BANCO DE DADOS 2: TRABALHO FINAL

ALUNO: LUIS FERNANDO PIZZIRANI

TSI23

DESCRIÇÃO DO ESTUDO:
Banho e Tosa

- ***LEVANTAMENTO DE REQUISITOS:***

Um salão de beleza para pets apresenta um catálogo composto por pacotes de banho, banho e tosa, e tosa higiênica, tendo como opção o pacote fiidelidade (anual), pacote mensal, e o pacote unitário.

Para realizar o serviço, é necessário que o cliente cadastre seu animal de estimação, como cachorro ou gato, identifiicando nome, raça e o seu porte, sendo estes como porte pequeno, porte médio ou grande.

No pet-shop trabalham fiuncionários responsáveis pelo firete e que cuidam do banho e tosa, sendo responsáveis pela retirada e devolução do animal. Deve-se saber o número de telefione e a identifiicação de cada fiuncionário, e suas respectivas tosas contendo data e horário agendado.

O transporte (firete) destina-se a um bairro. Para cada bairro existe um valor de firete máximo de acordo com a distância da empresa até o local sendo percursos com viagens curtas, médias e viagem de longa distância que implicam o valor total do serviço.

PRINCIPAIS CONSULTAS:

Listar as datas de consultas do Pets.

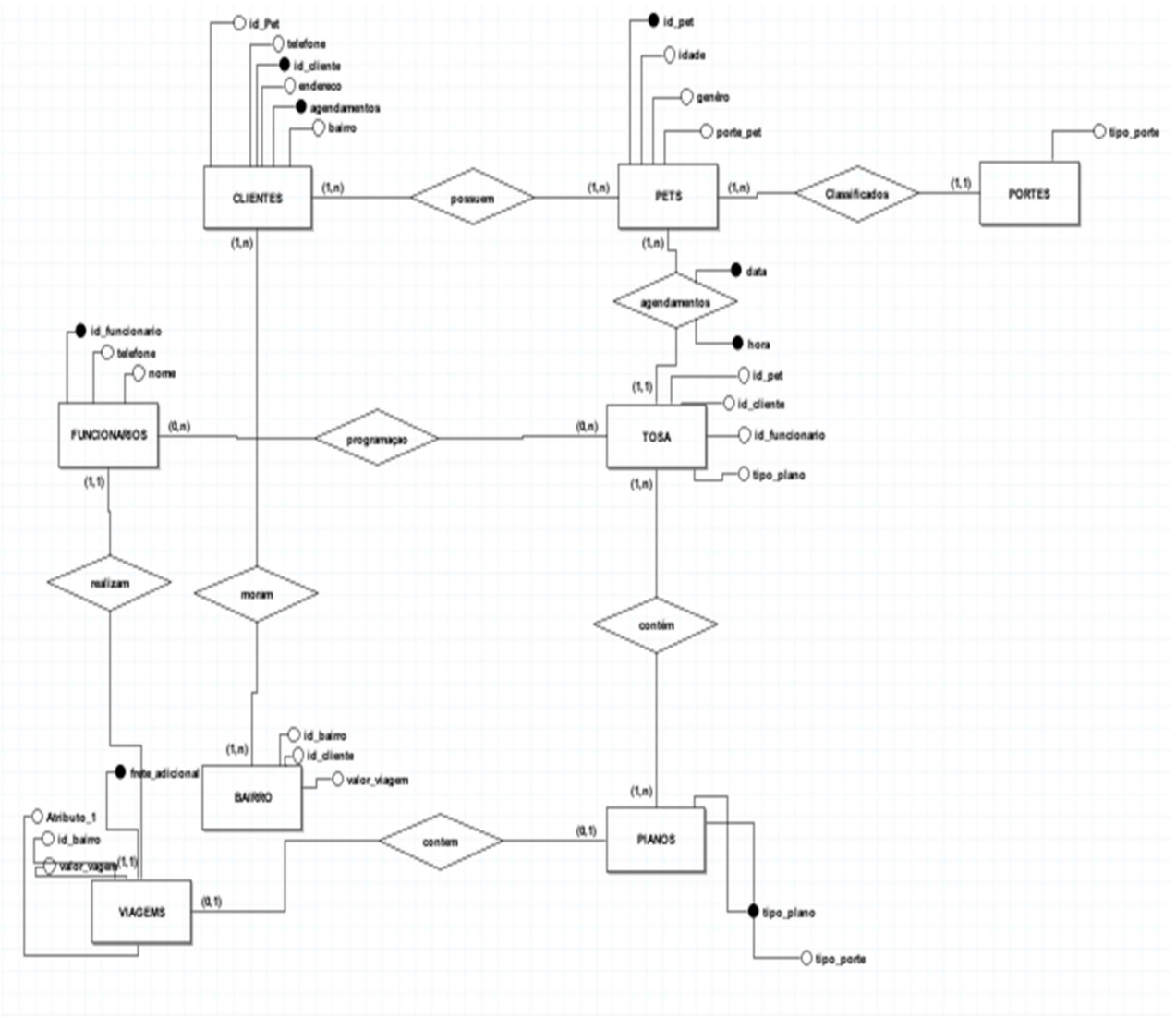
Listar os pets cadastrados para cada cliente

Consultar o valor de transporte do pet para o bairro cadastrado do cliente

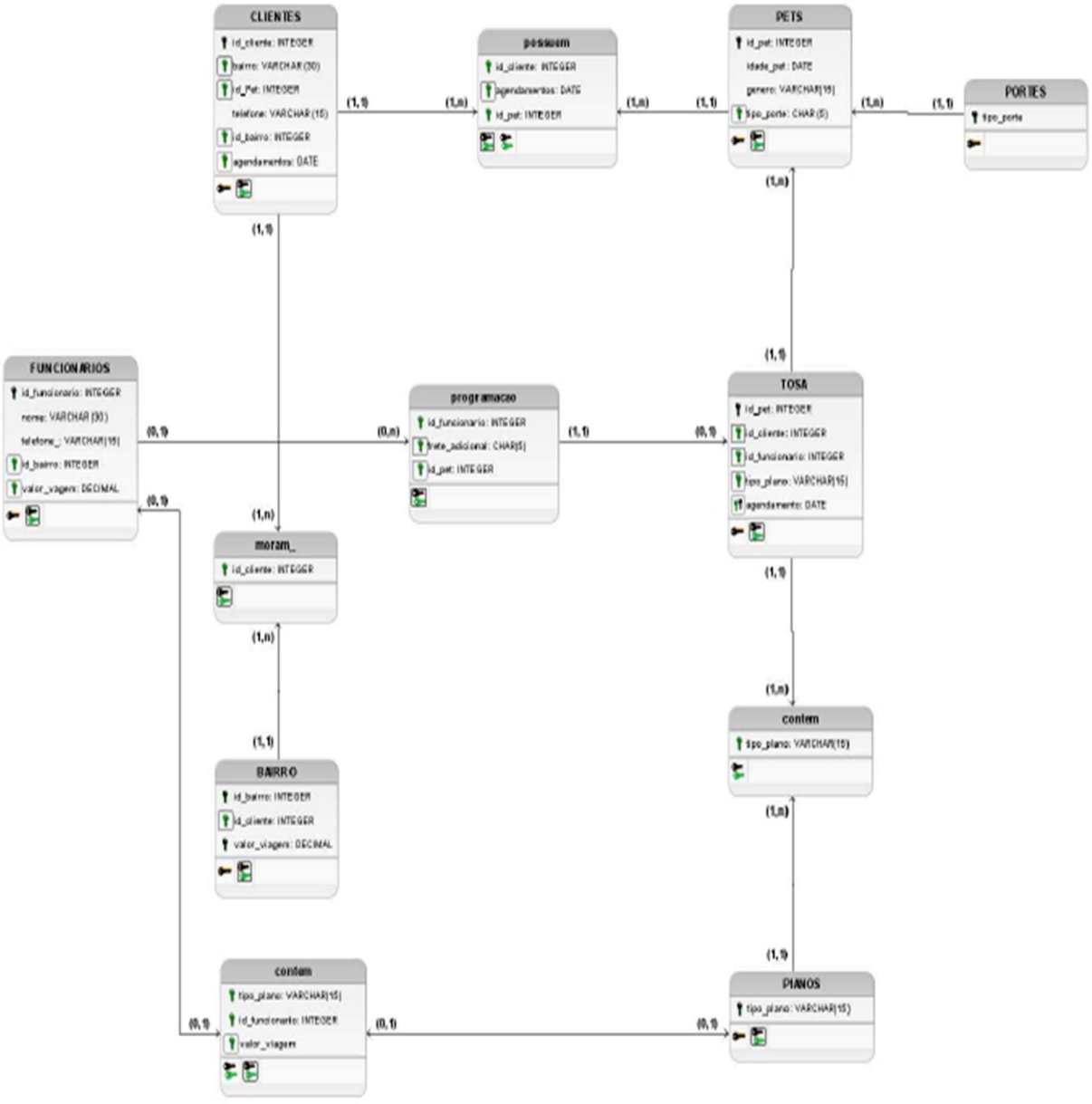
Consultar o plano escolhido pelo cliente

Consultar o fiuncionário responsável pela tosa e transporte do pet.

- ***MODELAGEM CONCEITUAL- BANHO E TOSA***



- ***MODELAGEM CONCEITUAL- BANHO E TOSA***



- ***PROJETO FISICO: BANHO E TOSA***

```
CREATE TABLE clientes (
id_cliente INTEGER NOT
NULL, nome VARCHAR(30)
NOT NULL,
id_bairro INTEGER NOT NULL,
telefone VARCHAR(15) NOT
NULL,
agendamentos
TIMESTAMP, PRIMARY KEY
(id_cliente)
);
CREATE TABLE bairro (
id_bairro INTEGER NOT
NULL, id_cliente INTEGER
NOT NULL,
valor_transporte DECIMAL NOT
NULL, PRIMARY KEY (id_bairro),
FOREIGN KEY (id_cliente) REFERENCES clientes (id_cliente)
);
CREATE TABLE residem (
id_cliente INTEGER NOT
NULL, id_bairro INTEGER
NOT NULL,
PRIMARY KEY (id_cliente, id_bairro),
FOREIGN KEY (id_cliente) REFERENCES clientes
(id_cliente), FOREIGN KEY (id_bairro) REFERENCES bairro
(id_bairro)
```

```
);
CREATE TABLE petz (
id_pet INTEGER NOT
NULL,
id_cliente INTEGER NOT
NULL, idade TIMESTAMP,
genero VARCHAR(30),
tipo_porte CHAR NOT
NULL,
```

```
PRIMARY KEY (id_pet),
FOREIGN KEY (id_cliente) REFERENCES clientes (id_cliente)
);

CREATE TABLE funcionarios (
id_funcionario SERIAL PRIMARY
KEY, nome VARCHAR(30) NOT
NULL,
telefone VARCHAR(15),
id_bairro INTEGER NOT
NULL,
valor_transporte DECIMAL NOT NULL
);

CREATE TABLE porte (
tipo_porte SERIAL PRIMARY KEY
);

CREATE TABLE tosa (
agendamentos TIMESTAMP NOT
NULL, id_pet INTEGER NOT NULL,
id_cliente INTEGER NOT NULL,
id_funcionario INTEGER NOT
NULL, tipo_plano CHAR(5) NOT
NULL,
PRIMARY KEY (agendamentos, id_pet, id_cliente,
id_funcionario), FOREIGN KEY (id_pet) REFERENCES petz
(id_pet),
FOREIGN KEY (id_cliente) REFERENCES clientes (id_cliente),
FOREIGN KEY (id_funcionario) REFERENCES funcionarios (id_funcionario)
```

```
);

CREATE TABLE planos (
tipo_plano SERIAL PRIMARY
KEY
);

CREATE TABLE agendamentos (
  agendamentos id SERIAL PRIMARY KEY, agendamentos TIMESTAMP NOT NULL,
    id_funcionario INTEGER NOT NULL, id_pet INTEGER NOT NULL
    valor_transporte DECIMAL,
    FOREIGN KEY (id_funcionario) REFERENCES funcionarios
    (id_funcionario), FOREIGN KEY (id_pet) REFERENCES petz (id_pet)
    );

    CREATE TABLE transporte (
    tipo_plano INTEGER NOT NULL,
    id_funcionario INTEGER NOT
    NULL, valor_transporte
    DECIMAL NOT NULL,
    PRIMARY KEY (tipo_plano, id_funcionario),
    FOREIGN KEY (tipo_plano) REFERENCES planos (tipo_plano),
    FOREIGN KEY (id_funcionario) REFERENCES funcionarios (id_funcionario)
    );
```

```
CREATE TABLE possuem (
id_cliente INTEGER NOT
NULL,
agendamentos_id INTEGER NOT
NULL, id_pet INTEGER NOT NULL,
PRIMARY KEY (id_cliente, agendamentos_id, id_pet),
FOREIGN KEY (id_cliente) REFERENCES clientes
(id_cliente),
FOREIGN KEY (agendamentos_id) REFERENCES agendamentos
(agendamentos_id), FOREIGN KEY (id_pet) REFERENCES petz (id_pet)
```

INSERINDO DADOS NAS TABELAS:

–Inserindo dados na tabela cliente

```
NSERT INTO clientes (id_cliente, nome, id_bairro, telefone, agendamentos)
VALUES
  (2, 'Antonio Lopes', 1, '123456789', '2023-12-13 10:00:00'),
  (3, 'Francisco Souza', 2, '987654321', '2023-12-14 15:30:00'),
  (4, 'Roberto Cruz', 3, '555111222', NULL),
  (5, 'Maria Silva', 4, '333444555', '2023-12-15 08:45:00'),
  (6, 'Carlos Oliveira', 5, '777888999', '2023-12-16 14:20:00');
```

```
-- Inserindo dados na tabela bairro
INSERT INTO bairro (id_bairro, id_cliente, valor_transporte)
VALUES
  (1, 2, 50.00),
  (2, 3, 30.00),
  (3, 4, 25.00),
  (4, 5, 40.00),
  (5, 6, 20.00);
```

```
-- É preciso atualizar dados na tabela clientes para adicionar o id_bairro correspondente
UPDATE clientes SET id_bairro = 1 WHERE id_cliente = 2;
UPDATE clientes SET id_bairro = 2 WHERE id_cliente = 3;
UPDATE clientes SET id_bairro = 3 WHERE id_cliente = 4;
UPDATE clientes SET id_bairro = 4 WHERE id_cliente = 5;
UPDATE clientes SET id_bairro = 5 WHERE id_cliente = 6;
```

```
--Inserindo dados relacionados na tabela residem
INSERT INTO residem (id_cliente, id_bairro)
VALUES
  (2, 1),
```

```
 (3, 2),


 (4, 3),
 (5, 4),
 (6, 5);

-- Percebi que faltava inserir uma coluna nome_pet na tabela petz então realizei este ALTER TABLE

ALTER TABLE petz
ADD COLUMN nome_pet VARCHAR(50);

-- Inseririndo dados na tabela petz
INSERT INTO petz (id_pet, id_cliente, idade, genero, tipo_porte, nome_pet)
VALUES
 (1, 2, '2021-01-15', 'Macho', 'Pequeno', 'Buddy'),
 (2, 3, '2019-05-20', 'Fêmea', 'Médio', 'Luna'),
 (3, 4, '2020-10-08', 'Macho', 'Grande', 'Rocky'),
 (4, 5, '2018-03-12', 'Fêmea', 'Pequeno', 'Coco'),
 (5, 6, '2017-08-25', 'Macho', 'Médio', 'Max');

-- Percebi que aloquei um CHAR erroneamente então vou ter que alterar novamente a tabela:
-- Alterarei o tamanho do campo tipo_porte para acomodar valores maiores
ALTER TABLE petz
ALTER COLUMN tipo_porte TYPE VARCHAR(10);

-- AGORA CONSEGUI INSERIR OS DADOS.


-- Inseririndo dados na tabela funcionarios
INSERT INTO funcionarios (nome, telefone, id_bairro, valor_transporte)
VALUES
 ('João Silva', '987654321', 1, 50.00),
 ('Ana Santos', '123456789', 2, 30.00),
 ('Carlos Oliveira', '555111222', 3, 25.00),
 ('Mariana Lima', '333444555', 4, 40.00),
```

```
 ('Pedro Rocha', '777888999', 5, 20.00);

  -- Alterando tipo de dado na coluna tipo_plano para VARCHAR
ALTER TABLE tosa
ALTER COLUMN tipo_plano TYPE VARCHAR(10);

-- Inserindo dados na tabela tosa
INSERT INTO tosa (agendamentos, id_pet, id_cliente, id_funcionario, tipo_plano)


VALUES
 ('2023-12-18 10:00:00', 1, 2, 1, 'Básico'),



 ('2023-12-19 15:30:00', 2, 3, 2, 'Premium'),
 ('2023-12-20 14:00:00', 3, 4, 3, 'Standard'),
 ('2023-12-21 09:45:00', 4, 5, 4, 'Básico'),
 ('2023-12-22 13:20:00', 5, 6, 5, 'Premium');

-- Alterar tipo de dado da coluna tipo_plano para VARCHAR
ALTER TABLE tosa
ALTER COLUMN tipo_plano TYPE VARCHAR(10); --



 -- Inserindo dados na tabela agendamentos
INSERT INTO agendamentos (agendamentos, id_funcionario, id_pet,
valor_transporte)
VALUES
 ('2023-12-23 11:30:00', 1, 1, 20.00),
 ('2023-12-24 14:45:00', 2, 2, 30.00),
 ('2023-12-25 09:00:00', 3, 3, 25.00),
 ('2023-12-26 13:15:00', 4, 4, 40.00),
 ('2023-12-27 16:30:00', 5, 5, 15.00);

 -- Adicionei uma coluna id_funcionario e categoria_plano na tabela planos
ALTER TABLE planos
```

```
ADD COLUMN id_funcionario INTEGER REFERENCES funcionarios (id_funcionario),
ADD COLUMN categoria_plano VARCHAR(30) UNIQUE;

-- Inserir dados de exemplo na tabela planos
INSERT INTO planos (id_funcionario, categoria_plano)
VALUES
  (1, 'Categoria1'),
  (2, 'Categoria2'),
  (3, 'Categoria3');

  UPDATE planos
SET categoria_plano =
  CASE tipo_plano
    WHEN 1 THEN 'Básico'
    WHEN 2 THEN 'Intermediário'
    WHEN 3 THEN 'Completo'
 END;


-- Insert transportes
INSERT INTO transporte (tipo_plano_new, id_funcionario, valor_transporte)

VALUES
  (1, 1, 100.00),
  (2, 2, 120.00),
  (3, 3, 150.00);



INSERT INTO possuem (id_cliente, agendamentos_id, id_pet)
VALUES
  (2, 1, 1),
  (3, 2, 2),
  (4, 3, 3);

  --Insert dados na tabela contem
INSERT INTO contem (tipo_plano)
VALUES
```

```
  (1),
  (2),
  (3);

  -- Insert na tabela programacao
INSERT INTO programacao (id_funcionario, id_pet)
VALUES
  (1, 1),
  (2, 2),
  (3, 3);
```

CONSULTAS COM JOIN

1 A consulta obtém informações sobre os agendamentos de tosa, nome do cliente, a data do agendamento e o nome do funcionário responsável pela tosa:

```
SELECT c.nome AS cliente_nome, t.agendamentos, f.nome AS funcionario_nome
FROM clientes c
JOIN tosa t ON c.id_cliente = t.id_cliente
JOIN funcionarios f ON t.id_funcionario = f.id_funcionario;
```

	cliente_nome character varying (30)	agendamentos timestamp without time zone	funcionario_nome character varying (30)
1	Antonio Lopes	2023-12-18 10:00:00	João Silva
2	Francisco Souza	2023-12-19 15:30:00	Ana Santos
3	Roberto Cruz	2023-12-20 14:00:00	Carlos Oliveira
4	Maria Silva	2023-12-21 09:45:00	Mariana Lima
5	Carlos Oliveira	2023-12-22 13:20:00	Pedro Rocha

2 . Consulta para Listar Pets e Suas Informações de Tosa Agendada:

```
SELECT p.nome_pet, t.agendamentos
FROM petz p
LEFT JOIN tosa t ON p.id_pet = t.id_pet;
```

	nome_pet character varying (50)	agendamentos timestamp without time zone
1	Buddy	2023-12-18 10:00:00
2	Luna	2023-12-19 15:30:00
3	Rocky	2023-12-20 14:00:00
4	Coco	2023-12-21 09:45:00
5	Max	2023-12-22 13:20:00

3 Consulta para relacionar clientes aos seus pets:

```
SELECT c.nome AS cliente_nome, p.nome_pet
FROM clientes c
JOIN petz p ON c.id_cliente = p.id_cliente;
```

	cliente_nome character varying (30)	nome_pet character varying (50)
1	Antonio Lopes	Buddy
2	Francisco Souza	Luna
3	Roberto Cruz	Rocky
4	Maria Silva	Coco
5	Carlos Oliveira	Max

4 Consulta que seleciona informações específicas sobre clientes e seus bairros correspondentes

```
SELECT c.id_cliente, c.nome, b.valor_transporte, c.telefone, c.agendamentos
FROM clientes c
INNER JOIN bairro b ON c.id_bairro = b.id_bairro;
```

	id_cliente integer	nome character varying (30)	valor_transporte numeric	telefone character varying (15)	agendamentos timestamp without time zone
1	2	Antonio Lopes	50.00	123456789	2023-12-13 10:00:00
2	3	Francisco Souza	30.00	987654321	2023-12-14 15:30:00
3	4	Roberto Cruz	25.00	555111222	[null]
4	5	Maria Silva	40.00	333444555	2023-12-15 08:45:00
5	6	Carlos Oliveira	20.00	777888999	2023-12-16 14:20:00

5 .Consulta para Verificar Agendamentos Futuros de Tosa:

```
SELECT c.nome AS cliente_nome, t.agendamentos
FROM clientes c
JOIN tosa t ON c.id_cliente = t.id_cliente
WHERE t.agendamentos > CURRENT_TIMESTAMP;
```

	cliente_nome character varying (30)	agendamentos timestamp without time zone
1	Antonio Lopes	2023-12-18 10:00:00
2	Francisco Souza	2023-12-19 15:30:00
3	Roberto Cruz	2023-12-20 14:00:00
4	Maria Silva	2023-12-21 09:45:00
5	Carlos Oliveira	2023-12-22 13:20:00

STORED PROCEDURES

```
-- AGENDAMENTO
-- Este procedimento insere um novo agendamento para um cliente específico.
-- Parâmetros:
--   - id_pet: Identificador único do pet (que seria equivalente ao cliente).
--   - agendamento_timestamp: Data e hora do agendamento.

  CREATE OR REPLACE PROCEDURE inserir_agendamento(
   id_pet INT,
   agendamento_timestamp TIMESTAMP
) AS $$
BEGIN
   -- Insere um novo agendamento na tabela 'agendamentos'.
   INSERT INTO agendamentos(id_pet, agendamento_timestamp)
   VALUES (id_pet, agendamento_timestamp);
END;
$$ LANGUAGE plpgsql;


-- Consultando os agendamentos para verificar se o novo foi inserido

SELECT * FROM agendamentos;
```

	agendamentos_id [PK] integer	agendamentos timestamp without time zone	id_funcionario integer	id_pet integer	valor_transporte numeric
1	1	2023-12-23 11:30:00	1	1	20.00
2	2	2023-12-24 14:45:00	2	2	30.00
3	3	2023-12-25 09:00:00	3	3	25.00
4	4	2023-12-26 13:15:00	4	4	40.00
5	5	2023-12-27 16:30:00	5	5	15.00

— *Atualização do Valor do Transporte em um Agendamento*

```
-- Parâmetros:
--   - agendamento_id: Identificador único do agendamento a ser atualizado.
--   - novo_valor_transporte: Novo valor do transporte a ser atribuído ao agendamento.


CREATE OR REPLACE PROCEDURE atualizar_valor_transporte(
   agendamento_id INT,
   novo_valor_transporte NUMERIC
) AS $$
BEGIN
   -- Atualiza o valor do transporte na tabela 'agendamentos'.
   UPDATE agendamentos
   SET valor_transporte = novo_valor_transporte
   WHERE agendamentos_id = agendamento_id;
END;
$$ LANGUAGE plpgsql;

-- Chamando o Procedimento Armazenado para Atualizar Valor do Transporte
CALL atualizar_valor_transporte(1, 20.00);

--Consultando os dados do agendamento atualizado
SELECT * FROM agendamentos WHERE agendamentos_id = 1;
```

Data Output | Messages | Notifications

	agendamentos_id [PK] integer	agendamentos timestamp without time zone	id_funcionario integer	id_pet integer	valor_transporte numeric
1	1	2023-12-23 11:30:00	1	1	20.00

VIEWS

```
---- View que fornece informações sobre os agendamentos futuros.

CREATE OR REPLACE VIEW view_agendamentos_futuros AS
SELECT
   a.agendamentos_id,
   a.agendamentos AS agendamento_timestamp,
   f.nome AS nome_funcionario,
   c.nome AS nome_cliente,
   p.nome_pet,
   p.genero AS genero_pet,
   p.tipo_porte AS tipo_porte_pet
FROM
   agendamentos a
   JOIN funcionarios f ON a.id_funcionario = f.id_funcionario
   JOIN petz p ON a.id_pet = p.id_pet
   JOIN clientes c ON p.id_cliente = c.id_cliente
WHERE
   a.agendamentos >= NOW(); -- Somente agendamentos futuros

    --consulta agendamentos futuros
    SELECT * FROM view_agendamentos_futuros;
```

	agendamentos_id integer	agendamento_timestamp timestamp without time zone	nome_funcionario character varying (30)	nome_cliente character varying (30)	nome_pet character varying (50)	genero_pet character varying (30)	tipo_porte_pet character varying (10)
1	2	2023-12-24 14:45:00	Ana Santos	Francisco Souza	Luna	Fêmea	Médio
2	3	2023-12-25 09:00:00	Carlos Oliveira	Roberto Cruz	Rocky	Macho	Grande
3	4	2023-12-26 13:15:00	Mariana Lima	Maria Silva	Coco	Fêmea	Pequeno
4	5	2023-12-27 16:30:00	Pedro Rocha	Carlos Oliveira	Max	Macho	Médio
5	1	2023-12-23 11:30:00	João Silva	Antonio Lopes	Buddy	Macho	Pequeno
6	6	2023-12-16 10:00:00	João Silva	Antonio Lopes	Buddy	Macho	Pequeno
7	7	2023-12-17 14:30:00	Ana Santos	Francisco Souza	Luna	Fêmea	Médio

```
-- View que fornece informações sobre os agendamentos atrasados.
CREATE OR REPLACE VIEW view_agendamentos_atrasados AS
SELECT
    a.agendamentos_id,
    a.agendamentos AS agendamento_timestamp,
    f.nome AS nome_funcionario,
    c.nome AS nome_cliente,
    p.nome_pet,
    p.genero AS genero_pet,
    p.tipo_porte AS tipo_porte_pet
FROM
    agendamentos a
    JOIN funcionarios f ON a.id_funcionario = f.id_funcionario
    JOIN petz p ON a.id_pet = p.id_pet
    JOIN clientes c ON p.id_cliente = c.id_cliente
WHERE
    a.agendamentos < NOW(); -- Somente agendamentos atrasados

     --consulta agendamentos atrasados
     SELECT * FROM view_agendamentos_atrasados;
```

	agendamentos_id integer	agendamento_timestamp timestamp without time zone	nome_funcionario character varying (30)	nome_cliente character varying (30)	nome_pet character varying (50)	genero_pet character varying (30)	tipo_porte_pet character varying (10)
1	11	2023-11-25 14:30:00	Ana Santos	Francisco Souza	Luna	Fêmea	Médio
2	12	2023-11-30 12:00:00	João Silva	Roberto Cruz	Rocky	Macho	Grande
3	13	2023-12-05 11:15:00	Carlos Oliveira	Maria Silva	Coco	Fêmea	Pequeno
4	14	2023-12-10 15:30:00	Ana Santos	Carlos Oliveira	Max	Macho	Médio