**Sprint V – BD: Create e Read**

**Descrevendo o Projeto de Banco de Dados**

A empresa XYZ deseja que seja criado um banco de dados de um sistema de controle de veículos e consultores que visitam os clientes. Este sistema será utilizado pela empresa e alimentado pelos próprios consultores que precisam lançar a visita informando a data que ocorreu a visita, qual o cliente que foi visitado, qual o veículo utilizado e qual o KM de saída e de retorno do veículo.

**Restrições e premissas:**

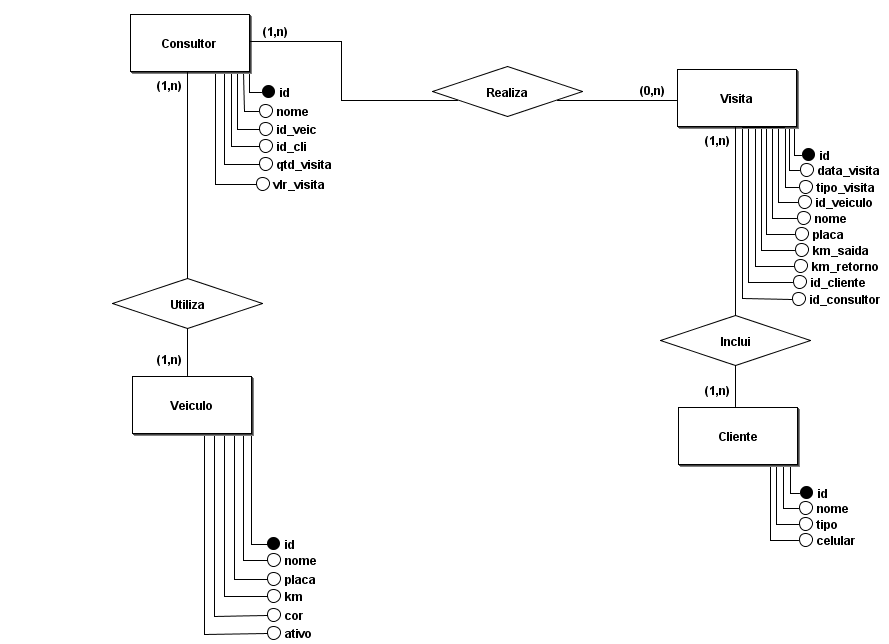
■ Uma Visita poderá conter apenas um Cliente.

■ Uma Visita poderá conter apenas um Consultor.

■ Um Consultor poderá utilizar um Veículo por vez.

■ Não teremos cadastros de Empresas, pois não temos Filiais.

**DER (Diagrama de Entidade e Relacionamento)**

****

**TRABALHANDO COM SQL**

Neste tópico, vamos trabalhar com a implementação das entidades criadas no DER, abordando tanto as metodologias do modelo DDL (Data Definition Language) e DML (Data Manipulation Language).

**TRABALHANDO COM DDL – DATA DEFINITION LANGUAGE**

■ Criação da Database

CREATE DATABASE "empresa\_XYZ"

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'Portuguese\_Brazil.1252'

LC\_CTYPE = 'Portuguese\_Brazil.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;

■ Criação da Schema

CREATE SCHEMA LC;

■ Criando as Tabelas

■ Cliente

CREATE TABLE LC.Cliente (

id serial NOT NULL PRIMARY KEY,

nome VARCHAR (50) NOT NULL,

tipo CHAR NOT NULL);

COMMENT ON COLUMN LC.Cliente.tipo IS 'F - Física / J - Jurídica';

ALTER TABLE LC.Cliente ADD COLUMN celular VARCHAR(20);

■ Veículo

CREATE TABLE LC.Veiculo (

id serial NOT NULL,

nome VARCHAR (50) NOT NULL,

placa VARCHAR (12) NOT NULL,

km integer NOT NULL,

cor VARCHAR (20) NOT NULL,

ativo CHAR (1),

PRIMARY KEY (id));

■ Consultor

CREATE TABLE LC.Consultor (

id bigserial NOT NULL PRIMARY KEY,

nome VARCHAR (50) NOT NULL,

id\_veic integer NOT NULL,

id\_cli integer NOT NULL,

qtd\_visita integer NOT NULL,

vlr\_visita DOUBLE PRECISION NOT NULL,

FOREIGN KEY (id\_veic) REFERENCES LC.Veiculo (id),

FOREIGN KEY (id\_cli) REFERENCES LC.Cliente (id));

■ Visita

CREATE TABLE LC.Visita (

id bigserial NOT NULL PRIMARY KEY,

data\_visita DATE NOT NULL DEFAULT now(),

tipo\_visita CHAR NOT NULL,

id\_veiculo serial NOT NULL,

nome VARCHAR (50) NULL,

placa VARCHAR (12) NULL,

km\_saida integer NOT NULL,

km\_retorno integer NOT NULL,

id\_cliente integer NOT NULL,

id\_consultor integer NOT NULL,

FOREIGN KEY (id\_veiculo) REFERENCES LC.Veiculo (id),

FOREIGN KEY (id\_cliente) REFERENCES LC.Cliente (id),

FOREIGN KEY (id\_consultor) REFERENCES LC.Consultor (id)

);

COMMENT ON COLUMN LC.Visita.tipo\_visita IS 'C - CONSULTORIA / A - ACESSORIA';

**TRABALHANDO COM DML – DATA MANIPULATION LANGUAGE**

■ POPULANDO DADOS TABELA CLIENTE – INSERT

INSERT INTO LC.Cliente (nome, tipo, celular)

VALUES ('Gustavo', 'F', '(93)91445589');

INSERT INTO LC.Cliente (nome, tipo, celular)

VALUES ('PEDRO OURES', 'J', '(92)93445589');

INSERT INTO LC.Cliente (nome, tipo, celular)

VALUES ('OTAVIO MESQUITA', 'J', '(99)93355589');

INSERT INTO LC.Cliente (nome, tipo, celular)

VALUES ('EFRAIN AGUIAR', 'F', '(93)91538504');

INSERT INTO LC.Cliente (nome, tipo, celular)

VALUES ('CLEUDIMAR MORAES', 'J', '');

SELECT \* FROM LC.Cliente;

■ POPULANDO DADOS TABELA VEÍCULO – INSERT

INSERT INTO LC.Veiculo (nome, placa, km, cor, ativo)

VALUES ('GOL 1000', 'RTY-2344', '0', 'AZUL', 'S');

INSERT INTO LC.Veiculo (nome, placa, km, cor, ativo)

VALUES ('SAVEIRO CROSS', 'ATR-3234', '0', 'AZUL', 'S');

INSERT INTO LC.Veiculo (nome, placa, km, cor, ativo)

VALUES ('FIAT SIENA', 'XXI-5994', '0', 'AZUL', 'S');

INSERT INTO LC.Veiculo (nome, placa, km, cor, ativo)

VALUES ('FIAT PALIO', 'ABC-1234', '0', 'AZUL', 'S');

INSERT INTO LC.Veiculo (nome, placa, km, cor, ativo)

VALUES ('FIAT PALIO', 'ABX-3356', '0', 'AZUL', 'S');

SELECT \* FROM LC.Veiculo;

■ POPULANDO DADOS TABELA CONSULTOR – INSERT

INSERT INTO LC.Consultor (nome, id\_veic, id\_cli, qtd\_visita, vlr\_visita)

VALUES ('GUSTAVO BIAL', '1', '1', '1', '1200');

INSERT INTO LC.Consultor (nome, id\_veic, id\_cli, qtd\_visita, vlr\_visita)

VALUES ('ROMARIO ABREU', '2', '2', '1', '2000');

INSERT INTO LC.Consultor (nome, id\_veic, id\_cli, qtd\_visita, vlr\_visita)

VALUES ('ROGERIO ALVES', '3', '3', '1', '1500');

INSERT INTO LC.Consultor (nome, id\_veic, id\_cli, qtd\_visita, vlr\_visita)

VALUES ('GUSTAVO BRETA', '4', '4', '1', '1890');

INSERT INTO LC.Consultor (nome, id\_veic, id\_cli, qtd\_visita, vlr\_visita)

VALUES (‘FELIPE SIQUEIRA’, '5', '5', '1', '2100');

SELECT \* FROM LC.Consultor;

■ POPULANDO DADOS TABELA VISITA – INSERT

INSERT INTO LC.Visita (data\_visita, tipo\_visita, id\_veiculo, nome, placa,

km\_saida, km\_retorno, id\_cliente, id\_consultor)

VALUES ('01/10/2018', 'C', '1', 'GOL 1000', 'RTY-2344',

'0', '300', '1', '1');

INSERT INTO LC.Visita (data\_visita, tipo\_visita, id\_veiculo, nome, placa,

km\_saida, km\_retorno, id\_cliente, id\_consultor)

VALUES ('05/10/2018', 'C', '2', 'SAVEIRO CROSS', 'ATR-3234',

'0', '250', '2', '2');

INSERT INTO LC.Visita (data\_visita, tipo\_visita, id\_veiculo, nome, placa,

km\_saida, km\_retorno, id\_cliente, id\_consultor)

VALUES ('20/10/2018', 'C', '3', 'FIAT SIENA', 'XXI-5994',

'0', '350', '3', '3');

INSERT INTO LC.Visita (data\_visita, tipo\_visita, id\_veiculo, nome, placa,

km\_saida, km\_retorno, id\_cliente, id\_consultor)

VALUES ('25/10/2018', 'C', '4', 'FIAT PALIO', 'ABC-1234',

'0', '120', '4', '4');

INSERT INTO LC.Visita (data\_visita, tipo\_visita, id\_veiculo, nome, placa,

km\_saida, km\_retorno, id\_cliente, id\_consultor)

VALUES ('29/10/2018', 'C', '5', 'FIAT PALIO', 'ABX-3356',

'0', '170', '5', '5');

-- CONSULTA BÁSICA - SELECT

SELECT \* FROM LC.Visita;

SELECT \* FROM LC.Consultor;

-- CONSULTA BÁSICA - UTILIZANDO WHERE - SELECT

SELECT \* FROM LC.Cliente

WHERE id = 5;

■ REALIZANDO CONSULTAS UTILIZANDO JOINs UNINDO AS TABELAS

SELECT LC.Consultor.nome as "CONSULTOR",

LC.Visita.data\_visita as "DATA VISITA",

LC.Cliente.nome as "CLIENTE",

LC.Veiculo.placa as "VEÍCULO",

LC.Visita.km\_saida as "KM SAÍDA",

LC.Visita.km\_retorno as "KM RETORNO"

FROM LC.Visita

INNER JOIN LC.Consultor ON (LC.Consultor.id = LC.Visita.id\_consultor)

INNER JOIN LC.Cliente ON (LC.Cliente.id = LC.Visita.id\_cliente)

INNER JOIN LC.Veiculo ON (LC.Veiculo.id = LC.Visita.id\_veiculo);

■ Criando TRIGGER before insert com função na coluna data\_visita na tabela LC.Visita para inserir data e hora caso consultor não insira

CREATE FUNCTION data\_visita()

RETURNS TRIGGER AS $$

BEGIN

NEW.data\_visita = NOW();

RETURN NEW;

END;

$$ language 'plpgsql';

CREATE TRIGGER datas\_visitas

BEFORE INSERT

ON LC.Visita

FOR EACH ROW

EXECUTE PROCEDURE data\_visita();

--Inserindo dados de visita sem informar data de visita executando Procedure

INSERT INTO LC.Visita (tipo\_visita, id\_veiculo, nome, placa,

km\_saida, km\_retorno, id\_cliente, id\_consultor)

VALUES ('A', '1', 'GOL 1000', 'RTY-2344', '300', '500', '2', '1');

INSERT INTO LC.Visita (tipo\_visita, id\_veiculo, nome, placa,

km\_saida, km\_retorno, id\_cliente, id\_consultor)

VALUES ('A', '1', 'GOL 1000', 'RTY-2344', '500', '700', '6', '1');

SELECT \* FROM LC.Visita;

■ Criando Funcções multiplica

CREATE FUNCTION multiplica(num1 INT, num2 FLOAT)

RETURNS FLOAT AS $$

BEGIN

RETURN num1 \* num2;

END

$$ LANGUAGE 'plpgsql';

■ Aplicando a função multiplica em uma Consulta simples

SELECT multiplica(3, 1500.00)

■ Criando Funcções aplicar\_desconto

CREATE FUNCTION aplicar\_desconto(vlr\_visita FLOAT, desconto FLOAT)

RETURNS FLOAT AS $$

BEGIN

RETURN vlr\_visita \* (1 - desconto);

END

$$ LANGUAGE 'plpgsql'

■ Excluindo a função aplicar\_desconto

DROP FUNCTION aplicar\_desconto

■ Recriando a Função aplicar\_desconto

CREATE OR REPLACE FUNCTION aplicar\_desconto(vlr\_visita FLOAT, desconto FLOAT)

RETURNS FLOAT AS $$

BEGIN

RETURN vlr\_visita \* (1 - desconto);

END

$$ LANGUAGE 'plpgsql'

■ Aplicando a função aplicar\_desconto em uma Consult Joins

SELECT LC.Consultor.nome AS "Nome", aplicar\_desconto(vlr\_visita, 0.20)

FROM LC.Consultor

■ Criação da Função que retorna tipoVisita CONFORME VALOR DA VISITA

CREATE FUNCTION tipoVisita(vlr\_visita MONEY)

RETURNS TEXT AS $$

BEGIN

IF vlr\_visita < 2000 THEN

RETURN 'Acessoria';

ELSIF vlr\_visita > 2000 THEN

RETURN 'Consultoria';

ELSE

RETURN 'Não parametizado';

END IF;

END;

$$ language 'plpgsql';

■ EXCLUINDO A FUNÇÃO tipoVisita com o DROP

DROP FUNCTION tipoVisita(MONEY);

■ Recriando a função tipoVisita excluida

CREATE OR REPLACE FUNCTION tipoVisita(vlr\_visita DOUBLE PRECISION)

RETURNS TEXT AS $$

BEGIN

IF vlr\_visita < 2000 THEN

RETURN 'Acessoria';

ELSIF vlr\_visita > 2000 THEN

RETURN 'Consultoria';

ELSE

RETURN 'Não parametizado';

END IF;

END;

$$ language 'plpgsql';

■ Realizando Consulta simples da tabela Consultor executando a Função tipoVisita e função multiplica

SELECT

LC.Consultor.id AS "ID",

LC.Consultor.nome AS "AGENTE",

LC.Consultor.qtd\_visita AS "QTD VISITA",

tipoVisita(vlr\_visita),

multiplica(qtd\_visita, vlr\_visita) AS "Valor Total"

FROM LC.Consultor;

■ REALIZANDO CONSULTAS UTILIZANDO INNER JOIN UNINDO AS TABELAS E EXECUTANDO AS FUNÇÕES tipoVisita e multiplica

SELECT LC.Consultor.nome as "CONSULTOR",

LC.Visita.data\_visita as "DATA VISITA",

LC.Cliente.id as "Código",

LC.Cliente.nome as "CLIENTE",

LC.Cliente.celular as "Fone",

LC.Veiculo.placa as "VEÍCULO",

LC.Visita.km\_saida as "KM SAÍDA",

LC.Visita.km\_retorno as "KM RETORNO",

tipoVisita(vlr\_visita) as "TIPO",

LC.Consultor.qtd\_visita as "QTD",

LC.Consultor.vlr\_visita as "Comissão",

multiplica(qtd\_visita, vlr\_visita) AS "Valor Total"

FROM LC.Visita

INNER JOIN LC.Consultor ON (LC.Consultor.id = LC.Visita.id\_consultor)

INNER JOIN LC.Cliente ON (LC.Cliente.id = LC.Visita.id\_cliente)

INNER JOIN LC.Veiculo ON (LC.Veiculo.id = LC.Visita.id\_veiculo)

■ REALIZANDO CONSULTAS UTILIZANDO INNER LEFT JOIN UNINDO AS TABELAS COM OS DADOS DA ESQUERDA E EXECUTANDO AS FUNÇÕES tipoVisita e multiplica

SELECT LC.Cliente.ID AS "Código",

LC.Cliente.nome AS "Nome",

LC.Cliente.celular AS "Fone",

LC.Visita.data\_visita AS "Data",

LC.Consultor.id AS "CONSULTOR",

LC.Visita.id AS "VISITA",

tipoVisita(vlr\_visita) as "TIPO",

LC.Veiculo.nome AS "Carro",

multiplica(qtd\_visita, vlr\_visita) AS "Valor Total"

FROM LC.Cliente

LEFT JOIN LC.Consultor ON (LC.Consultor.id\_cli = LC.Cliente.id)

LEFT JOIN LC.Visita ON (LC.Visita.id\_cliente = LC.Cliente.id)

LEFT JOIN LC.Veiculo ON (LC.Veiculo.id = LC.Visita.id\_veiculo)

■ REALIZANDO CONSULTAS UTILIZANDO RIGHT JOIN UNINDO AS TABELAS COM OS DADOS DA DIREITA E EXECUTANDO AS FUNÇÕES tipoVisita e multiplica

SELECT LC.Cliente.ID AS "Código",

LC.Cliente.nome AS "Nome",

LC.Cliente.celular AS "Fone",

LC.Consultor.id AS "Consultor",

LC.Veiculo.nome AS "Carro",

LC.Visita.data\_visita AS "Data",

LC.Visita.id AS "VISITA",

tipoVisita(vlr\_visita) as "TIPO",

multiplica(qtd\_visita, vlr\_visita) AS "Valor Total"

FROM LC.Cliente

RIGHT JOIN LC.Visita ON (LC.Visita.id\_cliente = LC.Cliente.id)

RIGHT JOIN LC.Veiculo ON (LC.Veiculo.id = LC.Visita.id\_veiculo)

RIGHT JOIN LC.Consultor ON (LC.Consultor.id = LC.Visita.id\_consultor)