

# EEL871 Banco de Dados - Trabalho Prático

Bruno Dantas, Carlos Henrique Ferreira Brito Filho, Rafael Martins, Thiago Guimarães

29 de maio de 2021

## Questão 5

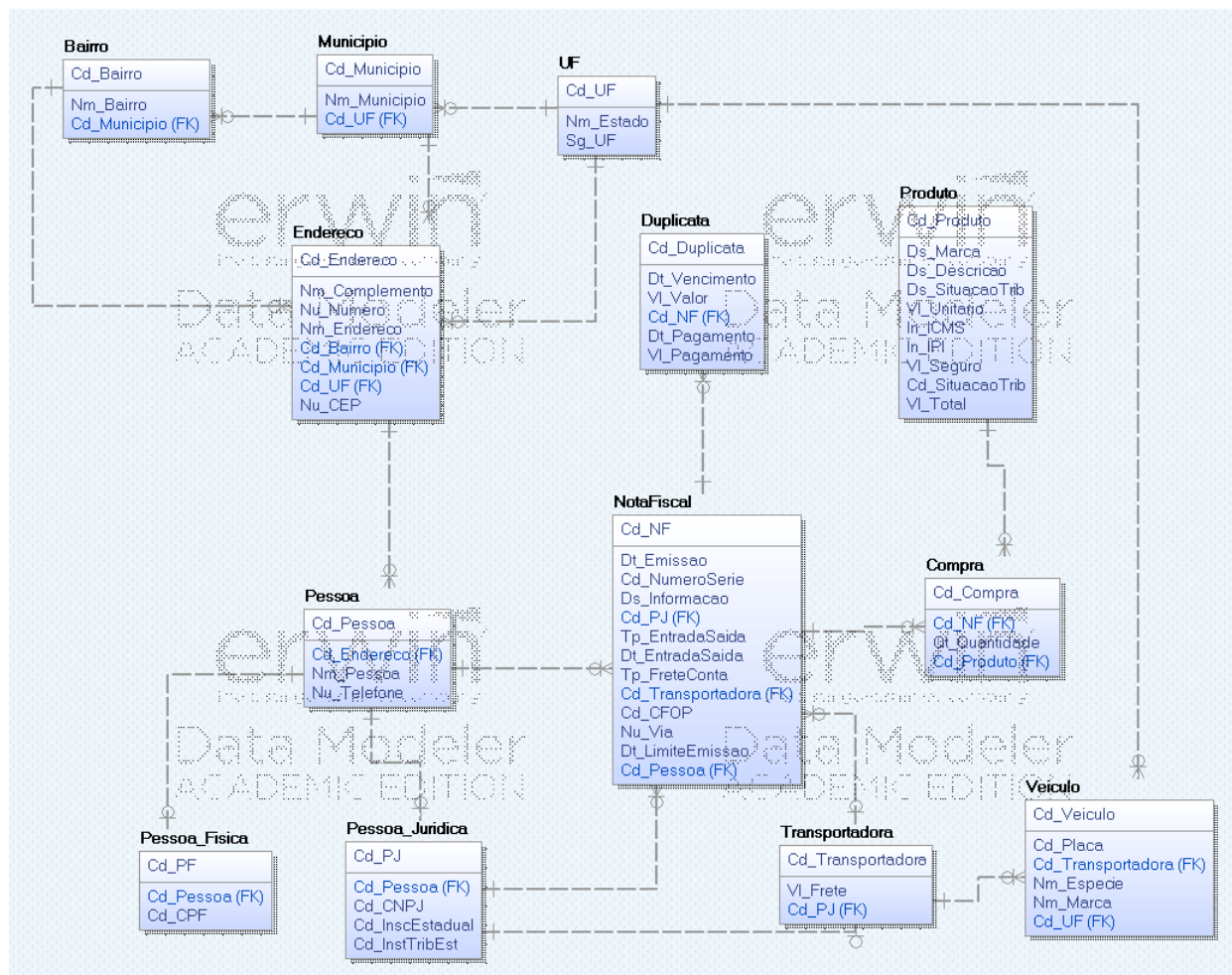


Figura 1: Modelo de Nota Fiscal ERWin

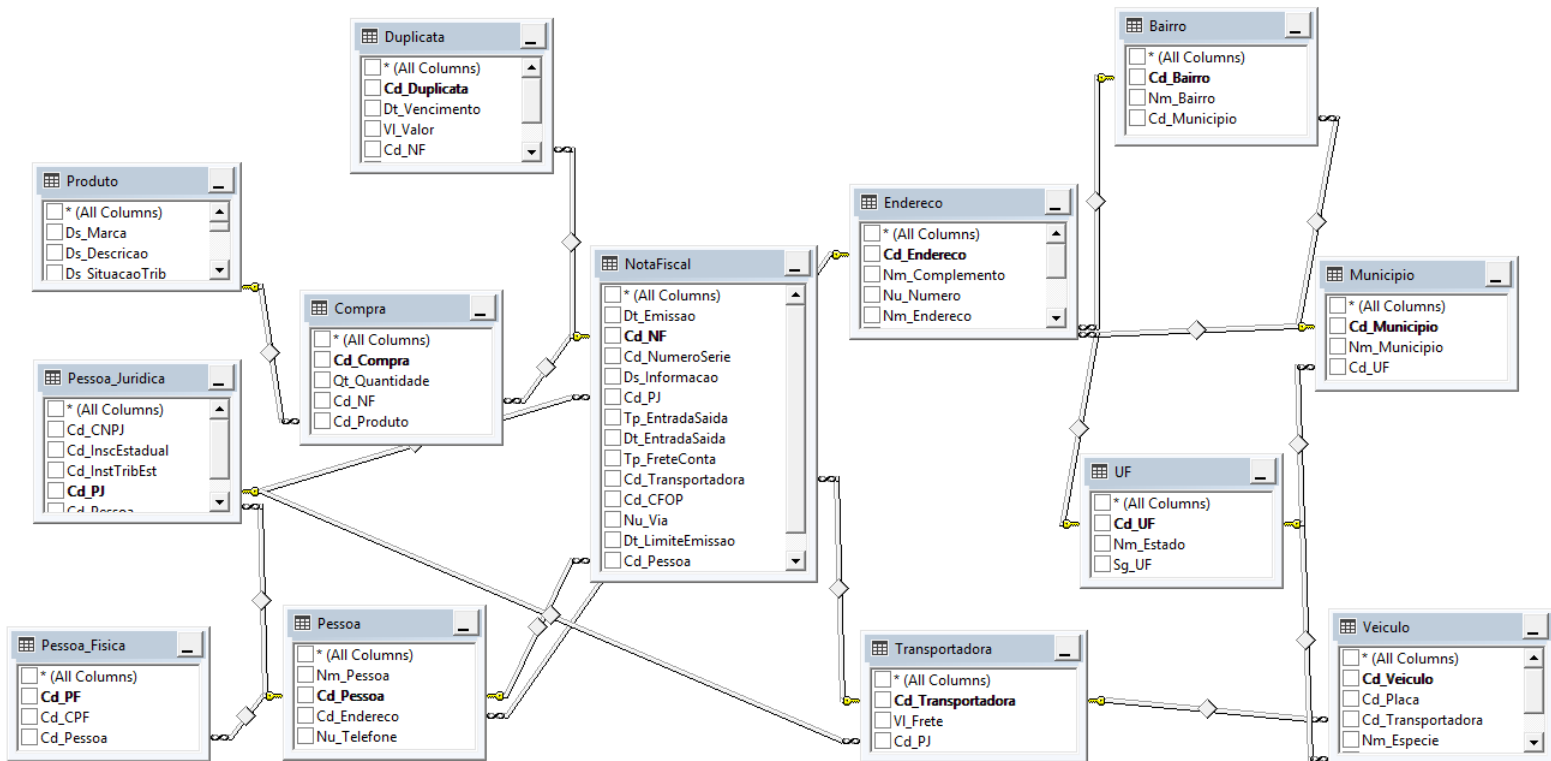


Figura 2: View do Modelo de Nota Fiscal SQL Server

Para o código dessa questão, optamos por não colocar no relatório devido ao seu tamanho, então clique aqui para ver esse código: Github.

## Questão 6

Tal como a questão anterior, devido ao tamanho do código, clique aqui para vê-lo em outra página: Github.

## Questão 7

```
CREATE LOGIN palma
WITH PASSWORD = 'senha';
GO
```

```
CREATE USER palma
FOR LOGIN palma;
GO
```

```
GRANT SELECT
ON dbo.uf TO palma;
```

## Questão 8

```
INSERT INTO [Compra] (Qt_Quantidade, Cd_NF, Cd_Produto)
VALUES (2,1,2), (2,1,3), (2,1,4), (2,1,5)
```

```
INSERT INTO [Compra] (Qt_Quantidade, Cd_NF, Cd_Produto)
VALUES (2,2,1), (2,2,3), (2,2,4), (2,2,5)
```

```
SELECT
    A.Nm_Pessoa
FROM
    (
        SELECT
            Compra.Cd_Compra,
            NotaFiscal.Cd_NF,
            Produto.Cd_Produto,
            Pessoa.Cd_Pessoa,
            Nm_Pessoa
        FROM [Compra] Compra
        INNER JOIN PRODUTO Produto
            ON Compra.Cd_Produto = Produto.Cd_Produto
        INNER JOIN [NotaFiscal] NotaFiscal
            ON Compra.Cd_NF = NotaFiscal.Cd_NF
        INNER JOIN [Pessoa] Pessoa
            ON Pessoa.Cd_Pessoa = NotaFiscal.Cd_Pessoa
    ) A
JOIN [Produto] Produto
    ON Produto.Cd_Produto = A.Cd_Produto
GROUP BY A.Cd_Pessoa, A.Nm_Pessoa
HAVING COUNT(DISTINCT A.Cd_Produto) >=
    (
        SELECT COUNT(DISTINCT Cd_Produto)
        FROM Produto
    )
```

## Questão 9

```
BEGIN TRANSACTION insercaoVeiculo
INSERT INTO [dbo].[ Veiculo]
(
    [Cd_Placa],
    [Cd_Transportadora],
    [Nm_Especie],
    [Nm_Marca],
    [Cd_UF]
)
VALUES
(
    N'LAB2048',
    1,
    N'Conjugado',
    N'Volkswagen',
    1
)
IF (
    SELECT COUNT(Nm_Especie)
    FROM Veiculo
    WHERE Nm_Especie=N'Conjugado'
) > 1
BEGIN
    ROLLBACK TRANSACTION
    PRINT 'Apenas um veiculo conjugado permitido.'
END
ELSE
BEGIN
    PRINT 'Inclusao bem sucedida'
    COMMIT TRANSACTION insercaoVeiculo
END
```

## Questão 11

```
INSERT INTO NotaFiscal
(
    Dt_Emissao ,
    Cd_NumeroSerie ,
    Ds_Informacao ,
    Cd_PJ ,
    Tp_EntradaSaida ,
    Dt_EntradaSaida ,
    Tp_FreteConta ,
    Cd_Transportadora ,
    Cd_CFOP ,
    Nu_Via ,
    Dt_LimiteEmissao
)
VALUES
(
    CAST( '2021-03-02T00:00:00.000 ' AS DateTime) ,
    321 ,
    'Xpto ' ,
    1 ,
    0 ,
    CAST( '2021-03-04T00:00:00.000 ' AS DateTime) ,
    0 ,
    3 ,
    '3232 ' ,
    23444 ,
    CAST( '2021-03-06T00:00:00.000 ' AS DateTime)
)
GO
```

## Questão 12

```
INSERT INTO Duplicata
(
    Dt_Vencimento , Vl_Valor ,
    Cd_NF, Dt_Pagamento ,
    Vl_Pagamento
)
VALUES
(
    CAST( '2022-01-01 ' AS DATETIME) , 40 ,
    3 , CAST( '2021-09-09 ' AS DATETIME) ,
    40
)
```

```

INSERT INTO Duplicata
(
    Dt_Vencimento , Vl_Valor ,
    Cd_NF, Dt_Pagamento ,
    Vl_Pagamento
)
VALUES
(
    CAST( '2021-01-01 ' AS DATETIME) , 50 ,
    2, NULL,
    NULL
)

INSERT INTO Duplicata
(
    Dt_Vencimento , Vl_Valor ,
    Cd_NF, Dt_Pagamento ,
    Vl_Pagamento
)
VALUES
(
    CAST( '2022-01-01 ' AS DATETIME) , 40 ,
    1, NULL,
    NULL
)

SELECT *
FROM Duplicata

CREATE VIEW Cliente_Com_Duplicata_Vencida AS
(
    SELECT DISTINCT
        Pessoa.Cd_Pessoa ,
        Pessoa.Nm_Pessoa
    FROM Pessoa
    LEFT JOIN NotaFiscal
        ON Pessoa.Cd_Pessoa = NotaFiscal.Cd_Pessoa
    INNER JOIN
    (
        SELECT Cd_NF
        FROM Duplicata
        WHERE
            Duplicata.Dt_Vencimento <= SYSUTCDATETIME()
            AND Duplicata.Dt_Pagamento IS NULL
    ) AS Duplicata_Vencida
        ON NotaFiscal.Cd_NF = Duplicata_Vencida.Cd_NF
)

```

## Questão 13

```
CREATE PROCEDURE topThree AS
    SELECT
        TOP 3 sum(Qt_Quantidade) AS QuantidadeDeVendas ,
        Cd_Produto
    FROM Compra
    GROUP BY Cd_Produto
    ORDER BY QuantidadeDeVendas DESC

EXEC topThree
```

## Questão 14

```
CREATE TRIGGER change_payment
    ON Duplicata AFTER UPDATE AS IF UPDATE (Vl_Pagamento)
BEGIN
    UPDATE
        Duplicata
    SET Duplicata.Dt_Pagamento = GETDATE()
    FROM
        Duplicata ,
        Inserted
    WHERE
        Duplicata.Cd_Duplicata = Inserted.Cd_Duplicata
END

SELECT
    Dt_Pagamento
FROM
    Duplicata
WHERE Cd_Duplicata = 1

UPDATE Duplicata
    SET Vl_Pagamento = 9
    WHERE Cd_Duplicata = 1

SELECT
    Dt_Pagamento
FROM
    Duplicata
WHERE Cd_Duplicata = 1
```

## Questão 15

```
SELECT *  
FROM Duplicata
```

```
INSERT INTO Duplicata
```

```
(  
    Dt_Vencimento ,  
    Vl_Valor ,  
    Cd_NF ,  
    Dt_Pagamento ,  
    Vl_Pagamento  
)  
VALUES  
(  
    CAST(DATEADD(day, 15, GETDATE())) AS DATETIME) ,  
    40 ,  
    1 ,  
    NULL ,  
    NULL  
) ,  
(  
    CAST(DATEADD(day, -15, GETDATE())) AS DATETIME) ,  
    50 ,  
    2 ,  
    NULL ,  
    NULL  
)
```

```
SELECT *  
FROM Duplicata
```

```
SELECT *  
FROM Duplicata  
WHERE
```

```
    Duplicata.Dt_Vencimento >= CAST(DATEADD(day, 30, GETDATE())) AS DATETIME)
```

```
UNION
```

```
SELECT *  
FROM Duplicata  
WHERE
```

```
    Duplicata.Dt_Vencimento <= CAST(DATEADD(day, -30, GETDATE())) AS DATETIME)
```

## Questão 16

```
SELECT *  
FROM NotaFiscal nf  
FULL OUTER JOIN Pessoa  
    ON nf.Cd_Pessoa = Pessoa.Cd_Pessoa
```



## 1 Questão 17

```
ALTER TABLE
    dbo.Produto
ADD CONSTRAINT
    Vl_Max_Prod CHECK (Vl_Unitario < 1001);

UPDATE
    dbo.Produto
SET
    Vl_Unitario = 1001
WHERE
    Cd_Produto = 4;
```

## Questão 18

```
DECLARE
    @cd_produto int ,
    @ds_marca VARCHAR(MAX) ,
    @ds_descricao VARCHAR(MAX) ,
    @ds_situacao_trib VARCHAR(MAX) ,
    @vl_unitario int ,
    @in_icms int ,
    @in_ipi int ,
    @vl_seguro int ,
    @cd_situacao_trib int ,
    @vl_total int ;

DECLARE product_cursor CURSOR
FOR SELECT
    Cd_Produto ,
    Ds_Marca ,
    Ds_Descricao ,
    Ds_SituacaoTrib ,
    Vl_Unitario ,
    In_ICMS ,
    In_IPI ,
    Vl_Seguro ,
    Cd_SituacaoTrib ,
    Vl_Total
FROM
    Produto;
```

```

OPEN product_cursor;

FETCH NEXT FROM product_cursor INTO
    @cd_produto,
    @ds_marca,
    @ds_descricao,
    @ds_situacaoatrib,
    @vl_unitario,
    @in_icms,
    @in_ipi,
    @vl_seguro,
    @cd_situacaoatrib,
    @vl_total;

WHILE @@FETCHSTATUS = 0
    BEGIN
        PRINT
            CAST(@cd_produto AS VARCHAR) + ',' +
            @ds_marca + ',' +
            @ds_descricao + ',' +
            @ds_situacaoatrib + ',' +
            CAST(@vl_unitario AS VARCHAR) + ',' +
            CAST(@in_icms AS VARCHAR) + ',' +
            CAST(@in_ipi AS VARCHAR) + ',' +
            CAST(@vl_seguro AS VARCHAR) + ',' +
            CAST(@cd_situacaoatrib AS VARCHAR) + ',' +
            CAST(@vl_total AS VARCHAR);

        FETCH NEXT FROM product_cursor INTO
            @cd_produto,
            @ds_marca,
            @ds_descricao,
            @ds_situacaoatrib,
            @vl_unitario,
            @in_icms,
            @in_ipi,
            @vl_seguro,
            @cd_situacaoatrib,
            @vl_total;

    END;

CLOSE product_cursor;

DEALLOCATE product_cursor;

```

## Questão 19

```
ALTER TABLE NotaFiscal  
ADD XML_InfComp XML;
```

```
DECLARE @myXML XML;  
SET @myXML = '  
<NotaFiscal>  
.....<AgumonFlamenguista>True</AgumonFlamenguista>  
.....<DeVerdade>Nao</DeVerdade>  
.....<MaisInformacoes>NAO_HA MAIS INFORMACOES</MaisInformacoes>  
</NotaFiscal>  
';  
INSERT INTO NotaFiscal  
(  
    Dt_Emissao ,  
    Cd_NumeroSerie ,  
    Ds_Informacao ,  
    Cd_PJ ,  
    Cd_Pessoa ,  
    Tp_EntradaSaida ,  
    Dt_EntradaSaida ,  
    Tp_FreteConta ,  
    Cd_Transportadora ,  
    Cd_CFOP ,  
    Nu_Via ,  
    Dt_LimiteEmissao ,  
    XML_InfComp  
)  
values  
(  
    CAST( '2020-01-23 ' AS DATE) ,  
    717420 ,  
    'OXML' ,  
    1 ,  
    3 ,  
    1 ,  
    CAST( '2020-01-23 ' AS DATE) ,  
    1 ,  
    1 ,  
    1100 ,  
    1 ,  
    CAST( '2020-01-25 ' AS DATE) ,  
    @myXML  
)
```