2020-07-13 - Guilherme Lima

**OBJETIVO DA PROCEDURE:**

Está procedure tem o objetivo de realizar as seguintes operações com os usuários:

* Cadastrar.
* Editar.
* Apagar.

**ARGUMENTOS:**

Segue abaixo a relação com o significado de cada argumento que deve ser inserido para execução da procedure:

* **editarAdicionar:** Este argumento indica qual das ações possíveis será executada na chamada da procedure, os seus valores são:

1. Para cadastrar um novo usuário.
2. Para apagar um usuário existente.
3. Para editar um usuário existente.

* **nomeUsuario:** Ao nick ou nome utilizado no sistema para identificar o novo usuário, ou em caso de edição o novo nick ou nome do usuário já cadastrado.
* **nome**: Ao nome da pessoa a se tornar um novo usuário, ou em caso de edição o nome da pessoa que já é um usuário com as alterações.
* **email**:Ao email da pessoa a se tornar um novo usuário, ou em caso de edição o email do usuário cadastrado com as alterações.
* **tipoUsuario**: A um dos tipos de usuário contidos em Nuvem\_Tipo\_Usuarios, em caso de EXCLUSÃO de um usuário teve conter uma string vazia.
* **senha:** A senha para o novo usuário, que poderá ser trocada pelo mesmo posteriormente no site, ou uma nova senha para um usuário já cadastrado.
* **qtdeUnidades**: Esse é um argumento utilizado para auxiliar os algoritmos empregados nesta procedure,e deve conter (sempre) o número total de códigos de unidades que está sendo enviado no argumento (unidades), ou ser um zero caso esteja acontecendo uma EXCLUSÃO de usuário.
* **unidades:** Esse argumento vai simular uma lista ou array uma vez que MYSQL não possui essa estrutura, deve então ser feita uma string semelhante ao exemplo:
  + '1,2,3,4,5,6' - Cada um dos números na string se refere ao código de uma unidade cadastrada em Nuvem\_Unidades\_Clientes e como no exemplo o'que deve separá las é uma vírgula (,).

Outra característica deste argumento é como devem ser organizados os dados dentro dele, considerando que podem ser feitos CADASTROS, EDIÇÕES E EXCLUSÕES segue exemplos para cada caso respectivamente:

* + '1,2,10,16' - Note que a string a esquerda não possui valores negativos pois todos os códigos serão vinculados ao novo usuário.
  + '3,4,-10,-2' - Agora nas EDIÇÕES deve ser seguida a seguintes regras:
    - Unidades a serem vinculadas devem estar localizadas a esquerda da string como sendo números positivos.
    - Unidades a serem desvinculadas devem estar direita da string com sinal de negativo (-) para identificação no algoritmo.
    - Para o caso de somente desvincular todos os códigos devem conter o sinal de negativo (-).
    - Para o caso de somente vincular todos os códigos devem estar positivo.
  + ' ' - Caso for feito uma EXCLUSÃO de um usuário esta string deve ser vazia, porém deve existir.
* **codUsuarioCadastrado:** Este argumento deve conter o código do usuário existente que está a sofrer alterações seja de qualquer dos seus dados ou vínculos a unidades existentes na base de dados.

Quando for feito um novo cadastro este argumento deve estar com uma string vazia.

**BLOCOS LÓGICOS:**

**editarAdicionar = 1:**

1. Será feito um INSERT em Nuvem\_Usuarios com todos os dados do mesmo informados na página de cadastros pelo usuário que está realizando o cadastro.
2. Uma variável é definida como @codUsuario contendo um SELECT do código do novo usuário cadastrado a partir do email.
3. Um loop WHILE iniciará verificando se o argumento unidades é maior que zero, indicando assim que tem códigos de unidades que precisam ser vinculados ao novo usuário.
4. Para cada código de unidade será feito um insert em Nuvem\_Usuarios\_Unidades incluindo o código do novo usuário ao código da unidade que é sempre o último item contido na string (unidades) que simula uma lista utilizando o método SUBSTRING\_INDEX(unidades, ',',-1) que fatia a string em cada caractere que for uma vírgula e pega o último elemento.
5. O argumento que contém a lista de códigos de unidades deve ser atualizado retirando o último item desta lista uma vez que ele já foi vinculado ao novo usuário, para isso foi utilizado o método: set unidades=SUBSTRING\_INDEX(unidades,',',qtdeUnidades-1);
6. O argumento qtdeUnidades também deve ser atualizado uma vez que a quantidade de unidades a serem vinculadas ao usuário foi decrementada em 1

para isso utiliza-se o seguinte método: set qtdeUnidades=qtdeUnidades-1;

**editarAdicionar = 2:**

1. É feito um DELETE de todas as unidades vinculadas ao usuário que será excluído na operação.
2. É feito um DELETE de todos os dados do usuário contidos em Nuvem\_Usuários.

**editarAdicionar = 4:**

1. Inicia-se um loop WHILE verificando o argumento unidades se é maior que 0 indicando que existem códigos de unidades a serem editados.
2. A condição para ser executado o seguinte bloco é a string contida em unidades possuir códigos positivos e negativos, desta maneira deve conter elementos no final da lista com sinal de (-) e elementos no início da lista sem o sinal de (-).

O motivo para existir tão bloco de algoritmo se deve ao desempenho alcançado no processamento do banco de dados de tais informações, pois se a lista unidades cumprir tal condição isso significa que terá vínculos a serem desfeitos e novos vínculos a serem feitos, se fossem feitos com INSERT e DELETE somente, teriamos uma queda considerável de desempenho pois ao deletar e ao inserir dados de uma tabela os índices da mesma devem ser atualizados, então neste caso é feito um UPDATE para que não tenha que ocorrer uma atualização de índice desnecessária.

1. É então feito um UPDATE no registro que contém a unidade a ser desvinculada o código do usuário cadastrado e, colocando o novo código de unidade para vínculo como no exemplo a seguir: unidades = (1,2,3,4,-10,-12,-6) neste caso o código 1 entraria no lugar do 6.
2. É então atualizado os argumentos unidades e qtdeUnidades uma vez que o último e o primeiro elemento de unidades passaram pelo processo de atualização, para isso é feito o seguinte:

* Atualiza o argumento unidade retirando o último item, seguindo o exemplo ficaria assim:

set unidades = (1,2,3,4,-10,-12,-6)

set unidades = SUBSTRING\_INDEX(unidades,',',qtdeUnidades-1);

Este método retorna todos os elementos sem o último.

unidades = (1,2,3,4,-10,-12).

* Então é feita uma atualização em qtdeUnidades decrementado 1 indicando assim que qtdeUnidades corresponde a quantidade exata de códigos de unidades a serem alterados.

set qtdeUnidades = qtdeUnidades-1;

* Em seguida o argumento unidade é novamente atualizado desta vez com todos os elementos com exceção do primeiro elemento set unidades = (1,2,3,4,-10,-12). set unidades = SUBSTRING\_INDEX( unidades,',',qtdeUnidades-qtdeUnidades-qtdeUnidades+1); unidades = (2,3,4,-10-12)

* E por fim atualiza o argumento qtdeUnidades decrementando 1 de seu número total pois a lista unidades perdeu dois dos elementos. set qtdeUnidades = qtdeUnidades-1;

1. A condição para o seguinte bloco lógico é o último elemento de unidades conter sinal negativo:
2. É deletado o registro que contém o código de unidade da vez a ser desvinculado do usuário.
3. É atualizado a variável unidades com o seguinte método: set unidades = SUBSTRING\_INDEX(unidades,',',qtdeUnidades-1) que retorna todos os itens com exceção do último.
4. É atualizado o argumento qtdeUnidades pois unidades já não possui a mesma quantidade de códigos.

1. A condição para o seguinte bloco lógico é o último elemento de unidades não conter sinal negativo:
2. É inserido um registro em Nuvem\_Usuarios\_Unidades vínculando o código de usuário ao código de unidade da vez(loop).
3. É atualizado o argumento unidades retirando o último elemento.
4. É atualizado o argumento qtdeUnidades pois unidades não possui o mesmo número de códigos de unidade.
5. Ao finalizar o loop se o mesmo ocorrer, é então feito um UPDATE em Nuvem\_Usuarios com todos os dados informados na página

web tais como nome, nome usuario, tipo etc...