

Stored Procedure - Exemplo

```
create or replace function FConsulta1(codigo  
IN INTEGER) return varchar2 is  
nome VARCHAR2(30);  
BEGIN  
select e.nomeeditora into nome from editoras  
e where ideditora=codigo;  
Return nome;  
end;
```

Stored Procedure Exercícios

Stored Procedure - Exercícios

Criar um procedimento para formatar nome de autores de acordo com citações em referências bibliográficas formatado (Último, Primeiro)

Exemplo: Machado de Assis = Assis, Machado.

Criar uma função para retornar as estatísticas de vogais, espaços em branco e consoantes.

```
SELECT FEstatistica('Universidade Federal de Mato  
Grosso') FROM dual, resultado:
```

Vogais:40%, Consoantes:49 %, Espacos:11%

Função FORMULA(N) para retornar o valor H, sendo

$$H = 1 + 1/2 + 3 + 1/4 + \dots + (N + 1/N).$$

Stored Procedure - Exercícios

Nome do Autor Formatado (Último, Primeiro)

```
CREATE OR REPLACE Procedure Formatar(PNome IN VARCHAR2) AS
primeiro VARCHAR(20);          ultimo VARCHAR(20);
i NUMBER(2); p NUMBER(2); u NUMBER(2);
BEGIN
  p:=0; u:=0;
  FOR i IN 1..Length(PNome)
  loop
    IF (SubStr(PNome,i,1)=' ') AND (p=0) THEN
      p:=i;
    END IF;
    IF (SubStr(PNome,i,1)=' ') THEN
      u:=i;
    END IF;
  END loop;
  primeiro:=SubStr(PNome,1,p);
  ultimo:=SubStr(PNome,u,u+(Length(PNome)));
  dbms_output.put_line(Trim(ultimo||', '||primeiro));
END;
```

Stored Procedure - Exercícios

Nome do Autor Formatado (Último, Primeiro)

```
CREATE OR REPLACE Function FFormatar(PNome IN VARCHAR2) RETURN
varchar2 is
primeiro VARCHAR(20);          ultimo VARCHAR(20);
i NUMBER(2); p NUMBER(2); u NUMBER(2);
BEGIN
  p:=0; u:=0;
  FOR i IN 1..Length(PNome)
  loop
    IF (SubStr(PNome,i,1)=' ') AND (p=0) THEN
      p:=i;
    END IF;
    IF (SubStr(PNome,i,1)=' ') THEN
      u:=i;
    END IF;
  END loop;
  primeiro:=SubStr(PNome,1,p);
  ultimo:=SubStr(PNome,u,u+(Length(PNome)));
  RETURN Trim(ultimo||', '||primeiro);
END;
```

PL/SQL - Exercícios

```
CREATE OR REPLACE FUNCTION FEstadistica(P_Nome IN VARCHAR2) RETURN VARCHAR2 is
i NUMBER(3);
vogais NUMBER(3);      espacos NUMBER(3);      consoantes NUMBER(3);
perc_vogais NUMBER(3); perc_espacos NUMBER(3); perc_consoantes NUMBER(3);
BEGIN
vogais:=0; espacos:=0; consoantes:=0;
FOR i IN 1..Length(P_Nome)
LOOP
IF (SubStr(P_nome,i,1)=' ') THEN
    espacos:=espacos+1;
END IF;
IF (SubStr(Upper(P_nome),i,1) IN ('A','E','I','O','U')) THEN
    vogais:=vogais+1;
END IF;
END LOOP;
consoantes:=Length(P_Nome)-(vogais+espacos);
perc_vogais:=(vogais*100)/Length(P_Nome);
perc_espacos:=(espacos*100)/Length(P_Nome);
perc_consoantes:=(consoantes*100)/Length(P_Nome);
RETURN 'Vogais:'||perc_vogais||'%', Consoantes:'||perc_consoantes||' %,
    Espacos:'||perc_espacos||'%';
END;
```

PL/SQL - Exercícios

```
CREATE TABLE valores(id NUMBER(10),valor1 NUMBER(4,2),valor2  
NUMBER(4,2),valor3 NUMBER(4,2),PRIMARY KEY (id));
```

```
CREATE OR REPLACE PROCEDURE Preencher(PTotal IN NUMBER) AS  
aux NUMBER(10);  
BEGIN  
    aux:=0;  
    SELECT Nvl(Max(v.id),0) INTO aux FROM valores v;  
    FOR i IN 1..PTotal  
    LOOP  
        aux:=aux+1;  
        INSERT INTO valores(id,valor1,valor2,valor3) VALUES  
(aux,dbms_random.value(1,10),dbms_random.value(1,10),dbms_random.value(1,10));  
    END LOOP;  
END;
```

PL/SQL - Exercícios

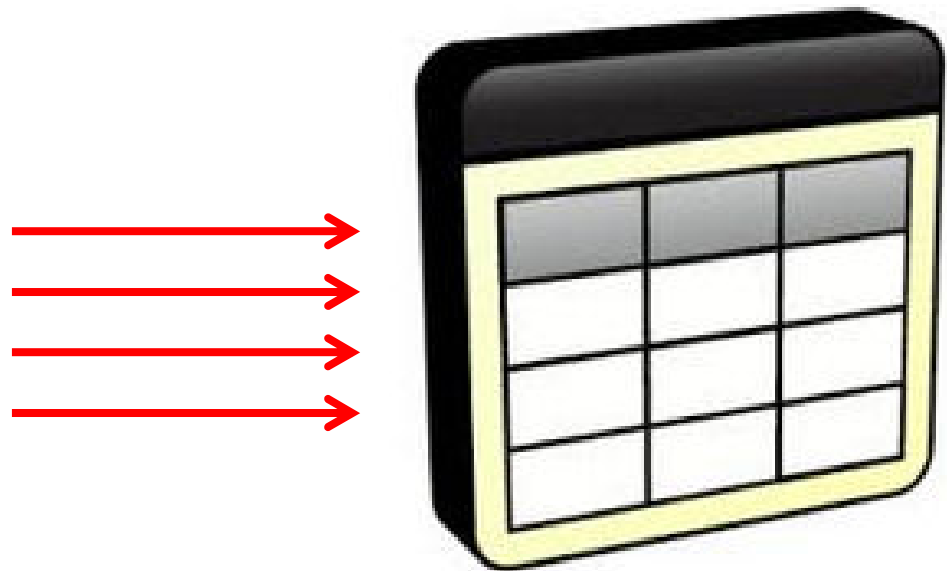
Dado a tabela: CREATE TABLE valores(id NUMBER(10),valor1 NUMBER(4,2),valor2 NUMBER(4,2),valor3 NUMBER(4,2),PRIMARY KEY (id));
Implemente um procedimento para preencher as colunas da tabela, sendo: Id sequencial.
valor1, valor2 e valor3 valores randomicos entre 1 e 10.

Função para retornar o total de vendas realizadas parametrizada pelo ID do cliente.

Procedimento para exibir as vendas por estado, o estado deverá ser informado via parâmetro.

Função para retornar o primeiro nome de uma pessoa informada via parâmetro, utilize o espaço em branco como referência.

Cursores



PL/SQL - Cursores

Sentença SQL
Select * from tabela



Linha1
Linha2
Linha3
...
Linha N

Sentença SQL (Procedimento)

Select * from tabela



Tratamento dos Dados (Buffer)



Linha1
Linha2
Linha3
...
Linha N

PL/SQL - Cursores

```
create or replace procedure Lista_Vendedor(PCodigo IN NUMBER) AS
AuxNome VARCHAR2(40);
BEGIN
    SELECT nomevendedor INTO AuxNome FROM exercicios3.vendedores v WHERE
        v.idvendedor=PCodigo;
    Dbms_Output.put_line(AuxNome);
end;
```

```
create or replace procedure Lista_Vendedores as
    CURSOR CursorVendedores IS SELECT nomevendedor, idvendedor
                                     FROM vendedores;
BEGIN
    FOR linhas IN CursorVendedores
    LOOP
        DBMS_OUTPUT.PUT_LINE(linhas.nomevendedor);
    END LOOP;
end;
```

Cursores com Stored Procedures

```
create or replace procedure Lista_Vendedores is  
    CURSOR CursorVendedores  
    IS SELECT nomevendedor, idvendedor  
        FROM vendedores WHERE idvendedor >= 1 AND  
        idvendedor <= 5;  
BEGIN  
    FOR linhas IN CursorVendedores  
    LOOP  
        DBMS_OUTPUT.PUT_LINE(linhas.nomevendedor);  
    END LOOP;  
end;
```

Cursores Parametrizados com Stored Procedures

create or replace procedure consultacidades1(codigo in integer) is

CURSOR cursorcidades

IS SELECT nomecidade, idcidade

FROM cidades WHERE idcidade=codigo;

BEGIN

FOR linhas IN cursorcidades

LOOP

DBMS_OUTPUT.PUT_LINE(linhas.nomecidade);

END LOOP;

end;

Cursos Parametrizados com Stored Procedures

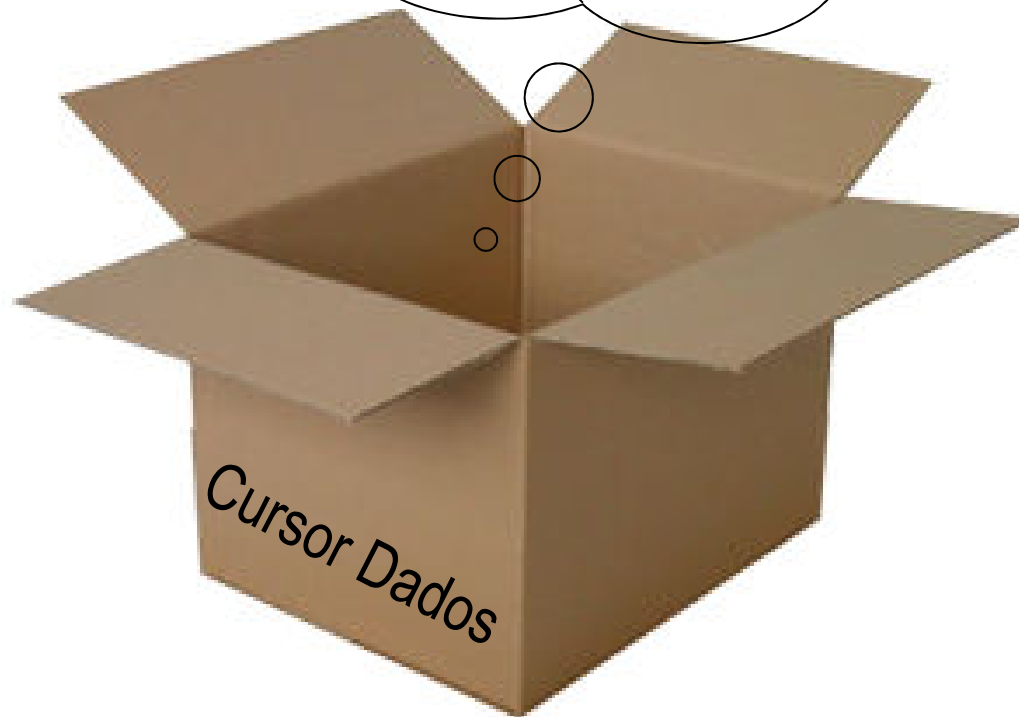
```
create or replace procedure consultacidades2(nome in  
  varchar2) is CURSOR cursorcidades  
  IS SELECT nomecidade, idcidade  
    FROM cidades WHERE upper(nomecidade) LIKE  
    upper(nome) || '%';  
BEGIN  
  FOR linhas IN cursorcidades  
  LOOP  
    DBMS_OUTPUT.PUT_LINE(linhas.nomecidade);  
  END LOOP;  
end;
```

Cursores Parametrizados com Stored Procedures

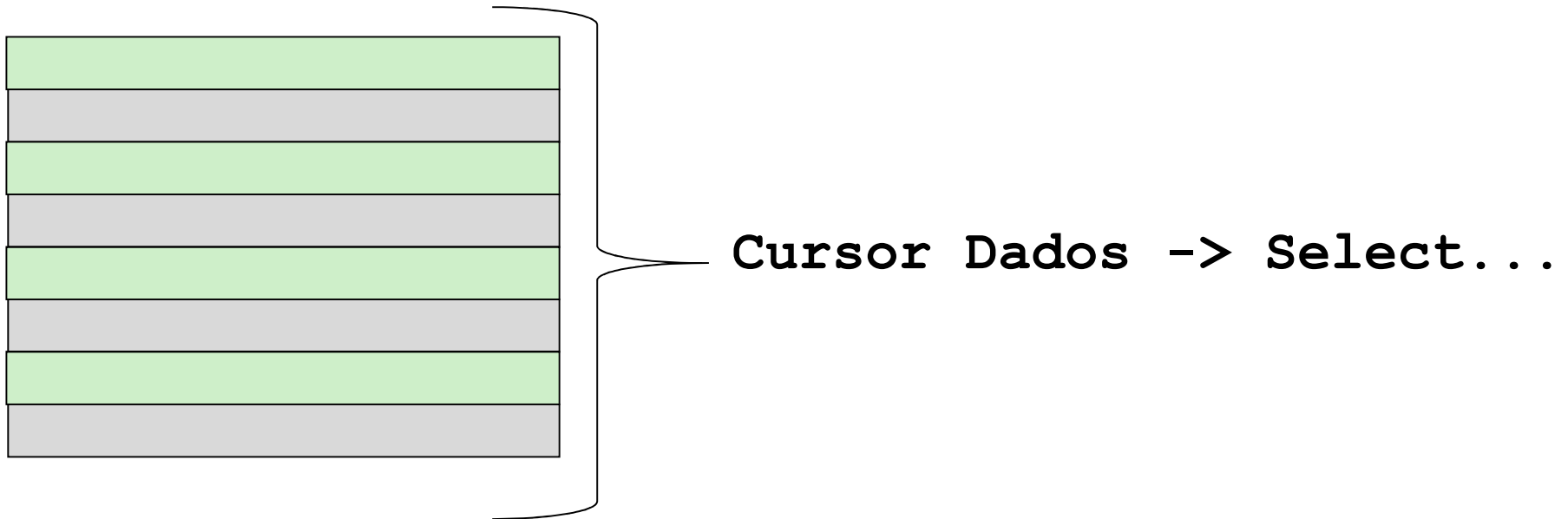
```
create or replace procedure ConsultaProduto(codigo in integer) is
  CURSOR CursosProdutos
  IS SELECT vd.datavenda,cl.nomecliente,ve.nomevendedor,pr.modelo
     FROM vendas vd,clientes cl,vendedores ve,veiculos pr
     WHERE vd.idcliente=cl.idcliente AND vd.idvendedor=ve.idvendedor
     AND pr.idveiculo=vd.idveiculo AND pr.idveiculo=codigo;

BEGIN
  FOR linhas IN CursosProdutos
  LOOP
    DBMS_OUTPUT.PUT_LINE('DataVenda..' || linhas.datavenda);
    DBMS_OUTPUT.PUT_LINE('Cliente....:' || linhas.nomecliente);
    DBMS_OUTPUT.PUT_LINE('Vendedor...:' || linhas.nomevendedor);
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;
end;
```

Select * from ...

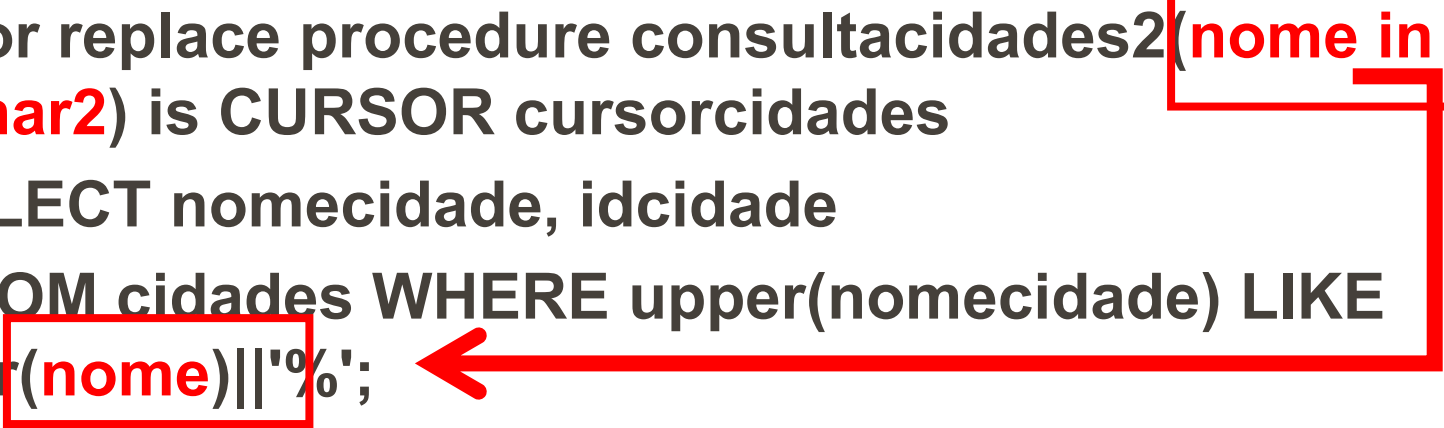


Cursores Parametrizados



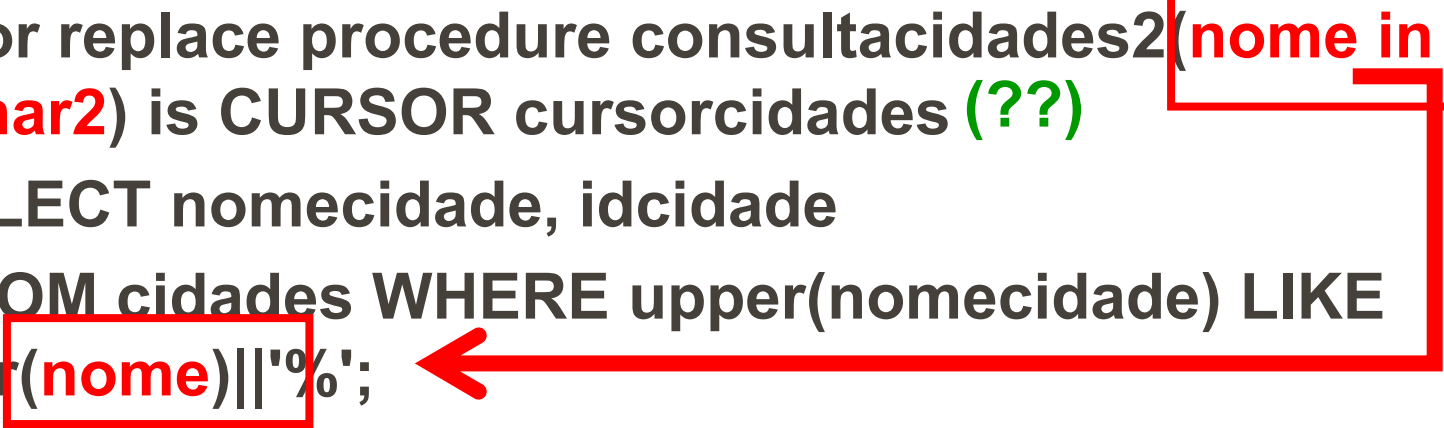
Cursores Parametrizados com Stored Procedures

```
create or replace procedure consultacidades2(nome in  
  varchar2) is CURSOR cursorcidades  
  IS SELECT nomecidade, idcidade  
    FROM cidades WHERE upper(nomecidade) LIKE  
    upper(nome)||'%%';  
BEGIN  
  FOR linhas IN cursorcidades  
  LOOP  
    DBMS_OUTPUT.PUT_LINE(linhas.nomecidade);  
  END LOOP;  
end;
```



Cursos Parametrizados com Stored Procedures

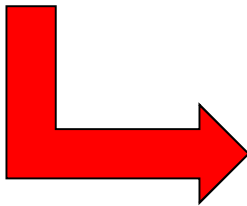
```
create or replace procedure consultacidades2(nome in  
  varchar2) is CURSOR cursorcidades (??)  
  IS SELECT nomecidade, idcidade  
    FROM cidades WHERE upper(nomecidade) LIKE  
    upper(nome)||'%%';  
BEGIN  
  FOR linhas IN cursorcidades  
  LOOP  
    DBMS_OUTPUT.PUT_LINE(linhas.nomecidade);  
  END LOOP;  
end;
```



Cursoros Parametrizados com Stored Procedures

Editoras

Thomson Learning
Arcadia
Arte e Cultura
Ática
Barracuda



Livros

Catherine Blair - A NOIVA FRANCESA.pdf
Catherine Coulter - Noites II - NOITES DE SOMBRAS.pdf
Catherine Fisher - INCARCERON I.pdf
Catherine George - A PROVA DO PECADO.rtf
Catherine George - DEPOIS DO BAILE.doc
Catherine George - MAGICO RETORNO.doc

Cursores Parametrizados com Stored Procedures

```
CREATE OR REPLACE PROCEDURE Editoras_Livros AS
CURSOR C_Editoras is SELECT ed.nomeeditora,ed.ideditora FROM editoras ed;
CURSOR C_Livros (P_Editora IN NUMBER) is SELECT li.titulo,li.ideditora FROM livros li
WHERE li.ideditora=P_Editora;
```

```
BEGIN
```

```
DBMS_OUTPUT.ENABLE(1000000); --set serveroutput on size 1000000
```

```
FOR Dados_Editora IN C_Editoras
```

```
LOOP
```

```
    Dbms_Output.put_line(Dados_Editora.nomeeditora);
```

```
    FOR Dados_Livros IN C_Livros(Dados_Editora.ideditora)
```

```
    LOOP
```

```
        Dbms_Output.put_line('    '||Dados_Livros.titulo);
```

```
    END LOOP;
```

```
END LOOP;
```

```
END;
```