

Aula 09

Cursores

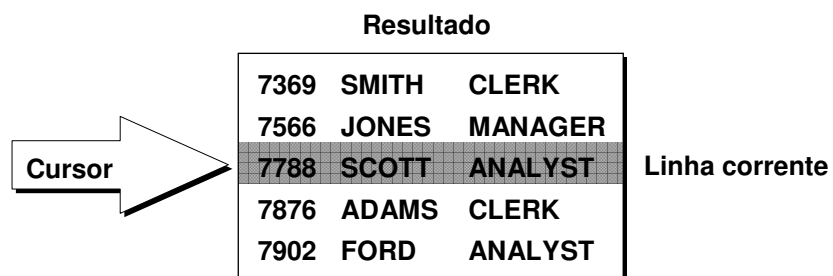
Banco de Dados - Computação - UNISUL

Cursores

Todo comando executado pelo Oracle Server tem um cursor individual associado a ele:

- **Cursores implícitos: declarado por todo comando DML e SELECT do PL/SQL.**
- **Cursores explícitos: declarado pelo programador.**

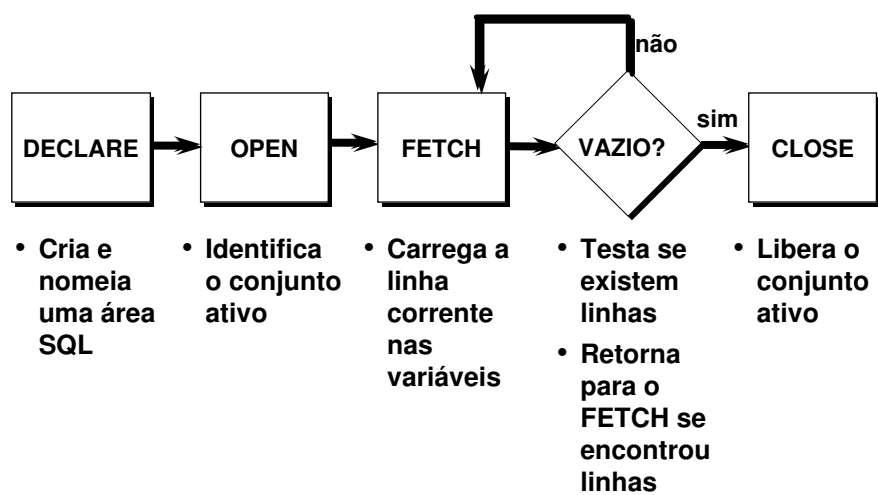
Funções de Cursor Explícito



3

Banco de Dados - Computação - UNISUL

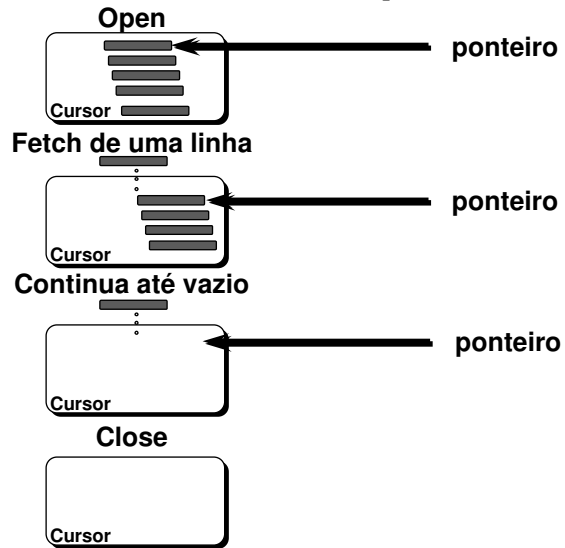
Controlando um Cursor Explícito



4

Banco de Dados - Computação - UNISUL

Controlando um Cursor Explícito



5

Banco de Dados - Computação - UNISUL

Declarando o Cursor

Sintaxe

```
CURSOR nome_cursor IS  
    comando_select;
```

- Não incluir a cláusula INTO na declaração do cursor.
- Se for necessário uma ordem específica de registros, pode ser usado a cláusula ORDER BY.

6

Banco de Dados - Computação - UNISUL

Declarando o Cursor

Exemplo

```
DECLARE
  CURSOR c1 IS
    SELECT empno, ename
    FROM   emp;

  CURSOR c2 IS
    SELECT *
    FROM   dept
    WHERE  deptno = 10;
BEGIN
  ...
```

Abrindo o Cursor

Sintaxe

```
OPEN nome_cursor;
```

- Open do cursor executa a query e identifica o conjunto ativo de linhas.
- Se a query não retorna linhas, não ocorre exception.
- Use atributo de teste para o cursor depois do fetch.

Buscando dados do Cursor

Sintaxe

```
FETCH nome_cursor INTO [variavel1, variavel2, ...]  
                        / nome_registro];
```

- Recupera os valores da linha corrente nas variáveis de saída.
- Cada variável corresponde a uma posição de coluna definida na declaração do cursor.
- Teste para ver se o cursor contem linhas.

Buscando dados do Cursor

Exemplos

```
FETCH c1 INTO v_empno, v_ename;
```

```
...  
OPEN nome_cursor;  
LOOP  
    FETCH nome_cursor INTO variaveis_definidas  
    EXIT WHEN ...;  
    ...  
    -- processa o dado recuperado  
    ...  
END;
```

Fechando o Cursor

Sintaxe

```
CLOSE nome_cursor;
```

- Feche o cursor depois de completar o processamento das linhas.
- Reabra o cursor se for necessário
- Não é possível recuperar uma linha de um cursor que já tenha sido fechado.

Atributos de Cursor Explícito

Atributo	Tipo	Descrição
%ISOPEN	Boolean	TRUE se o cursor estiver aberto
%NOTFOUND	Boolean	TRUE se não retornaram linhas no último fetch
%FOUND	Boolean	TRUE se retornaram linhas no último fetch; complemento do %NOTFOUND
%ROWCOUNT	Number	Número de linhas retornadas

Controlando múltiplos Fetches

- Processa várias linhas de um cursor explícito usando um loop.
- Fetch de uma linha a cada iteração (laço).
- Usa o atributo %NOTFOUND para escrever um teste para um fetch sem sucesso.
- Usa atributos de cursor explícito para testar o sucesso de cada fetch.

O atributo %ISOPEN

- Fetch só pode ser feito com o cursor aberto.
- Use o atributo %ISOPEN antes de executar um fetch para verificar se o cursor está aberto.

Exemplo

```
IF NOT c1%ISOPEN THEN  
    OPEN c1;  
END IF;  
LOOP  
    FETCH c1...
```

%NOTFOUND e %ROWCOUNT

- Use o atributo **%ROWCOUNT** para saber o número exato de linhas recuperadas.
- Use o atributo **%NOTFOUND** para determinar a saída do loop.

Cursor e Registros

Registros PL/SQL podem ser usados para receber valores de linhas de um fetch.

Exemplo

```
...  
CURSOR c1 IS  
    SELECT empno, ename  
    FROM emp;  
    registro_emp c1%ROWTYPE;  
BEGIN  
    OPEN c1;  
    ...  
    FETCH c1 INTO registro_emp;
```


Cursor FOR Loops

Sintaxe

```
FOR nome_registro IN nome_cursor LOOP  
    comando1;  
    comando2;  
    . . .  
END LOOP;
```

- Open, fetch, e close ocorrem de maneira implícita.
- Não declarar o registro; isto é feito implicitamente.

Cursor FOR Loops

Recupera um empregado de cada vez até que não existam mais.

Exemplo

```
DECLARE  
    CURSOR c1 IS  
        SELECT empno, ename  
        FROM emp;  
BEGIN  
    FOR registro_emp IN c1 LOOP  
        -- open e fetch implícitos  
        IF registro_emp.empno = 7839 THEN  
            ...  
        END LOOP; -- close implícito  
    END;  
END;
```

Cursorões com Parâmetros

Sintaxe

```
CURSOR nome_cursor  
  [(nome_parametro nome_tipodado, ...)]  
IS  
  comando_select;
```

- Passa valores por parâmetro para um cursor quando ele é aberto e a query é executada.
- Abre um cursor explícito várias vezes com um conjunto diferente de linhas a cada vez.

Cursorões com Parâmetros

Passa o número do departamento e o cargo para a cláusula WHERE.

Exemplo

```
DECLARE  
  CURSOR c1  
  (v_deptno NUMBER, v_job VARCHAR2) IS  
  SELECT empno, ename  
  FROM emp  
  WHERE deptno = v_deptno  
  AND job = v_job;  
BEGIN  
  OPEN c1(10, 'CLERK');  
  ...
```