 <p>INSTITUTO FEDERAL Paraíba Campus João Pessoa</p> <p>Curso: Superior de Tecnologia em Sistemas para Internet Disciplina: Banco de Dados II Professor: Alex Sandro da Cunha Rêgo Período: 3º Período</p>
---	---

Projeto 2019.1 (versão 1.0 – 22/05/2019)

Implementação de rotinas de banco de dados

Sistema de Contabilidade - SICON

1. Objetivo:

Implementação de rotinas automáticas de processamento em banco de dados utilizando *stored procedures*, cursores, triggers, visões e suporte transacional para garantia da atomicidade.

2. Banco de Dados

O projeto deve ser implementado, obrigatoriamente, no banco de dados PostgreSQL.

3. Equipe

Até 3 (três) pessoas.

O projeto será defendido por todos os membros da equipe.

As notas poderão ser diferenciadas, de acordo o grau de envolvimento do aluno no desenvolvimento do projeto.

#	Equipe
1	
2	
3	
4	

4. Data de Entrega (envio do código):

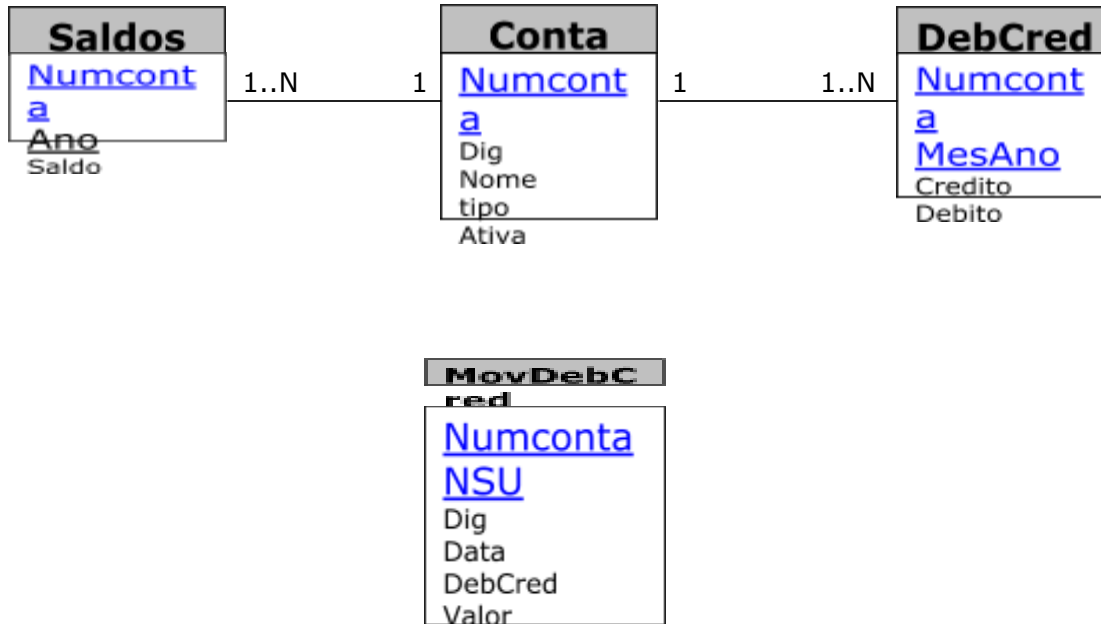
18 de junho de 2019. Submissão via Google Classroom.

Os projetos submetidos após o prazo terão uma penalidade de -1,5 pontos.

A implementação do projeto deve ser submetida sob a forma de script SQL (arquivo com extensão .sql), contendo toda a sequência de instruções para montagem do banco, criação dos objetos de banco de dados e seus respectivos testes de funcionamento e comentários ao longo do código.

5. Esquema de Banco de Dados:

Considere o seguinte esquema relacional para um sistema de contabilidade:



Cada tabela é definida sucintamente da seguinte maneira:

- **Contas:** armazena informações sobre as contas do sistema, com números exclusivos;
- **Saldos:** armazena o saldo inicial de cada ano para cada conta;
- **DebCred:** armazena o total de débito e crédito de cada conta em um determinado mês e ano;
- **MovDebCred:** tabela responsável por toda a movimentação financeira das contas para cada mês e ano, onde cada conta pode receber inúmeras movimentações.

Estas tabelas podem ser povoadas da seguinte forma (segundo sua cardinalidade):

Visão dos dados: Tabela "Conta"

NumConta	Dig	Nome	Tipo	Ativa
11010100100	0	Banco do Brasil	A	S
11010100200	2	Caixa matriz	A	S
11010100300	8	Fundo de participação	S	S
...

Tipo: A (Analítica) e S (Sintética). Uma conta do tipo analítica pode receber lançamento de crédito ou débito. Uma conta sintética só pode ter seu saldo evoluído (automaticamente) de acordo com o lançamento realizados nas contas filhas

Ativa: S (Sim) e N (Não). As contas que estão inativas não podem ser movimentadas (ter seu saldo ou evolução de crédito/débito) no sistema

Toda conta é constituída por 11 dígitos numéricos.

ilustração dos dados: Tabela "Saldos"

NumConta	Ano	Saldo
11010100100	2018	2500.00
11010100100	2019	-3300.00
11010100200	2018	1200.00
11010100200	2019	300.00
...

ilustração dos dados: Tabela "MovDebCred"

NumConta	NSU	Data	DebCred	Valor
11010100100	1	01-01-2019	C	20.00
11010100100	2	01-01-2019	D	30.00
11010100100	3	04-01-2019	D	40.00
11010100100	4	05-01-2019	C	50.00
11010100200	5	01-01-2019	C	10.00
11010100200	6	01-01-2019	C	30.00
11010100200	7	02-02-2019	D	20.00
11010100300	8	03-01-2019	C	100.00
...

ilustração dos dados: Tabela "DebCred"

NumConta	MesAno	Credito	Debito
11010100100	012019	0.00	100.00
11010100100	022019	150.00	50.00
...
11010100100	122019	260.00	130.00
11010100200	012019	350.00	220.00
11010100200	022019	130.00	60.00
...
11010100200	122019	1000.00	3000.00

(outra seqüência de registros para a próxima conta)

6. Dicionário de Dados

Para haver uma padronização no nome e tipo dos campos, o esquema de banco de dados deve obedecer rigorosamente ao dicionário de dados definido nesta prática.

TABELA CONTAS

Campo	Tipo	Permite Nulo	Def .	Restr.	CONTEÚDO
NumConta	Char(11)	Não	-		Número da conta com 11 dígitos (obrigatório). CHECK (LEN (numconta) = 11)
*** A chave primária é composta pelos campos acima ***					
Dig	Char(1)	Não	-	0..9 e &	Dígito de verificação da conta Aceita os dígitos numéricos de 0 a 9 e o "&"
Nome	Varchar(50)	Não	-	-	Nome da conta
Tipo	Char(1)	Não	A	A,S	A – Analítica S – Sintética
Ativa	Char(1)	Não	S	S,N	Situação da conta S – Ativa N - Inativa

TABELA SALDOS

Campo	Tipo	Permite Nulo	Def.	Restr.	CONTEÚDO
NumConta	Char(11)	Não	-	FK	Número da conta Chave estrangeira para a tabela <i>conta</i>
Ano	Int	Não	-	-	Ano do processamento
*** A chave primária é composta pelos campos acima ***					
Saldo	numeric(9,2)	Não	0	-	Saldo inicial da conta para o respectivo ano

TABELA DEBCRED

Campo	Tipo	Permite Nulo	Def.	Restr.	CONTEÚDO
NumConta	Char(11)	Não	-	FK	Número da conta Chave estrangeira para a tabela <i>conta</i>
MesAno	Char(6)	Não	-	-	Mês/Ano da movimentação
*** A chave primária é composta pelos campos acima ***					
Credito	numeric(11,2)	Não	0	-	Movimento total de crédito da conta para um referido mês e ano
Debito	numeric(11,2)	Não	0		Movimento total de débito da conta para um referido mês e ano

TABELA MOVDEBCRED

Campo	Tipo	Permite Nulo	Def.	Restr.	CONTEÚDO
NumConta	Char(11)	Não	-	-	Número da conta
NSU	int	Não	-	Identity	Número sequencial gerado pelo sistema
*** A chave primária é composta pelos campos acima ***					
Dig	Char(1)	Não	-	-	Dígito da conta
Data	DateTime	Não	-	-	Data da movimentação
DebCred	Char(1)	Não	-	D,C	Tipo da movimentação da conta D – Débito C – Crédito
Valor	numeric(11,2)	Não	0		Valor da movimentação

7. Lógica do Negócio

7.1 O cálculo do dígito

Consideremos a conta 11010100100. A fórmula para cálculo do dígito obedece às seguintes regras:

- Calcule a soma dos produtos dos onze dígitos utilizando os seguintes pesos: 2, 7, 6, 5, 4, 3, 2, 7, 6, 5 e 4.

Exemplo: **soma** = $(1*2) + (1*7) + (0*6) + \dots + (0*5) + (0*4) = 23$

- Em seguida, obtenha o resto da divisão da **soma** por 11:
Resto = **soma % 11** (resto = 1)
- O dígito é obtido através da diferença do número 11 com o resto:
Dígito = **11 - Resto** (dígito = 10)

- O dígito final é obtido pelas condições:
Se dígito = 10 -> **dígito final = 0**
Se dígito = 11 -> **dígito final = &**
Para as outras situações -> **dígito final = dígito**
- Então, concluímos que para a conta 11010100100, o dígito calculado é **0**.

7.2 Os lançamentos de débito/crédito

Na tabela **MovDebCred**, o campo **numconta** não é chave estrangeira para a tabela **contas**. Esquisito não? Mas é assim mesmo; o usuário pode inserir errado uma conta (ou dígito) para que a **crítica**¹ aponte as inconsistências.

A tabela **MovDebCred** armazena o movimento financeiro das contas (de débito ou crédito) relacionados a diferentes meses/anos. Insira diferentes lançamentos em contas específicas para simular seu povoamento.

7.3 A crítica do movimento

Considerando que todos os lançamentos de débito e crédito, para um referido mês ano, foram inseridos no banco de dados, é hora de realizar a **crítica do movimento**.

A **crítica do movimento** consiste em certificar se os registros lançados não violam as regras de integridade do domínio do problema. A realização da crítica é um passo que antecede a **atualização do plano de contas**. Sem sua execução, poderemos ter uma evolução dos saldos que não condiz com a realidade dos números.

Na realização da **crítica do movimento**, analisando registro por registro, devem ser identificadas irregularidades do tipo:

- Conta não cadastrada (não existe a integridade referencial);
- Dígito incorreto (dígito informado não é compatível com o dígito real);
- Conta sintética recebendo lançamento.

7.4 Atualização do Plano de Contas

Uma vez certificada a validade dos lançamentos de débito e crédito, através da rotina de **crítica do movimento**, é chegada a hora de realizar a **atualização das contas**. Em um sistema real, isto é feito após o fechamento de cada mês/ano.

A execução da atualização do Plano de Contas envolve as tabelas **movdebcred** (consulta), **debcred** e **conta** (modificação).

Não é obrigatório que todas as contas cadastradas tenham lançamentos em todos os meses.

A **atualização do plano de contas** é feita através do processamento dos lançamentos de débito e crédito de um mês/ano específico.

A lógica da atualização compreende a seguinte sequência lógica de passos:

¹ Será explicado posteriormente

- Determine o mês/ano de processamento. Vamos considerar como exemplo o mês de **abril/2019 (04/2019)**
- Selecione todos os registros da tabela **MovDebCred** que correspondem ao mês **04/2019**.
- Certifique que os lançamentos realizados são consistentes, aplicando a rotina de realização da **crítica do movimento**. Enquanto erros forem apontados, faça a devida correção até que não haja qualquer inconsistência.
- Para cada conta distinta que recebeu lançamento de débito ou de crédito na tabela **MovDebCred**, totalize separadamente todos os valores de débito e crédito. É normal que uma conta receba 0 ou n lançamentos. Por exemplo:

Conta	Total de crédito	Total de Débito
11010100100	70.00	70.00
11010100200	40.00	20.00
...		

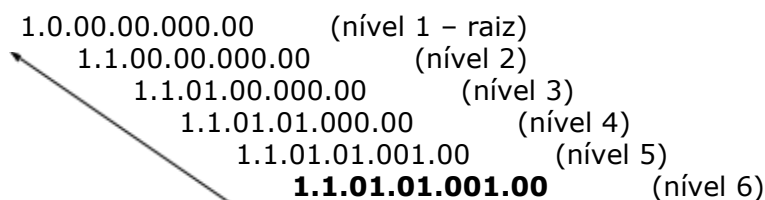
- Uma vez computado o total de crédito e débito de uma conta, replique esses totalizadores na tabela **DebCred**. A totalização mencionada, de acordo com as contas movimentadas, resulta na inserção dos seguintes:

Conta	MêsAno	Total de Débito	Total de Crédito
11010100100	042019	70.00	70.00
11010100200	042019	40.00	20.00
...			

- Determinados os totalizadores de débito e crédito da conta analítica que recebeu lançamentos no mês/ano em questão, é hora de propagar estes totalizadores nas contas superiores. Há uma hierarquia de níveis que determina a relação pai/filha entre contas. Por exemplo, a conta (formatada com pontos que atuam como separadores entre os níveis da conta)

1.1.01.01.001.01

tem o nível de profundidade **6**, sendo dependente da existência das contas de todos os seus níveis superiores.



Portanto, a totalização de crédito e débito da conta 1.1.01.01.001.00 implica na atualização automática (acrescentar o totalizador correspondente ao valor que já existe na conta superiora) dos totais de débito e crédito de todas as contas superiores, ou seja, da conta 1.1.01.01.001.00 até a conta 1.0.00.00.000.00. Esta deve ser uma tarefa **atômica**, isto é, tudo ou nada.

- Após o processamento de todas as contas, a atualização é encerrada.

7.5 A tabela Saldos

A tabela **saldos** só deve ser atualizada quando todos os meses de um ano de referência forem processados. Então, é necessário transportar o saldo final de todas as contas referentes ao mês de dezembro para o saldo inicial do ano seguinte. Se não houver registro correspondente, um novo será criado. Observe a ilustração para a conta 1.1.01.01.001.01:

Tabela **Saldo**:

Ano	Saldo Inicial
2017	300.00
2018	1000.00

Observação: o saldo inicial (2018) corresponde ao saldo final do ano de exercício anterior (2017), após as sucessivas atualizações.

Tabela **DebCred**:

Total	01/18	02/18	03/18	04/18	05/18	06/18	07/18	08/18	09/18	10/18	11/18	12/18
Cred	50.00	100.0	-	10.0	20.0	-	10.0	20.0		80.0	-	50.0
Deb	-	30.0	80.0	40.0	-	-			50.0	10.0	10.0	
<div><div>50.0</div><div>+50.0</div><div>+70.0</div><div>-80.0</div><div>-30.0</div><div>+20.0</div><div>-</div><div>+10.0</div><div>+20.0</div><div>-50.0</div><div>+70.0</div><div>-10.0</div><div>+</div></div>												

Saldo Final da conta 1.1.01.01.001.01 ao final de 2018:

1000.0 + (50.0 + 70.0 - 80.0 - 30.0 + 20.0 + 10.0 + 20.0 - 50.0 + 70.0 - 10.0 + 50.0 = **1120.0**

Ou seja, o valor de R\$ 1120.00 correspondente ao exercício 2018 deve ser transportado para a tabela saldo por meio da inserção de um registro contendo a conta, o ano seguinte (2019) e este saldo (que agora vai ser o inicial para 2019)

Tabela **Saldo**:

Ano	Saldo Inicial
2017	300.00
2018	1000.00
2019	1120.00

7. Tarefas

7.1 Trigger

- Criar um trigger na tabela **conta** que só permita adicionar uma nova conta se todas as suas contas superiores já estiverem na tabela.

7.2 Visão

- Criar uma visão que mostre, para cada conta distinta, as seguintes informações (ano corrente):

Conta | Ano | Saldo Inicial | Total de Crédito | Total de Débito | Saldo Final

7.3 Stored Procedure

- Implementar uma stored procedure que execute o processamento de **atualização do movimento mensal das contas**. A procedure deve receber como argumento o mês e ano para atualização.
- Implemente uma stored procedure que transporte o saldo final (dezembro/yyyy) de um ano que teve seu processamento encerrado para o ano posterior. O transporte de saldo deve ser feito para todas as contas do plano de contas
- Implementar uma stored procedure/stored function que realize a crítica do movimento de débito e crédito. A stored procedure deve sinalizar quando não houver inconsistência nos registros analisados.

7.4 Stored Function

- Criar uma função que receba uma conta como argumento e retorne seu respectivo dígito
- Criar uma função (que retorne uma tabela) que, para uma conta e um mês ano passado como argumento, retorne os valores (evoluídos) conforme ilustrado abaixo:

Conta 11010100100 (05/2018):

Saldo Anterior	Tot Crédito	Tot Débito	Saldo Atual
1.010,00	20,00	0,00	1.030,00

Reforçando: o **saldo anterior** é resultante do **saldo inicial da conta no início do ano + total de crédito de janeiro - total de débito de janeiro + total de crédito de fevereiro - total de débito de fevereiro + total de crédito de março - total de débito de março + total de crédito de abril - total de débito de abril**. Em outras palavras, o saldo anterior é evoluído de acordo com os meses anteriores.

O Total de crédito e débito do mês da consulta não é utilizado na evolução; eles serão coletados diretamente para determinar o saldo atual da conta para o final do mês de maio/2018 segundo o nosso exemplo.