

1 - O trabalho será realizado em grupo de até **3 alunos**;

* OBS: Para cada aluno a mais na formação do grupo corresponderá a -30% da nota.

2 - O valor do trabalho é de **30 pontos**;

3 - A data de entrega de TODOS os grupos será no dia **17/11/2022**;

* OBS: Para cada dia corrido de atraso na entrega do trabalho será descontado 50% da nota.

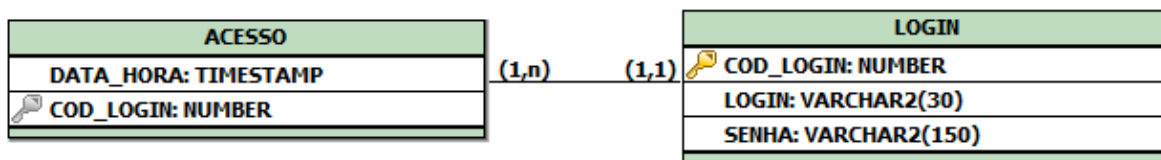
4 - O trabalho deverá ser entregue via **CLASSROOM**;

5 - Descrição do trabalho:

O trabalho consiste em criar rotinas que serão responsáveis por criptografar e descriptografar os valores armazenados no campo SENHA da tabela LOGIN.

Descrição:

Para esse trabalho o grupo deverá obedecer ao modelo de dados abaixo e para tal o grupo terá que criar os objetos correspondentes.



Regras de Negócio:

1 – Criar a função FN_CRIPTOGRAFIA() que receba como parâmetros a SENHA que será criptografada e o ACRÉSCIMO que será usado seguindo a regra de criptografia abaixo.

2 – Regra para criptografia:

A criptografia será realizada em 3 etapas.

2.1) a função deverá embaralhar os caracteres da senha usando uma matriz 3xN, ou seja, caso a senha usada seja **COTEMIG123** o resultado do embaralhamento será:

CEG3
OM1
TI2

➡

CEG3OM1TI2

2.2) os caracteres deverão ser convertidos para seu correspondente na tabela ASCII.

CEG3OM1TI2



067069071051079077049084073050

OBS: Perceba que todos os números tem 3 posições.

2.3) Deverá ser acrescido em cada número gerado o valor passado no parâmetro ACRESCIMO

Supondo que o valor de ACRESCIMO seja 3, teríamos:

070072074054082080052087076053

Esse número é que será registrado no campo SENHA da tabela LOGIN para o usuário em questão.

3 - Criar a função FN_DESCRITOGRRAFIA() que receba como parâmetros a SENHA criptografada e o ACRESCIMO, então desfaça a criptografia realizada com base no item 2.

4 – crie a procedure PR_INSERE_LOGIN() que receba como parâmetros 3 valores COD_LOGIN, LOGIN e SENHA, então a procedure deverá:

4.1) certificar que não haja valores de COD_LOGIN e LOGIN repetidos.

4.2) usar a função FN_CRIPTOGRAFIA() para embaralhar a senha passada pelo usuário.

DICA:

Crie uma variável para armazenar o momento do insert.

DATA_HORA TIMESTAMP;

DATA_HORA := SYSTIMESTAMP;

Use o comando abaixo como valor para o parâmetro (ACRESCIMO).

to_char(DATA_HORA,'FF1')

4.3) fazer a inserção do login na tabela LOGIN

4.4) fazer a inserção do acesso na tabela ACESSO, você deve saber que o valor de DATA_HORA será o mesmo da variável DATA_HORA acima.

5 – crie a procedure PR_VALIDA_LOGIN() que receba como parâmetros 2 valores LOGIN e SENHA, então a procedure deverá:

5.1) certificar que o LOGIN exista na tabela LOGIN.

5.2) usar a função FN_DESCRITOGRRAFIA() para desembaralhar a senha passada pelo usuário.

DICA:

O valor de acréscimo passado no parâmetro deve ser recuperado a partir do campo DATA_HORA da tabela ACESSO referente ao LOGIN informado pelo usuário.

5.3) caso as senhas sejam iguais, atualizar o valor do campo SENHA na tabela LOGIN usando o novo valor de ACRESCIMO (momento atual)

5.4) atualizar o campo DATA_HORA na tabela ACESSO com o mesmo valor de data usado para o ACRESCIMO.

5.5) emitir mensagem compatível.