



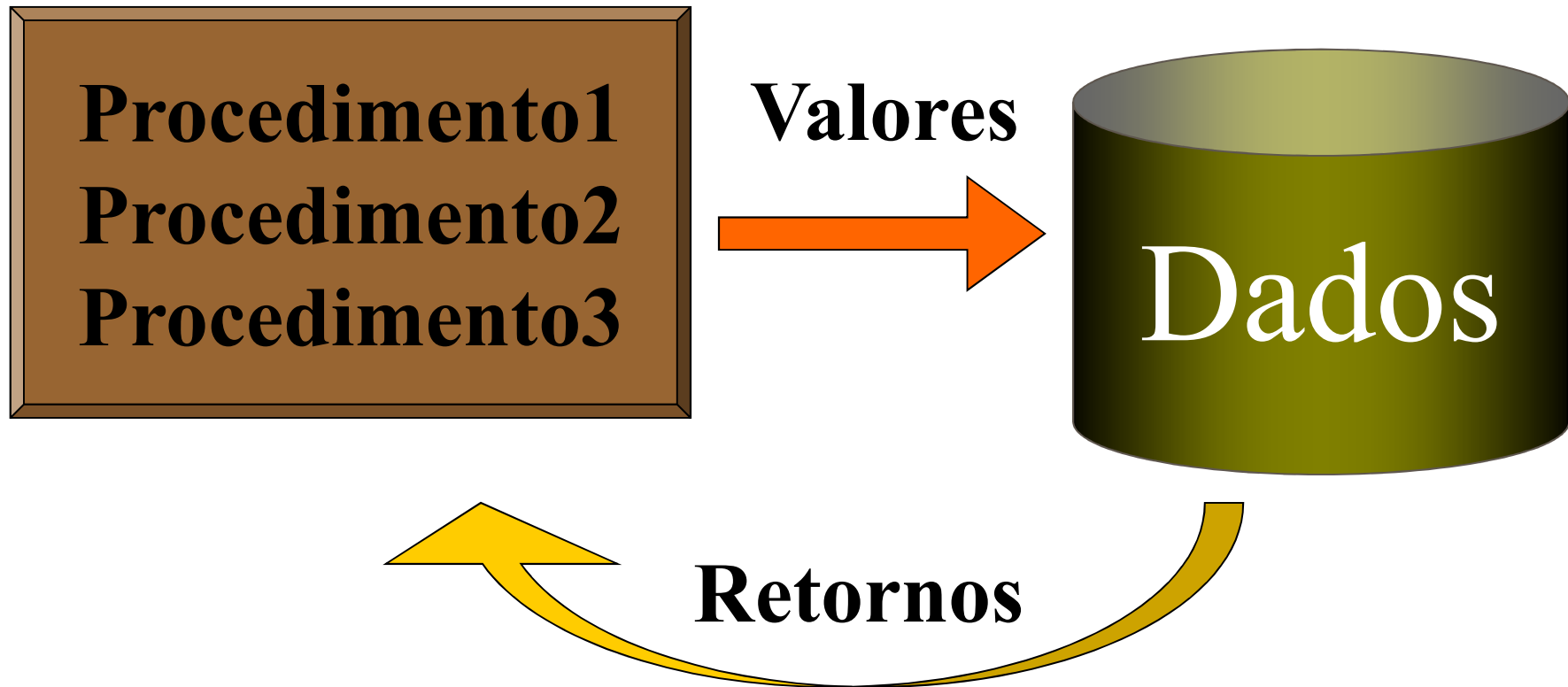
Universidade Federal de Mato Grosso-UFMT
Sistemas de Informação
Laboratório de Banco de Dados
Prof. Clóvis Júnior

Stored Procedures

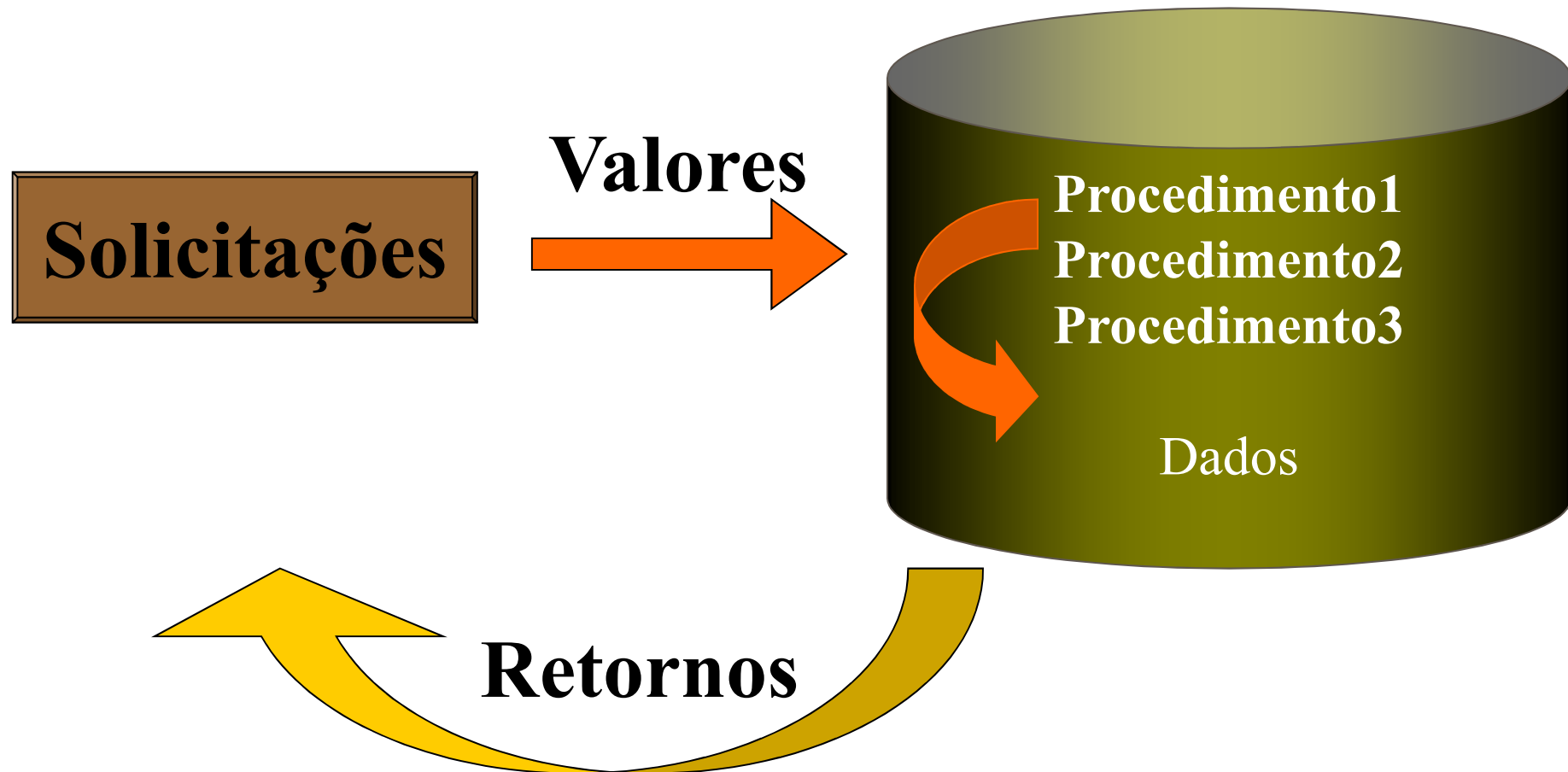
(Procedimentos Armazenados)



Procedures - Arquitetura



Stored Procedure - Arquitetura



Stored Procedure - Visão Geral

- **Vantagens:**
 - **Redução do tráfego de rede;**
 - **Possibilidade de implementação em outras linguagens (Java, C++);**
 - **Front-End Independente;**
 - **Velocidade na execução do SQL (sentenças pré-compilada)**

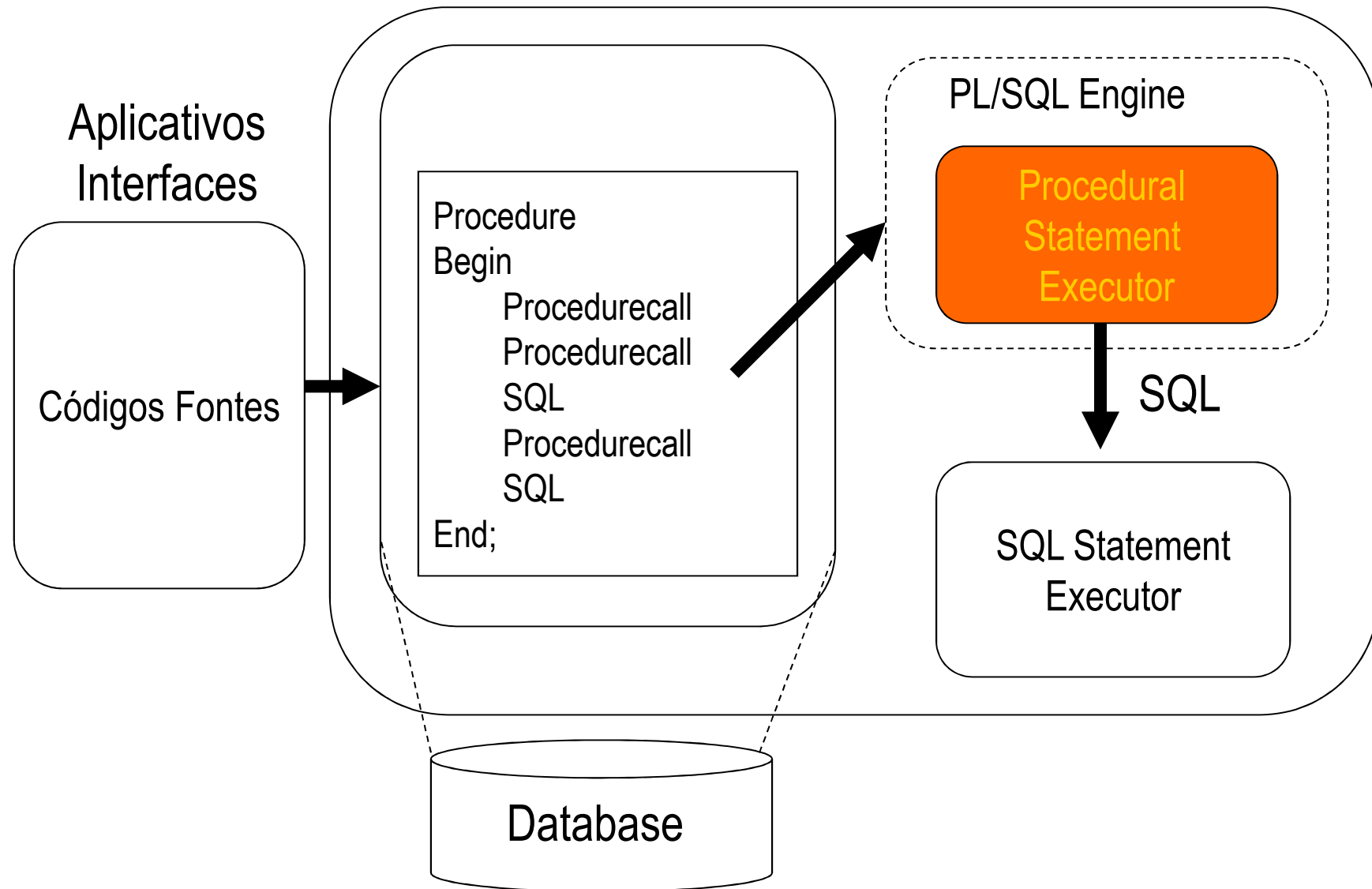
Stored Procedure - Visão Geral

- **Vantagens:**
- **Programação Modular;**
- **Automatização de transações sensivelmente complexas;**
- **Suporte a estruturas como: if, else, while loops, goto, variaveis locais etc.**

Stored Procedure - Visão Geral

- **Desvantagens**
 - **Código específico, cada SGBD tem uma linguagem de programação própria.**

Stored Procedure - Arquitetura



Stored Procedure - Sintaxe

create or replace

procedure nome_do_procedimento as

begin

Implementação (códigos);

end;

Salvando, Compilando, e Executando Códigos PL/SQL

- ⌘ Barra (slash) (/): Executa a última linha digitada;
- ⌘ START MyProg.SQL: Executa sentenças ou procedimentos contidos em arquivos externos;
- ⌘ SHOW ERRORS: Apresenta mensagens de erro;
- ⌘ EXEC MyProg ('parameter1', 'parameter2'): Executa o procedimento;

Procedimentos

```
create or replace procedure saida is  
mensagem VARCHAR2(30);  
BEGIN  
mensagem:='UFMT';  
Dbms_Output.put_line(mensagem);  
end;
```

PL/SQL - Funções

Length: Tamanho da string

Substr: Gera substring

Min: Retorna valor mínimo

Max: Retorna Valor máximo;

Count: Total de ocorrências

Translate: Substituição de strings

PL/SQL - Funções

SELECT

```
nomecliente, SubStr (nomecliente, 1, 5) Sub  
, Length (nomecliente) Tamanho FROM  
clientes c
```

Funções

```
CREATE OR REPLACE FUNCTION  
somar(v1 IN NUMBER,v2 IN NUMBER)  
RETURN NUMBER IS  
BEGIN  
    RETURN v1+v2;  
END;
```

Stored Procedure - IF

Cálculo da Média de 4 Notas

```
CREATE OR REPLACE FUNCTION FResultado(p_n1 IN  
    NUMBER, p_n2 IN NUMBER, p_n3 IN NUMBER, p_n4  
    IN number) RETURN VARCHAR2 IS  
v_resultado VARCHAR2(30);  
v_media NUMBER(4,2);  
BEGIN  
    v_media:=(p_n1+p_n2+p_n3+p_n4)/4;  
    IF v_media<5 THEN v_resultado:='Reprovado';  
    ELSE v_resultado:='Aprovado';  
    END IF;  
    RETURN v_resultado;  
END;
```

Stored Procedure - IF

Cálculo da Média de 4 Notas - Teste

SELECT FResultado(3,4,7,8) FROM dual  **Aprovado**

SELECT FResultado(3,4,2,1) FROM dual  **Reprovado**

**CREATE TABLE NotasTeste(matricula NUMBER(4),nota1
NUMBER(4,2),nota2 NUMBER(4,2),nota3
NUMBER(4,2),nota4 NUMBER(4,2),PRIMARY
KEY(matricula));**

	MATRICULA	NOTA1	NOTA2	NOTA3	NOTA4
1	1	5	7	8	3
2	2	3	4	5	3
3	3	1	9	5	2

**SELECT nota1,nota2,nota3,nota4,FResultado(nota1,nota2,nota3,nota4)
Resultado FROM NotasTeste**

	NOTA1	NOTA2	NOTA3	NOTA4	RESULTADO
1	5	7	8	3	Aprovado
2	3	4	5	3	Reprovado
3	1	9	5	2	Reprovado

Funções

```
CREATE OR REPLACE FUNCTION calcular (v1 IN NUMBER,v2 IN
NUMBER,op IN char) RETURN NUMBER IS
resultado NUMBER(10,3);
BEGIN
CASE op
WHEN '+' THEN resultado:=v1+v2;
WHEN '-' THEN resultado:=v1-v2;
WHEN '/' THEN IF v2<>0 THEN resultado:=v1/v2;
                ELSE resultado:=0;
            END IF;
WHEN '*' THEN resultado:=v1*v2;
END CASE;
RETURN resultado;
END;
```


Funções

```
CREATE OR REPLACE FUNCTION FData (P_Cidade IN varchar2) RETURN varchar2 IS  
dia VARCHAR2(2); ano VARCHAR2(4); mes VARCHAR2(2); nome_mes VARCHAR2(12);  
BEGIN  
dia:=To_Char(SYSDATE,'DD');  
mes:=To_Char(SYSDATE,'MM');  
ano:=To_Char(SYSDATE,'YYYY');  
  
CASE mes  
WHEN 1 THEN nome_mes:='Janeiro';   WHEN 2 THEN nome_mes:='Fevereiro';  
WHEN 3 THEN nome_mes:='Março';     WHEN 4 THEN nome_mes:='Abril';  
WHEN 5 THEN nome_mes:='Maio';      WHEN 6 THEN nome_mes:='Junho';  
WHEN 7 THEN nome_mes:='Julho';     WHEN 8 THEN nome_mes:='Agosto';  
WHEN 9 THEN nome_mes:='Setembro';  WHEN 10 THEN nome_mes:='Outubro';  
WHEN 11 THEN nome_mes:='Novembro'; WHEN 12 THEN nome_mes:='Dezembro';  
END CASE;  
RETURN P_Cidade||', '||dia||' de '||nome_mes||' de '||ano;  
END;
```

Stored Procedure - IF

Função ParImpar

```
CREATE FUNCTION ParImpar(Valor IN INTEGER)  
RETURN VARCHAR2 IS  
aux VARCHAR2(15);  
BEGIN  
    aux:='Impar';  
    IF Mod(Valor,2)=0  
        THEN aux:='Par';  
    END IF;  
    RETURN aux;  
END;
```

Stored Procedure

Contador Simples

```
CREATE OR REPLACE PROCEDURE contador  
(valor IN NUMBER) IS  
begin  
  for variavel in 1..valor  
  loop  
    dbms_output.put_line (variavel);  
  end loop;  
end;
```

Stored Procedure

Contador para Valores Aleatórios

**CREATE OR REPLACE PROCEDURE ContadorRandomico (valor IN
NUMBER) IS**

begin

for variavel in 1..valor

loop

dbms_output.put_line (trunc(dbms_random.value(1,100)));

end loop;

end;

Stored Procedure

Contador para Valores Aleatórios

```
CREATE OR REPLACE PROCEDURE ContadorRandomico2(valor IN  
    NUMBER) aS  
total NUMBER(4);  
valor2 NUMBER(3);  
BEGIN  
    total:=0; valor2:=0;  
    for variavel in 1..valor  
    LOOP  
        valor2:=trunc(dbms_random.value(1,100));  
        total:=total+valor2;  
        dbms_output.put_line (valor2);  
    end loop;  
    dbms_output.put_line ('-----');  
    dbms_output.put_line ('Total:' || total);  
end;
```

Stored Procedure

Total de Caracteres em Texto

```
CREATE OR REPLACE FUNCTION TotalCar(PTexto IN VARCHAR2,PCar IN  
  CHAR) RETURN INTEGER IS  
  x NUMBER(3);  
  total NUMBER(3);  
  BEGIN  
    total:=0;  
    FOR x IN 1..Length(PTexto)  
    LOOP  
      IF upper(SubStr(PTexto,x,1))=upper(PCar) THEN  
        total:=total+1;  
      END IF;  
    END LOOP;  
    RETURN total;  
  END;
```

Stored Procedure

Cálculo do Fatorial

```
CREATE OR REPLACE FUNCTION Ffatorial (n IN NUMBER) RETURN varchar2 IS
f INTEGER;      c NUMBER(4);  aux VARCHAR2(120);
BEGIN
    aux:='Erro no cálculo, valor inválido';
    f:=1; c:=1;
    IF (n>=0) AND (n<=33) THEN
    BEGIN
        WHILE c<=n
        LOOP
            f:=f*c;
            c:=c+1;
        END LOOP;
        aux:=To_Char(f);
    END;
    END IF;
    RETURN aux;
END;
```

Stored Procedure com Sequences

```
create or replace procedure inserir (n in  
  varchar2) is  
begin  
insert into alunos(matricula,nome) values  
  (seq_matricula.nextval,n);  
end;
```


Stored Procedure – Exemplo Parametrizado

```
create or replace procedure inserir (n in  
    varchar2) is  
    ultimo integer;  
begin  
    select nvl(max(alunos.matricula),0) into ultimo  
    from alunos;  
    insert into alunos(matricula,nome) values  
        (ultimo+1,n);  
end;
```

Stored Procedure - Exemplo

```
create or replace procedure consulta1(codigo  
IN INTEGER) as  
nome VARCHAR2(30);  
BEGIN  
select c.nomecliente into nome from clientes c  
where c.idcliente=codigo;  
Dbms_Output.put_line(nome);  
end;
```