

CREATE DATABASE Loja -----CRIANDO BANCO-----

USE Loja

-----CRIANDO AS TABELAS-----

CREATE TABLE Cliente(

IdCliente INT PRIMARY KEY IDENTITY (1,1),

Nome VARCHAR(60) NOT NULL,

CpfCnpj BIGINT NOT NULL UNIQUE ,

)

GO

CREATE TABLE Email(

IdEmail INT PRIMARY KEY IDENTITY (1,1),

Email VARCHAR(254) NOT NULL UNIQUE,

AceitaPropaganda BIT,

IdCliente INT CONSTRAINT FK_IdCliente FOREIGN KEY (IdCliente)

REFERENCES Cliente (IdCliente)

)

GO

CREATE TABLE TipoTelefone (

IdTipoTelefone TINYINT PRIMARY KEY IDENTITY(1,1),

Tipo VARCHAR(11)

)

GO

CREATE TABLE Telefone(

IdTelefone INT PRIMARY KEY IDENTITY (1,1),

Numero BIGINT UNIQUE,

IdCliente INT CONSTRAINT FK_IdCliente_IdCliente FOREIGN KEY (IdCliente)

REFERENCES Cliente(IdCliente),

IdTipoTelefone TINYINT CONSTRAINT FK_TipoTelefone_IdTipoTelefone FOREIGN KEY (IdTipoTelefone)

REFERENCES TipoTelefone(IdTipoTelefone)

)GO

CREATE TABLE Produto(

IdProduto INT PRIMARY KEY IDENTITY (1,1),

Nome VARCHAR(50) UNIQUE NOT NULL,

Estoque int NOT NULL ,

Preco MONEY NOT NULL

```

)
GO

CREATE TABLE Venda(

    IdVenda INT PRIMARY KEY IDENTITY (1,1),

    DataVenda Smalldatetime NOT NULL,

    IdCliente INT CONSTRAINT FK_IdCliente_IdClienteVenda FOREIGN KEY (IdCliente)

REFERENCES Cliente(IdCliente)

)

```

```

GO

CREATE TABLE ItemVenda(

    IdItemVenda          INT PRIMARY KEY IDENTITY(1,1),

    IdProduto INT CONSTRAINT FK_IdProduto_IdProduto FOREIGN KEY (IdProduto)

REFERENCES Produto (IdProduto),

    IdVenda INT CONSTRAINT FK_IdVenda_IdVenda FOREIGN KEY (IdVenda)

REFERENCES Venda(IdVenda)

)

```

-----POPULANDO AS TABELAS-----

```

insert into Cliente Values('Bruna Cristina',51610987080)

SELECT * FROM Cliente

```

```

INSERT INTO Produto VALUES('Coberta',100,'49')

SELECT * FROM Produto

```

```

INSERT INTO TipoTelefone VALUES ('Comercial')

SELECT * FROM TipoTelefone

INSERT INTO Email VALUES ('junior@gmail.com',0,22)

SELECT * FROM Email

```

```

INSERT INTO Telefone VALUES (37027040,13,2)

SELECT * FROM Telefone

```

```

INSERT INTO Venda VALUES (1,'2000-02-10')

SELECT *FROM Venda

```

```

INSERT INTO ItemVenda VALUES(3,16,8)

SELECT *FROM ItemVenda

```

-----EXERCICIO 1-----

--1. Ranking de produtos mais vendidos por período.

--a. Filtros : Data inicial(obrigatório) e data final(obrigatório);

--b. Retorno : Lista contendo: nome do produto, quantidade de vendas, valor total das Vendas;

--c. Observações : deverá ser possível ordenar os resultados por quantidade de vendas;

```
SELECT RESULTADO.Produto,
      SUM(RESULTADO.Quantidade) AS Quantidade,
      SUM(RESULTADO.Total) AS Total
FROM (
      SELECT P.Nome AS Produto,
            ITV.Quantidade,
            (P.Preco * ITV.Quantidade) AS Total
      FROM ItemVenda AS ITV WITH(NOLOCK)
      INNER JOIN Produto AS P WITH(NOLOCK)
            ON ITV.IdProduto = P.IdProduto
      INNER JOIN Venda AS V WITH(NOLOCK)
            ON V.IdVenda= ITV.IdVenda
      WHERE V.DataVenda BETWEEN '2019-12-04' AND '2019-12-09'
    ) AS RESULTADO
GROUP BY RESULTADO.Quantidade, RESULTADO.Produto
ORDER BY RESULTADO.Quantidade DESC
```

-----EXERCICIO2-----

--2. Lista de produtos vendidos por período.

--a. Filtros : Data inicial(obrigatório) e data final(obrigatório);

--b. Retorno : Lista contendo: nome do produto, data da venda, valor do produto;

--c. Observações : deverá ser possível ordenar os resultados por nome do produto e data

--da venda. Não deverão aparecer produtos sem venda.

```
DECLARE @DataInicial      date = '2000-12-29',
        @DataFinal        date = '2020-12-30'
```

```
SELECT P.Nome AS Produto,
      V.DataVenda,
      P.Preco AS Preço_Produto
```

```

FROM Produto AS P
    INNER JOIN ItemVenda as ITV WITH(NOLOCK)
        ON ITV.IdProduto = P.IdProduto
    INNER JOIN Venda AS V WITH(NOLOCK)
        ON V.IdVenda=ITV.IdVenda
    WHERE V.DataVenda BETWEEN @DataInicial AND @DataFinal
ORDER BY P.Nome,V.DataVenda DESC

```

-----EXERCICIO3-----

- 3. Ranking de vendas por mês.
- a. Filtros : ano (não obrigatório).
- b. Retorno : Lista contendo: nome do mês, quantidade de vendas, valor total de vendas
- c. Observações : deverá ser possível ordenar os resultados por nome do produto e data
- da venda. Nos meses sem vendas de anos que possuem vendas, os valores deverão vir zerados.

```

DECLARE @Ano SMALLINT = NULL;

```

```

WITH mes (id,nome)

```

```

AS (

```

```

    SELECT 1, 'Janeiro'

```

```

    UNION

```

```

    SELECT 2, 'Fevereiro'

```

```

    UNION

```

```

    SELECT 3, 'Março'

```

```

    UNION

```

```

    SELECT 4, 'Abril'

```

```

    UNION

```

```

    SELECT 5, 'Maio'

```

```

    UNION

```

```

    SELECT 6, 'Junho'

```

```

    UNION

```

```

    SELECT 7, 'Julho'

```

```

    UNION

```

```

    SELECT 8, 'Agosto'

```

```

    UNION

```

```

        SELECT 9, 'Setembro'

        UNION

        SELECT 10, 'Outubro'

        UNION

        SELECT 11, 'Novembro'

        UNION

        SELECT 12, 'Dezembro'

    ),

    Dados AS

    (

        SELECT P.Nome AS Produto,

                MONTH(V.DataVenda) AS Mes,

        YEAR(V.DataVenda) AS Ano,

                ITV.Quantidade,

                (P.Preco * ITV.Quantidade) AS Total

        FROM ItemVenda AS ITV WITH(NOLOCK)

                INNER JOIN Produto AS P WITH(NOLOCK)

                        ON ITV.IdProduto = P.IdProduto

                INNER JOIN Venda AS V WITH(NOLOCK)

                        ON V.IdVenda= ITV.IdVenda

        WHERE (@Ano IS NULL OR month(V.DataVenda) = @Ano)

    ),

    AnoVenda (ano) AS

    (

        SELECT DISTINCT

                YEAR(V.DataVenda) AS ano

        FROM Venda AS V WITH(NOLOCK)

    ),

    AnoMes (ano, mes) AS

    (

        SELECT ano, null

                from AnoVenda

        UNION ALL

        SELECT av.ano,

                m.id

```

```

FROM mes m
        INNER JOIN AnoVenda av
                ON 1 = 1
)
select am.ano,
        am.mes AS numeroMes,
        m.nome AS nomeMes,
        ISNULL(SUM(d.Quantidade), 0) AS Quantidade,
        ISNULL(SUM(d.Total), 0) as ValorTotal
from AnoMes am
        INNER JOIN mes m
                ON m.id = am.mes
        LEFT JOIN Dados d
                ON am.ano = d.ano
                AND am.mes = d.mes
GROUP BY am.ano, am.mes, m.nome
ORDER BY am.ano, am.mes

```

-----EXERCICIO4-----

- 4. Estoque inicial dos produtos.
- a. Filtros : ano (não obrigatório).
- b. Retorno : Lista contendo: nome do produto, estoque inicial.
- c. Observações : Considerando o estoque atual dos produtos, e as vendas efetuadas, a
- API deve retornar o estoque inicial dos produtos.

```

SELECT
        P.Nome,
        P.Estoque AS EstoqueAtual,
        sum(ITV.Quantidade) as QuantidadesVendidas ,
        (P.Estoque + ISNULL(SUM(ITV.Quantidade), 0)) AS EstoqueInicial

FROM Produto AS P WITH(NOLOCK)
        LEFT JOIN ItemVenda AS ITV WITH(NOLOCK)
                ON ITV.IdProduto = P.IdProduto

GROUP BY P.Nome,P.Estoque

```

EXERCICIO5

--5. Ranking de clientes.

--a. Filtros : CNPJ do cliente (não obrigatório).

--b. Retorno : Lista contendo: nome do cliente, produtos que o cliente comprou [nome

--do produto, valor do produto], quantidade total de produtos comprados, valor total

--de produtos comprados.

--ordenado por quem mais gastou.

SELECT

C.Nome AS Cliente,

STRING_AGG(p.Nome,',') AS Produto ,

STRING_AGG(ITV.Quantidade,',') AS Quantidade,

SUM(ITV.Quantidade) AS TotalItem,

STRING_AGG(p.Preco,',') AS PreçoUnitario,

SUM(ITV.Quantidade*p.Preco) AS Total

FROM

ItemVenda AS ITV WITH(NOLOCK)

INNER JOIN Venda AS V WITH(NOLOCK)

ON ITV.IdVenda =V.IdVenda

LEFT JOIN Cliente AS C WITH(NOLOCK)

ON C.IdCliente =V.IdCliente

INNER JOIN Produto AS P WITH(NOLOCK)

ON ITV.IdProduto = P.IdProduto

GROUP BY C.Nome

ORDER BY Total DESC

EXERCICIO 5-Parte 2

--um result set, retorna nome do cliente, total de compras, total de itens, valor total das compras, valor da maior compra

--e no outro resultset

--retorna a lista de compras dele

; WITH CTE_EXER2 AS (

```

SELECT C.IdCliente ,
       V.IdVenda,
       0 AS QUANTIDADE_ITEM,
       (SELECT SUM(P.Preco) * SUM(iv.Quantidade)
        FROM ItemVenda iv
        INNER JOIN Produto P
        ON P.IdProduto = iv.IdProduto
        WHERE iv.IdVenda = v.IdVenda
       ) AS VALOR_TOTAL_COMPRA
FROM Cliente AS C
      INNER JOIN Venda AS V
      ON V.IdCliente = C.IdCliente

```

UNION ALL

```

SELECT T.IdCliente,
       NULL AS IdVenda,
       itv.Quantidade AS QUANTIDADE_ITEM,
       CAST(0 AS money) AS VALOR_TOTAL_COMPRA
FROM CTE_EXER2 AS T
      INNER JOIN ItemVenda AS ITV
      ON ITV.IdVenda = T.IdVenda

```

)

```

SELECT C.Nome,
       COUNT(T.IdVenda) AS QUANTIDADE_VENDA,
       SUM(T.QUANTIDADE_ITEM) AS Qtd_TotalItens,
       SUM(T.VALOR_TOTAL_COMPRA) AS Vlr_TotalItens,
       MAX(T.VALOR_TOTAL_COMPRA) AS Vlr_MaiorCompra
FROM CTE_EXER2 AS T
      INNER JOIN Cliente AS C
      ON C.IdCliente = T.IdCliente
GROUP BY C.Nome

```