

BANCO DE DADOS

CURSORES

RICARDO SONAGLIO ALBANO



## CURSORES

- Ao executamos um comando SELECT, UPDATE ou DELETE em uma tabela do banco de dados, são acessados vários registros de uma só vez;
- O número de registros retornados depende do tamanho da tabela e da forma com que são buscados em conjunto com a cláusula WHERE, que realiza uma filtragem nos dados selecionados.
- As operações com conjuntos de linhas é uma das características que define um banco de dados relacional;



## **CURSORES**

- Um conjunto pode ser vazio ou poderá conter uma única linha, mas ainda assim será um conjunto;
- Existem situações em que trazer os registros de uma só vez não é conveniente ou possível para realizar certos tipos de operações, onde é necessário obter resultado de cada linha uma a uma;
- Nestes casos os SGBD's fornecem um recurso bastante interessante chamado CURSOR.



## CURSORES

- É uma instrução SELECT que será acessada linha a linha através de um laço WHILE e alguns comandos específicos para cursores;
- Utilizado normalmente em procedimentos armazenados;
- Permitem a realização de operações sobre linhas individuais de um resultado.
- Podem executar operações como: atualizar, excluir ou mover dados;



## **CURSORES**

- São criados usando o comando DECLARE CURSOR;
- Algumas Características dos cursores:
  - Capacidade de refletir alterações em dados subjacentes;
  - Capacidade de rolar pelo conjunto de linhas;
  - Capacidade de atualizar o conjunto de linhas.



## CURSORES

- O conjunto de linhas para o qual um cursor aponta é definido pelo comando SELECT.;
- Algumas restrições SELECT ao se criar um cursor T-SQL:
  - Não poderá retornar vários conjuntos de linhas;
  - Não poderá conter a cláusula INTO para criar uma nova tabela;
  - Não poderá conter a cláusulas COMPUTE ou COMPUTE BY;
  - Poderá conter funções agregadas, tais como AVG.



## **CURSORES**

- Etapas para utilização do cursor:
  - 1) Declaração do cursor;
  - 2) Abertura do cursor;
  - 3) Execução do cursor;
  - 4) Fechamento do cursor;
  - 5) Dessalocar o cursor.



## CURSORES

- Declarando um Cursor:

DECLARE **NOME\_CURSOR** CURSOR

1. [ LOCAL | GLOBAL ]
2. [ FORWARD\_ONLY | SCROLL ]
3. [ STATIC | KEYSET | DYNAMIC | FAST\_FORWARD ]
4. [ READ\_ONLY | SCROLL\_LOCKS | OPTIMISTIC ]
5. [ TYPE\_WARNING ]
6. FOR select\_statement
7. [ FOR UPDATE [ OF column\_name [ ,...n ] ] ]



## CURSORES

- Opções da declaração do cursor:
  - **LOCAL**: O escopo do cursor é local;
  - **GLOBAL**: O escopo do cursor é global para a conexão;
  - **FORWARD\_ONLY**: Só pode ser rolado da primeira à última linha. FETCH NEXT é a única opção de busca com suporte.
  - **SCROLL**: Todas as operações de movimentação poderão ser realizadas. Especifica que todas as opções de busca (FIRST, LAST, PRIOR, NEXT, RELATIVE, ABSOLUTE).



## CURSORES

- FETCH - Leitura das linhas de cursor:
  - Realiza a movimentação em um cursor, permitindo percorrê-lo linha a linha.
  - Sintaxe:
  - `FETCH [ [NEXT | PRIOR | FIRST | LAST |`
  - `ABSOLUTE n | RELATIVE n ] FROM ] nome_do_cursor`
  - `[INTO @variável1, @variavel2, ...]`



## CURSORES

- FETCH - Leitura das linhas de cursor:
  - NEXT: move para a próxima linha do cursor ou para a primeira;
  - PRIOR: move para a linha anterior;
  - FIRST: move para a primeira linha;
  - LAST: move para última linha;
  - ABSOLUTE n: move para a linha de posição n no cursor (se for positivo, a contagem inicia na primeira linha, se negativo, na última);
  - RELATIVE n: move para n linhas para frente ou para trás.



## CURSORES

- FETCH - Leitura das linhas de cursor:
  - INTO *@variavel1[, @variavel2...]*: permite associar cada coluna do cursor a uma variável declarada;
  - Cada variável listada no comando FETCH deverá estar relacionada a uma coluna do cursor;
  - A variável deve possuir o mesmo tipo da coluna, não sendo realizadas conversões implícitas.



## **CURSORES**

- **FETCH** - Leitura das linhas de cursor:
  - Para ler todo o cursor, ou seja, todas as linhas que estão armazenadas utiliza-se o comando **WHILE**;
  - A variável **@@FETCH\_STATUS** retorna informações sobre o último comando **FETCH** que foi lançado:
    - 0: O **FETCH** foi realizado com sucesso;
    - -1: O **FETCH** falhou;
    - -2: O registro trazido foi perdido.



## CURSORES

- Exemplo:
  - Ler a tabela cidade e mostrar todas as cidades com os respectivos estados, mostrando a seguinte mensagem:

A cidade de Curitiba - Pertence ao estado: Paraná

Comando:

```
SELECT c.nome, e.nome from cidades c, estados e  
WHERE c.estado = e.sigla
```



## CURSORES

- Resolvendo o exemplo:
- Etapas de um cursor;
  - 1) Declaração do cursor;
  - 2) Abertura do cursor;
  - 3) Execução do cursor;
  - 4) Fechamento do cursor;
  - 5) Desalocar o cursor



## CURSORES

- Resolvendo o exemplo:

```
open exemplo
```

```
declare exemplo cursor local forward_only for
```

```
    select c.nome, e.nome from cidades c, estados e where  
    c.estado = e.sigla
```

```
open exemplo
```

```
declare @cidade char(50), @estado char(50);
```

```
fetch exemplo into @cidade, @estado;
```



## CURSORES

- Resolvendo o exemplo:

```
while @@fetch_status = 0
```

```
Begin
```

```
    print = 'a cidade de ' + ltrim(@cidade) + ' – pertence ao estado: ' +  
    ltrim(@estado);
```

```
    fetch next from exemplo into @cidade, @estado;
```

```
End
```

```
close exemplo;
```

```
deallocate exemplo;
```



**Obrigado por sua atenção!**

Ricardo Sonaglio Albano