

Encontro 2 – Cursors e Manipulação de Dados (4h)

Objetivos

- Compreender o uso de **cursores implícitos e explícitos** no PL/SQL.
- Iterar sobre resultados de consultas.
- Atualizar e manipular dados com **comandos DML** em blocos PL/SQL.

1. Cursores no PL/SQL

Um **cursor** é uma área de memória que armazena o resultado de uma consulta SQL. No PL/SQL, podemos trabalhar com:

- **Cursores implícitos:** criados automaticamente pelo Oracle quando executamos instruções SQL (INSERT, UPDATE, DELETE, SELECT INTO).
- **Cursores explícitos:** criados pelo programador para percorrer várias linhas de uma consulta.

1.1 Cursores Implícitos

São criados automaticamente quando usamos **DML** ou **SELECT INTO**.

Atributos úteis:

Atributo	Descrição
SQL%ROWCOUNT	Número de linhas afetadas
SQL%FOUND	Verdadeiro se alguma linha foi afetada
SQL%NOTFOUND	Verdadeiro se nenhuma linha foi afetada
SQL%ISOPEN	Sempre FALSE para cursores implícitos

Exemplo:

```
BEGIN
  UPDATE empregados
  SET salario = salario * 1.1
  WHERE departamento_id = 10;
```

```

    IF SQL%FOUND THEN
        DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' funcionários
atualizados. ');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Nenhum funcionário encontrado. ');
    END IF;
END;

```

1.2 Cursores Explícitos

Usados quando precisamos **percorrer várias linhas**.

Etapas:

1. **DECLARE** o cursor
2. **OPEN** o cursor (executa a consulta)
3. **FETCH** (busca linha por linha)
4. **CLOSE** o cursor (libera memória)

Exemplo básico:

```

DECLARE
    CURSOR c_emp IS
        SELECT nome, salario FROM empregados;
    v_nome empregados.nome%TYPE;
    v_salario empregados.salario%TYPE;
BEGIN
    OPEN c_emp;
    LOOP
        FETCH c_emp INTO v_nome, v_salario;
        EXIT WHEN c_emp%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_nome || ' - ' || v_salario);
    END LOOP;
    CLOSE c_emp;
END;

```

1.3 Cursor FOR Loop

O **FOR loop** simplifica o uso do cursor. Não é preciso abrir, buscar ou fechar manualmente.

```

BEGIN
    FOR rec IN (SELECT nome, salario FROM empregados) LOOP

```

```

        DBMS_OUTPUT.PUT_LINE(rec.nome || ' - ' || rec.salario);
    END LOOP;
END;

```

1.4 Manipulação de Dados (INSERT, UPDATE, DELETE) em PL/SQL

Podemos executar DML dentro de blocos PL/SQL e usar variáveis para controlar a lógica.

Exemplo: Inserindo dados

```

DECLARE
    v_nome empregados.nome%TYPE := 'Carlos';
    v_salario empregados.salario%TYPE := 2500;
BEGIN
    INSERT INTO empregados (id, nome, salario)
    VALUES (seq_empregados.NEXTVAL, v_nome, v_salario);
    DBMS_OUTPUT.PUT_LINE('Empregado ' || v_nome || ' inserido.');
```

```
END;
```

Exemplo: Atualizando com lógica

```

BEGIN
    FOR rec IN (SELECT id, salario FROM empregados WHERE departamento_id
    = 20) LOOP
        UPDATE empregados
        SET salario = rec.salario * 1.15
        WHERE id = rec.id;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Salários reajustados.');
```

```
END;
```

2. Exemplos Práticos Guiados

Exemplo 1 – Listar nomes e salários

```

DECLARE
    CURSOR c_emp IS SELECT nome, salario FROM empregados;
    v_nome empregados.nome%TYPE;
    v_salario empregados.salario%TYPE;
BEGIN
    OPEN c_emp;
    LOOP
        FETCH c_emp INTO v_nome, v_salario;
        EXIT WHEN c_emp%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_nome || ' - ' || v_salario);
```

```
END LOOP;  
CLOSE c_emp;  
END;
```

Exemplo 2 – Usando FOR Cursor Loop

```
BEGIN  
  FOR emp IN (SELECT nome, salario FROM empregados WHERE salario < 3000)  
  LOOP  
    DBMS_OUTPUT.PUT_LINE(emp.nome || ' - ' || emp.salario);  
  END LOOP;  
END;
```

Exemplo 3 – Atualizar salários

```
BEGIN  
  UPDATE empregados  
  SET salario = salario * 1.1  
  WHERE departamento_id = 30;  
  
  DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' salários reajustados.');
```

END;

3. Exercícios Práticos

Exercício 1 – Cursor para reajuste de salário

Crie um bloco PL/SQL que:

- Percorra a tabela **empregados** usando um cursor.
- Aumente o salário em **5%** se o salário for menor que **2000**.
- Exiba uma mensagem com o nome e o novo salário.

Exercício 2 – Relatório de funcionários

Exibir um relatório com nome, cargo e salário de todos os empregados do departamento 10.

Exercício 3 – Exclusão condicional

Excluir todos os empregados com salário abaixo de 1000 e
exibir o total excluído.

Exercício 4 – Inserção com lógica

Criar um bloco que insira um novo empregado apenas se não existir
outro com o mesmo nome.