**Atividade Extra - Triggers**

Neste momento iniciamos a parte intermediária do módulo de Oracle, iremos trabalhar com triggers (gatilhos). A utilização de triggers faz com que ações sejam executadas antes ou depois de um comando: INSERT, UPDATE ou DELETE, triggers não funcionam com o comando SELECT.

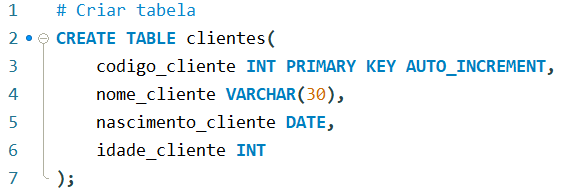
Ao criar uma trigger temos que especificar o momento que ela será executada, podendo ser:

* Before: Antes de realizar a ação SQL;
* After: Depois de realizar a ação SQL.

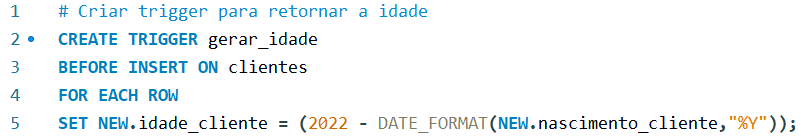
**Importante! Temos algumas restrições ao utilizar o before e o after:**

* Before: Você pode alterar ou cadastrar algum valor antes de inserir na tabela. Valores que já estejam cadastrados na tabela não conseguem ser alterados. É interessante utilizar o before para validar ações, exemplo:
  + Antes de realizar um cadastro, verificar se o e-mail informado já existe.
  + Alterar um CPF, apenas se for válido e não pertencer a outro cliente registrado.
* After: Essa expressão funciona apenas em informações já cadastradas em tabelas. Utilize o after quando precisar realizar um cadastro, alteração ou remoção em uma outra tabela, exemplo:
  + Após cadastrar uma venda, altere a quantidade de um determinado produto em outra tabela.
  + Após realizado um cadastro, dispare um e-mail dando boas vindas para o novo usuário.

Vamos criar um pequeno exemplo, onde seja possível realizar o auto incremento de uma tabela, inicialmente crie uma tabela chamada **clientes**:



O próximo passo é criar uma trigger para automaticamente retornarmos a idade, utilizaremos uma lógica simples, onde será pego o ano de 2022 e será subtraído pelo ano de nascimento do cliente, veja a estrutura da trigger:



Inicialmente criamos uma trigger chamada **gerar\_idade**, na segunda linha utilizamos o comando **BEFORE INSERT** para informar que antes de realizar a inserção, será disponibilizado na coluna **idade\_cliente** a idade desse cliente que será registrado.

Na quarta linha temos o comando **FOR EACH ROW**, utilizamos essa expressão caso seja realizado uma inserção múltipla, exemplo:

INSERT INTO clientes (codigo\_cliente, nome\_cliente, nascimento\_cliente) VALUES

(1, "Ana", "2002-03-04"),

(2, "Bianca", "2007-16-10"),

(3, "Caio", "1995-01-22");

Já na quinta linha, como ainda não efetuamos o registro do cliente, utilizamos o comando **NEW** para manipular o valor informado.

Trigger criada, podemos cadastrar um cliente:



Em seguida podemos selecionar:





Exercícios

1. Criar uma tabela **alunos** com a seguinte estrutura:
   * codigo\_aluno - Int e PK
   * nome\_aluno - Varchar(30) Not Null
   * nota1\_aluno - Double Not Null
   * nota2\_aluno - Double Not Null
   * media\_aluno - Double
   * situacao\_aluno - Varchar(15)
2. Criar uma trigger para gerar a média do aluno automaticamente.
3. Com base na média do aluno, faça uma trigger para retornar a situação do aluno. Caso a média seja 7 ou superior, estará aprovado(a) e caso contrário reprovado(a).  
   [Clique aqui](https://www.techonthenet.com/mysql/loops/if_then.php) para ver como implementar uma condicional em uma trigger.
4. Exclua as triggers e em seguida as tabelas.
   * Para excluir uma tabela: **DROP TABLE** **tabela**;
   * Para excluir uma trigger: **DROP TRIGGER trigger**;
5. Criar uma tabela chamada **produtos** contendo a seguinte estrutura:
   * codigo\_produto - Int AI, e PK
   * nome\_produto - Varchar(30) Not Null
   * valor\_produto - Double Not Null
   * valor\_promocional - Double
6. Criar uma trigger para cadastrar o **valor\_promocional**, deverá ser cadastrado o valor do produto com 10% de desconto. Sempre que o usuário informar o produto e seu valor, deverá ser registrado: código, nome do produto, valor do produto e valor promocional.
7. Realize uma inserção, informando o nome do produto e o valor.
8. Remova a trigger e a tabela.
9. Criar uma tabela chamada **produtos** com a seguinte estrutura
   1. codigo\_produto - Int AI e PK
   2. nome\_produto - Varchar(30)
   3. segmento\_produto - Varchar(20)
10. Crie uma outra tabela chamada **qtd\_segmentos** com a seguinte estrutura:
    1. nome\_segmento - Varchar(30)
    2. quantidade\_segmento - Int
11. Cadastre três segmentos e informe a quantidade zero na tabela **qtd\_segmentos**.
12. Crie uma trigger para cadastro, quando efetuado o cadastro de um novo produto (nome do produto e o segmento), a tabela **qtd\_segmento** deverá incrementar aquele determinado segmento.
13. Remove a trigger e as tabelas.