro De Oliveira - @ynd-icaro

Jean Lucas Preis - @JeanPrei

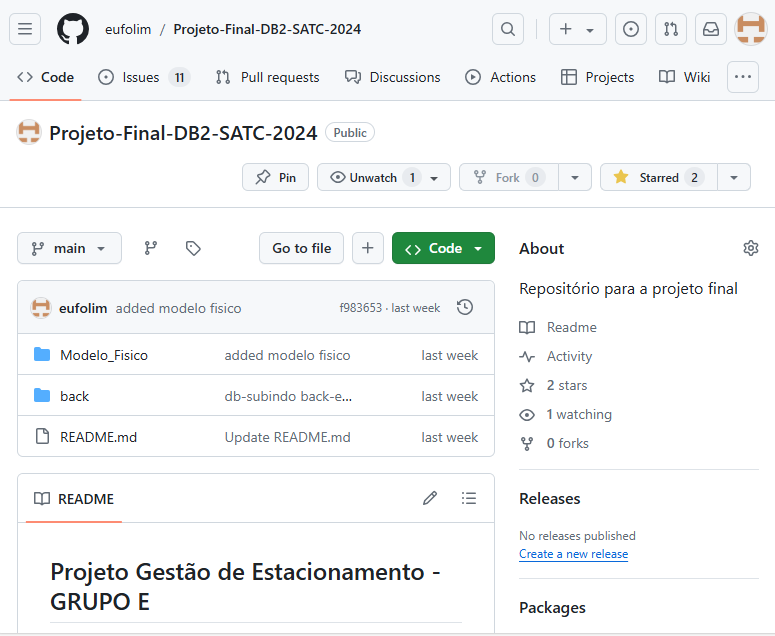
Luiz Antônio Frey Cristiano - @eufolim

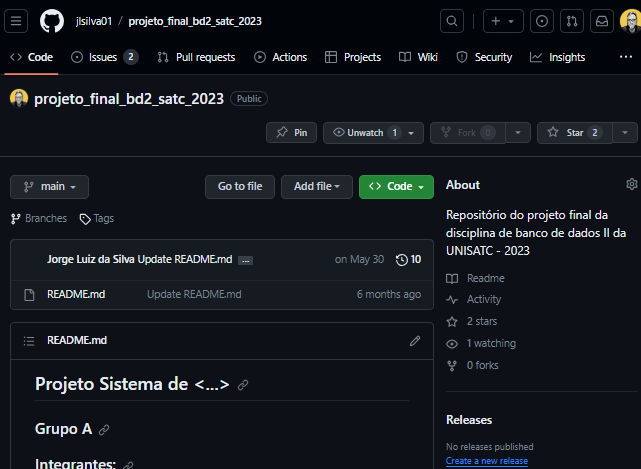
Matheus Araildi - @Araldi42<br>

Criciúma, 26/06/2024

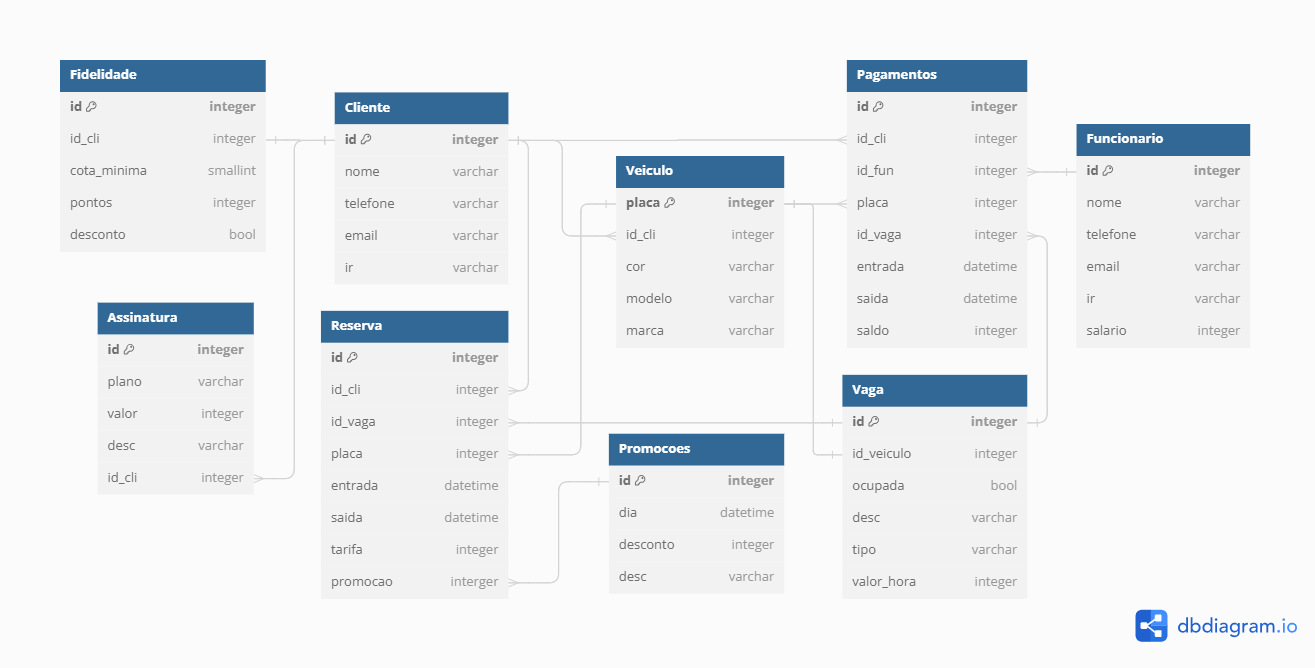
**URL do projeto no GitHub**

<https://github.com/eufolim/Projeto-Final-DB2-SATC-2024>

****

****

**Modelo ER Físico**

****

**Dicionário de Dados**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Vaga | | | | | | |
| **Descrição** |  | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id | | int | 1 – 2,147,483,647 | NOT NULL | X |  |  |
| Id\_veiculo | | char | 7 - 7 | NOT NULL |  | X |  |
| ocupada | | bool | 0 - 1 | NOT NULL |  |  |  |
| desc | | varchar | 1 - 255 | NOT NULL |  |  |  |
| tipo | | varchar | 1 - 255 | NOT NULL |  |  |  |
| Valor\_hora | | int | 1 – 10,000 | NOT NULL |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Veiculo | | | | | | |
| **Descrição** |  | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| placa | | char | 7 - 7 | NOT NULL | X |  |  |
| Id\_cli | | int | 1 - 2,147,483,647 | NOT NULL |  | X |  |
| cor | | varchar | 1 - 50 | NOT NULL |  |  |  |
| modelo | | varchar | 1 - 100 | NOT NULL |  |  |  |
| marca | | varchar | 1 - 50 | NOT NULL |  |  |  |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Cliente | | | | | | |
| **Descrição** |  | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id | | int | 1 – 2,147,483,647 | NOT NULL | X |  |  |
| nome | | varchar | 1 - 110 | NOT NULL |  |  |  |
| telefone | | int | 8 - 13 | NOT NULL |  |  |  |
| email | | varchar | 1 - 255 | NOT NULL |  |  |  |
| ir | | varchar | 1 - 255 | NOT NULL |  |  |  |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Funcionario | | | | | | |
| **Descrição** |  | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id | | int | 1 – 2,147,483,647 | NOT NULL | X |  |  |
| nome | | varchar | 1 - 110 | NOT NULL |  |  |  |
| telefone | | int | 8 - 13 | NOT NULL |  |  |  |
| email | | varchar | 1 - 255 | NOT NULL |  |  |  |
| ir | | varchar | 1 - 255 | NOT NULL |  |  |  |
| salario | | int | 1 - 2,147,483,647 | NOT NULL |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Assinatura | | | | | | |
| **Descrição** |  | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id | | int | 1 – 2,147,483,647 | NOT NULL | X |  |  |
| plano | | varchar | 1 - 110 | NOT NULL |  |  |  |
| valor | | int | 1 - 2,147,483,647 | NOT NULL |  |  |  |
| desck | | varchar | 1 - 255 | NOT NULL |  |  |  |
| Id\_cli | | int | 1 - 2,147,483,647 | NOT NULL |  | X |  |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Pagamentos | | | | | | |
| **Descrição** |  | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id | | int | 1 – 2,147,483,647 | NOT NULL | X |  |  |
| Id\_cli | | int | 1 – 2,147,483,647 | NOT NULL |  | X |  |
| Id\_fun | | int | 1 - 2,147,483,647 | NOT NULL |  | X |  |
| placa | | char | 7 - 7 | NOT NULL |  | X |  |
| Id\_vaga | | int | 1 - 2,147,483,647 | NOT NULL |  | X |  |
| entrada | | datetime | 1000-01-01 00:00:00 - 9999-12-31 23:59:59 |  |  |  |  |
| saida | | datetime | 1000-01-01 00:00:00 - 9999-12-31 23:59:59 |  |  |  |  |
| saldo | | int | 1 - 2,147,483,647 |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |
| **Tabela** | Reserva | | | | | | |
| **Descrição** |  | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id | | int | 1 – 2,147,483,647 | NOT NULL | X |  |  |
| Id\_cli | | varchar | 1 - 2,147,483,647 | NOT NULL |  | X |  |
| Id\_vaga | | int | 1 - 2,147,483,647 | NOT NULL |  | X |  |
| placa | | varchar | 7 - 7 | NOT NULL |  | X |  |
| entrada | | datetime | 1000-01-01 00:00:00 - 9999-12-31 23:59:59 | NOT NULL |  |  |  |
| saida | | datetime | 1000-01-01 00:00:00 - 9999-12-31 23:59:59 | NOT NULL |  |  |  |
| tarifa | | int | 1 - 2,147,483,647 | NOT NULL |  |  |  |
| promoçao | | int | 1 - 2,147,483,647 |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Promoções | | | | | | |
| **Descrição** |  | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id | | int | 1 – 2,147,483,647 | NOT NULL | X |  |  |
| Id\_cli | | int | 1 - 2,147,483,647 | NOT NULL |  | X |  |
| desconto | | int | 1 - 2,147,483,647 | NOT NULL |  |  |  |
| desck | | varchar | 1 - 255 | NOT NULL |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |
| **Tabela** | Fidelidade | | | | | | |
| **Descrição** |  | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id | | int | 1 – 2,147,483,647 | NOT NULL | X |  |  |
| Id\_cli | | int | 1 - 2,147,483,647 | NOT NULL |  | X |  |
| cota\_minima | | smallint | 1 - 32,767 | NOT NULL |  |  |  |
| pontos | | int | 1 - 2,147,483,647 | NOT NULL |  |  |  |
| desconto | | bool | False - true | NOT NULL |  |  |  |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |

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

Create DATABASE estacionamentdb;

CREATE TABLE Cliente (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(255),

telefone VARCHAR(20),

email VARCHAR(255),

ir VARCHAR(20)

);

CREATE TABLE Veiculo (

placa VARCHAR(20) PRIMARY KEY,

id\_cli INT,

cor VARCHAR(50),

modelo VARCHAR(50),

marca VARCHAR(50),

FOREIGN KEY (id\_cli) REFERENCES Cliente(id)

);

CREATE TABLE Vaga (

id INT PRIMARY KEY AUTO\_INCREMENT,

id\_veiculo VARCHAR(20),

ocupada BOOL,

desc VARCHAR(255),

tipo VARCHAR(50),

valor\_hora INT,

FOREIGN KEY (id\_veiculo) REFERENCES Veiculo(placa)

);

CREATE TABLE Funcionario (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(255),

telefone VARCHAR(20),

email VARCHAR(255),

ir VARCHAR(20),

salario INT

);

CREATE TABLE Assinatura (

id INT PRIMARY KEY AUTO\_INCREMENT,

plano VARCHAR(50),

valor INT,

desc VARCHAR(255),

id\_cli INT,

FOREIGN KEY (id\_cli) REFERENCES Cliente(id)

);

CREATE TABLE Pagamentos (

id INT PRIMARY KEY AUTO\_INCREMENT,

id\_cli INT,

id\_vaga INT,

placa VARCHAR(20),

entrada DATETIME,

saida DATETIME,

saldo INT,

total\_horas INT,

FOREIGN KEY (id\_cli) REFERENCES Cliente(id),

FOREIGN KEY (id\_vaga) REFERENCES Vaga(id),

FOREIGN KEY (placa) REFERENCES Veiculo(placa)

);

CREATE TABLE Promocoes (

id INT PRIMARY KEY AUTO\_INCREMENT,

dia DATE,

desconto INT,

desc VARCHAR(255)

);

CREATE TABLE Reserva (

id INT PRIMARY KEY AUTO\_INCREMENT,

id\_cli INT,

id\_vaga INT,

placa VARCHAR(20),

entrada DATETIME,

saida DATETIME,

tarifa INT,

promocao INT,

FOREIGN KEY (id\_cli) REFERENCES Cliente(id),

FOREIGN KEY (id\_vaga) REFERENCES Vaga(id),

FOREIGN KEY (placa) REFERENCES Veiculo(placa),

FOREIGN KEY (promocao) REFERENCES Promocoes(id)

);

-- Tabela de fidelidade e recompensas

CREATE TABLE Fidelidade (

id INT PRIMARY KEY AUTO\_INCREMENT,

id\_cli INT,

pontos INT,

FOREIGN KEY (id\_cli) REFERENCES Cliente(id)

);

-- Trigger para calcular total de horas

DELIMITER //

CREATE TRIGGER calcula\_total\_horas

BEFORE INSERT ON Pagamentos

FOR EACH ROW

BEGIN

DECLARE horas INT;

SET horas = TIMESTAMPDIFF(HOUR, NEW.entrada, NEW.saida);

SET NEW.total\_horas = horas;

END;

//

DELIMITER ;

-- Trigger para atualizar pontos de fidelidade ao adicionar um pagamento

DELIMITER //

CREATE TRIGGER atualiza\_pontos\_pagamento

AFTER INSERT ON Pagamentos

FOR EACH ROW

BEGIN

DECLARE pontos\_atual INT;

SET pontos\_atual = NEW.total\_horas; -- Cada hora equivale a 1 ponto

IF EXISTS (SELECT \* FROM Fidelidade WHERE id\_cli = NEW.id\_cli) THEN

UPDATE Fidelidade

SET pontos = pontos + pontos\_atual

WHERE id\_cli = NEW.id\_cli;

ELSE

INSERT INTO Fidelidade (id\_cli, pontos)

VALUES (NEW.id\_cli, pontos\_atual);

END IF;

END;

//

DELIMITER ;

-- Procedure para resgatar pontos de fidelidade

DELIMITER //

CREATE PROCEDURE resgatar\_pontos(IN p\_id\_cli INT, IN p\_pontos INT)

BEGIN

DECLARE pontos\_atual INT;

SELECT pontos INTO pontos\_atual

FROM Fidelidade

WHERE id\_cli = p\_id\_cli;

IF pontos\_atual >= p\_pontos THEN

UPDATE Fidelidade

SET pontos = pontos - p\_pontos

WHERE id\_cli = p\_id\_cli;

SELECT 'Pontos resgatados com sucesso!' AS resultado;

ELSE

SELECT 'Pontos insuficientes para resgate.' AS resultado;

END IF;

END;

//

DELIMITER ;

INSERT INTO Cliente (nome, telefone, email, ir) VALUES ('João Silva', '99999-9999', 'joao@example.com', '123456789');

-- Inserir um veículo

INSERT INTO Veiculo (placa, id\_cli, cor, modelo, marca) VALUES ('ABC1234', 1, 'Azul', 'Civic', 'Honda');

-- Inserir um pagamento com entrada e saída

INSERT INTO Pagamentos (id\_cli, id\_vaga, placa, entrada, saida, saldo) VALUES (1, 1, 'ABC1234', '2024-06-14 08:00:00', '2024-06-14 10:00:00', 200);

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

/\* ADICIONA ALUNOS \*/

--INSERT INTO aluno (nm\_aluno, email)

-- VALUES ('FULANO DE TAL', 'fulano@satc.edu.br');

--GO

insert into aluno (nm\_aluno, email)

select 'MARISA MELO OLIVEIRA', 'email@satc.edu.br' ;

insert into aluno (nm\_aluno, email)

select 'MURILO CARVALHO CARDOSO', 'email@satc.edu.br' ;

insert into aluno (nm\_aluno, email)

select 'VINICIUS ROCHA RODRIGUES', 'email@satc.edu.br' ;

/\* ADICIONA AS ALTERNATIVAS DE CADA QUESTAO\*/

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (1, 'Dados / Comunicação / Conhecimento' ,0);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (1, 'Análise / Dados / Conhecimento' ,0);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (1, 'Informação / Compreensão / Comunicação' ,0);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (1, 'Dados / Informação / Conhecimento' ,1);

GO

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (2, 'F, V, F, F, V', 1);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (2, 'V, F, F, V, F', 0 );

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (2, 'V, V, V, F, V', 0);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (2, 'F, F, V, V, F', 0);

GO

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

--13. (Aberta) Utilizando a linguagem SQL, execute consultas que possam responder a seguinte questão: Qual(is) Carro(s) possui(em) mais sinistros cadastrados?

select s.placa, COUNT(\*)

from sinistro s inner join carro c

on s.placa = c.placa

group by s.placa

order by COUNT(\*) desc

--14. (Aberta) Quantos clientes cadastrados possuem letra “a” no meio e não no final?

select \* --count(\*)

from cliente

where nome like '%a%'

and nome not like '%a'

--15. (Aberta) Quantos clientes não possuem apólice?

select nome, cod\_apolice --\* --count(\*)

from cliente c left join apolice a

on c.cod\_cliente = a.cod\_cliente

where cod\_apolice is null

select nome, cod\_apolice

from cliente c inner join apolice a on c.cod\_cliente = a.cod\_cliente

--16. (Aberta) Quais carros não possuem apólice?

select \* --count(\*)

from carro c left join apolice a on c.placa = a.placa

where cod\_apolice is null

select \*

from apolice

where placa not in (select placa from carro)