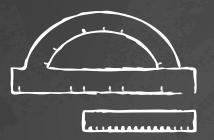
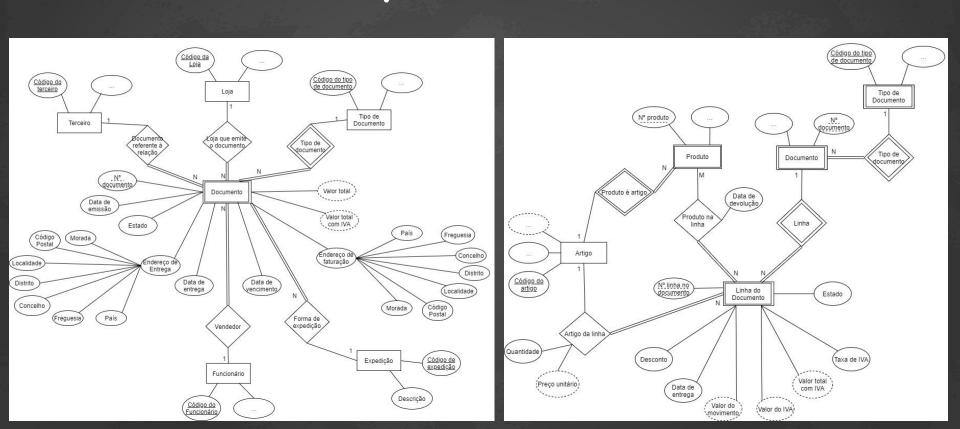


Joana Martins (84628) e Orlando Macedo (94521)



## Primeiro relembrar

### ... Principais Entidades ...





#### O1 SQL-DDL

Sql-DDL associados à definição da estrutura da bd.

#### 03 Formulários gráticos

Exemplos de formulários gráficos e de diferentes usos de SQL-DML

#### 0Z Normalização

Resultados retirados da análise do ER.

#### 04 Indexação

Exemplos de usos de índices para maior rapidez de consultas.



## O1 SQL-DDL

#### Criação de tabelas

Aspectos que procurámos ter em atenção:

- Nomear as constraints para serem mais fáceis de manipular após a criação das tabelas.
- Ter em atenção os valores de Chaves Estrangeiras e atribuir-lhes o valor mais sensato caso sejam removidos tuplos que as envolvam.
- Verificar os valores a serem introduzidos através de "Check".

```
CREATE TABLE ClubeVideo.FUNCIONARIO
   Cod Funcionario
   Nome Funcionario
                                VARCHAR(40) NOT NULL.
   Data Nascimento
                               DATE,
   Nacionalidade
                               VARCHAR(30),
   Sexo
                               VARCHAR(1),
   Estado Civil
                               VARCHAR(20).
   Telemovel
                                                    NOT NULL UNIQUE,
                                VARCHAR(9)
   E mail
                               VARCHAR(256)
                                                    NOT NULL UNIQUE.
   Vencimento
                               MONEY.
   Data Admissao
                               DATE.
   Cod endereco
                               INT.
   Cod Situacao
                               INT,
   Cod profissao
                               INT.
   Cod tipo pagamento
                                INT.
   NTF
                                VARCHAR(9)
                                                    UNIQUE,
   NIB
                               VARCHAR(21),
   No doc identificacao
                               VARCHAR(14)
                                                    UNIQUE,
   Cod_empresa
   PRIMARY KEY(Cod_Funcionario),
   CONSTRAINT FKEMPRESA FOREIGN KEY(Cod empresa)
       REFERENCES ClubeVideo.Empresa(nEmpresa) ON DELETE SET NULL,
   CONSTRAINT FKENDERECO FOREIGN KEY(Cod endereco)
       REFERENCES ClubeVideo.Endereco(nEndereco) ON DELETE SET NULL,
   CONSTRAINT FKSITUACAO FOREIGN KEY(Cod Situacao)
       REFERENCES ClubeVideo.SITUACAO(Cod Situacao) ON DELETE SET NULL,
   CONSTRAINT FKPROFISSAO FOREIGN KEY(Cod profissao)
       REFERENCES ClubeVideo.PROFISSAO(Cod Profissao) ON DELETE SET NULL,
   CONSTRAINT FKPAGAMENTO FOREIGN KEY(Cod tipo pagamento)
       REFERENCES ClubeVideo.PAGAMENTO(Cod Pagamento) ON DELETE SET NULL,
   CONSTRAINT CHECKTELEFUN CHECK(LEN(Telemovel) = 9).
   CONSTRAINT CHECKNIFFUN CHECK(LEN(NIF) = 9),
   CONSTRAINT CHECKNIBFUN CHECK(LEN(NIB) = 21),
   CONSTRAINT CHECKSEXOFUN CHECK(Sexo = 'F' OR Sexo = 'M')
```

Criação da tabela "Funcionário".

#### Mais exemplos ...

```
CREATE TABLE ClubeVideo.Documento(
   nDocumento
                   INT IDENTITY (1, 1),
   tipoDoc
                   INT
                               NOT NULL,
                   INT
   empresa
                               NOT NULL,
   terceiro
                   INT
                               NOT NULL,
   loja
                   INT
                               NOT NULL,
   dataEmissao
                   DATE
                               NOT NULL,
   dataEntrega
                   DATE
                               NOT NULL,
   nVendedor
                   INT
                               NOT NULL,
   dataVencimento DATE
                               NOT NULL.
   fExpedicao
                   INT
                               NOT NULL.
   endEntrega
                   INT
                               NOT NULL,
   endFaturacao
                   INT
                               NOT NULL,
   valorTotal
                   MONEY
                               DEFAULT NULL,
   valorTotalIVA
                   MONEY
                               DEFAULT NULL,
   estado
                   BIT
                               DEFAULT 0.
   PRIMARY KEY (empresa, tipoDoc, nDocumento),
   CONSTRAINT FKEMPRESADOC FOREIGN KEY (tipoDoc, empresa)
       REFERENCES ClubeVideo.TipoDocumento(nTipoDoc,empresa) ON DELETE CASCADE,
   CONSTRAINT FKTERCEIRODOC FOREIGN KEY (terceiro)
       REFERENCES ClubeVideo.TERCEIRO(Cod terceiro) ON DELETE CASCADE.
   CONSTRAINT FKLOJADOC FOREIGN KEY (loja)
       REFERENCES ClubeVideo.LOJA(Cod_loja) ON DELETE CASCADE,
   CONSTRAINT FKVENDEDORDOC FOREIGN KEY (nVendedor)
       REFERENCES ClubeVideo.FUNCIONARIO(Cod Funcionario) ON DELETE CASCADE.
   CONSTRAINT FKEXPEDICAODOC FOREIGN KEY (fExpedicao)
       REFERENCES ClubeVideo.Expedicao(nExpedicao) ON DELETE CASCADE,
   CONSTRAINT FKENDERECOENTDOC FOREIGN KEY (endEntrega)
       REFERENCES ClubeVideo. Endereco(nEndereco) ON DELETE NO ACTION,
   CONSTRAINT FKENDERECOFACDOC FOREIGN KEY (endFaturacao)
       REFERENCES ClubeVideo.Endereco(nEndereco) ON DELETE NO ACTION,
```

#### Mais exemplos ...

```
CREATE TABLE ClubeVideo.LinhaDocumento(
   nLinha
                   INT,
   documento
                   INT.
   tipoDoc
                   INT,
   empresa
                   INT,
   artigo
                   INT
                           NOT NULL,
   quantidade
                   INT
                           NOT NULL.
   precoUnit
                   MONEY,
   taxaIVA
                   INT,
                   DATE
                           NOT NULL,
   dataEntrega
   valorMov
                   MONEY.
   valorTVA
                   MONEY,
   valorTotalIVA
                   MONEY,
   desconto
                   INT
                           DEFAULT 0,
   estado
                           DEFAULT 0.
   PRIMARY KEY (empresa, tipoDoc, documento, nLinha),
   CONSTRAINT FKDOCUMENTOLINHA FOREIGN KEY (empresa, tipoDoc, documento)
       REFERENCES ClubeVideo.Documento(empresa,tipoDoc,nDocumento) ON DELETE CASCADE,
   CONSTRAINT FKARTIGOLINHA FOREIGN KEY (artigo)
       REFERENCES ClubeVideo.Artigo(nArtigo) ON DELETE CASCADE,
   CONSTRAINT CHECKQTLINHA CHECK(quantidade>0),
   CONSTRAINT CHECKIVALINHA CHECK(taxaIVA BETWEEN 0 and 100),
   CONSTRAINT CHECKDESCONTOLINHA CHECK(desconto BETWEEN 0 and 100),
```

Criação da tabela "Linha do Documento".

#### Mais exemplos ...

```
CREATE TABLE ClubeVideo.Produto LinhaDocumento(
    produto
                    INT,
   artigo
                    INT,
   linha
                    INT.
   documento
                    INT,
   tipoDoc
                    INT,
    empresa
                    INT,
   dataDevolucao
                   DATE
                            DEFAULT NULL,
   PRIMARY KEY (empresa, tipoDoc, documento, linha, artigo, produto),
   CONSTRAINT FKPRODUTOPROD FOREIGN KEY (artigo, produto)
        REFERENCES ClubeVideo.Produto(artigo, nProduto),
   CONSTRAINT FKLINHAPROD FOREIGN KEY (empresa, tipoDoc, documento, linha)
       REFERENCES ClubeVideo.LinhaDocumento(empresa,tipoDoc,documento,nLinha)
       ON DELETE CASCADE,
```

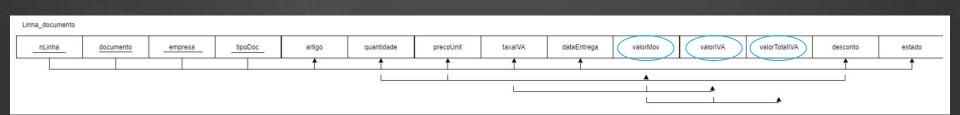
Criação da tabela "Produto da Linha do Documento".

#### Normalização



2ª Forma Normal

- Os atributos são atómicos.
- Não existem relações dentro de relações.
- Atributos não pertencentes a chaves candidatas dependem totalmente da chave primária.
- Não existem dependências transitivas





Formulários Gráficos e operações com SQL-DML

## Video demonstrativo

https://drive.google.com/file/d/1XWWIQLoYwoxCsTsoaok10RPOSAmDw6w2/view?usp=sharing



## Formulário gráfico de "Login"

 Para ser possível utilizar a aplicação é primeiro necessário efetuar um login que dependendo do mesmo redireciona para uma página de empregado de empresa ou de loja.

```
CREATE PROCEDURE ClubeVideo.pr_VERIFICAR_LOGIN @Cod_funcionario INT, @Passe NVARCHAR(50), @Msg_resposta NVARCHAR(250)='' OUTPUT
AS
BEGIN

SET NOCOUNT ON

SET @Cod_funcionario = REPLACE(@Cod_funcionario, '''', ''''');

DECLARE @cod INT

IF EXISTS (SELECT TOP 1 Cod_func FROM ClubeVideo.[LOGIN] WHERE Cod_func_@Cod_funcionario)

BEGIN

SET @cod=(SELECT Cod_func FROM ClubeVideo.[LOGIN] WHERE Cod_func_@Cod_funcionario AND Palavra_passe=HASHBYTES('SHA2_512', @Passe))

IF(@cod IS NULL)

SET @Msg_resposta='Password Incorreta'

ELSE

SET @Msg_resposta='Login efetuado'

END

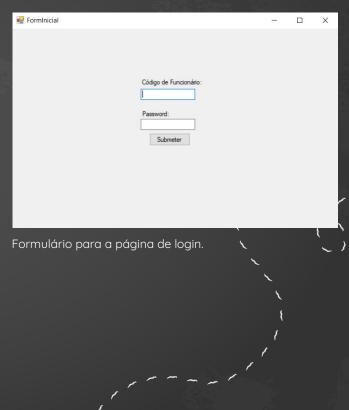
END

END

END

END

END
```



Procedimento usado para verificar veracidade dos dados introduzidos.

#### Exemplos de stored procedures

```
CREATE PROCEDURE ClubeVideo.pr SearchProdutosLoja (@loja INT, @nProduto INT = null, @artigo INT = null, @disponibilidade INT = null)
BEGIN
   DECLARE @sqlCommand nvarchar(1000);
   SET @sqlCommand = 'SELECT * FROM ClubeVideo.Produto';
   DECLARE @v loja nvarchar(100);
   SET @v loja = @loja;
   SET @sqlCommand += ' WHERE loja = ' + @v loja;
   IF (NOT (@artigo is NULL))
                                                                                                                                         Procedimento que retorna as lojas de uma empresa
       DECLARE @v artigo nvarchar(100);
       SET @v artigo = @artigo;
                                                                                                                                                             ou de um funcionário em específico.
       SET @sqlCommand += ' AND artigo = ' + @v artigo:
   FND
   IF (NOT (@nProduto is NULL))
                                                                                 CREATE PROCEDURE ClubeVideo.pr Lojas (@Cod empresa INT, @Cod funcionario INT = null)
       DECLARE @v nProduto nvarchar(100);
       SET @v nProduto = @nProduto:
       SET @sqlCommand += ' AND nProduto = ' + @v nProduto;
                                                                                     IF (@Cod funcionario is null)
   END
   IF (NOT (@disponibilidade is NULL))
                                                                                        SELECT * FROM ClubeVideo.LOJA WHERE Cod empresa=@Cod empresa;
                                                                                     END
       DECLARE @v disponibilidade nvarchar(100);
                                                                                     ELSE
                                                                                     BEGIN
       SET @v disponibilidade = @disponibilidade;
                                                                                         SELECT LL.Cod loja, LL.Nome loja, LL.E mail, LL.Telefone, LL.Cod endereco, LL.Cod empresa, LL.ativa
       SET @sqlCommand += ' AND disponibilidade = ' + @v disponibilidade;
                                                                                        FROM ClubeVideo, FUNCIONARIO AS F JOIN ClubeVideo, FUNCIONARIOS EM LOJAS AS L ON F.Cod Funcionario = L.Cod funcionario
   FND
                                                                                         JOIN ClubeVideo.LOJA AS LL ON L.Cod loja=LL.Cod loja
                                                                                         WHERE F.Cod Funcionario=@Cod funcionario;
   EXEC (@sqlCommand)
                                                                                 GO
```

### Exemplos de triggers

```
REATE TRIGGER t criarLinha CM ClubeVideo.LinhaDocumento
  DECLARE @nLinha AS INT;
  SELECT @ntinha-ntinha FROM inserted;
  SELECT @documento-documento FROM inserted;
  DECLARE StipoDoc AS INT:
  SELECT @tipoDoc-tipoDoc FROM inserted;
  DECLARE Gemoresa AS INT:
  SELECT @empresa-empresa FROM inserted;
  DECLARE Gartieo AS INT:
  SELECT Bartigo artigo FROM inserted:
  DECLARE @quantidade AS INT;
  SELECT @quantidade-quantidade FROM inserted:
  DECLARE @dataEntrega AS DATE;
  SELECT @dataEntrega-dataEntrega FROM inserted;
  DECLARE @desconto AS INT;
  SELECT @desconto desconto FROM inserted;
  DECLARE Mestado AS RIT:
  SELECT Westado-estado FROM inserted
  DECLARE Sterceiro AS INT:
  FROM ClubeVideo.Documento AS D 301N inserted AS L ON (D.empresa-L.empresa AND D.tipoDoc-L.tipoDoc AND D.nDocumento-L.documento);
  SELECT @precoUnit = Preco
   FROM Clubevideo.TERCEIRO AS T 301N Clubevideo.PRECO_ARTIGO AS P ON T.Cod_lista_precos-P.Cod_lista_precos
   WHERE Cod_terceiro-@terceiro AND Cod_artigo-@artigo;
  DECLARE STAXATVA AS INT:
   SELECT @taxaIVA - percentagem
  FROM Clubevideo Artigo 301% Clubevideo TipoTVA CW TVA-nTVA
  WERE marties Garties:
   DECLARE SyalorMov AS MONEY:
  SELECT @valorMov - (@quantidade * @precoUnit);
   DECLARE @per_taxaTVA AS DECIMAL(5,2);
  SELECT @per_taxaIVA = Cast(@taxaIVA AS DECIMAL(5,2))/100
   DECLARE @valorIVA AS MONEY;
  DECLARE @valorTotalIVA AS MONEY;
   IF (@desconto is null)
      SELECT @valorIVA = (@valorMov * @per_taxaIVA);
      SELECT @valorTotalTVA = (@valorMov + @valorTVA):
      INSERT INTO ClubeVideo.LinhaDocumento (empresa,tipoDoc,documento,nLinha,artigo,quantidade,precoUnit,taxaTVA,dataEntrega,
      valorMov, valorIVA, valorTotalIVA) SELECT empresa, tipoDoc, documento, @mlinha, artigo, quantidade, @precoUnit, @taxaIVA, dataEntrega,
      @valorMov,@valorIVA,@valorTotalIVA FROM inserted;
      DECLARE Open desconto AS DECIMAL(5,2):
      SELECT @per_desconto - Cast(@desconto AS DECIMAL(5,2))/100;
      SELECT @valorMov = (@valorMov = @valorMov*@per_desconto);
SELECT @valorIVA = (@valorMov * @per_taxaIVA);
      SELECT @valorTotalIVA = (@valorMov + @valorIVA);
      INSERT INTO ClubeVideo.LinhaGocumento (empresa,tipoGoc,documento,nLinha,artigo,quantidade,precoUnit,taxaTVA,dataEntrega
       walorMov,valorIVA,valorTotalIVA,desconto) SELECT empresa,tipoDoc,documento,@mlinha,artigo,quantidade,@precoUnit.@taxaIVA,
      dataEntrega,@valorMov.@valorIVA,@valorTotalIVA,desconto FROM inserted;
```

```
CREATE TRIGGER ClubeVideo.retirarLoginDesp ON ClubeVideo.FUNCIONARIO
AFTER UPDATE
                                -- retirar um empregado dos logins assim que o mesmo é despedido
AS
BEGIN
   DECLARE @codSituacao AS INT:
   DECLARE @codFuncionario AS INT;
   SELECT @codFuncionario = Cod Funcionario, @codSituacao = Cod Situacao FROM INSERTED;
   IF @codSituacao = 4
       IF EXISTS (SELECT TOP 1 Cod_func FROM ClubeVideo.[Login] WHERE Cod_func = @codFuncionario)
       BEGIN
           DELETE FROM ClubeVideo.[login] WHERE Cod func = @codFuncionario;
           PRINT 'Funcionário com código ' + CAST(@codFuncionario AS VARCHAR(10)) + ' deixou de ter acesso à empresa';
       END
   END
END
```

Trigger que verifica a situação de um funcionário aquando de uma atualização.

#### Exemplos de UDF's

```
CREATE FUNCTION ClubeVideo.f_Linhas (@documento INT, @tipoDoc INT, @empresa INT) RETURNS Table
AS
RETURN(SELECT * FROM ClubeVideo.LinhaDocumento WHERE empresa=@empresa AND tipoDoc=@tipoDoc AND documento=@documento)
GO
```

UDF que retorna as linhas de um documento.

```
CREATE FUNCTION ClubeVideo.f_ProdutosLinha (@linha INT, @documento INT, @tipoDoc INT, @empresa INT) RETURNS Table
AS
RETURN(SELECT * FROM ClubeVideo.Produto_LinhaDocumento
WHERE empresa=@empresa AND tipoDoc=@tipoDoc AND documento=@documento AND linha=@linha)
GO
```

UDF que retornar os produtos de uma linha de um documento.

```
CREATE FUNCTION ClubeVideo.f_EnderecoCompleto (@nEndereco INT) RETURNS Table
AS
RETURN(SELECT * FROM ClubeVideo.Endereco WHERE nEndereco=@nEndereco)
GO
```

UDF que retorna toda a informação sobre um endereço.

#### Uso de Transações

```
CREATE PROCEDURE ClubeVideo.pr deleteLinha (@nLinha INT, @documento INT, @tipoDoc INT, @empresa INT)
   SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
   BEGIN TRANSACTION:
   DECLARE @estadoD AS BIT:
   SELECT @estadoD = estado
   FROM ClubeVideo.Documento
   WHERE empresa =@empresa AND tipoDoc=@tipoDoc AND nDocumento=@documento;
   IF (@estadoD=1)
   BEGIN
       COMMIT TRANSACTION;
       RAISERROR('Não é possivel eliminar a Linha no Documento', 16, 1):
   END
   ELSE
   BEGIN
       DELETE ClubeVideo.LinhaDocumento
       WHERE empresa = @empresa AND tipoDoc = @tipoDoc AND documento = @documento AND nLinha = @nLinha;
       COMMIT TRANSACTION:
   END
```

Uso de transações na operação de remoção de uma linha.

```
CREATE PROCEDURE ClubeVideo.pr NovaLinha (@nLinha INT, @documento INT, @tipoDoc INT, @empresa INT,
   @artigo INT, @quantidade INT, @dataEntrega DATE, @desconto INT = null)
   SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
   REGIN TRANSACTION:
   CREATE TABLE #Documento aux(
       dataEntrega DATE.
       dataVencimento DATE.
       estado
   INSERT INTO #Documento aux (dataEntrega,dataVencimento,estado)
   SELECT dataEntrega, dataVencimento, estado
   FROM ClubeVideo.Documento
   WHERE empresa=@empresa AND tipoDoc=@tipoDoc AND nDocumento=@documento;
   DECLARE @estadoD AS BIT;
   SELECT @estadoD = estado
   FROM #Documento aux;
   DECLARE @dataEntregaD AS DATE;
   SELECT @dataEntregaD = dataEntrega
   FROM #Documento aux;
   DECLARE @dataVencimentoD AS DATE:
   SELECT @dataVencimentoD = dataVencimento
   FROM #Documento aux;
   IF (@estadoD=0 AND (@dataEntrega>@dataEntregaD) AND (@dataEntrega<@dataVencimentoD))</pre>
       IF (@desconto is null)
           INSERT INTO ClubeVideo.LinhaDocumento (empresa, tipoDoc, documento, nLinha, artigo, quantidade, dataEntrega) VALUES
           (@empresa, @tipoDoc, @documento, @nLinha, @artigo, @quantidade, @dataEntrega);
           COMMIT TRANSACTION:
       END
       ELSE
           INSERT INTO ClubeVideo.LinhaDocumento (empresa, tipoDoc, documento, nLinha, artigo, quantidade, dataEntrega, desconto) VALUES
           (@empresa, @tipoDoc, @documento, @nLinha, @artigo, @quantidade, @dataEntrega, @desconto);
           COMMIT TRANSACTION
       END
   END
   FLSE
   REGIN
      COMMIT TRANSACTION:
       RAISERROR('Não é possivel inserir a Linha no Documento', 16, 1);
```

#### Indexação

CREATE INDEX idx\_Loja ON ClubeVideo.LOJA(Cod\_empresa,ativa) WITH (FILLFACTOR = 90, PAD\_INDEX = ON);

Índice non\_clustered

CREATE INDEX idx Produto ON ClubeVideo.Produto(loja,artigo) WITH (FILLFACTOR = 65, PAD INDEX = ON);

Índice non\_clustered

CREATE INDEX idx\_Funcionario ON ClubeVideo.FUNCIONARIO(Cod\_empresa,Nome\_Funcionario) WITH (FILLFACTOR = 85, PAD\_INDEX = ON);

Índice non-clustered

CREATE INDEX idx\_ListaPrecos ON ClubeVideo.LISTA\_PRECOS(Cod\_empresa) WITH (FILLFACTOR = 85, PAD\_INDEX = ON);

Índice non\_clustered

# Lonclusão



... Informação mais detalhada no relatório.

