/\* CRIANDO BANCO E ACESSANDO \*/

CREATE DATABASE TESTE\_ACCENTURE2

GO

USE TESTE\_ACCENTURE2

GO

/\* CRIAÇÃO DAS TABELAS \*/

CREATE TABLE tbContrato(

IDContrato INT NOT NULL IDENTITY,

CodContrato INT NOT NULL,

NomeContrato VARCHAR(20) NOT NULL,

TipoInstrumento VARCHAR(10) NOT NULL,

Vencimento DATE NOT NULL,

Emissao DATE NOT NULL,

Moeda CHAR(3) NULL,

Indexador VARCHAR (30) NULL,

Calendario VARCHAR (50) NOT NULL,

Basis VARCHAR (30) NOT NULL,

Pais CHAR (6) NOT NULL,

SeriePreco VARCHAR(30) NULL

CONSTRAINT PK\_IDCONTRATO PRIMARY KEY (IDContrato)

)

GO

CREATE TABLE tbCronograma(

IDContrato INT NOT NULL,

DataFoto DATE NOT NULL,

IDTranche TINYINT NOT NULL,

Tipo INT NOT NULL,

DatBase DATE NOT NULL,

DataBaixa DATE NOT NULL,

DataEvento DATE NOT NULL,

Projetado DECIMAL (38,2) NOT NULL,

Realizado DECIMAL (38,2) NOT NULL,

TaxaCambio DECIMAL (38,2) NOT NULL,

TaxaVariante DECIMAL (38,2) NOT NULL,

Taxa DECIMAL (38,2) NOT NULL,

FoiRealizado BIT NOT NULL

)

GO

/\* FOREIGN KEY DE CRONOGRAMA PARA CONTRATO \*/

ALTER TABLE tbCronograma

ADD CONSTRAINT FK\_Cronograma\_Contrato

FOREIGN KEY (IDContrato) REFERENCES tbContrato(IDContrato)

GO

/\* ------------------------------------------------\*/

/\* POPULAÇÃO DAS TABELAS \*/

/\* TABELA CONTRATO \*/

BULK INSERT tbContrato

FROM 'C:\Teste Accenture\ETL arquivos\Arquivo1\_Contratos.txt'

WITH (KEEPIDENTITY,

FIRSTROW = 2,

ROWTERMINATOR ='\n',

MAXERRORS=0);

GO

SELECT \* FROM tbContrato

GO

/\* TABELA CRONOGRAMA \*/

BULK INSERT tbCronograma

FROM 'C:\Teste Accenture\ETL arquivos\Arquivo2\_Cronograma2.csv'

WITH (FIRSTROW = 2,

fIELDTERMINATOR =';',

ROWTERMINATOR ='\n',

MAXERRORS=0);

GO

SELECT \* FROM tbCronograma

GO

/\* ------------------------------------------------\*/

/\* ADICIONAMENTO TABELA DATA DO ULTIMO FLUXO \*/

ALTER TABLE tbContrato

ADD DataUltimoFluxo DATE NULL

GO

/\* identificador de fluxo do cronograma do contrato \*/

/\*----------------- CHECKSUM -----------------------\*/

ALTER TABLE tbCronograma

ADD IDFluxo AS CHECKSUM(IDContrato,IDTranche,Tipo,DatBase)

GO

--ALTER TABLE tbCronograma

-- ADD IDFluxo AS CONCAT(IDContrato,IDTranche,Tipo,convert(varchar, "DatBase", 112),IdCronograma)

--GO

--https://docs.microsoft.com/pt-br/sql/t-sql/functions/checksum-transact-sql?view=sql-server-ver15

--https://stackoverflow.com/questions/45468766/checksum-example-explanation-in-sql-server-t-sql

--https://www.mssqltips.com/sqlservertip/1023/checksum-functions-in-sql-server-2005/

/\*----------------------------------------------------------------\*/

-- ADICIONAMENTO DE INDICES

CREATE CLUSTERED INDEX IX\_TbCronograma

on tbCronograma (idfluxo)

GO

ALTER INDEX IX\_TbCronograma

ON TBCRONOGRAMA

REBUILD WITH(FILLFACTOR = 80)

GO

CREATE NONCLUSTERED INDEX NIX\_TbCronograma

ON tbCronograma (IDContrato, DataFoto)

INCLUDE(IDFLUXO, TIPO, DATBASE, DATAEVENTO, FOIREALIZADO, PROJETADO, REALIZADO, TaxaCambio, TaxaVariante, Taxa)

GO

CREATE NONCLUSTERED INDEX NIX\_TbContrato

ON tbContrato (CodContrato)

Include (NomeContrato, tipoinstrumento, Moeda, Indexador, Calendario, Basis, Pais, SeriePreco, DataUltimoFluxo) --ADIÇÃO DO INCLUDE

GO

--https://stackoverflow.com/questions/1307990/why-use-the-include-clause-when-creating-an-index

/\*----------------------------------------------------------------\*/

/\* Contrução de uma sotred procedure que receberá como parâmetro @IDContrato e @DataFoto,

e deve retornar o cronograma do contrato na data foto. Formato de saída abaixo: \*/

-- RETORNAR O MAIS RECENTE ATÉ A DATA FOTO.

-- OU ADICIONAR UM "TOP (1)" CASO SEJÁ REQUERIDO APENAS O PRIMEIRO RESULTADO

CREATE OR ALTER PROCEDURE SP\_CONTRATO\_DATAFOTO

@IDContrato INT, @DataFoto DATE

AS

SELECT IDCONTRATO, DataFoto, IDFLUXO, TIPO, DATBASE, DATABAIXA, DATAEVENTO,

FOIREALIZADO, PROJETADO, TaxaCambio, TaxaVariante, Taxa

FROM tbCronograma

WHERE IDCONTRATO = @IDContrato AND DATAFOTO = @DataFoto

ORDER BY DataBaixa DESC

GO

-----------

-- Chamar a proc inserindo Id do Contrato e Data da Foto

EXEC SP\_CONTRATO\_DATAFOTO 1502135, '19000101'

GO

/\*------------------------------------------------------------------\*/

/\* Construção de uma stored procedure que apartir dos

dados do arquivo 3 irá inserir novas fotos no modelo. \*/

/\*------------------------------------------------------------------\*/

/\* Construção de uma stored procedure que retorne na primeira coluna o Nome do contrato,

na segunda coluna o indexador em ordem crescente e na terceira coluna o indexador em ordem

decrescente. \*/

CREATE OR ALTER PROCEDURE SP\_NomeContrato\_Indexador2

--@NomeContrato VARCHAR(30)

AS

SELECT NomeContrato,

ROW\_NUMBER() OVER(ORDER BY Indexador DESC) as Ind\_Decrescente,

ROW\_NUMBER() OVER(ORDER BY Indexador ASC) as Ind\_Crescente,

Indexador

FROM tbContrato

ORDER BY IDContrato

GO

EXEC SP\_NomeContrato\_Indexador2

GO

/\*

--DROP PROCEDURE SP\_NomeContrato\_Indexador

CREATE or ALTER PROCEDURE SP\_NomeContrato\_Indexador

@NomeContrato VARCHAR(30)

AS

SELECT NomeContrato, Indexador AS IndexadorCrescente,

Indexador as IndexadorDecrescente2

FROM tbContrato

WHERE NomeContrato LIKE ''+@NomeContrato+''

ORDER BY IndexadorCrescente ASC,

IndexadorDecrescente2 DESC

GO

-- Chamar Proc com o nome exato do contrato

-- ou adicionando uma fração do nome entre aspas simples e sinais de % ( '%Exemplo%' )

EXEC SP\_NomeContrato\_Indexador '%13%'

GO

\*/

--MANIPULAÇÃO DE FUNÇÕES XML

--Criar uma consulta que retorne para cada contrato um xml com os atributos seguindo o schema abaixo

--e insira em uma nova tabela contendo duas colunas: IDContrato, xmlgerado

--<contract>

--<[tipoinstrumento] @issuedate @calendar @basis/>

--<currency @index @currency />

--<series @priceseries>

--</[tipoinstrumento]>

--</contract>

/\* CRIAÇÃO DOS XMLs \*/

SELECT \* FROM tbContrato

DECLARE @IDCONTRATO INT

SET @IDCONTRATO = 372586

SELECT

IDContrato AS '@Contrato',

(

SELECT

Emissao AS '@issuedate',

Calendario AS'@calendar',

Basis AS '@basis'

FROM tbContrato

WHERE IDContrato = @IDCONTRATO

FOR XML PATH ('Date'), TYPE

),

(

SELECT

Indexador AS '@index',

Moeda AS '@currency'

FROM tbContrato

WHERE IDContrato = @IDCONTRATO

FOR XML PATH ('Currency'), TYPE

),

(

SELECT

SeriePreco AS '@priceseries'

FROM tbContrato

WHERE IDContrato = @IDCONTRATO

FOR XML PATH ('PriceSeries'), TYPE

)

FROM tbContrato

WHERE IDContrato = @IDCONTRATO

FOR XML PATH ('Contrato'), ROOT ('ACCENTURE')

GO

CREATE TABLE XML\_DATA(

IDContrato INT UNIQUE,

XMLGERADO XML

)

GO

--CRIAÇÃO DA PROC QUE ADICIONA OS XMLS A TABELA XML\_DATA

CREATE PROCEDURE SP\_MANIPULA\_XML

@IDCONTRATO INT

AS

DECLARE @xmlgerado xml

SET @xmlgerado = ( SELECT -- Definição da variável como o resultado da criação do XML.

IDContrato AS '@Contrato',

(

SELECT

Emissao AS '@issuedate',

Calendario AS'@calendar',

Basis AS '@basis'

FROM tbContrato

WHERE IDContrato = @IDCONTRATO

FOR XML PATH ('Date'), TYPE

),

(

SELECT

Indexador AS '@index',

Moeda AS '@currency'

FROM tbContrato

WHERE IDContrato = @IDCONTRATO

FOR XML PATH ('Currency'), TYPE

),

(

SELECT

SeriePreco AS '@priceseries'

FROM tbContrato

WHERE IDContrato = @IDCONTRATO

FOR XML PATH ('PriceSeries'), TYPE

)

FROM tbContrato

WHERE IDContrato = @IDCONTRATO -- Recebe a variável da Proccedure

FOR XML PATH ('Contrato'), ROOT ('ACCENTURE')) -- tranforma o xml e o ")" fecha o resultado da definição da variável @xmlgerado

SELECT @Xmlgerado -- adiciona o resultado da váriavel a tabela XML\_DATA recebendo as duas variáveis.

INSERT INTO XML\_DATA (IDContrato, XMLGERADO)

VALUES (@IDCONTRATO, @xmlgerado)

GO

SELECT IDCONTRATO FROM tbContrato

GO

EXEC SP\_MANIPULA\_XML 11546068

GO

SELECT \* FROM XML\_DATA

GO

--https://www.informit.com/articles/article.aspx?p=389112&seqNum=2

--http://www.linhadecodigo.com.br/artigo/3705/representando-dados-em-xml-no-sql-server.aspx

--http://www.linhadecodigo.com.br/artigo/3705/representando-dados-em-xml-no-sql-server.aspx#ixzz6sOhQEDab

--https://docs.microsoft.com/pt-br/sql/relational-databases/xml/examples-using-path-mode?view=sql-server-ver15

--https://stackoverflow.com/questions/48338477/insert-into-select-in-sql-server

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--Crie um script que insere um novo atributos chamado “calendarname” \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*---nos xmls dessa nova tabela. O Atributo deve ser o valor da tag @calender sem o prefixo ‘CAL’ \*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\* QUERY \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

DECLARE @CALENDARNAME NVARCHAR (200), -- receberá o novo nome do contrato na substring

@IDCONTRATO INT, -- Recebe Id do contrato

@x XML, -- recebe a estrutura da coluna xml.

@tag varchar(100) = 'calendarname' -- nova tag

SET @IDCONTRATO = 170374

SET @CALENDARNAME = (SELECT SUBSTRING( c.Calendario, 4, 6) AS CALENDARNAME

FROM XML\_DATA AS X

INNER JOIN tbContrato AS C

ON X.IDContrato = C.IDContrato

WHERE x.IDContrato = @IDCONTRATO

)

SET @x = (

SELECT XMLGERADO

FROM XML\_DATA

WHERE IDContrato = @IDCONTRATO

)

DECLARE @new\_element XML='<' + @tag +'>' + @CALENDARNAME + '</' + @tag + '>';

SET @x.modify('insert sql:variable("@new\_element") as first into (/Contrato)[1]');

SELECT @x;

UPDATE XML\_DATA

SET XMLGERADO = @X

WHERE IDContrato = @IDCONTRATO

GO

select xmlgerado from XML\_DATA

go

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*STORED PROCEDURE\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE or ALTER PROCEDURE SP\_ALTERA\_CALENDAR

@IDCONTRATO INT

AS

DECLARE @CALENDARNAME NVARCHAR (200), -- receberá o novo nome do contrato na substring

--@IDCONTRATO INT, -- Recebe Id do contrato

@x XML, -- recebe a estrutura da coluna xml.

@tag varchar(100) = 'calendarname' -- nova tag

SET @CALENDARNAME = (SELECT SUBSTRING( c.Calendario, 4, 13) AS CALENDARNAME

FROM XML\_DATA AS X

INNER JOIN tbContrato AS C

ON X.IDContrato = C.IDContrato

WHERE x.IDContrato = @IDCONTRATO

)

SET @x = (

SELECT XMLGERADO

FROM XML\_DATA

WHERE IDContrato = @IDCONTRATO

)

DECLARE @new\_element XML='<' + @tag +'>' + @CALENDARNAME + '</' + @tag + '>';

SET @x.modify('insert sql:variable("@new\_element") as first into (/ACCENTURE/Contrato)[1]');

SELECT @x;

UPDATE XML\_DATA

SET XMLGERADO = @X

WHERE IDContrato = @IDCONTRATO

GO

-------------------

select IDContrato,xmlgerado from XML\_DATA

go

EXEC SP\_ALTERA\_CALENDAR 175702

GO

11546068

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--Crie um script que remova dos xmls a nova tag criada.

UPDATE XML\_DATA

SET xmlgerado.modify('delete (/ACCENTURE/Contrato/calendarname)')

WHERE IDContrato = 441000