SQL/DQL – *Stored Procedures* (Procedimentos armazenados)

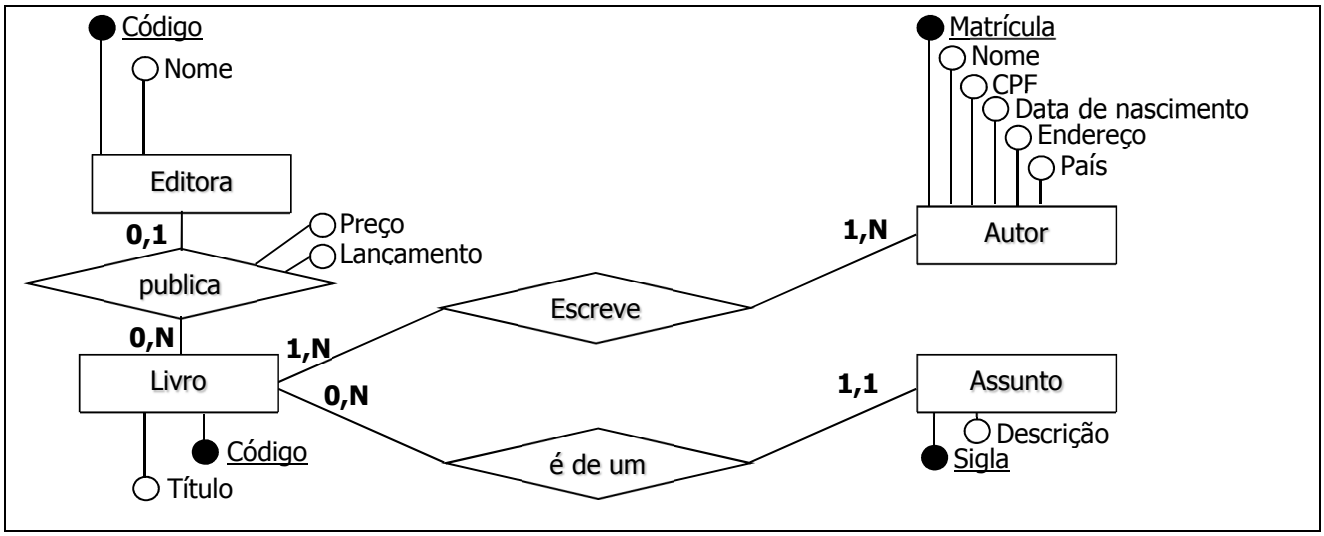
- *Stored Procedure* é um conjunto de instruções escrito numa linguagem própria para procedures (procedimentos) e *triggers* (gatilhos) do SGBD, e que é armazenado como parte do banco de dados.
- *Stored Procedures* reduzem o tráfego na rede, pois são executadas pelo SGBD na máquina servidora de banco de dados.
- *Stored Procedures* não permitem instruções DDL.
- *Triggers* são quase a mesma coisa que *Stored Procedures*, exceto pelo modo como são chamadas (e mais alguns pequenos detalhes):
  - *Stored Procedures* podem ser chamadas por aplicações cliente, outras *Stored Procedures* ou *triggers*.
  - *Triggers* são chamados automaticamente quando uma alteração em uma linha da tabela em questão ocorre.

Sintaxe básica de Stored Procedure:

```
CREATE or ALTER PROCEDURE <NomedaProcedure>
        ( <parâmetros de entrada> )
RETURNS
        ( <parâmetros de saída> )
AS
        <declaração de variáveis locais>
BEGIN
        <comandos da procedure>
END
```

Parâmetros de entrada:	Valores iniciais, que servem para estabelecer o comportamento do procedimento (todos os tipos, exceto BLOB ou ARRAY).
Parâmetros de saída:	Valores que retornam os resultados desejados, executados pelo procedimento (idem ao acima).
Comandos da procedure:	Conjunto de instruções SQL/DML e DQL.

Um livro é escrito por um ou mais escritores, e os registram em uma única Editora, que pode editar nenhum, um ou vários livros. Porém, nem todos estes livros conseguem ser lançados, embora já estejam registrados. Cada livro está associado a um determinado assunto, e pode ser escrito por um ou vários autores, os quais, por sua vez, podem escrever tantos livros quantos queiram.



/*=====================

32.1.	Criar o *Stored Procedure*, SP_00, para executar as instruções acima, porém, para o autor, cujo código será informado via parâmetro.

====================== */

-- Antes, mostrar o nome, data de nascimento e país do autor, cujo código é 501:

```
select  nome, nascim, pais
from    Autor
where   matricula = 501
```

```
set term^;
CREATE or ALTER PROCEDURE SP_00
    (i_matricula    smallint)
RETURNS
    (o_nome     varchar (80),
     o_nascim   date,
     o_pais     char (02)
     )
AS
BEGIN
    select  nome, nascim, pais
    from    Autor
    where   matricula = :i_matricula
    INTO    :o_nome, :o_nascim, :o_pais;
    SUSPEND;
END^
commit;
```

---------- Executando...

```
EXECUTE PROCEDURE SP_00 (502);-- Experimente com outros códigos, válidos ou não...

SELECT  * from SP_00 (502);    -- Experimente com outros códigos, válidos ou não...
```

/*=====================

32.2.	Criar o Stored Procedure, SP_01, para mostrar os nomes, datas de nascimento e países dos autores do país, cuja sigla será informada.

```
=====================*/

        select  nome, nascim, pais
        from    Autor
        where   UPPER (pais) = UPPER ('br')
-----------------------------------------

        set term^;
        CREATE or ALTER PROCEDURE SP_01
            (i_pais    char (02))
        RETURNS
            (o_nome     varchar (80),
             o_nascim   date,
             o_pais     char (02)
             )
        AS
        BEGIN
          FOR
            select  nome, nascim, pais
            from    Autor
            where   UPPER (pais) = UPPER (:i_pais)
            INTO    :o_nome, :o_nascim, :o_pais
          DO
          SUSPEND;
        END^
        set term;^
        commit;
----------------------------------------- Executando...

        SELECT  * from SP_01 ('us');   -- Não encontra, pois deveria ser "US"
        SELECT  * from SP_01 ('US');   -- Agora, sim!
/*=====================
```

32.3.	Mostre os códigos e nomes dos livros, e nomes das editoras que os publicaram, porém somente dos livros publicados a partir de uma data informada como argumento de entrada no Stored Procedure

```
===================== */

-- 1º) Mostrar todos os dados dos livros lançados após uma determinada data

        Select   *
        from     livro
        where    lancamento > '2013/10/10'
-----------------------------------------

-- 2º) Idem, mas deve-se também mostrar o nome da editora correspondente

        Select   *
        From     Livro L INNER JOIN Editora E ON L.codedit = E.codedit
        Where    lancamento > '2013/10/10'
-----------------------------------------

-- 3º) Destes, mostrar somente o código e nome do livro, e nome da Editora

        Select   L.codlivro, L.titulo, E.nome
        From     Livro L INNER JOIN Editora E ON L.codedit = E.codedit
        Where    lancamento > '2013/10/10'
-----------------------------------------

-- 4º) Por fim, criar a SP_02, utilizando esta lógica...
```

```
set term^;
CREATE or ALTER PROCEDURE SP_02
  (i_data   date)
RETURNS
  (o_codlivro   smallint,
    o_editora   varchar (80),
    o_titulo      varchar (80))
AS
BEGIN
   FOR
        select  L.codlivro, L.titulo, E.nome
        from    Livro L INNER JOIN Editora E
                   ON L.codedit = E.codedit
        where   lancamento > :i_data

        INTO    :o_codlivro, :o_titulo, :o_editora

    DO
    SUSPEND;
END^
set term;^
commit;
```

-- Experimente, com "2013/10/10" e outras datas...

```
Select  * from SP_02 ('2013/10/10');
```

32.4.	Crie o Stored Procedure para mostrar os nomes, datas de nascimento dos autores de algum país

```
select  nome, nascim , pais
from    Autor
where   pais = 'BR'
```

```
set term^;
CREATE or ALTER PROCEDURE SP_Dados_Autor2
    (i_pais     varchar(30))
RETURNS
    (o_nome     varchar (80),
     o_nascim   date,
     o_pais     varchar (30)
     )
AS
BEGIN
    Select    nome, nascim, pais
    From      Autor
    Where     pais = :i_pais
    INTO      :o_nome , :o_nascim , :o_pais ;

    SUSPEND;
END^
set term;^
COMMIT;

select  *
from    SP_Dados_Autor2 ('BR');      -- Oops! Há mais que 1 Autor de 'BR'
```

-- Corrigir, pois há mais que 1 autor de 'BR'.

-- Usar o conjunto "**FOR**...**DO**..."

```
…
BEGIN
    FOR
    select  nome, nascim, pais
    from    Autor
    where   pais = :i_pais
    INTO    :o_nome , :o_nascim , :o_pais     -- Retirar o ";"!
    DO
    SUSPEND;
…

select  *
from    SP_Dados_Autor2 ('BR');
```

-- Experimente com 'us'

-- Faça funcionar com 'US', em qualquer caixa (maiúsculas ou minúsculas), sempre.

/* ============================

32.5.	Crie o procedimento "AVGLivro" para calcular o preço médio dos livros publicados pela editora, cujo nome será informado externamente

============================ */

-- 1º) Encontrar o preço médio dos livros publicados por determinada editora

```
select  avg(preco) from    Livro
where   codedit =
    (select codedit from Editora
      where nome = 'Marketing Books')
```

-- 2º) Criar o SP correspondente...

```
set term^;
CREATE or ALTER PROCEDURE AVGLivro
    (i_nomeEditora  varchar (80))
RETURNS
    (o_precomedio   numeric (10,2))
AS
BEGIN
    select  avg(preco) from    Livro
    where   codedit =
              (select codedit from Editora
              where UPPER (nome) = UPPER (:i_nomeEditora)
              )
        INTO    :o_precoMedio;
    SUSPEND;
END^
```

```
        set term;^
        commit;
---------------------------------------- Executando...

        SELECT  * from AVGLivro ('Marketing Books');
        SELECT  o_precomedio from AVGLivro ('Marketing Books');
/* ============================
```

32.6. Mostre os títulos, datas de lançamento e valores de todos os livros, cujos preços sejam superiores ao preço médio, encontrado no Procedimento "AVGLivro" */

```
============================ */

        Select      titulo, lancamento, preco
        from    Livro
        where   preco >
            (SELECT  o_precomedio from AVGLivro ('Marketing Books'))
----------------------------------------
        set term^;
        CREATE or ALTER PROCEDURE SP_03
          (i_nomeEditora    varchar (80))
        RETURNS
          (o_preco  numeric (10,2),
           o_titulo     varchar (80),
           o_lancamento date)
        AS
        BEGIN
            FOR
                Select      titulo, lancamento, preco
                from    Livro
                where   preco >
                (SELECT  o_precomedio from AVGLivro (:i_nomeEditora))
                INTO :o_titulo, :o_lancamento, :o_preco
            DO
            SUSPEND;
        END^
        set term;^
----------------------------------------
        SELECT  * from SP_03 ('Marketing Books');
/* ============================
```

32.7. 10. Crie o Stored Procedure "SP_Busca_Livros" para mostrar os códigos, títulos, datas de lançamento (se vazia, mostrar "-- não lançado") e preço, por ordem de preço decrescente, de todos os livros, cujos títulos contenham uma determinada palavra informada, em qualquer caixa, em qualquer posição do título.

```
   -- Sugestão: Faça o SELECT primeiramente...

============================*/
```