MODELAGEM DE ESTRUTURA RELACIONAL

# Descrição

Você deve criar um modelo de banco de dados para atender os requisitos de uma integração que fornecerá arquivos contendo informações de contratos financeiros e cronogramas desses contratos respeitando uma visualização temporal.

Você deve desenvolver o método de captura desses arquivos (ETL) para importá-los no modelo criado, e desenvolver as consultas solicitadas nos requisitos abaixo.

O método de captura dos arquivos é livre, mas o modelo de banco deve ser desenvolvido no SQL Server (qualquer versão a partir 2016).

# REQUISITOS

## rEQUISITOS OBRIGATÓRIOS

### mODELAGEM DA ESTRUTURA DE BANCO DE DADOS

O arquivo 1 contém os contratos financeiros que devem ser importados.

O arquivo 2 contém o cronograma dos contratos financeiros com seus fluxos em seu estado inicial.

O arquivo 3 contém fotos adicionais de cronograma.

Deve ser criado um modelo de banco de dados relacional para receber a importação dos arquivos e atender as condições abaixo:

* Contratos são únicos (*OK*)
* Contratos possuem um cronograma, mas o estado do cronograma pode evoluir no tempo. (*quais são os estados?*)  
    
  Por exemplo, quando um fluxo é pago na data X (*Data Foto?*), o cronograma do contrato recebe um novo estado (*Nova Linha na TB\_Cronograma*) onde este fluxo (*Fluxo = cronograma?*) está realizado, mas o estado anterior continua registrado em sua data original.

*Não entendi muito bem essa parte, eu fiz uma consulta aqui considerando os campos informados abaixo, porém as duas linhas estão como “FoiRealizado” = 0, pelo que entendi uma das linhas deveria ser 1  
SELECT \* FROM TB\_Cronograma tc WHERE tc.IDContrato = 11546204 AND tc.IDTranche = 4 AND tc.Tipo = 3 AND tc.[DataBase] = '2020-04-15'*

* Deve ser gerado um identificador de fluxo do cronograma do contrato (o que determina o fluxo são os campos IDContrato, IDTranche, Tipo e dataBase)
* Fluxos possuem propriedades estáticas no tempo: Tipo, IDTranche e dataBase. As demais propriedades podem ser alteradas em novas fotos.
* Fluxos podem ser realizados apenas uma vez.
* O identificador do fluxo é único para todas as fotos em que estiver inserido
* Quando fluxos forem alterados uma coluna DataUltimoFluxo na tbContrato deve ser atualizada contendo a DataBase do fluxo de maior DataBase deste contrato.

*Fiz a tratativa desse caso na* ***P\_SistemaDeContratos\_InsereFoto*** *ao inserir uma nova foto (fluxo/contrato?) atualiza na TB\_Contrato o ultimo fluxo*

Após criação do modelo de banco, deve construir um processo de extração e carga dos arquivos 1 e 2 para preencher as tabelas.

### stored procedures de consulta e inserção

Construção de uma stored procedure que receberá como parâmetro @IDContrato e @DataFoto, e deve retornar o cronograma do contrato na data foto. Formato de saída abaixo: (OK **P\_SistemaDeContratos\_ListaCronograma**)



Construção de uma stored procedure que a partir dos dados do arquivo 3 irá inserir novas fotos no modelo. (*nesse ponto não entendi se a procedure deve apenas importar a carga para a tabela principal ou se irá receber como parâmetro de entrada os dados da tabela*).   
**Procedure: P\_SistemaDeContratos\_InsereFoto**

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. (*Alguns indexadores estavam nulos, deixei no retorno da procedure como “Não Especificado”*) **Procedure: P\_SistemaDeContratos\_ListaContratos**

### monitoria de banco

Para prevenir gargalos que impeçam os processos do banco de executarem com normalidade, você deve descrever meios de monitorar o banco para identificar esses pontos críticos.

No meu dia a dia consumo utilizar as duas procedures abaixo, uma já é bem conhecida disponibilizada pelo Adam Machanic e outra para identificar processos em lock.

EXEC sp\_WhoIsActive

EXEC stpLock\_Raiz

## Requisitos opcionais

### indexação e plano de execução

Seu modelo de banco deve possuir uma estrutura de indexação ótima, e você pode apresentar os planos de execução das consultas e explicá-los.

Pelas consultas realizadas pelas procedures, criei apenas esse índice abaixo, a query para criação está dentro do script.  
  
**IX\_TB\_Cronograma\_IDContrato\_DataFoto\_Includes**

### Integração contínua e scripts de versionamento

Durante o processo de desenvolvimento você deve utilizar alguma estratégia de versionamento de scripts. Sugestão git + projeto sql (visual studio ou azure data studio)

No momento não tenho experiência com sistema de versionamento para SQL server

### Testes automatizados

Criar testes unitários de banco de dados para as procedures desenvolvidas, para que rodem no build de uma solução SQL.

### visualização do dado

Você pode apresentar os resultados das consultas/procedures em uma ferramenta de visualição (por exemplo: PowerBI).

### manipulação de funções xml **(ok)**

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>

Não costumo usar arquivos XML, estou mais habituado com os JSON mas consegui desenvolver conforme solicitado: **P\_SistemaDeContratos\_GeraXMLContrato**

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’

Procedure para atualização de todos os XML da tabela TB\_Contratos\_XML: **P\_SistemaDeContratos\_AlteraXMLContrato\_AdicionaTAG**

Crie um script que remova dos xmls a nova tag criada.  
Procedure para remover a tag criada anteriormente: **P\_SistemaDeContratos\_AlteraXMLContrato\_RemoveTAG**