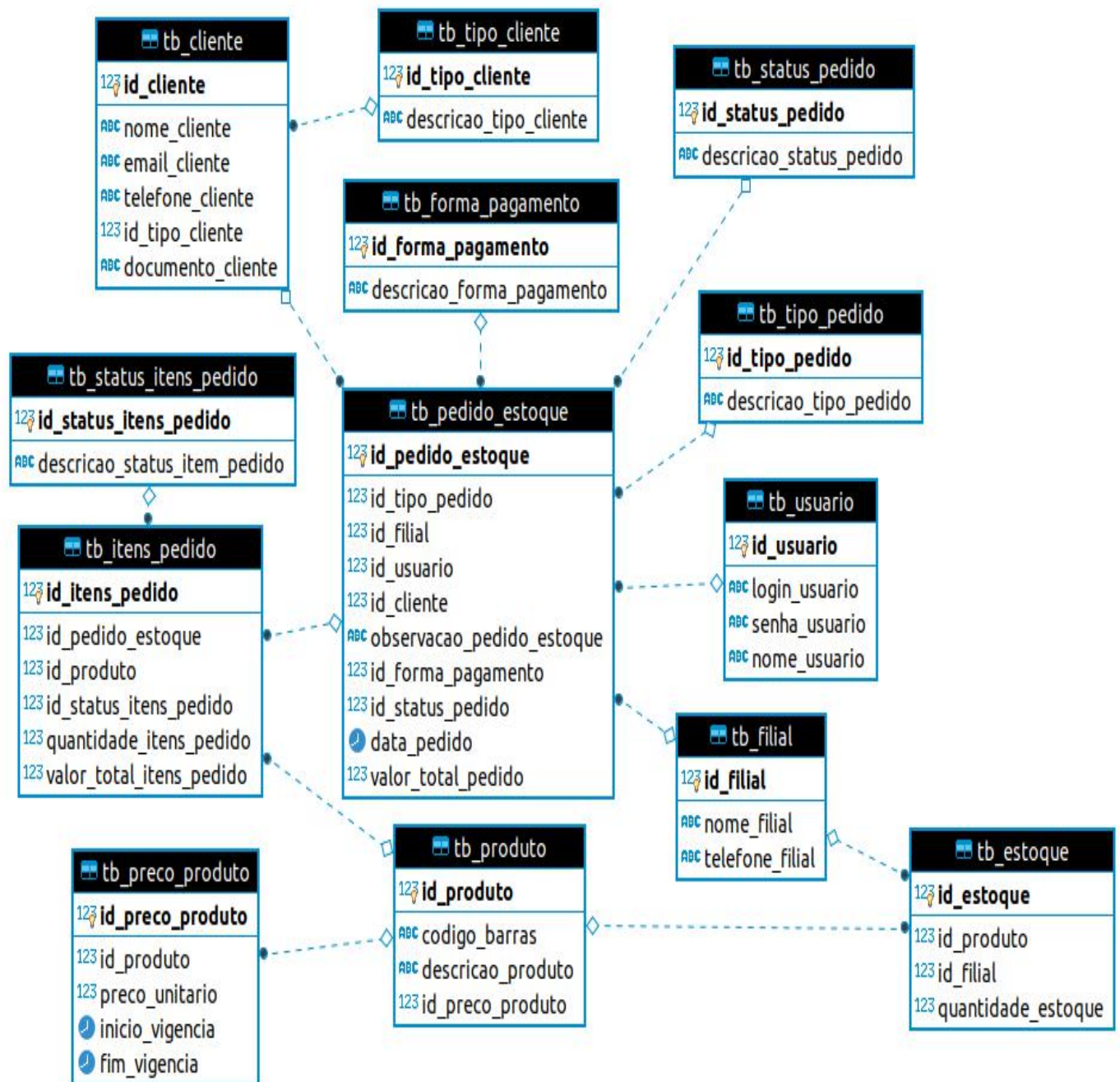


Documentação Final do Projeto

Diagrama de Entidade-Relacionamento:



Tabelas de Constantes

TB_TIPO_PESSOA

	123 id_tipo_cliente T	ABC descricao_tipo_cliente T
1	1	PESSOA FÍSICA
2	2	PESSOA JURÍDICA

TB_TIPO_PEDIDO

	123 id_tipo_pedido T	ABC descricao_tipo_pedido T
1	1	ENTRADA
2	2	SAÍDA

TB_STATUS_PEDIDO

	123 id_status_pedido T	ABC descricao_status_pedido T
1	1	ABERTO
2	2	PROCESSANDO PAGAMENTO
3	3	PAGO
4	4	PROCESSANDO ENTREGA
5	5	ENTREGUE
6	6	CANCELADO

TB_STATUS_ITEM_PEDIDO

	123 id_status_itens_pedido T	ABC descricao_status_item_pedido T
1	1	ATIVO
2	2	CANCELADO
3	3	PROCESSADO

TB_FORMA_PAGAMENTO

	123 id_forma_pagamento	ABC descricao_forma_pagamento
1	1	À VISTA
2	2	BOLETO
3	3	CARTÃO

SQLs das Consultas

1) Escrever uma consulta que retorne todos os produtos com quantidade maior ou igual a 100:

--Vinculando consulta às filiais.

```
select
    tf.nome_filial,
    tp.descricao_produto
from
    tb_estoque te
    inner join tb_filial tf on (te.id_filial = tf.id_filial)
    inner join tb_produto tp on (te.id_produto = tp.id_produto)
where
    te.quantidade_estoque >= 100
```

--Vinculando consulta independente das filiais.

```
select
    tp.descricao_produto
from
    tb_estoque te
    inner join tb_produto tp on (te.id_produto = tp.id_produto)
where
    te.quantidade_estoque >= 100
group by
    tp.descricao_produto
```

2) Escrever uma consulta que traga todos os produtos que têm estoque para a filial de código 60 (Adaptando para código filial 1):

```
select
    tp.descricao_produto
from
    tb_estoque te
        inner join tb_filial tf on (te.id_filial = tf.id_filial)
        inner join tb_produto tp on (te.id_produto = tp.id_produto)
where
    tf.id_filial = 1-- ou 60
```

3) Escrever consulta que liste todos os campos para o domínio PedidoEstoque e ItensPedido filtrando apenas o produto de código 7993 (Adaptando para código produto 2)

```
select
    tpe.*,
    tip.*
from
    tb_pedido_estoque tpe
        inner join tb_itens_pedido tip on (tpe.id_pedido_estoque =
tip.id_pedido_estoque )
        inner join tb_produto tp on (tip.id_produto = tp.id_produto)
where
    tp.id_produto = 2-- ou 7993
```

4) Escrever uma consulta que liste os pedidos com suas respectivas formas de pagamento:

```
select
    tpe.*,
    tfp.*
from
    tb_pedido_estoque tpe
        inner join tb_forma_pagamento tfp on (tpe.id_forma_pagamento = tfp.id_forma_pagamento)
```

5) Escrever uma consulta para sumarizar e bater os valores da capa **do** pedido com os valores dos ítems de pedido:

select

```
tpe.id_pedido_estoque,  
tpe.valor_total_pedido,  
sum(tip.valor_total_itens_pedido) as valor_total_itens_pedido,  
sum(tip.quantidade_itens_pedido * tpp.preco_unitario) as valor_calculado_por_preco_produto,  
case  
    when tpe.valor_total_pedido = sum(tip.valor_total_itens_pedido) and  
         tpe.valor_total_pedido = sum(tip.quantidade_itens_pedido * tpp.preco_unitario)  
    then  
        'SEM DIVERGÊNCIAS'  
    else  
        'VALORES DIVERGENTES'  
end as batimento
```

from

```
tb_pedido_estoque tpe  
    inner join tb_itens_pedido tip on (tpe.id_pedido_estoque = tip.id_pedido_estoque)  
    inner join tb_produto tp on (tip.id_produto = tp.id_produto)  
    inner join tb_preco_produto tpp on (tp.id_produto = tpp.id_produto and  
        (  
            (tpe.data_pedido >= tpp.inicio_vigencia and tpe.data_pedido < tpp.fim_vigencia) or  
            (tpe.data_pedido >= tpp.inicio_vigencia and tpp.fim_vigencia is null)  
        )  
    )
```

group by

```
tpe.id_pedido_estoque,  
tpe.valor_total_pedido
```

6) Escrever uma consulta para sumarizar o total dos itens por pedido e que filtre apenas os pedidos **no** qual a soma total da quantidade de ítems de pedido seja maior que 10

select

```
tpe.id_pedido_estoque,  
sum(tip.quantidade_itens_pedido) as total_itens_pedido
```

from

```
tb_pedido_estoque tpe  
    inner join tb_itens_pedido tip on (tpe.id_pedido_estoque = tip.id_pedido_estoque)
```

group by

```
tpe.id_pedido_estoque
```

having

sum(tip.quantidade_itens_pedido) > 10