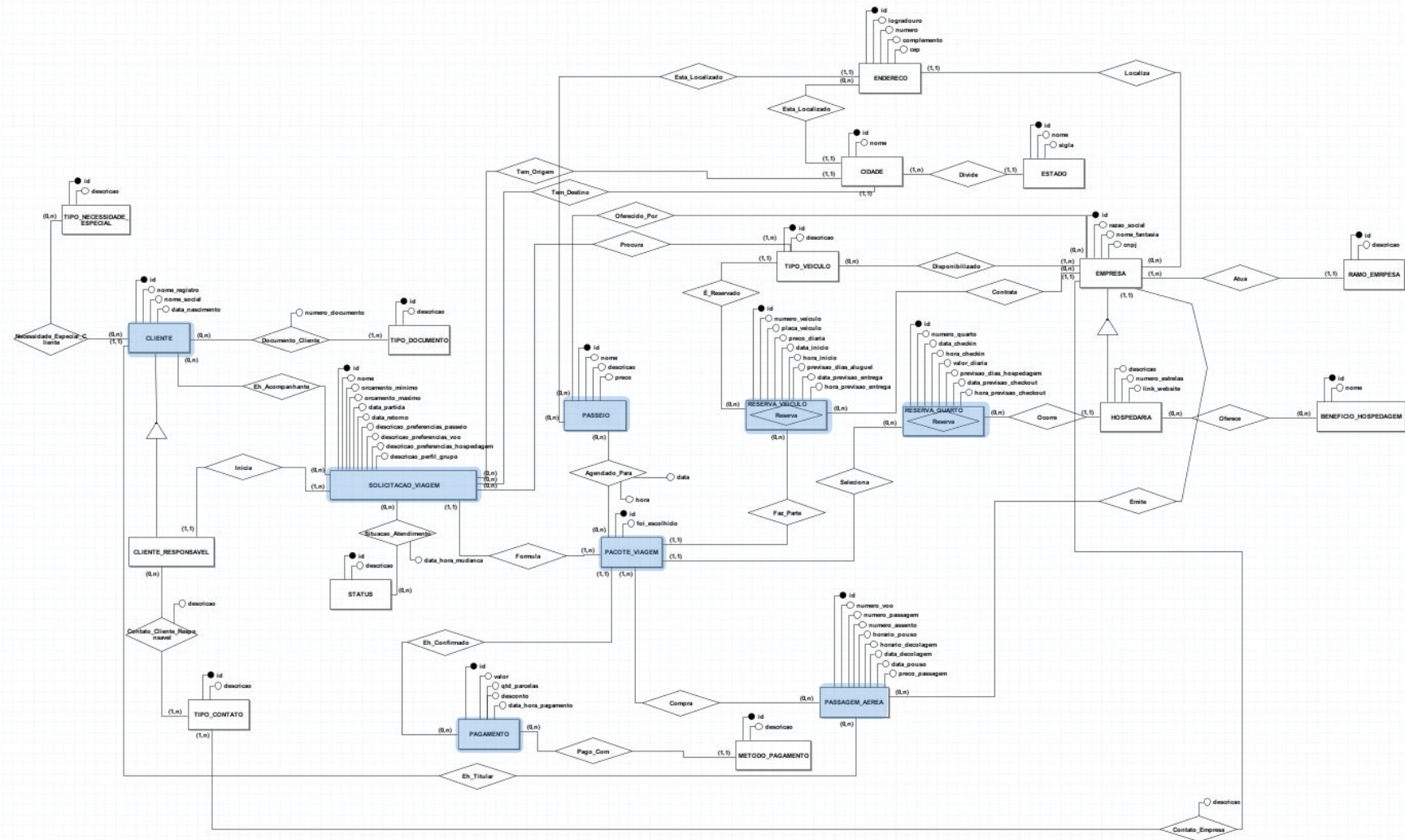




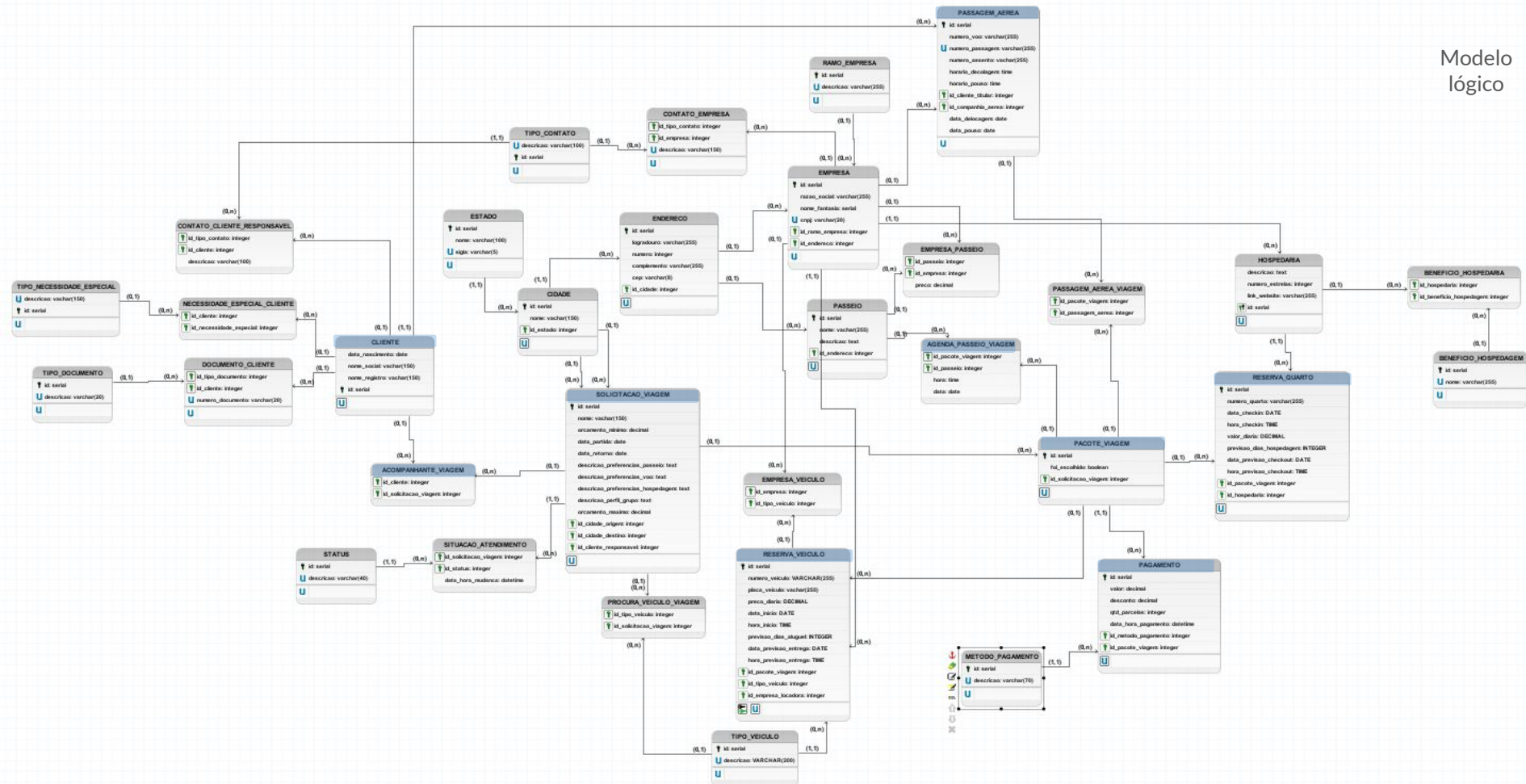
Banco de Dados - Viagens BSI



Modelo conceitual



3



Modelo físico (V4)

```

D > Beatrix > Documentos > BSI > 4.Periodo > banco de dados > trabalho >
442 (10, 'Chão de Floresta MEI', 'Chão de Floresta', '47973577000130', 2, 10),
443 (11, 'Villa da Montanha SA', 'Villa da Montanha', '42454553000159', 2, 11),
444 (12, 'Pousada do Rio EPP Ltda', 'Pousada do Rio', '60727233000112', 2, 12),
445 (13, 'Hotel Spa das Termas EPP Ltda', 'Hotel Spa das Termas', '22915540000160', 2, 13),
446 (14, 'Villa Tropical SA', 'Villa Tropical', '57826263000187', 2, 14),
447 (15, 'Resort Paradiso SA', 'Resort Paradiso', '93420343000119', 2, 15),
448 (16, 'Daybreak Hotel SA', 'Daybreak Hotel', '95848263000157', 2, 16),
449 (17, 'TopGun Hotel SA', 'TopGun Hotel', '63126380000190', 2, 17),
450 (18, 'Argos Hotel SA', 'Argos Hotel', '22328472000131', 2, 18),
451 (19, 'Cleonice Bochimpani Furtado 258.534.321-92 MEI', 'Urban Retreat', '33449767000151', 2),
452 (20, 'Gustavo Santos Marica 042.178.212-98 MEI', 'Casa do Mar', '53211467000147', 2, 20),
453 (21, 'Resort dos Sonhos SA', 'Resort dos Sonhos', '95844141000185', 2, 21),
454 (22, 'Hotel Charmes SA', 'Hotel Charmes', '93821387000174', 2, 22),
455 (23, 'Hotel Plus SA', 'Hotel Plus', '82958722000171', 2, 23),
456 (24, 'Pousada das Estrelas EPP Ltda', 'Pousada das Estrelas', '08214569000161', 2, 24),
457 (25, 'Hotel Colonial SA', 'Hotel Colonial', '53353587000101', 2, 25),
458 (26, 'Releio das Montanhas SA', 'Releio das Montanhas', '26834177000111', 2, 26),
459 (27, 'Pousada da Serra EPP Ltda', 'Pousada da Serra', '32031115000133', 2, 27),
460 (28, 'Pousada Cava SA', 'Pousada Cava', '19464577000195', 2, 28),
461 (29, 'Bluejay Pousada SA', 'Bluejay Pousada', '62718773000165', 2, 29),
-- Empresas de Passos
463 (30, 'Beto Carrero Mundo Parque SA', 'Beto Carrero World', '62508357300016', 4, 30), -- emp
464 (31, 'Turismo em Paraty SA', 'Turismo Amigo Paraty', '4150380000129', 4, 31), -- empresa
465 (32, 'Turismo em Buriões SA', 'Turismo Legal Buriões', '41525280000102', 4, 32), -- empresa
466 (33, 'Turismo em Foz do Iguaçu SA', 'Turismo Interessante Parati', '43785315000102', 4, 33)
467 (34, 'Turismo em Porto de Galinhas SA', 'Turismo Exploracion Pernambuco', '21738150000133',
468
469 INSERT INTO HOSPEDARIA (id, descricao, numero_estrelas, link_website) VALUES
470 (1, 'Como o melhor nem diz: Gem Hotel!! Lugo, Espanha, anos 1899. Inovação... Chilas ge
471 (9, 'Hotel de alto padrão, em uma de vantagem no meio da cidade, próximo ao mar e ao centro
472 (19, 'Chão Individual Isolados. Todos tem banheira, alguns tem churrasqueira, é possível comprar i
473 (11, 'Pousada muito confortável, possui em uma vila de chácaras individuais. Tem café da manhã
474 (12, 'Pousada bastante tranquila, com café da manhã ok, nível standards, mas de baixo orçamento
475 (14, 'Hotel bonito, elegante, simples, com café da manhã e buffet disponível a um preço razoáve
476 (16, 'Boa hotel para quem quer equilíbrio entre o conforto e o preço, com café da manhã e ab
477 (15, 'Resort perfeito para passar com a família, amigos ou em casal. All inclusive, piscinas
478 (16, 'Ótimo custo-benefício, com café da manhã excelente.', 2, 'www.daybreak.com.br'), -- ho
479 (17, 'Parati Internacional, boa localização.', 4, 'www.topgun.com.br'), -- hospedagem spa paul
480 (18, 'Aconchegante, confortável, good vibes.', 3, 'www.argos.com.br'), -- hospedagem lenç
481 (19, 'Casa de temporada', 4, 'www.urbanretreat.com.br'), -- hospedagem lenç
482 (20, 'Casa de temporada', 3, 'www.casadoram.com.br'), -- hospedagem salvador
483 (21, 'Resort do fervor!', 5, 'www.resortdossonhos.com.br'), -- hospedagem salvador
484 (22, 'Boa custo-benefício.', 2, 'www.hotelchamarnal.com.br'), -- hospedagem natal
485 (23, 'Hotel de qualidade um pouco superior, custo-benefício para boa localização.', 3, 'www
486 (24, 'Aconchegante.', 3, 'www.pousadastascsp.com.br'), -- hospedagem foz do iguaçu
487 (25, 'Aconchegante, tranquilo, estável.', 3, 'www.colonialhotelparana.com.br'), -- hospede
488 (26, 'Hotel muito bom, com uma família.', 4, 'www.refugiadosasmontanhas.com.br'), -- hospede
489 (27, 'Econômica.', 2, 'www.pousadade.com.br'), -- hospedagem dosingos martins

```

```

D > Beatrix > Documentos > BSI > 4.Periodo > banco de dados > trabalho > -- fisico_v4
1 -- Logica_v2_ViagensBSI:
2
3 BEGIN;
4
5 CREATE TABLE SOLICITACAO_VIAGEM (
6     id SERIAL UNIQUE NOT NULL,
7     nome VARCHAR(150),
8     orcamento decimal DECIMAL,
9     data_partida DATE,
10    data_retorno DATE,
11    descricao_preferencias_passo TEXT,
12    descricao_preferencias_hospedagem TEXT,
13    descricao_perfil_grupo TEXT,
14    orcamento_maximo DECIMAL,
15    id_cidade_origem INTEGER,
16    id_cidade_destino INTEGER,
17    id_cliente_responsavel INTEGER
18 );
19
20 CREATE TABLE PACOTE_VIAGEM (
21     id SERIAL UNIQUE NOT NULL,
22     id_solicitacao_viagem DEFAULT false NOT NULL,
23     id_solicitacao_viagem INTEGER NOT NULL,
24 );
25
26 CREATE TABLE EMPRESA (
27     id SERIAL NOT NULL UNIQUE,
28     razao_social VARCHAR(255) NOT NULL,
29     nome_fantasia VARCHAR(255),
30     cnpj VARCHAR(14) NOT NULL UNIQUE,
31     id_ramo_empresa INTEGER NOT NULL,
32     id_endereco INTEGER
33 );
34
35 CREATE TABLE RESERVA_QUARTO (
36     id SERIAL UNIQUE NOT NULL,
37     numero_quarto VARCHAR(255),
38     data_checkin DATE,
39     hora_checkin TIME,
40     previsao_dias_hospedagem INTEGER,
41     data_previsao_checkout DATE,
42     hora_previsao_checkout TIME,
43     valor_diaria DECIMAL,
44     id_pacote_viagem INTEGER,
45     id_hospedaria INTEGER
46 );
47
48 CREATE TABLE RESERVA_VEICULO (
49     id SERIAL UNIQUE NOT NULL,
50     numero_veiculo VARCHAR(255),
51     placa_veiculo VARCHAR(255),
52     preco_diaria DECIMAL,
53     data_inicio DATE,
54     hora_inicio TIME,
55     previsao_dias_aluguel INTEGER,
56     data_previsao_entrega DATE,
57     hora_previsao_entrega TIME,
58     id_pacote_viagem INTEGER,
59     id_tipo_veiculo INTEGER,
60     id_empresa_locadora INTEGER
61 );
62
63 CREATE TABLE PASSO (
64     id SERIAL NOT NULL UNIQUE,
65     nome VARCHAR(255) NOT NULL,
66     descricao TEXT,
67     id_endereco INTEGER
68 );
69
70 CREATE TABLE ENDERECO (
71     id SERIAL NOT NULL UNIQUE,
72     logradouro VARCHAR(255),
73     numero INTEGER,
74     complemento VARCHAR(255),
75     cep VARCHAR(8),
76     id_cidade INTEGER NOT NULL
77 );
78
79 -- Adicionando as constraints para a tabela AGENDA_PASSEIO_VIAGEM
80 ALTER TABLE AGENDA_PASSEIO_VIAGEM ADD CONSTRAINT FK_AGENDA_PASSEIO
81 FOREIGN KEY (id_pacote_viagem)
82 REFERENCES PACOTE_VIAGEM (id);
83
84 ALTER TABLE AGENDA_PASSEIO_VIAGEM ADD CONSTRAINT FK_AGENDA_PASSEIO
85 FOREIGN KEY (id_passo)
86 REFERENCES PASSO (id);
87
88 -- Adicionando as constraints para a tabela CONTATO_EMPRESA
89 ALTER TABLE CONTATO_EMPRESA ADD CONSTRAINT FK_CONTATO_EMPRESA_TIPO
90 FOREIGN KEY (id_tipo_contato)
91 REFERENCES TIPO_CONTATO (id);
92
93 ALTER TABLE CONTATO_EMPRESA ADD CONSTRAINT FK_CONTATO_EMPRESA_EMPR
94 FOREIGN KEY (id_empresa)
95 REFERENCES EMPRESA (id);
96
97 -- Adicionando as constraints para a tabela ACOMPANHANTE_VIAGEM
98 ALTER TABLE ACOMPANHANTE_VIAGEM ADD CONSTRAINT FK_ACOMPANHANTE_VIA
99 FOREIGN KEY (id_cliente)
100 REFERENCES CLIENTE (id);
101
102 ALTER TABLE ACOMPANHANTE_VIAGEM ADD CONSTRAINT FK_ACOMPANHANTE_VIA
103 FOREIGN KEY (id_solicitacao_viagem)
104 REFERENCES SOLICITACAO_VIAGEM (id);
105
106 -- Adicionando as constraints para a tabela SITUACAO_ATENDIMENTO
107 ALTER TABLE SITUACAO_ATENDIMENTO ADD CONSTRAINT FK_SITUACAO_ATENDI
108 FOREIGN KEY (id_solicitacao_viagem)
109 REFERENCES SOLICITACAO_VIAGEM (id);
110
111 ALTER TABLE SITUACAO_ATENDIMENTO ADD CONSTRAINT FK_SITUACAO_ATENDI
112 FOREIGN KEY (id_status)
113 REFERENCES STATUS (id);
114
115 -- Adicionando as constraints para a tabela PROCURA_VEICULO_VIAGEM
116 ALTER TABLE PROCURA_VEICULO_VIAGEM ADD CONSTRAINT FK_PROCURA_VEICU
117 FOREIGN KEY (id_tipo_veiculo)
118 REFERENCES TIPO_VEICULO (id);
119
120 ALTER TABLE PROCURA_VEICULO_VIAGEM ADD CONSTRAINT FK_PROCURA_VEICU
121 FOREIGN KEY (id_solicitacao_viagem)
122 REFERENCES SOLICITACAO_VIAGEM (id);
123
124 COMMIT;

```

Inserts

Contagem de solicitações de viagem por cidade de destino

```
res = pd.read_sql_query("""
SELECT C.nome AS cidade_destino, COUNT(SV.id) AS total_solicitacoes
FROM CIDADE C
LEFT JOIN SOLICITACAO_VIAGEM SV ON C.id = SV.id_cidade_destino
GROUP BY C.nome
ORDER BY total_solicitacoes desc;
""", conn)
```

res

<ipython-input-52-f2e55fe026f6>:3: UserWarning: pandas only supports
res = pd.read_sql_query("""

	cidade_destino	total_solicitacoes
0	Paraty	2
1	Foz do Iguaçu	1
2	Balneário Camboriú	1
3	São Paulo	1
4	Lençóis	1
5	Búzios	1
6	Natal	1
7	Porto de Galinhas	1
8	Salvador	1
9	Chapada Diamantina	1
10	Domingos Martins	1

[] # Contagem de clientes por estado

```
res = pd.read_sql_query("""
SELECT E.nome AS estado, COUNT(CL.id) AS total_clientes
FROM ESTADO E
LEFT JOIN CIDADE C ON E.id = C.id_estado
LEFT JOIN CLIENTE CL ON C.id = CL.id
GROUP BY E.nome
ORDER BY total_clientes desc;
""", conn)
```

res

<ipython-input-61-ba2d795d1747>:4: UserWarning: pandas only
res = pd.read_sql_query("""

	estado	total_clientes
0	Bahia	10
1	Rio de Janeiro	6
2	Pernambuco	4
3	Ceará	3
4	Santa Catarina	3
5	Paraná	2
6	Amazonas	1
7	São Paulo	1
8	Rio Grande do Sul	0
9	Rio Grande do Norte	0
10	Espírito Santo	0

Média de valor das diárias de hospedagem por cidade

```
res = pd.read_sql_query("""
SELECT c.nome AS cidade,
      AVG(rq.valor_diaria) AS media_valor_diaria
FROM RESERVA_QUARTO rq
JOIN HOSPEDARIA h ON rq.id_hospedaria = h.id
JOIN CIDADE c ON h.id = c.id
GROUP BY c.nome
ORDER BY media_valor_diaria desc;
""", conn)
```

res

<ipython-input-60-195552d9d5b4>:3: UserWarning: pandas
res = pd.read_sql_query("""

	cidade	media_valor_diaria
0	São Paulo	350.0
1	Porto Seguro	300.0
2	Florianópolis	250.0
3	Jericoacoara	220.0
4	Morro de São Paulo	200.0
5	Recife	160.0
6	Arraial d'Ajuda	150.0
7	Ilhéus	120.0
8	Chapada Diamantina	100.0
9	Olinda	90.0

Group By e funções de
agrupamento

Valor total de pagamentos por método de pagamento

```
res = pd.read_sql_query("""
SELECT MP.descricao AS metodo_pagamento, SUM(P.valor) AS total_pagamentos
FROM METODO_PAGAMENTO MP
LEFT JOIN PAGAMENTO P ON MP.id = P.id_metodo_pagamento
GROUP BY MP.descricao
ORDER BY total_pagamentos desc;
""", conn)
```

res

<ipython-input-62-0dadab91513f>:3: UserWarning: pandas only supports SQLAlchemy
res = pd.read_sql_query("""

	metodo_pagamento	total_pagamentos
0	Cartão de crédito	6725.0
1	Picpay	2800.0
2	Paypal	1800.0
3	Cartão de débito	1660.0
4	DOC	1425.0
5	Pix	1412.0
6	Dinheiro	600.0
7	TED	540.0
8	Ame Digital	360.0

Group By e funções de agrupamento

Valor máximo de desconto em pagamento por pacote de viagem

```
res = pd.read_sql_query("""
SELECT PV.id, MAX(P.desconto) AS max_desconto
FROM PACOTE_VIAGEM PV
INNER JOIN PAGAMENTO P ON PV.id = P.id_pacote_viagem
GROUP BY PV.id
ORDER BY MAX(P.desconto) DESC;
""", conn)
```

res

<ipython-input-65-e0d9b91f416d>:3: UserWarning: pandas only supports SQLAlchemy
res = pd.read_sql_query("""

	id	max_desconto
0	3	50.0
1	10	0.0
2	6	0.0
3	2	0.0
4	11	0.0
5	9	0.0
6	12	0.0
7	5	0.0
8	7	0.0
9	4	0.0
10	1	0.0

Quantidade de reservas de veículos por tipo de veículo

```
res = pd.read_sql_query("""
SELECT TV.descricao AS tipo_veiculo, COUNT(RV.id) AS total_reservas
FROM TIPO_VEICULO TV
LEFT JOIN RESERVA_VEICULO RV ON TV.id = RV.id_tipo_veiculo
GROUP BY TV.descricao
ORDER BY total_reservas desc;
""", conn)
```

res

<ipython-input-63-300e5488a626>:3: UserWarning: pandas only supports SQLAlchemy
res = pd.read_sql_query("""

	tipo_veiculo	total_reservas
0	Carro esportivo	2
1	Carro básico econômico com ar condicionado	2
2	Carro básico econômico sem ar condicionado	1
3	Carro automático	1
4	Caminhonete	0
5	Carro compacto	0
6	Pickup	0
7	Minivan	0
8	Triciclo	0
9	Vespa	0
10	Carro familiar	0
11	Carro blindado	0
12	Quadriciclo	0

Lista de solicitações de viagem e número do quarto reservado

```
res = pd.read_sql_query("""
SELECT SV.id, SV.nome AS nome_viagem, COALESCE(RQ.numero_quarto, 'Sem reserva') AS numero_quarto
FROM SOLICITACAO_VIAGEM SV
LEFT JOIN RESERVA_QUARTO RQ ON SV.id = RQ.id_pacote_viagem;
""", conn)
```

res

<ipython-input-53-d52a1fe9442d>:3: UserWarning: pandas only supports SQLAlchemy connectable (engine,

```
res = pd.read_sql_query("""
```

	id	nome_viagem	numero_quarto
0	1	Patrícia: Arraial do Cabo - Balneário Camboriu	101
1	2	Patrícia: Arraial do Cabo - Chapada Diamantina	205
2	3	Patrícia: Arraial do Cabo - Paraty	103
3	4	Patrícia: Arraial do Cabo - Búzios	302
4	4	Patrícia: Arraial do Cabo - Búzios	303
5	4	Patrícia: Arraial do Cabo - Búzios	304
6	5	Kaique da Mata: Recife - São Paulo	415
7	5	Kaique da Mata: Recife - São Paulo	416
8	5	Kaique da Mata: Recife - São Paulo	417
9	6	Viagem de amigas de Antônia: Rio de Janeiro - ...	101
10	7	Viagem em família César Novaes : Serra - Salvador	201
11	9	Luíz Lopes Viagem de casal: Florianópolis - Fo...	208
12	10	Otávio Conceição viagem com irmãos: Serra - Do...	206

Lista de reservas de quartos e nome da solicitação de viagem associada

```
res = pd.read_sql_query("""
SELECT RQ.id, RQ.numero_quarto, COALESCE(SV.nome, 'Sem solicitação associada') AS nome_viagem
FROM RESERVA_QUARTO RQ
RIGHT JOIN SOLICITACAO_VIAGEM SV ON SV.id = RQ.id_pacote_viagem;
""", conn)
```

res

<ipython-input-54-b57e6c5cb1d3>:3: UserWarning: pandas only supports SQLAlchemy connectable (engine,

```
res = pd.read_sql_query("""
```

	id	numero_quarto	nome_viagem
0	18.0	101	Patrícia: Arraial do Cabo - Balneário Camboriu
1	19.0	205	Patrícia: Arraial do Cabo - Chapada Diamantina
2	20.0	103	Patrícia: Arraial do Cabo - Paraty
3	21.0	302	Patrícia: Arraial do Cabo - Búzios
4	22.0	303	Patrícia: Arraial do Cabo - Búzios
5	23.0	304	Patrícia: Arraial do Cabo - Búzios
6	24.0	415	Kaique da Mata: Recife - São Paulo
7	25.0	416	Kaique da Mata: Recife - São Paulo
8	26.0	417	Kaique da Mata: Recife - São Paulo
9	27.0	101	Viagem de amigas de Antônia: Rio de Janeiro - ...
10	28.0	201	Viagem em família César Novaes : Serra - Salvador
11	29.0	208	Luíz Lopes Viagem de casal: Florianópolis - Fo...
12	30.0	306	Otávio Conceição viagem com irmãos: Serra - Do...
13	31.0	307	Otávio Conceição viagem com irmãos: Serra - Do...
14	32.0	308	Otávio Conceição viagem com irmãos: Serra - Do...
15	33.0	402	Olivia Anarício: Caivaze do Sul - Porto de Cal

Left, Right e Full Join

Consulta usando LEFT JOIN para visualizar todas as hospedarias e suas reservas de quartos associadas

```
res = pd.read_sql_query("""
SELECT H.id AS id_hospedaria, E.nome_fantasia AS nome_hospedaria, COALESCE(RQ.numero_quarto, 'Sem reserva') AS numero_quarto
FROM HOSPEDARIA H
LEFT JOIN RESERVA_QUARTO RQ ON H.id = RQ.id_hospedaria
LEFT JOIN EMPRESA E ON H.id = E.id;
""", conn)
```

res

<ipython-input-56-11f65d64e055>:3: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database str

res = pd.read_sql_query("""

	id_hospedaria	nome_hospedaria	numero_quarto
0	8	Gran Hotel	101
1	11	Villa da Montanha	205
2	13	Hotel Spa das Termas	103
3	15	Resort Paradise	302
4	15	Resort Paradise	303
5	15	Resort Paradise	304
6	16	Daybreak Hotel	415
7	16	Daybreak Hotel	416
8	16	Daybreak Hotel	417
9	18	Argos Hotel	101
10	20	Casa do Mar	201
11	25	Hotel Colonial	208
12	26	Refúgio das Montanhas	306
13	26	Refúgio das Montanhas	307
14	26	Refúgio das Montanhas	308
15	28	Parque das Águas	402

Left, Right e Full Join

Correspondência entre solicitações de viagem e reservas de quartos

```
res = pd.read_sql_query("""
SELECT SV.id AS id_solicitacao, RQ.id AS id_reserva
FROM SOLICITACAO_VIAGEM SV
FULL JOIN RESERVA_QUARTO RQ ON SV.id = RQ.id_pacote_viagem;
""", conn)
```

res

<ipython-input-55-33b1364d415b>:3: UserWarning: pandas only supports SQLA

res = pd.read_sql_query("""

	id_solicitacao	id_reserva
0	1	18.0
1	2	19.0
2	3	20.0
3	4	21.0
4	4	22.0
5	4	23.0
6	5	24.0
7	5	25.0
8	5	26.0
9	6	27.0
10	7	28.0
11	9	29.0
12	10	30.0
13	10	31.0
14	10	32.0
15	11	33.0
16	12	34.0


```

res = pd.read_sql_query("""
CREATE OR REPLACE VIEW Detalhes_Reserva_Quarto AS
SELECT rq.id AS id_reserva, rq.numero_quarto, rq.data_checkin, rq.data_previsao_checkout,
       h.id AS id_hospedaria, h.descricao AS descricao_hospedaria, h.numero_estrelas,
       pv.id AS id_pacote_viagem, pv.foi_escolhido
FROM RESERVA_QUARTO rq
LEFT JOIN HOSPEDARIA h ON rq.id_hospedaria = h.id
LEFT JOIN PACOTE_VIAGEM pv ON rq.id_pacote_viagem = pv.id;

CREATE OR REPLACE VIEW Detalhes_Reserva_Hotel_Cliente AS
SELECT
    drq.id_reserva,
    drq.numero_quarto,
    drq.data_checkin,
    drq.data_previsao_checkout,
    e.razao_social AS nome_hotel,
    c.nome_social AS nome_cliente
FROM Detalhes_Reserva_Quarto drq
LEFT JOIN HOSPEDARIA h ON drq.id_hospedaria = h.id
LEFT JOIN EMPRESA e ON h.id = e.id
LEFT JOIN PACOTE_VIAGEM pv ON drq.id_pacote_viagem = pv.id
LEFT JOIN SOLICITACAO_VIAGEM sv ON pv.id_solicitacao_viagem = sv.id
LEFT JOIN CLIENTE c ON sv.id_cliente_responsavel = c.id;

SELECT * FROM Detalhes_Reserva_Hotel_Cliente;
""", conn)

res

```

```

query = """
SELECT
    viagem.nome as nome_viagem,
    cliente.nome_social as nome_acompanhante,
    EXTRACT(YEAR FROM AGE(CURRENT_DATE, cliente.data_nascimento)) as idade,
    tipo_necessidade_especial.descricao as necessidade_especial
FROM CLIENTE
LEFT JOIN necessidade_especial_cliente ON necessidade_especial_cliente.id_cliente = cliente.id
LEFT JOIN tipo_necessidade_especial ON tipo_necessidade_especial.id = necessidade_especial_cliente.id_necessidade_especial
INNER JOIN acompanhante_viagem ON acompanhante_viagem.id_cliente = cliente.id
INNER JOIN solicitacao_viagem as viagem ON viagem.id = acompanhante_viagem.id_solicitacao_viagem

WHERE viagem.id_cliente_responsavel IN (
    SELECT id FROM cliente
    WHERE EXTRACT(YEAR FROM AGE(CURRENT_DATE, cliente.data_nascimento)) BETWEEN 30 AND 45
);
"""

res = pd.read_sql_query(query, conn)
res

```

<ipython-input-87-42891b68e036>:19: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database

res = pd.read_sql_query(query, conn)

	nome_viagem	nome_acompanhante	idade	necessidade_especial
0	Kaique da Mata: Recife - São Paulo	Beatriz Leticia Lorena da Mota	23.0	None
1	Kaique da Mata: Recife - São Paulo	Noah Ian Carlos da Luz	20.0	None
2	Kaique da Mata: Recife - São Paulo	Osvaldo Manuel Figueiredo	32.0	None
3	Kaique da Mata: Recife - São Paulo	Luna Sueli da Rocha	30.0	None

<ipython-input-86-a61d91892ea9>:1: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite

res = pd.read_sql_query("""

	id_reserva	numero_quarto	data_checkin	data_previsao_checkout	nome_hotel	nome_cliente
0	18	101	2024-02-12	2024-02-18	Gran Hotel Hotelaria SA	Patrícia Luiza Jennifer Silveira
1	19	205	2024-07-24	2024-07-28	Villa da Montanha SA	Patrícia Luiza Jennifer Silveira
2	20	103	2024-12-28	2025-01-03	Hotel Spa das Termas EPP Ltda	Patrícia Luiza Jennifer Silveira
3	23	304	2024-09-12	2024-09-16	Resort Paradise SA	Patrícia Luiza Jennifer Silveira
4	23	303	2024-09-12	2024-09-16	Resort Paradise SA	Patrícia Luiza Jennifer Silveira

View e Subconsulta



Perguntas a serem respondidas

1

Quais são os lugares mais visitados pelos clientes, por período e idade dos clientes

3

Quais os passeios mais escolhidos pelos clientes dada uma lista de cidades

2

Quais os hotéis mais escolhidos pelos clientes no Brasil

4

Qual a quantidade de propostas vendidas e perdidas em um determinado período do tempo

5

Qual o faturamento da empresa em um determinado período de tempo

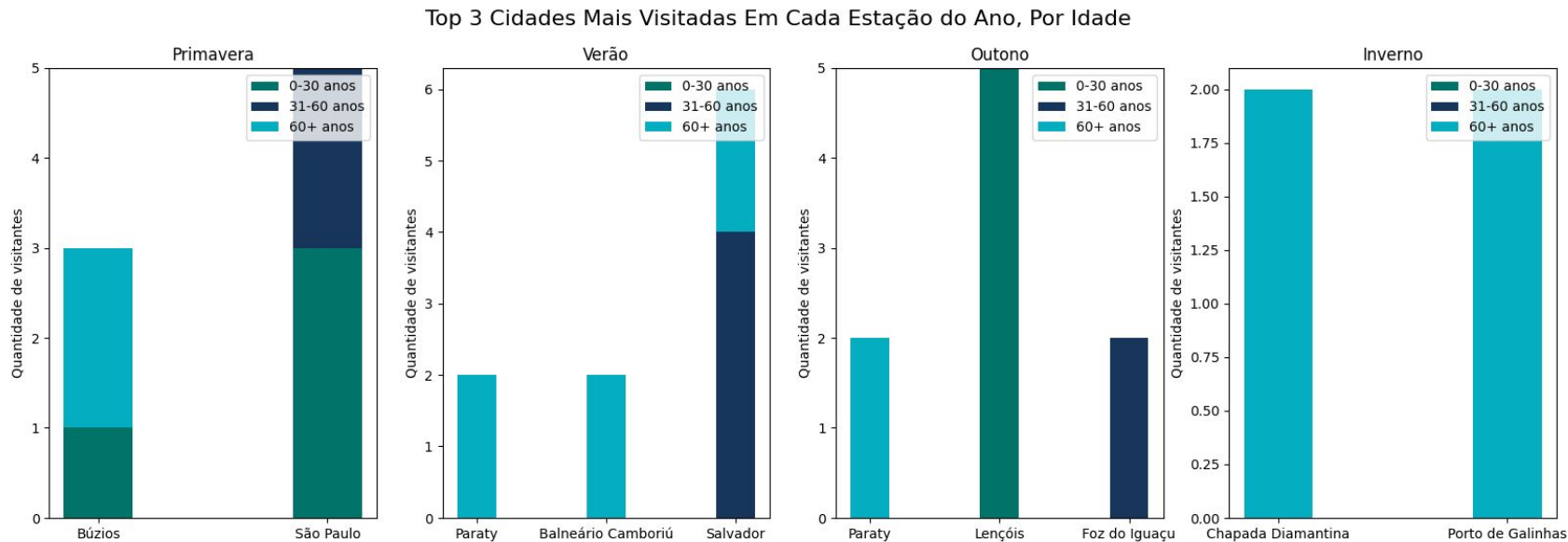
Relatório 1: Top 3 cidades mais visitadas em cada estação do ano, por idade

```
queryCidades = """
SELECT
    cidade.id as id_cidade,
    viagem.id as id_viagem,
    cidade.nome as nome_cidade,
    count(data_partida) as quantidade_viagens,
    COUNT(CASE WHEN DATE_PART('Year', AGE(current_date, cliente_responsavel.data_nascimento)) BETWEEN 0 AND 30 THEN 1 END) as pessoas_0_30,
    COUNT(CASE WHEN DATE_PART('Year', AGE(current_date, cliente_responsavel.data_nascimento)) BETWEEN 31 AND 60 THEN 1 END) as pessoas_31_60,
    COUNT(CASE WHEN DATE_PART('Year', AGE(current_date, cliente_responsavel.data_nascimento)) > 60 THEN 1 END) as pessoas_mais_60
FROM solicitacao_viagem as viagem
INNER JOIN cidade ON cidade.id = viagem.id_cidade_destino
INNER JOIN cliente as cliente_responsavel ON cliente_responsavel.id = viagem.id_cliente_responsavel
WHERE EXTRACT(MONTH FROM data_partida) IN %s
GROUP BY cidade.id, viagem.id
ORDER BY quantidade_viagens
LIMIT 3;
"""

queryAcompanhantes = """
SELECT
    id_cidade_destino,
    cidade.nome as nome_cidade,
    COUNT(CASE WHEN DATE_PART('Year', AGE(current_date, acompanhante.data_nascimento)) BETWEEN 0 AND 30 THEN 1 END) as pessoas_0_30,
    COUNT(CASE WHEN DATE_PART('Year', AGE(current_date, acompanhante.data_nascimento)) BETWEEN 31 AND 60 THEN 1 END) as pessoas_31_60,
    COUNT(CASE WHEN DATE_PART('Year', AGE(current_date, acompanhante.data_nascimento)) > 60 THEN 1 END) as pessoas_mais_60
FROM acompanhante_viagem
INNER JOIN cliente as acompanhante ON acompanhante.id = acompanhante_viagem.id_cliente
RIGHT JOIN solicitacao_viagem as viagem ON viagem.id = acompanhante_viagem.id_solicitacao_viagem
RIGHT JOIN cidade ON cidade.id = viagem.id_cidade_destino
WHERE id_cidade_destino IN %s AND viagem.id IN %s
GROUP BY id_cidade_destino, cidade.nome;
"""
```

	Primavera	Verão	Outono	Inverno
Búzios	[1, 0, 2]	NaN	NaN	NaN
São Paulo	[3, 2, 0]	NaN	NaN	NaN
Paraty	NaN	[0, 0, 2]	[0, 0, 2]	NaN
Balneário Camboriú	NaN	[0, 0, 2]	NaN	NaN
Salvador	NaN	[0, 4, 2]	NaN	NaN
Lençóis	NaN	NaN	[5, 0, 0]	NaN
Foz do Iguaçu	NaN	NaN	[0, 2, 0]	NaN
Chapada Diamantina	NaN	NaN	NaN	[0, 0, 2]
Porto de Galinhas	NaN	NaN	NaN	[0, 0, 2]

Relatório 1: Top 3 cidades mais visitadas em cada estação do ano, por idade



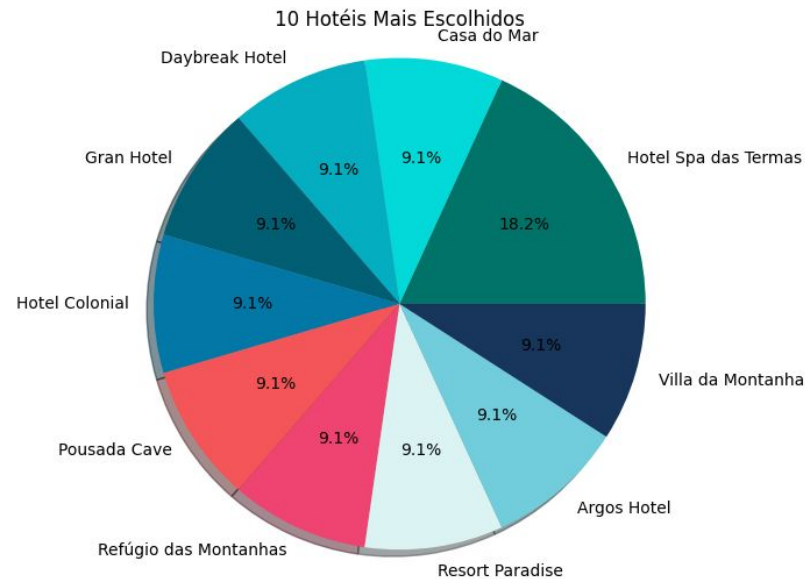
Relatório 2: Top 10 hotéis mais escolhidos pelos clientes no Brasil

```
res = pd.read_sql_query("""
SELECT
    E.nome_fantasia AS nome_hotel,
    CIDADE.nome AS cidade,
    COUNT(DISTINCT P.id) AS total_escolhas
FROM HOSPEDARIA H
INNER JOIN RESERVA_QUARTO R ON H.id = R.id_hospedaria
INNER JOIN PACOTE_VIAGEM P ON R.id_pacote_viagem = P.id
INNER JOIN EMPRESA E ON H.id = E.id
INNER JOIN ENDEREÇO ON E.id = ENDEREÇO.id
INNER JOIN CIDADE ON ENDEREÇO.id_cidade = CIDADE.id
GROUP BY E.nome_fantasia, CIDADE.nome
ORDER BY total_escolhas DESC
LIMIT 10;
""", conn)
res
```

	nome_hotel	cidade	total_escolhas
0	Hotel Spa das Termas	Paraty	2
1	Casa do Mar	Salvador	1
2	Daybreak Hotel	São Paulo	1
3	Gran Hotel	Balneário Camboriú	1
4	Hotel Colonial	Foz do Iguaçu	1
5	Pousada Cave	Porto de Galinhas	1
6	Refúgio das Montanhas	Domingos Martins	1
7	Resort Paradise	Búzios	1
8	Argos Hotel	Lençóis	1
9	Villa da Montanha	Chapada Diamantina	1



Relatório 2: Top 10 hotéis mais escolhidos pelos clientes no Brasil

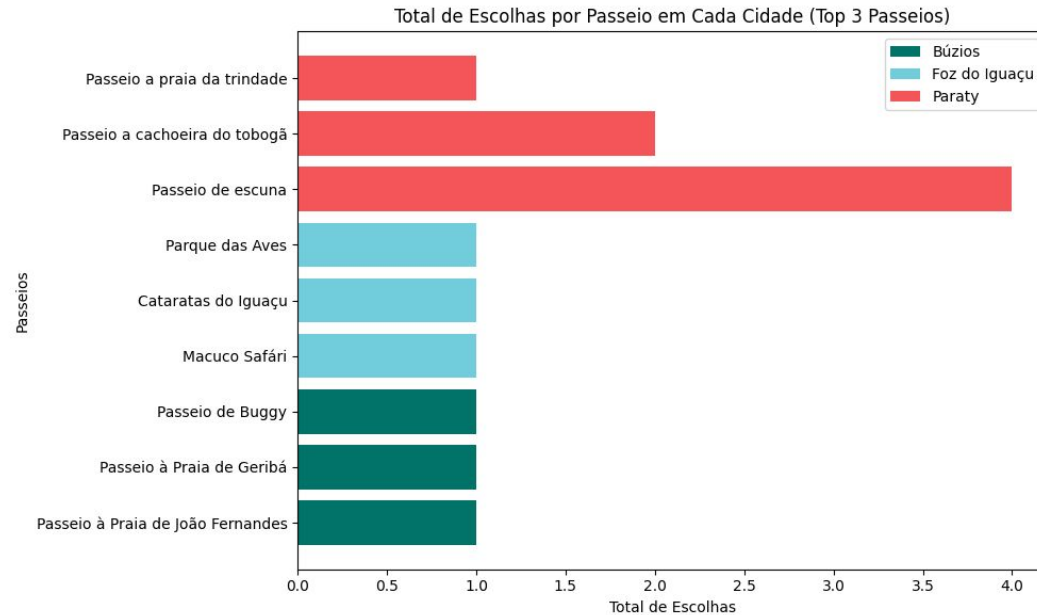


Relatório 3: Passeios mais escolhidos pelos clientes nas cidades mais visitadas

```
res = pd.read_sql_query("""
    WITH RankedPasseios AS (
    SELECT
        C.nome AS nome_cidade,
        P.nome AS nome_passeio,
        COUNT(*) AS total_escolhas,
        ROW_NUMBER() OVER(PARTITION BY C.nome ORDER BY COUNT(*) DESC) AS rank
    FROM PACOTE_VIAGEM PV
    INNER JOIN SOLICITACAO_VIAGEM SV ON PV.id_solicitacao_viagem = SV.id
    INNER JOIN CIDADE C ON SV.id_cidade_destino = C.id
    INNER JOIN AGENDA_PASSEIO_VIAGEM APV ON PV.id = APV.id_pacote_viagem
    INNER JOIN PASSEIO P ON APV.id_passeio = P.id
    WHERE C.id IN (
        SELECT id_cidade_destino
        FROM SOLICITACAO_VIAGEM
        GROUP BY id_cidade_destino
        ORDER BY COUNT(*) DESC
        LIMIT 3
    )
    GROUP BY C.nome, P.nome
    )
    SELECT nome_cidade, nome_passeio, total_escolhas
    FROM RankedPasseios
    WHERE rank <= 3
    ORDER BY nome_cidade, total_escolhas DESC; """,conn)
res
```

	nome_cidade	nome_passeio	total_escolhas
0	Búzios	Passeio à Praia de João Fernandes	1
1	Búzios	Passeio à Praia de Geribá	1
2	Búzios	Passeio de Buggy	1
3	Foz do Iguaçu	Macuco Safári	1
4	Foz do Iguaçu	Cataratas do Iguaçu	1
5	Foz do Iguaçu	Parque das Aves	1
6	Paraty	Passeio de escuna	4
7	Paraty	Passeio a cachoeira do tobogã	2
8	Paraty	Passeio a praia da trindade	1

Relatório 3: Passeios mais escolhidos pelos clientes nas cidades mais visitadas



Relatório 4: Quantidade de propostas vendidas e perdidas em 2023

```
inicio = "2023-01-01"
fim = "2023-12-31"
query = """
SELECT
    SUM(CASE WHEN sa.id_status = 3 THEN 1 ELSE 0 END) AS propostas_vendidas,
    SUM(CASE WHEN sa.id_status = 5 THEN 1 ELSE 0 END) AS propostas_perdidas
FROM SITUACAO_ATENDIMENTO sa
INNER JOIN SOLICITACAO_VIAGEM sv ON sa.id_solicitacao_viagem = sv.id
WHERE
    sa.data_hora_mudanca >= %s
    AND sa.data_hora_mudanca <= %s;
"""

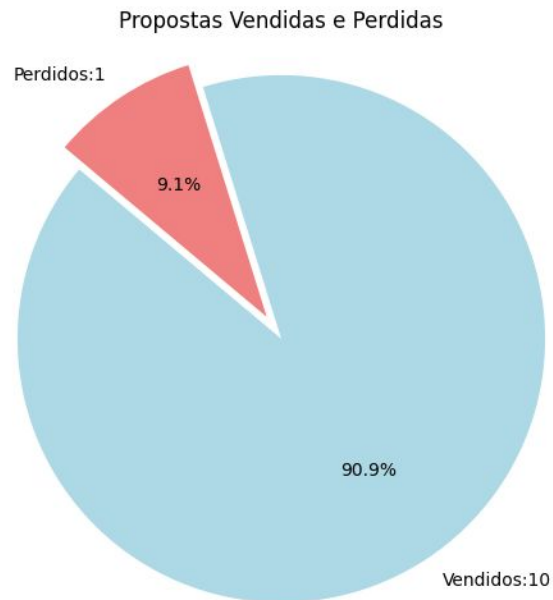
# Execução da consulta SQL com os parâmetros
res = pd.read_sql_query(query, conn, params=(inicio, fim))

print(res)
```

	propostas_vendidas	propostas_perdidas
0	10	1



Relatório 4: Quantidade de propostas vendidas e perdidas em 2023





Relatório 4: Renda bruta da empresa em 2023, por trimestre

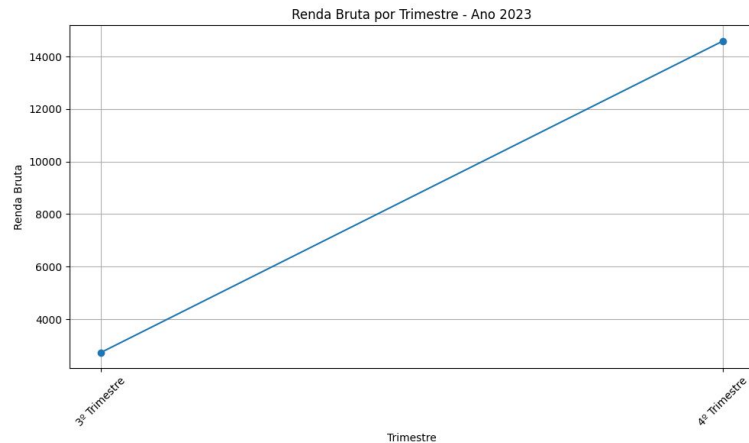
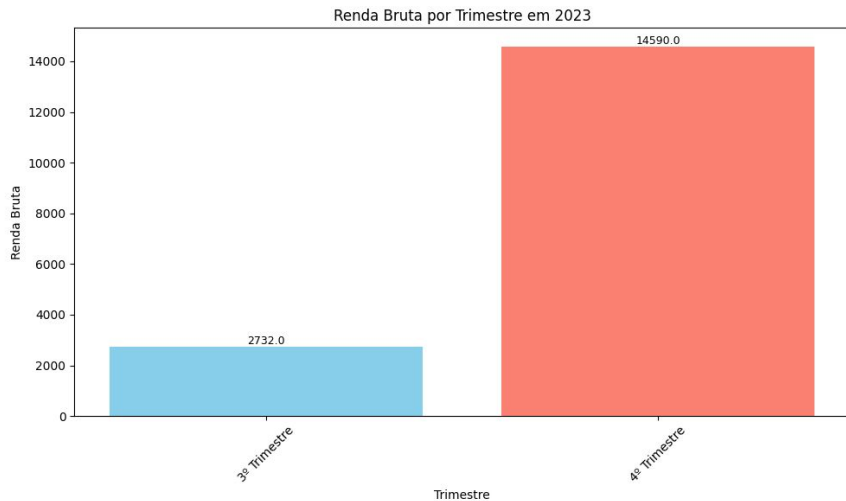
```
ano = "2023"
query = """
SELECT
    EXTRACT(QUARTER FROM p.data_hora_pagamento) AS trimestre,
    SUM(valor) AS renda_trimestral
FROM PAGAMENTO p
JOIN PACOTE_VIAGEM pv ON p.id_pacote_viagem = pv.id
JOIN SOLICITACAO_VIAGEM sv ON pv.id_solicitacao_viagem = sv.id
WHERE EXTRACT(YEAR FROM p.data_hora_pagamento) = %s
GROUP BY trimestre
ORDER BY trimestre;
"""
```

Renda Bruta 2023

trimestre	
3º Trimestre	2732.0
4º Trimestre	14590.0



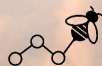
Relatório 4: Renda bruta da empresa em 2023, por trimestre



Obrigado!



VIAGENS BSI



DEVORA

POLLINATING IDEAS