



Cursores

Prof. Fábio Procópio
fabio.procopio@ifrn.edu.br



2

Relembrando...

- Na [aula passada](#), falamos sobre a construção e a utilização de um recurso poderoso dos SGBDs: os gatilhos.
- Nesta aula, falaremos sobre um assunto chamado cursores, o qual consiste em um objeto que aponta para uma determinada linha dentro de um conjunto de dados. Vamos nessa?!





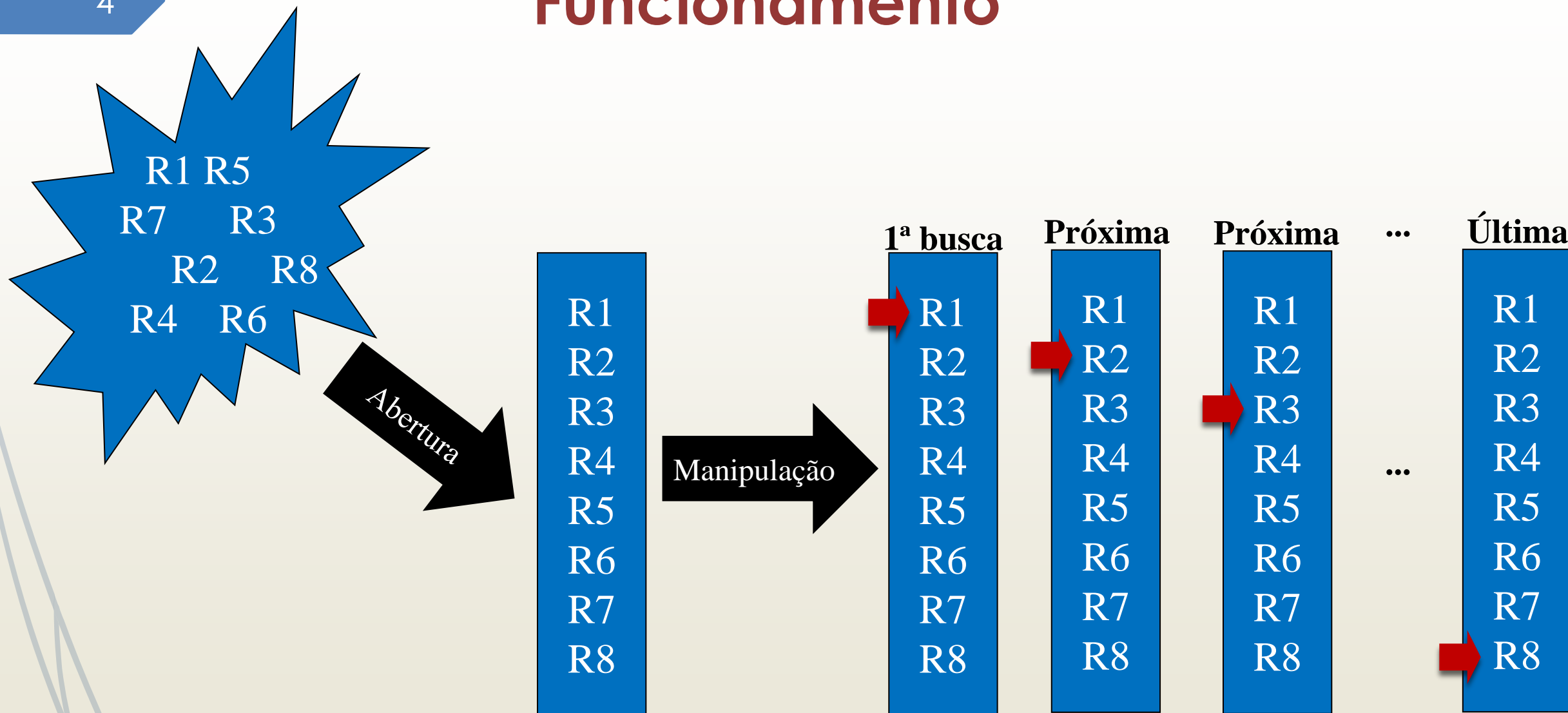
Introdução

- Um cursor funciona como um conjunto de resultados que mantém uma posição atual, à medida que o conjunto de resultados é percorrido;
- Exemplos:
 - Se você estivesse examinando nomes em um caderno de endereços, seu dedo atua como um cursor, um ponteiro para a entrada atual
 - Pense nas entradas na página atual do caderno de endereços como um conjunto de resultados de sua consulta
- Usar um cursor é como usar uma variável: 1) declare, 2) atribua um valor e, 3) use-o
- No entanto, ao contrário das variáveis locais, é preciso liberar explicitamente uma variável usada por um cursor e, depois destruí-la.



4

Funcionamento





Tipos – 1 de 3

➤ STATIC

- Registra uma “foto” dos dados especificados na instrução SELECT
- O armazena é feito no banco de dados tempdb
- Apenas leitura é permitida
- Pode ser: a) *forward-only* (apenas para frente) ou b) *scrollable* (rolável)

➤ KEYSET

- Armazena em tempdb apenas as colunas necessárias para identificar exclusivamente cada linha
- Linhas adicionadas às tabelas após a montagem do cursor não serão incluídas;
- Pode ser: a) *updatable* (atualizável), b) *read-only*, c) *scrollable* ou d) *forward-only*;



Tipos – 2 de 3

➤ DYNAMIC

- Comporta-se como se uma instrução SELECT fosse lançada novamente sempre que uma linha fosse referenciada;
- Os cursores dynamic refletem as alterações de valor tanto da associação quanto dos dados subjacentes, quer essas alterações tenham sido feitas pelo cursor ou por qualquer outro usuário;
- Possui algumas restrições:
 - a instrução SELECT só pode conter uma cláusula ORDER BY caso haja um índice contendo as colunas na cláusula ORDER BY.
 - Se uma cláusula ORDER BY não for apoiada por um índice, o SQL converterá o cursor para o cursor keyset



Tipos – 3 de 3

➤ FIREHOSE

- Declarado usando FAST_FORWARD, mas é mais conhecido como um cursor firehose.
- Possui duas restrições importantes. Se a instrução SELECT
 - referenciar colunas text, ntext ou image e contiver a cláusula TOP, o SQL Server converterá o cursor firehore em um cursor keyset.
 - combinar tabelas que contenham gatilhos com tabelas que não tenham gatilhos, cursor será convertido para um cursor static.



Sintaxe básica

```
DECLARE nome_cursor CURSOR
```

```
[LOCAL | GLOBAL]                                /*define o escopo do cursor*/
```

```
[FORWARD_ONLY | SCROLL]                        /*indica a rolagem a ser definida*/
```

```
[STATIC | KEYSER | DYNAMIC | FAST_FORWARD] /*indica o tipo de cursor*/
```

```
[READ_ONLY | SCROLL_LOCKS | OPTIMISTIC]        /*indica o tipo de bloqueio*/
```

```
[TYPE_WARNING] /*Envia mensagem se o cursor for convertido para outro tipo*/
```

```
FOR instrução_select                            /*indica as linhas que serão incluídas*/
```

```
/*Opcional. Por padrão, os cursores são atualizáveis a não ser que o  
parâmetro de bloqueio seja READ_ONLY. Especificam-se as linhas que podem  
sofrer atualização. Se omitidas, todas as colunas na instrução serão  
atualizáveis.*/
```

```
[FOR UPDATE [OF nomes_colunas]]
```




Operações

➤ Abertura

OPEN nome_cursor

➤ Fechamento

CLOSE nome_cursor

➤ Manipulação de linhas

FETCH nome_cursor

➤ **FETCH** pode ser combinado

➤ FETCH FIRST: retorna a primeira linha da variável

➤ FETCH NEXT: retorna a linha seguinte

➤ FETCH PRIOR: retorna a linha anterior

➤ FETCH RELATIVE n: retorna a linha n

➤ FETCH ABSOLUT n: especifica linhas antes da linha atual



Monitoramento – 1 de 2

➔ **@@CURSOR_ROWS**: retorna o número de linhas do último cursor aberto

Código	Descrição
-m	O cursor ainda não foi completamente preenchido
-1	O cursor é dinâmico e o número de linhas pode variar.
0	Uma das 3 opções: 1. Nenhum cursor foi aberto 2. Último cursor não foi fechado e liberado 3. Cursor não possui linhas
N	Número de linhas do cursor



Monitoramento – 2 de 2

- ➔ **@@FETCH_STATUS**: retorna informações sobre o último comando FETCH que foi lançado

Código	Descrição
0	FETCH realizado com sucesso
-1	FETCH falhou
2	Registro recuperado com sucesso



Exemplo 1

```
USE dbBiblioteca
GO

DECLARE crAutorObra CURSOR
FOR SELECT IdAutorObra, IdAutor, IdObra FROM TbAutorObra

OPEN crAutorObra

FETCH crAutorObra

CLOSE crAutorObra

DEALLOCATE crAutorObra
```



Exemplo 2

```
USE dbBiblioteca
GO

/*Variáveis para armazenar os valores retornados pelo cursor*/
DECLARE @IdAutorObra INT, @IdAutor SMALLINT, @IdObra INT

DECLARE crAutorObra CURSOR
FOR SELECT IdAutorObra, IdAutor, IdObra FROM TbAutorObra

OPEN crAutorObra

FETCH crAutorObra INTO @IdAutorObra, @IdAutor, @IdObra

/*Veja os valores armazenados nas variáveis*/
PRINT (@IdAutorObra)
PRINT (@IdAutor)
PRINT (@IdObra)

CLOSE crAutorObra
DEALLOCATE crAutorObra
```



Exemplo 3 – 1 de 2

```
/*SCROLL permite que o cursor pode rolar para frente e para trás*/  
DECLARE crAutorObra SCROLL CURSOR  
FOR SELECT a.NoPessoa AS Autor, o.NoObra AS Obra  
    FROM TbPessoa AS a  
    INNER JOIN TbAutorObra AS ao  
        ON a.IdPessoa = ao.IdAutor  
    INNER JOIN TbObra AS o  
        ON ao.IdObra = o.IdObra  
    ORDER BY a.NoPessoa  
  
OPEN crAutorObra
```



Exemplo 3 – 2 de 2

```
/*1ª linha*/  
FETCH NEXT FROM crAutorObra  
/*Última linha*/  
FETCH LAST FROM crAutorObra  
/*2ª linha a partir do topo do conjunto de dados*/  
FETCH ABSOLUTE 2 FROM crAutorObra  
/*3ª linha após a linha atual*/  
FETCH RELATIVE 3 FROM crAutorObra  
/*1ª linha*/  
FETCH FIRST FROM crAutorObra  
/*Linha seguinte, a partir da atual*/  
FETCH NEXT FROM crAutorObra  
/*Linha anterior, a partir da atual*/  
FETCH PRIOR FROM crAutorObra  
  
CLOSE crAutorObra  
DEALLOCATE crAutorObra
```



Exemplo 4

```
USE dbBiblioteca
GO

DECLARE crAutorObra CURSOR STATIC
FOR SELECT a.NoPessoa AS Autor, o.NoObra AS Obra
        FROM TbPessoa AS a
        INNER JOIN TbAutorObra AS ao
            ON a.IdPessoa = ao.IdAutor
        INNER JOIN TbObra AS o
            ON ao.IdObra = o.IdObra
        ORDER BY a.NoPessoa

OPEN crAutorObra

/*Retorna o número de linhas quando o cursor é do tipo STATIC ou KEYSET*/
PRINT ( @@CURSOR_ROWS )

CLOSE crAutorObra
DEALLOCATE crAutorObra
```




Exemplo 5 – 1 de 2

```
USE dbBiblioteca
GO

DECLARE @NoAutor VARCHAR(50), @NoObra VARCHAR(150)
DECLARE crAutorObra CURSOR STATIC
FOR SELECT a.NoPessoa AS Autor, o.NoObra AS Obra
      FROM TbPessoa AS a
      INNER JOIN TbAutorObra AS ao
            ON a.IdPessoa = ao.IdAutor
      INNER JOIN TbObra AS o
            ON ao.IdObra = o.IdObra
      ORDER BY a.NoPessoa, o.NoObra

OPEN crAutorObra
```



Exemplo 5 – 2 de 2

```
/*Preenche as variáveis com o primeiro registro do cursor*/  
FETCH crAutorObra INTO @NoAutor, @NoObra  
  
/*Percorre o cursor linha a linha, a partir da segunda*/  
WHILE @@FETCH_STATUS = 0  
    BEGIN  
        PRINT (@NoAutor + ' => ' + @NoObra)  
        FETCH crAutorObra INTO @NoAutor, @NoObra  
    END  
  
CLOSE crAutorObra  
DEALLOCATE crAutorObra
```



Exercício de fixação

1. Modifique o código do Exemplo 5 para que a listagem fique agrupada por autor, como a figura abaixo:

Brown, Dan:

La Fortaleza Digital

Ponto de Impacto

Coronel, Carlos:

Sistemas de Banco de Dados

Finlan, Christopher:

Professional Microsoft SQL Server 2016

Harrington, Roger:

ESCAPE FROM ALCATRAZ: True Stories of Jail Breaks and Great Escapes

How to get Away with Murder

Manzano, José:

Microsoft Essencial Windows 7 Professional

Introdução à Linguagem Lua

Programação Assembly: Padrão IBM-PC 8086/8088



Exercício de fixação

2. O sistema de biblioteca (dbBiblioteca) necessita adicionar, em todas as tabelas, um campo chamado InSituacao, do tipo TINYINT, o qual indica se o registro está ativo (1) ou inativo (0).

Construa uma rotina, usando cursores, para criar essa coluna. Após essa manutenção, ou seja, a criação do campo InSituacao, deve ser atribuído o valor 1 a essa nova coluna.



Referências

1. MICROSOFT. **DECLARE CURSOR (Transact-SQL)**. Disponível em: <https://docs.microsoft.com/pt-br/sql/t-sql/language-elements/declare-cursor-transact-sql?view=sql-server-ver15>. Acessado em: 16 nov. 2019.
2. MICROSOFT. **FETCH (Transact-SQL)**. Disponível em: <https://docs.microsoft.com/pt-br/sql/t-sql/language-elements/fetch-transact-sql?view=sql-server-2017>. Acessado em: 16 nov. 2019.
3. CRESPI. **Cursores no SQL Server**. Disponível em: <https://www.devmedia.com.br/cursores-no-sqlserver/5590>. Acessado em: 16 nov. 2019.
4. TECHFUNDA. **Scroll cursor example in SQL Server**. Disponível em: <http://techfunda.com/howto/913/scroll-cursor-example>. Acessado em: 17 nov. 2019.
5. DRAUSIO. **PD - Criar Cursor (Create Cursor) no SQL Server 2008 (Debug Passo a Passo)**. Disponível em: <https://www.youtube.com/watch?v=US31W2mRdr0>. Acessado em: 17 nov. 2019.