

Cursor é uma alternativa do ORACLE para manipular um conjunto de registros retornados por um comando SELECT de várias linhas.

Isto, vem como resposta a limitação de se trabalhar com uma variável "simples" que só recebe um único valor por vez.

A documentação da ORACLE recomenda: "Use cursores para processar individualmente cada linha retornada por uma instrução SELECT de várias linhas."

O conjunto retornado por uma consulta de várias linhas é chamado conjunto ativo e seu tamanho é o número de linhas que atende aos critérios da pesquisa. É função do cursor permitir que o usuário/programa processe o conjunto de linhas uma de cada vez.

De certa forma, um programa PL/SQL realiza os passos abaixo nessa ordem:

- 1- abre um cursor
- 2- processa linhas retornadas por uma consulta
- 3- fecha o cursor.

Você deve saber que a linha corrente do conjunto retornado pelo SELECT é marcada por um ponteiro.

Funções do cursor explícito:

- Pode processar além da primeira linha retornada pela consulta, linha por linha
- Controla que linha está sendo processada no momento
- Permite que o programador controle as linhas manualmente no bloco PL/SQL

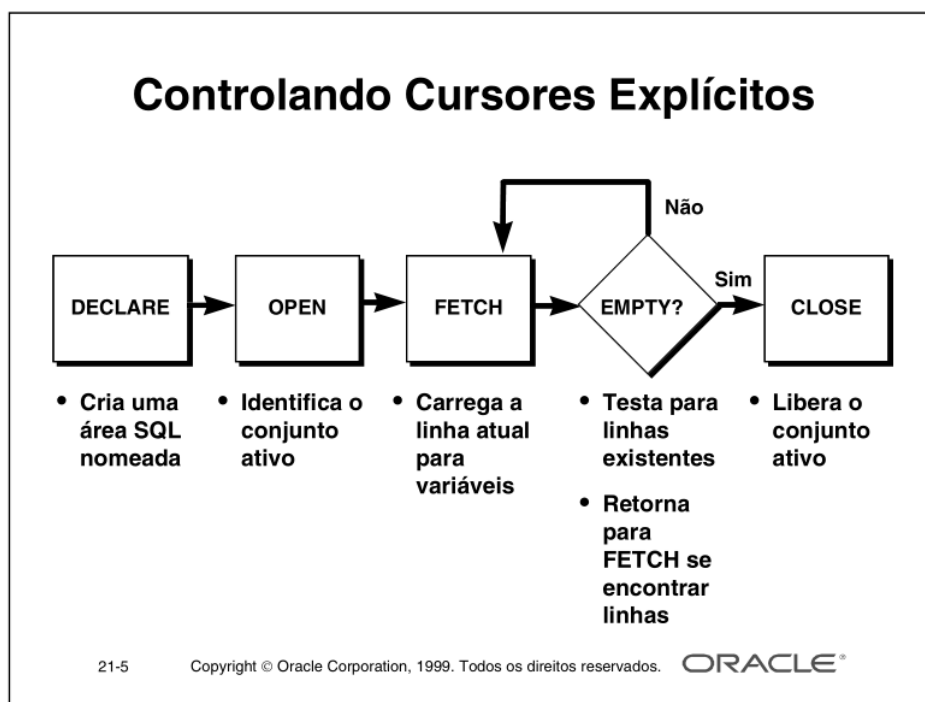


Figura 1 – Etapas do uso do cursor

A criação do cursor acontece na cláusula de declaração dos programas e obedece a sintaxe:

**CURSOR** **nome\_cursor** IS **comando\_select**;

onde:

**CURSOR** é o comando de criação;

**nome\_cursor** é nome que identificará o cursor

**IS** é a cláusula de ligação com o comando select

**comando\_select** é um comando select qualquer sem a cláusula INTO.

**ATENÇÃO:** caso seja necessário que o conjunto de linhas retornados pelo comando select obedeça a uma ordem específica a cláusula **ORDER BY** deverá ser usada.

## Declarando o Cursor

### Exemplo

```
DECLARE
  CURSOR emp_cursor IS
    SELECT empno, ename
    FROM   emp;

  CURSOR dept_cursor IS
    SELECT *
    FROM   dept
    WHERE  deptno = 10;
BEGIN
  ...
```

21-8

Copyright © Oracle Corporation, 1999. Todos os direitos reservados. ORACLE®

Você pode controlar o status do seu cursor com os atributos abaixo:

## Atributos do Cursor Explícito

Obter informações de status sobre um cursor.

Atributo	Tipo	Descrição
%ISOPEN	Booleano	Será avaliado para TRUE se o cursor estiver aberto
%NOTFOUND	Booleano	Será avaliado para TRUE se a extração mais recente não retornar uma linha
%FOUND	Booleano	Será avaliado para TRUE se a extração mais recente não retornar uma linha; complemento de %NOTFOUND
%ROWCOUNT	Número	Será avaliado para o número total de linhas retornadas até o momento

21-14

Copyright © Oracle Corporation, 1999. Todos os direitos reservados. ORACLE®

O exemplo abaixo mostra o uso de um cursor para calcular o total de alunos por turma. Repare que todas as etapas mostradas na figura 1 estão presentes no código.

```
set serveroutput on;
declare
--Na linha abaixo o cursor cr_alunos é criado com o conteúdo do
--comando SELECT NOME_ALUNO, COD_TURMA FROM TB_ALUNOS;
cursor cr_alunos is select nome_aluno, cod_turma from tb_alunos;

-- Na linha abaixo a variável LINHA é criada com o mesmo tipo do cursor cr_alunos
-- ou seja com duas colunas (NOME_ALUNO, COD_TURMA) e mesmo tipo (VARCHAR2(30), NUMBER)
respectivamente.
linha cr_alunos%rowtype;

total_turma1 number:=0;
total_turma2 number:=0;
begin
  -- na linha abaixo o cursor cr_alunos é aberto
  open cr_alunos;

  -- na linha abaixo o LOOP é usado para percorrer TODO o cursor
  loop

    -- agora será usado o comando FETCH para pegar o conteúdo da linha corrente do cursor
    -- e jogá-lo para dentro da variável LINHA
    -- em seguida o ponteiro do cursor passa para próxima linha
    fetch cr_alunos into linha;

    -- o LOOP será encerrado quando não houver mais linhas a serem percorridas
    exit when cr_alunos%notfound;
    if (linha.cod_turma = 1)
      then total_turma1:=total_turma1+1;
      else total_turma2:=total_turma2+1;
    end if;
  end loop;
  -- na linha abaixo o cursor cr_alunos é fechado.
  close cr_alunos;
  --exibição do resultado
  dbms_output.put_line('O total de alunos na turma 1 e: '||total_turma1);
  dbms_output.put_line('O total de alunos na turma 2 e: '||total_turma2);
end;
/
```

O uso do cursor também pode ser feito diretamente dentro da estrutura FOR. Perceba abaixo que nessa modalidade não é necessário se preocupar com declaração de variável, abertura do cursor, da execução dos fetchs e nem o fechamento do cursor.

O exemplo abaixo faz exatamente a mesma coisa do exemplo anterior.

```
set serveroutput on;
```

```
declare
```

```
--Na linha abaixo o cursor cr_alunos é criado com o conteúdo do
```

```
--comando SELECT NOME_ALUNO, COD_TURMA FROM TB_ALUNOS;
```

```
cursor cr_alunos is select nome_aluno, cod_turma from tb_alunos;
```

```
--variáveis que receberão a quantidade de alunos
```

```
total_turma1 number:=0;
```

```
total_turma2 number:=0;
```

```
begin
```

```
-- a linha abaixo mostra a chamada do cursor cr_alunos diretamente pelo for
```

```
-- nesse caso ele vai ler todas os registros do cursor até o seu final, isto porque não foi colocada uma  
condição de saída (EXIT).
```

```
for x in cr_alunos loop
```

```
  if (x.cod_turma = 1)
```

```
    then total_turma1:=total_turma1+1;
```

```
    else total_turma2:=total_turma2+1;
```

```
  end if;
```

```
end loop;
```

```
--exibição do resultado
```

```
dbms_output.put_line('O total de alunos na turma 1 e: '||total_turma1);
```

```
dbms_output.put_line('O total de alunos na turma 2 e: '||total_turma2);
```

```
end;
```

```
/
```