

Candy System

Componentes do grupo: Camila Volponi e Vitor da Silva

Motivação

- A criação do sistema visa gerar uma melhoria na qualidade do trabalho de pequenos e médios confeitadores e confeitarias. Provendo um sistema flexível capaz de auxiliar no gerenciamento de pedidos e produtos, capaz de ser usado por qualquer organização de pequeno e médio porte.

Minimundo

- Um confeitador deseja possuir um sistema para gerenciar sua empresa de confeitaria, registrando pedidos e produtos.
- Do produto, precisa saber: o nome, preço e quantidade, onde terá acesso aos ingredientes e aos modos de preparo.
- Dos ingredientes precisa-se do nome, quantidade e unidade de medida.
- Do modo de preparo precisa-se de cada passo para o preparo do produto e a ordem que será na receita.
- Do pedido, precisa ter: a data de entrega e horário previstos, forma de pagamento e ter os dados do cliente registrado, como nome, telefone e endereço.

Prototipação e Tabela de Dados

Candy System

https://candysystem.com.br/

INSERIR

Cliente

Nome: telefone

Endereço:

Pedido

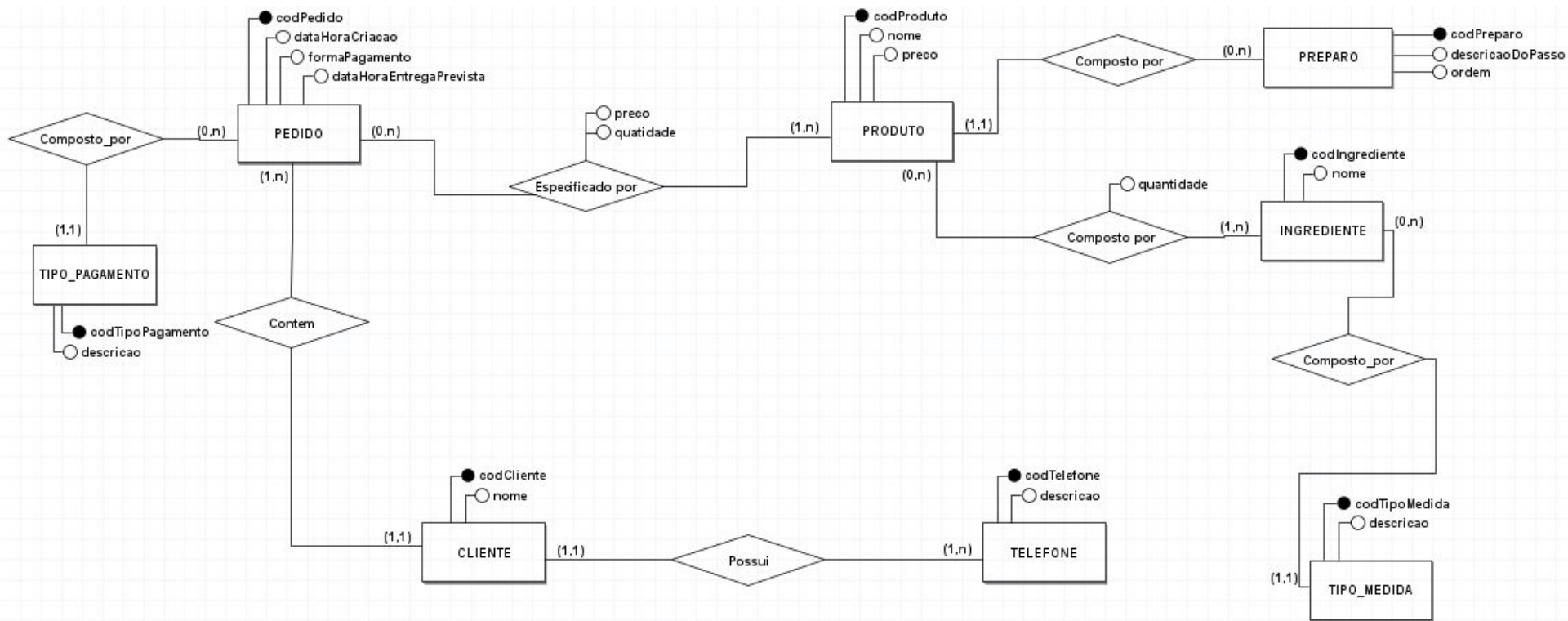
Para o dia: Para o horário: Forma de pagamento:

Produtos

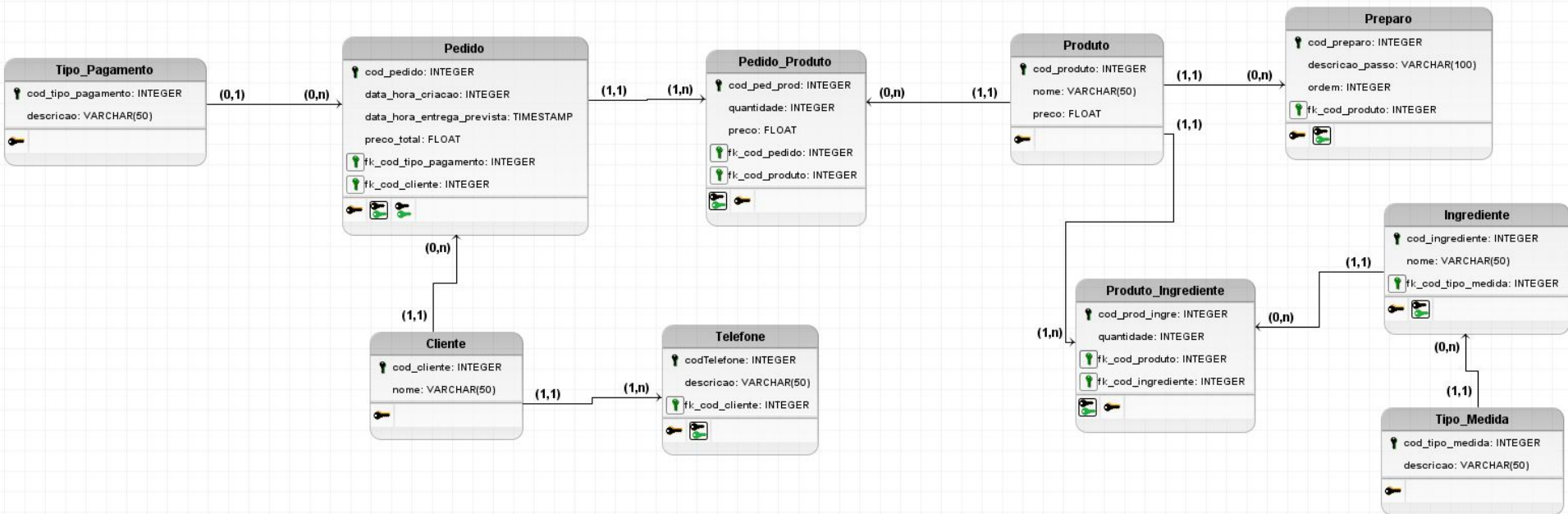
Produto	Quantidade	Ações
Pão	2,00	<div>Editar</div> <div>Excluir</div>

	F	G	H	I	J	K
	Pedido					
cc	cod_pedido	data_hora_criacao	data_hora_entrega_prevista	cod_tipo_pagamento	cod_CLIENTE_cod_pessoa	preco_total
	1	18/06/22 - 18:51	18/06/22 - 20:00	1	1	R\$ 50,00
	1	18/06/22 - 18:51	18/06/22 - 20:00	1	1	R\$ 50,00
	1	18/06/22 - 18:51	18/06/22 - 20:00	1	1	R\$ 50,00
	1	18/06/22 - 18:51	18/06/22 - 20:00	1	1	R\$ 50,00
	1	18/06/22 - 18:51	18/06/22 - 20:00	1	1	R\$ 50,00
	1	18/06/22 - 18:51	18/06/22 - 20:00	1	1	R\$ 50,00
	1	18/06/22 - 18:51	18/06/22 - 20:00	1	1	R\$ 50,00
	2	18/06/22 - 19:00	19/06/22 - 10:00	5	2	R\$ 50,00
	2	18/06/22 - 19:00	19/06/22 - 10:00	5	2	R\$ 50,00
	2	18/06/22 - 19:00	19/06/22 - 10:00	5	2	R\$ 50,00
	2	18/06/22 - 19:00	19/06/22 - 10:00	5	2	R\$ 50,00

Modelo Conceitual



Modelo Lógico



Modelo Físico

```
CREATE TABLE pedido(  
  cod_pedido INTEGER PRIMARY KEY,  
  data_hora_criacao TIMESTAMP,  
  data_hora_entrega_prevista TIMESTAMP,  
  preco_total FLOAT(4),  
  fk_cod_tipo_pagamento INTEGER,  
  fk_cod_cliente INTEGER  
);  
CREATE TABLE tipo_pagamento (  
  cod_tipo_pagamento INTEGER PRIMARY KEY,  
  descricao VARCHAR(50)  
);  
CREATE TABLE pedido_produto(  
  cod_ped_prod INTEGER PRIMARY KEY,  
  quantidade INTEGER,  
  preco FLOAT,  
  fk_cod_pedido INTEGER,  
  fk_cod_produto INTEGER  
);  
CREATE TABLE produto (  
  cod_produto INTEGER PRIMARY KEY,  
  nome VARCHAR(50),  
  preco FLOAT(4)  
);  
CREATE TABLE preparo (  
  cod_preparo INTEGER PRIMARY KEY,  
  descricao_passo VARCHAR(500),  
  ordem INTEGER,  
  fk_cod_produto INTEGER  
);
```

```
CREATE TABLE produto_ingrediente (  
  cod_prod_ingre INTEGER PRIMARY KEY,  
  quantidade INTEGER,  
  fk_cod_produto INTEGER,  
  fk_cod_ingrediente INTEGER  
);  
CREATE TABLE ingrediente (  
  cod_ingrediente INTEGER PRIMARY KEY,  
  nome VARCHAR(50),  
  fk_cod_tipo_medida INTEGER  
);  
CREATE TABLE tipo_medida (  
  cod_tipo_medida INTEGER PRIMARY KEY,  
  descricao VARCHAR(50)  
);  
CREATE TABLE cliente (  
  cod_cliente INTEGER PRIMARY KEY,  
  nome VARCHAR(50)  
);  
CREATE TABLE telefone (  
  cod_telefone INTEGER PRIMARY KEY,  
  descricao VARCHAR(50),  
  fk_cod_cliente INTEGER  
);
```

Modelo Físico - Inserts

```
INSERT INTO produto VALUES
```

```
(1, 'Cento de brigadeiro', 50),  
(2, 'Bolo de fuba', 30),  
(3, 'Cento de beijinho', 50),  
(4, 'Bolo de chocolate', 30),  
(5, 'Pudim', 20),  
(6, 'Maria mole', 10),  
(7, 'Brownie', 25),  
(8, 'Bolo de cenoura', 30),  
(9, 'Bolo de limão', 30),  
(10, 'Suspiro', 15),  
(11, 'Palha italiana', 5),  
(12, 'Cookie', 6),  
(13, 'Doce de pote', 13),  
(14, 'Bombom', 9),  
(15, 'Trufa', 7),  
(16, 'Pipoca doce jumbo', 17),  
(17, 'Rocamboles de prestígio', 50),  
(18, 'Cupcake', 10),  
(19, 'Mousse de maracujá', 20),  
(20, 'Mousse de morango', 20);
```

```
INSERT INTO pedido VALUES
```

```
(1, '2022-05-17 10:11', '2022-05-21 10:11', 100, 1, 1),  
(2, '2022-05-18 10:11', '2022-05-22 10:11', 50, 5, 2),  
(3, '2022-05-19 10:11', '2022-05-23 10:11', 150, 3, 3),  
(4, '2022-05-20 10:11', '2022-05-24 10:11', 100, 3, 4),  
(5, '2022-05-21 10:11', '2022-05-25 10:11', 50, 2, 5),  
(6, '2022-06-17 10:11', '2022-06-21 10:11', 140, 1, 9),  
(7, '2022-03-06 10:11', '2022-03-10 10:11', 140, 2, 14),  
(8, '2022-04-26 10:11', '2022-04-30 10:11', 168, 3, 17),  
(9, '2022-06-15 10:11', '2022-06-25 10:11', 300, 5, 2),  
(10, '2022-12-07 10:11', '2022-12-12 10:11', 93, 3, 6),  
(11, '2022-10-13 10:11', '2022-10-15 10:11', 170, 5, 18),  
(12, '2022-11-24 10:11', '2022-11-28 10:11', 47, 1, 20),  
(13, '2022-09-12 10:11', '2022-09-19 10:11', 110, 4, 9),  
(14, '2022-10-20 10:11', '2022-10-30 10:11', 100, 2, 7),  
(15, '2022-04-26 10:11', '2022-04-30 10:11', 20, 3, 17),  
(16, '2022-09-19 10:11', '2022-09-22 10:11', 47, 3, 5),  
(17, '2022-11-15 10:11', '2022-11-24 10:11', 100, 2, 11),  
(18, '2022-08-14 10:11', '2022-08-22 10:11', 70, 3, 19),  
(19, '2022-06-10 10:11', '2022-06-11 10:11', 40, 4, 5),  
(20, '2022-04-06 10:11', '2022-04-25 10:11', 51, 1, 19),  
(21, '2022-10-08 10:11', '2022-10-13 10:11', 150, 1, 9),  
(22, '2022-07-13 10:11', '2022-07-17 10:11', 69, 5, 7),  
(23, '2022-12-01 10:11', '2022-12-15 10:11', 30, 3, 2),  
(24, '2022-06-25 10:11', '2022-06-29 10:11', 64, 1, 13),  
(25, '2022-08-02 10:11', '2022-08-04 10:11', 30, 3, 2),  
(26, '2022-03-18 10:11', '2022-03-19 10:11', 100, 4, 5),  
(27, '2022-08-15 10:11', '2022-08-18 10:11', 24, 1, 10),  
(28, '2022-12-04 10:11', '2022-12-20 10:11', 68, 4, 9),  
(29, '2022-04-20 10:11', '2022-04-27 10:11', 100, 2, 2),  
(30, '2022-06-09 10:11', '2022-06-12 10:11', 40, 5, 3),  
(31, '2022-05-29 10:11', '2022-05-30 10:11', 290, 3, 16);
```


Consulta com Inner Join e Order By

```
SELECT cliente.nome as nome_cliente,  
       telefone.descricao as telefone,  
       produto.nome as produto,  
       pedido.data_hora_entrega_prevista,  
       pedido.preco_total as valor_total_pedido,  
       tipo_pagamento.descricao as tipo_pagamento  
FROM pedido  
INNER JOIN cliente ON (pedido.fk_cod_cliente = cliente.cod_cliente)  
INNER JOIN telefone ON (cliente.cod_cliente = telefone.fk_cod_cliente)  
INNER JOIN tipo_pagamento ON (tipo_pagamento.cod_tipo_pagamento = pedido.fk_cod_tipo_pagamento)  
INNER JOIN pedido_produto ON (pedido.cod_pedido = pedido_produto.fk_cod_pedido)  
INNER JOIN produto ON (produto.cod_produto = pedido_produto.fk_cod_produto)  
ORDER BY cliente.nome
```

Table queries ▾

Previous queries ▾

Execute ▶

nome_cliente	telefone	produto	data_hora_entrega_prevista	valor_total_pedido	tipo_pagamento
Camila Volponi	28 94852-1566	Cookie	2022-06-25 10:11:00 +0000	300	Dinheiro
Camila Volponi	28 94852-1566	Cento de brigadeiro	2022-05-22 10:11:00 +0000	50	Dinheiro
Camila Volponi	27 99999-2222	Rocambole de prestigio	2022-04-27 10:11:00 +0000	100	PicPay
Camila Volponi	28 94852-1566	Rocambole de prestigio	2022-04-27 10:11:00 +0000	100	PicPay
Camila Volponi	27 99999-2222	Maria mole	2022-08-04 10:11:00 +0000	30	Crédito
Camila Volponi	27 99999-2222	Bolo de limão	2022-12-15 10:11:00 +0000	30	Crédito
Camila Volponi	27 99999-2222	Cookie	2022-06-25 10:11:00 +0000	300	Dinheiro
Camila Volponi	27 99999-2222	Cento de brigadeiro	2022-05-22 10:11:00 +0000	50	Dinheiro
Camila Volponi	28 94852-1566	Bolo de limão	2022-12-15 10:11:00 +0000	30	Crédito
Camila Volponi	28 94852-1566	Maria mole	2022-08-04 10:11:00 +0000	30	Crédito

Consulta com Inner Join e Order By

```
SELECT cliente.nome as nome_cliente,  
       pedido.cod_pedido as numero_pedido,  
       produto.nome as nome_produto  
FROM cliente  
INNER JOIN pedido ON (pedido.fk_cod_cliente = cliente.cod_cliente)  
INNER JOIN pedido_produto ON (pedido.cod_pedido = pedido_produto.fk_cod_pedido)  
INNER JOIN produto ON (produto.cod_produto = pedido_produto.fk_cod_produto)  
ORDER BY cliente.nome
```

Table queries ▾

Previous queries ▾

Execute ▶

nome_cliente	numero_pedido	nome_produto
Camila Volponi	23	Bolo de limão
Camila Volponi	25	Maria mole
Camila Volponi	2	Cento de brigadeiro
Camila Volponi	9	Cookie
Camila Volponi	29	Rocambole de prestigio
Carol Soares	27	Cookie
Cecília Chaves	11	Pipoca doce jumbo
Clara Santos	10	Doce de pote
Clara Santos	10	Bolo de limão
Clara Santos	10	Maria mole

Consulta com Inner Join e Order By

```
SELECT produto.nome as nome_produto,  
       ingrediente.nome as nome_ingrediente,  
       produto_ingrediente.quantidade AS quantidade_ingrediente,  
       tipo_medida.descricao as tipo_medida  
FROM produto  
INNER JOIN produto_ingrediente ON (produto.cod_produto = produto_ingrediente.fk_cod_produto)  
INNER JOIN ingrediente ON (produto_ingrediente.fk_cod_ingrediente = ingrediente.cod_ingrediente)  
INNER JOIN tipo_medida ON (ingrediente.fk_cod_tipo_medida = tipo_medida.cod_tipo_medida)  
ORDER BY produto.nome
```

Table queries ▾

Previous queries ▾

Execute ▶

nome_produto	nome_ingrediente	quantidade_ingrediente	tipo_medida
Bolo de chocolate	Óleo	1	Copo
Bolo de chocolate	Ovo	1	Unidade
Bolo de chocolate	Leite	1	Copo
Bolo de chocolate	Margarina sem sal	1	Colher de sopa
Bolo de chocolate	Achocolatado	1	Colher de sopa
Bolo de chocolate	Achocolatado	3	Colher de sopa
Bolo de chocolate	Açúcar Refinado	1	Colher de sopa
Bolo de chocolate	Ovo	2	Unidade
Bolo de chocolate	Farinha de trigo	2	Colher de sopa
Bolo de chocolate	Fermento em pó	1	Colher de sopa
Bolo de fuba	Ovo	3	Unidade
Bolo de fuba	Leite	1	Copo

Consulta com Group By e Funções de agrupamento

```
SELECT tipo_pagamento.descricao as tipo_pagamento,  
       COUNT(pedido.fk_cod_tipo_pagamento) AS quantidade_utilizada_forma_pagamento  
FROM pedido  
INNER JOIN tipo_pagamento ON (tipo_pagamento.cod_tipo_pagamento = pedido.fk_cod_tipo_pagamento)  
GROUP BY tipo_pagamento.descricao
```

Table queries ▾

Previous queries ▾

Execute ▶

tipo_pagamento	quantidade_utilizada_forma_pagamento
Crédito	10
PIX	7
Débito	4
PicPay	5
Dinheiro	5

Consulta com Group By e Funções de agrupamento

```
SELECT pedido.cod_pedido AS numero_pedido,  
       cliente.nome AS nome_cliente,  
       COUNT(pedido_produto.fk_cod_produto) AS quantidade_produto_no_pedido  
FROM pedido  
INNER JOIN cliente ON (pedido.fk_cod_cliente = cliente.cod_cliente)  
INNER JOIN pedido_produto ON (pedido.cod_pedido = pedido_produto.fk_cod_pedido)  
GROUP BY pedido.cod_pedido, cliente.nome  
ORDER BY pedido.cod_pedido
```

Table queries ▾

Previous queries ▾

Execute ▶

numero_pedido	nome_cliente	quantidade_produto_no_pedido
1	Moises da Silva	2
2	Camila Volponi	1
3	Mariazinha Feirreira	2
4	Vitor da Silva	1
5	Marcelo da Vitória	2
6	Mariah Ferrari	3
7	Lais Barros	2
8	Iris Branco	4

Consulta com Group By e Funções de agrupamento

```
SELECT produto.nome AS nome_produto,  
       COUNT(produto_ingrediente.fk_cod_ingredientes) AS quantidade_ingredientes_no_produto  
FROM produto  
INNER JOIN produto_ingrediente ON (produto.cod_produto = produto_ingrediente.fk_cod_produto)  
GROUP BY produto.nome  
ORDER BY produto.nome
```

Table queries ▾

Previous queries ▾

Execute ▶

nome_produto	quantidade_ingredientes_no_produto
--------------	------------------------------------

Bolo de chocolate	10
-------------------	----

Bolo de fuba	7
--------------	---

Cento de beijinho	5
-------------------	---

Cento de brigadeiro	5
---------------------	---

Cookie	9
--------	---

Maria mole	4
------------	---

Pipoca doce jumbo	5
-------------------	---

Pudim	5
-------	---

Suspiro	2
---------	---

Consulta que utilize Left Join

```
select cliente.nome as nome_cliente,  
       pedido.cod_pedido  
from cliente  
left join pedido on (cliente.cod_cliente = pedido.fk_cod_cliente)  
order by pedido.cod_pedido desc
```

Table queries ▾

Previous queries ▾

Execute ▶

nome_cliente

cod_pedido

Gustavo Ferrari

Leticia Lima

Priscila Dias

Hellen Canejo

31

Mariazinha Feirreira

30

Camila Volponi

29

Mariah Ferrari

28

Carol Soares

27

Marcelo da Vitória

26

Camila Volponi

25

Leila Barros

24

Camila Volponi

23

Consulta que utilize View com subconsulta

```
create view precos_abaixo_media as  
select cod_ped_prod, preco from pedido_produto  
where preco < (select avg(pedido_produto.preco) from pedido_produto)  
order by preco desc;  
  
SELECT * FROM precos_abaixo_media;
```

Execute ▶

cod_ped_prod

preco

39	25
27	25
12	20
13	20
5	20
8	20
20	20
21	20
25	20
31	20
37	20

Perguntas a serem respondidas

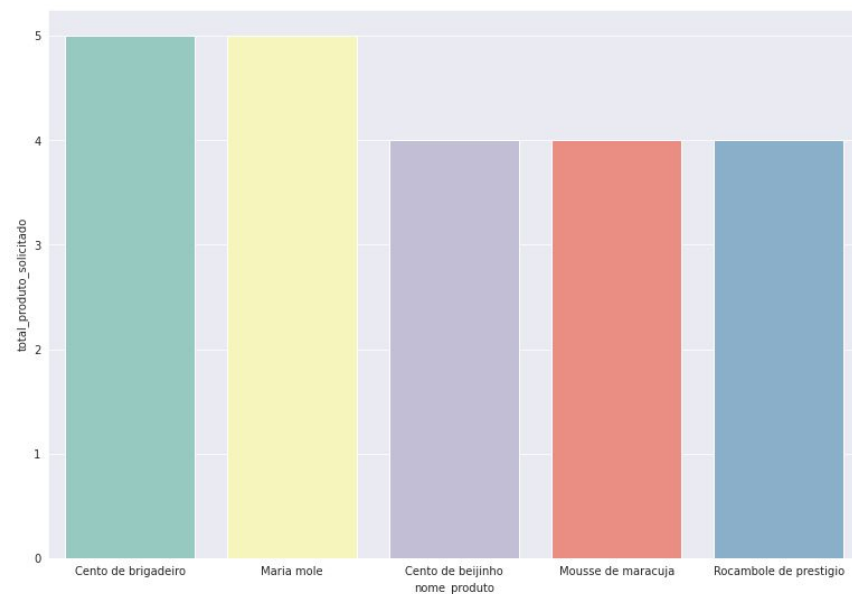
- Qual foi o total das vendas por mês ou outros períodos de tempo se necessário
- Quais produtos são mais pedidos
- Quais dos clientes mais gastam nesta confeitaria
- Quais dos clientes mais fazem pedidos
- Quais produtos tem o maior número de passos de preparo
- Quais pedidos foram feitos por um determinado cliente
- Quais pedidos tem entrega prevista para uma determinada semana (ou outro período de tempo).
- Quais produtos fazem parte de um determinado pedido.
- Quais os ingredientes, tipo de medida e quantidade dos ingredientes de um determinado produto.
- Quais são os passos do preparo de um produto, em ordem.



Relatório 1: Relativo aos 5 produtos mais escolhidos dentre todos os pedidos.

```
res = pd.read_sql_query(""" SELECT produto.nome as nome_produto,
                             COUNT(pedido_produto.fk_cod_produto) as total_produto_solicitado
                             FROM produto
                             INNER JOIN pedido_produto ON (produto.cod_produto = pedido_produto.fk_cod_produto)
                             GROUP BY produto.cod_produto, produto.nome
                             ORDER BY total_produto_solicitado DESC
                             LIMIT 5
                             """,conn)
```

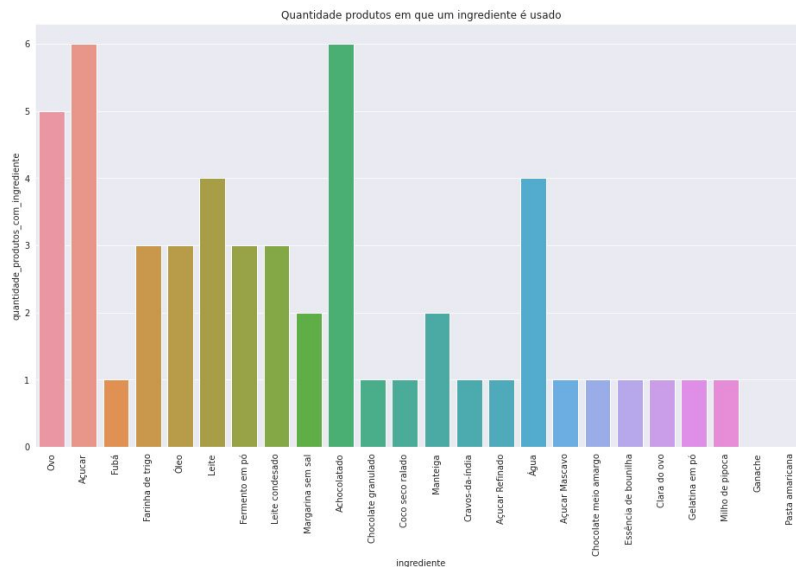
	nome_produto	total_produto_solicitado
0	Cento de brigadeiro	5
1	Maria mole	5
2	Cento de beijinho	4
3	Mousse de maracuja	4
4	Rocambole de prestigio	4



Relatório 2: Relativo à quantidade de produtos em que cada um dos ingredientes é utilizado

	ingrediente	quantidade_produtos_com_ingrediente
0	Ovo	5
1	Açúcar	6
2	Fubá	1
3	Farinha de trigo	3
4	Óleo	3
5	Leite	4
6	Fermento em pó	3
7	Leite condensado	3
8	Margarina sem sal	2
9	Achocolatado	6
10	Chocolate granulado	1
11	Coco seco ralado	1
12	Manteiga	2
13	Cravos-da-índia	1
14	Açúcar Refinado	1
15	Água	4
16	Açúcar Mascavo	1
17	Chocolate meio amargo	1
18	Essência de baunilha	1
19	Clara do ovo	1
20	Gelatina em pó	1
21	Milho de pipoca	1
22	Ganache	0
23	Pasta amarela	0

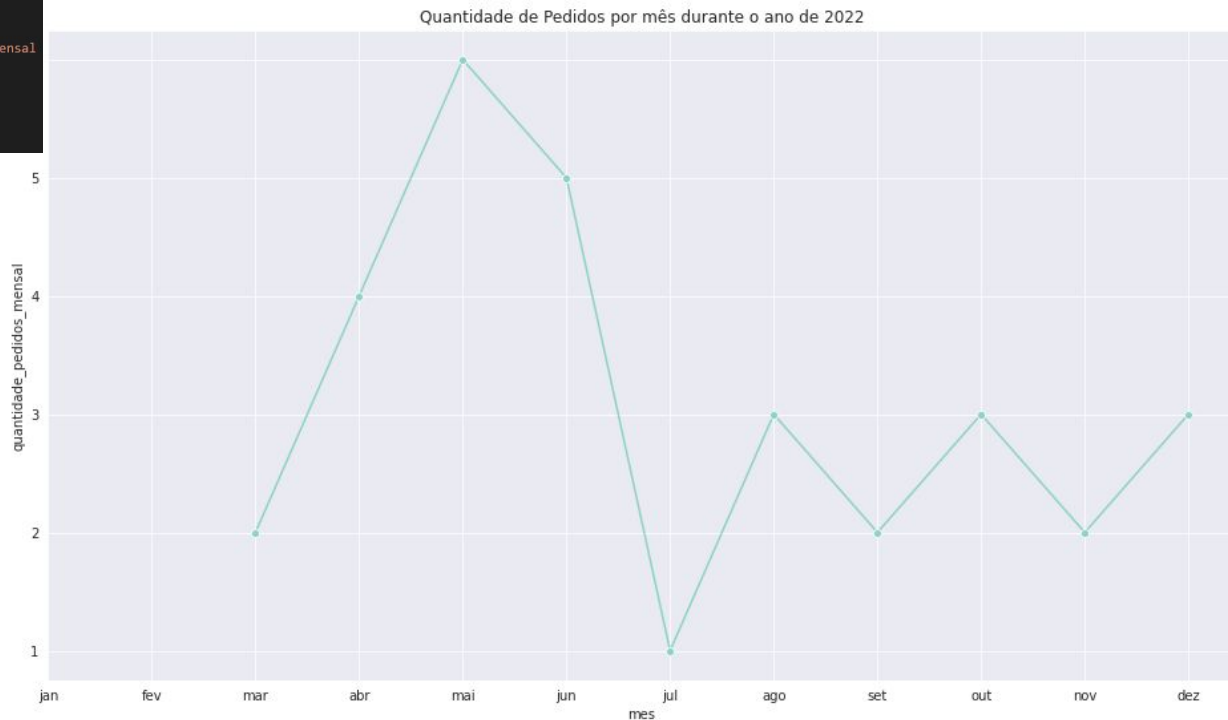
```
res = pd.read_sql_query("""
SELECT ing.nome AS ingrediente, count(prod_ing.fk_cod_ingrediente) AS quantidade_produtos_com_ingrediente
FROM ingrediente as ing
LEFT JOIN produto_ingrediente as prod_ing
ON (prod_ing.fk_cod_ingrediente = ing.cod_ingrediente)
GROUP BY ing.cod_ingrediente
ORDER BY ing.cod_ingrediente """,conn)
res
```



Relatório 3: Relatório que mostra a quantidade de pedidos por mês durante um ano (2022).

```
res = pd.read_sql_query("""
SELECT  EXTRACT('month' FROM p.data_hora_entrega_prevista) AS mes,
COUNT(EXTRACT('month' FROM p.data_hora_entrega_prevista)) AS quantidade_pedidos_mensal
FROM pedido as p
where extract('Year' from p.data_hora_criacao) = 2022
GROUP BY mes
ORDER BY mes""", conn)
res
```

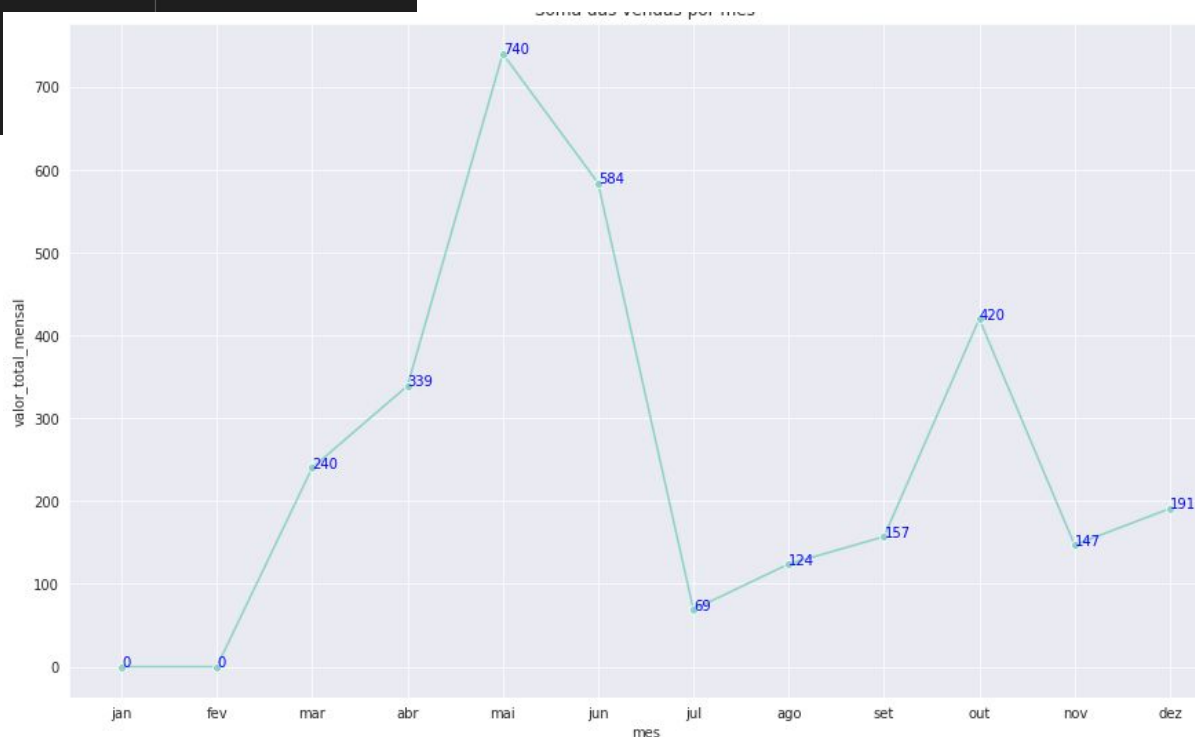
	mes	quantidade_pedidos_mensal
0	3.0	2
1	4.0	4
2	5.0	6
3	6.0	5
4	7.0	1
5	8.0	3
6	9.0	2
7	10.0	3
8	11.0	2
9	12.0	3



Relatório 4: Mostra a soma do valor dos pedidos por mês, durante todo um ano (2022)

```
res = pd.read_sql_query("""
SELECT EXTRACT('month' FROM p.data_hora_entrega_prevista) AS mes, SUM(p.preco_total) AS valor_total_mensal
FROM pedido as p
where extract('Year' from p.data_hora_criacao) = 2022
GROUP BY mes
UNION (VALUES (1,0),(2,0))
ORDER BY mes""", conn)
res
```

	mes	valor_total_mensal
0	1.0	0.0
1	2.0	0.0
2	3.0	240.0
3	4.0	339.0
4	5.0	740.0
5	6.0	584.0
6	7.0	69.0
7	8.0	124.0
8	9.0	157.0
9	10.0	420.0
10	11.0	147.0
11	12.0	191.0



Relatório 5: Obter um relatório que mostre para cada cliente, o total de compras que ele realizou em todo o ano de 2022, organizados do maior comprador para o menor comprador.

```
res = pd.read_sql_query("""
select c.cod_cliente, c.nome , sum(p.preco_total) as total_gastos_cliente
from cliente c inner join pedido p
on c.cod_cliente = p.fk_cod_cliente
where extract('Year' from p.data_hora_criacao) = 2022
group by c.cod_cliente,c.nome
order by total_gastos_cliente desc;""",conn)

res[["total_gastos_cliente"]].fillna(0,inplace=True)
res
```

	cod_cliente	nome	total_gastos_cliente
0	2	Camila Volponi	510.0
1	9	Mariah Ferrari	468.0
2	16	Hellen Canejo	290.0
3	5	Marcelo da Vitória	237.0
4	3	Mariazinha Feirreira	190.0
5	17	Iris Branco	188.0
6	18	Cecilia Chaves	170.0
7	7	Higor Silva	169.0
8	14	Lais Barros	140.0
9	19	Romeu Resende	121.0
10	11	Otávio Almeida	100.0
11	1	Moises da Silva	100.0
12	4	Vitor da Silva	100.0
13	6	Clara Santos	93.0
14	13	Leila Barros	64.0
15	20	Michele Meira	47.0
16	10	Carol Soares	24.0

