

Arquitetura e Administração de Bases de Dados
Oracle10g PL/SQL Programming

João Costa
jcosta@isec.pt

Cursorões

- ▶ Área de trabalho onde o comando SQL é processado
 - ▶ **Cursor implícito**
declarado automaticamente para comandos DML e SELECT.
 - ▶ **Cursor explícito**
declarado e gerido pelo programador para aceder a um conjunto de linhas devolvido por um SELECT.
- ▶ **Variável cursor**
 - ▶ Ponteiro ou referência para uma área de trabalho ou cursor.

Atributos dos Cursores

Attribute	Description
%FOUND	Returns TRUE if record was fetched successfully, Returns FALSE otherwise.
%NOTFOUND	Returns TRUE if record was not fetched successfully, Returns FALSE otherwise.
%ROWCOUNT	Returns number of records fetched from cursor at that point in time.
%ISOPEN	Returns TRUE if cursor is open, Returns FALSE otherwise.

▶ 3

2021/22 - LEI - AABD - PL/SQL

Cursores Implícitos

```

BEGIN
  UPDATE LIVROS
    SET QUANT_EM_STOCK = QUANT_EM_STOCK -1
  WHERE CODIGO_LIVRO = 21;

  IF SQL%NOTFOUND THEN
    DBMS_OUTPUT.PUT_LINE('LIVRO INEXISTENTE!');
  END IF;
END;
/

LIVRO INEXISTENTE!!

PL/SQL procedure successfully completed.
```

SQL%NOTFOUND

Para testar se a operação DML afetou algum registro

▶ 4

2021/22 - LEI - AABD - PL/SQL

Cursores Explícitos

- ▶ São o resultado de um comando **SELECT** ...
... ao qual é atribuído um nome.
- ▶ Este resultado pode ser manipulado linha a linha pelo programador
- ▶ Cursor definido, na secção declarativa

```
DECLARE
  CURSOR c1 IS
    SELECT *
    FROM   livros
    ORDER BY codigo_livro;
BEGIN
  -- ..
```

CURSOR nome IS
Declarar uma variável
que conterá o resultado
dum
SELECT

▶ 5

2021/22 - LEI - AABD - PL/SQL

Cursores Explícitos

Step	Activity	DESCRIPTION
1	DECLARE	Creates a named cursor identified by a SELECT statement. The SELECT statement does not include an INTO clause. Values in the cursor are moved to PL/SQL variables with the FETCH step.
2	OPEN	Processes the query and creates the active set of rows available in the cursor.
3	FETCH	Retrieves a row from the cursor into block variables. Each consecutive FETCH issued will retrieve the next row in the cursor until all rows have been retrieved.
4	CLOSE	Clears the active set of rows and frees the memory area used for the cursor.

▶ 6

2021/22 - LEI - AABD - PL/SQL

Cursor Explícito: Exemplo

```

DECLARE
  CURSOR c1 IS
    SELECT preco_tabela, titulo, genero
    FROM   livros;
  preco NUMBER;
  tit    VARCHAR2(100);
  gen    LIVROS.GENERO%TYPE;
BEGIN
  OPEN c1;
  LOOP
    FETCH c1 INTO preco, tit, gen;
    EXIT WHEN c1%NOTFOUND;
    IF preco > 100 THEN
      preco := preco * 1.1;
    ELSE
      preco := preco * 1.05;
    END IF;
    DBMS_OUTPUT.PUT_LINE(tit || '=' || preco);
  END LOOP;
  CLOSE c1;
END;
/

```

Declare cursor

Declare variables

Open cursor

Fetch a row from the cursor

Check if row returned from fetch

Calculate new price

Close cursor

▶ 7

2021/22 - LEI - AABD - PL/SQL

Cursores Explícitos

▶ utilizar o conjunto de instruções para o manipular

```

OPEN c1;
-- Abrir o cursor. O SELECT é executado neste momento para obter os resultados
LOOP
  -- deve-se usar um ciclo para iterativamente percorrer as várias linhas
  FETCH c1 INTO variáveis ;
  -- são necessárias tantas variáveis como as colunas do SELECT
  EXIT WHEN c1%NOTFOUND;
  -- teste de saída do cursor.
  -- %NOTFOUND é TRUE quando atingir o final do cursor
END LOOP;

CLOSE c1;
-- fechar o cursor de modo a libertar a memória ocupada

```

▶ 8

2021/22 - LEI - AABD - PL/SQL

Cursor Explícito: Variáveis do tipo do cursor

```

DECLARE
  CURSOR c1 IS
    SELECT preco_tabela, titulo
    FROM livros;

  r c1%ROWTYPE;

BEGIN
  OPEN c1;
  LOOP
    FETCH c1 INTO r;
    EXIT WHEN c1%NOTFOUND;
    IF r.preco_tabela > 100 THEN
      r.preco_tabela := r.preco_tabela * 1.1;
    ELSE
      r.preco_tabela := r.preco_tabela * 1.05;
    END IF;

    DBMS_OUTPUT.PUT_LINE(r.titulo || ' ' || r.preco_tabela);
  END LOOP;
  CLOSE c1;
END;

```

Diagram illustrating the explicit cursor process with annotations:

- Declare cursor (points to `CURSOR c1 IS`)
- Declare a ROWTYPE variable (points to `r c1%ROWTYPE;`)
- Open cursor (points to `OPEN c1;`)
- Fetch a row from the cursor (points to `FETCH c1 INTO r;`)
- Check if row returned from fetch (points to `EXIT WHEN c1%NOTFOUND;`)
- Calculate new price (points to `r.preco_tabela := r.preco_tabela * 1.1;` and `r.preco_tabela := r.preco_tabela * 1.05;`)
- Close cursor (points to `CLOSE c1;`)

▶ 9

2021/22 - LEI - AABD - PL/SQL

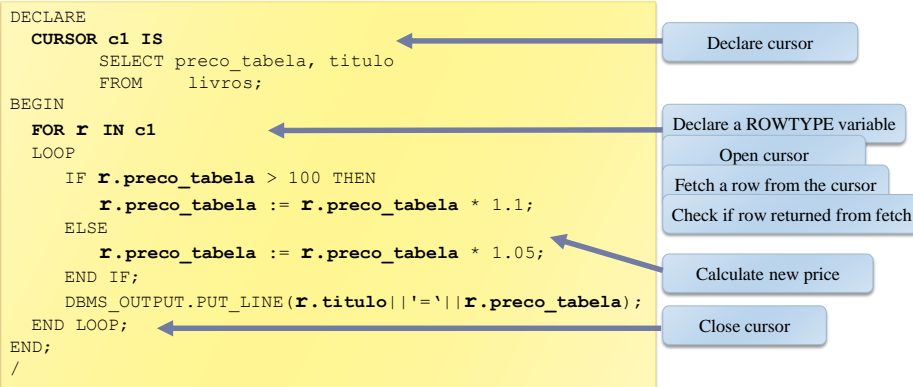
Ciclo FOR com cursores

- ▶ Faz automaticamente uma série de tarefas no processamento de cursores
 - ▶ Declaração da variável do tipo *record*
 - ▶ *Fetch* da próxima linha
 - ▶ Término do ciclo

▶ 10

2021/22 - LEI - AABD - PL/SQL

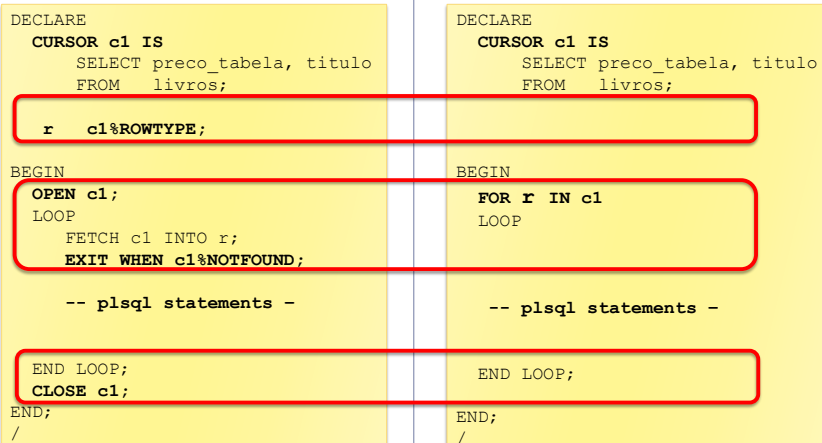
Ciclo FOR com cursores: Exemplo



▶ 11

2021/22 - LEI - AABD - PL/SQL

CURSOR (FOR vs OPEN ... FETCH)



▶ 12

2021/22 - LEI - AABD - PL/SQL

Ciclo FOR com cursores

- ▶ Para bloquear os registos, usar as cláusulas

FOR UPDATE e

WHERE CURRENT OF

▶ 13

2021/22 - LEI - AABD - PL/SQL

Cursor para atualizar a tabela

```
DECLARE
  CURSOR c1 IS
    SELECT preco_tabela, titulo, codigo_livro
    FROM   livros;

BEGIN
  FOR r IN c1
  LOOP
    IF r.preco_tabela > 100 THEN
      UPDATE LIVROS
        SET preco_tabela = preco_tabela * 1.1
        WHERE codigo_livro = r.codigo_livro;
    ELSE
      UPDATE LIVROS
        SET preco_tabela = preco_tabela * 1.05
        WHERE codigo_livro = r.codigo_livro;
    END IF;
  END LOOP;
END;
/
```

▶ 14

2021/22 - LEI - AABD - PL/SQL

Cursor para atualizar a tabela

```

DECLARE
  CURSOR c1 IS
    SELECT preco_tabela, titulo
    FROM livros
    FOR UPDATE OF preco_tabela;

BEGIN
  FOR r IN c1
  LOOP
    IF r.preco_tabela > 100 THEN
      UPDATE LIVROS
      SET preco_tabela = preco_tabela * 1.1
      WHERE CURRENT OF c1;
    ELSE
      UPDATE LIVROS
      SET preco_tabela = preco_tabela * 1.05
      WHERE CURRENT OF c1;
    END IF;
  END LOOP;
END;
/

```

The cursor is
for update of **preco_tabela**

Updates the table records
at the current position
of the cursor

► 15

2021/22 - LEI - AABD - PL/SQL

Cursor com parâmetros

```

DECLARE
  CURSOR c1 (coda NUMBER) IS
    SELECT titulo, preco_tabela
    FROM livros
    WHERE codigo_autor = coda;

BEGIN
  FOR r in c1 (21)
  LOOP
    -- plsql statements --
  END LOOP;

  FOR r in c1 (15)
  LOOP
    -- plsql statements --
  END LOOP;
END;

```

Declare the parameter

Use the parameter variable

Pass the parameter value to the cursor

Pass the parameter value to the cursor

► 16

2021/22 - LEI - AABD - PL/SQL

Ciclo FOR com cursor IMPLÍCITO

```
BEGIN
  FOR R IN (SELECT * FROM livros)
  LOOP
    DBMS_OUTPUT.PUT_LINE('TITULO' || R.TITULO);
  END LOOP;

  FOR R IN (SELECT * FROM AUTORES)
  LOOP
    DBMS_OUTPUT.PUT_LINE('NOME ' || R.NOME);
  END LOOP;
END;
/
```

cursor not declared.
Used by the Loop

cursor not declared.
Used by the Loop

Variáveis Cursor

- ▶ Manipula de forma mais eficiente os dados devolvidos por uma pesquisa já que devolve um ponteiro para a área de trabalho e não os dados
- ▶ A mesma variável cursor pode ser usada para várias pesquisas

Variáveis Cursor: Exemplo

```
DECLARE
  TYPE type_curvar IS REF CURSOR;
  cv_prod  type_curvar;
```

Definir um tipo de dados como sendo referência para um cursor

Declarar uma variável desse tipo

```
  rec_liv  livros%ROWTYPE;
  rec_aut  autores%ROWTYPE;
```

```
BEGIN
```

```
  OPEN cv_prod FOR SELECT * FROM livros;
```

Abrir um cursor com o resultado dum **SELECT**

```
  FETCH cv_prod INTO rec_liv;
```

```
  DBMS_OUTPUT.PUT_LINE('TITULO' || REC_LIV.TITULO);
```

```
  CLOSE cv_prod;
```

Fechar o cursor

```
  OPEN cv_prod FOR SELECT * FROM autores;
```

Abrir um cursor com o resultado dum **SELECT**

```
  FETCH cv_prod INTO rec_aut;
```

```
  DBMS_OUTPUT.PUT_LINE('NOME ' || REC_AUT.NOME);
```

```
  CLOSE cv_prod;
```

Fechar o cursor

```
END;
```

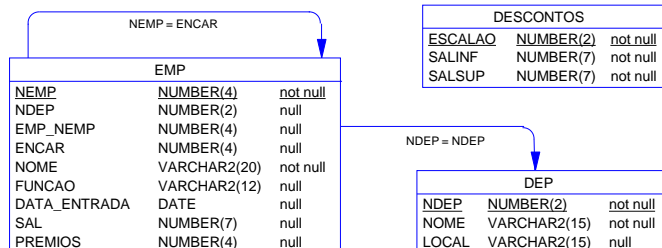
```
/
```

► 19

2021/22 - LEI - AABD - PL/SQL

Exercícios

1. Insira na tabela **TEMP** o nome e salário dos 10 empregados com salários mais altos
2. Para cada departamento, insira o nome e salário dos 2 empregados com salário mais baixo.



► 20

2021/22 - LEI - AABD - PL/SQL