

# – SQL – Pacotes

Laboratório de Bases de Dados  
Profa. Dra. Cristina Dutra de Aguiar Ciferri

# Pacotes

- Características
  - objetos do banco de dados
  - são equivalentes a bibliotecas
- Armazenam
  - procedimentos
  - funções
  - definições de cursores
  - declarações de variáveis e constantes
  - definição de exceções

# Motivação para uso de Pacotes

- Modularidade
  - subprogramas, exceções, etc. agrupados por funcionalidade
- Encapsulamento
  - oculta do usuário informações relevantes apenas no contexto interno do pacote
- Possibilidade de compilar subprogramas apenas com a especificação do pacote
  - antes mesmo da criação do corpo do pacote

# Motivação para uso de Pacotes

- Manutenção do valor das variáveis entre transações durante uma mesma seção
  - solução para o problema de tabela mutante
  - útil para manter informações de contexto utilizadas pelos subprogramas do pacote

# Motivação para uso de Pacotes

- Possibilita Overloading
  - exemplo

Início Pacote

procedure teste (**par1 in number**);

procedure teste (**par1 in varchar**);

procedure teste (**par1 in number, par2 in number**);

Fim pacote

# Composição dos Pacotes

- Especificação
  - contém as declarações **públicas** que estarão disponíveis para uso externo
- Corpo
  - contém as declarações **privadas** que estarão disponíveis apenas dentro do pacote
  - define as ações para os subprogramas (públicos ou privados)

# Especificação

```
CREATE OR REPLACE PACKAGE nome_pacote IS  
  
-- para procedimentos e funções, somente os  
-- cabeçalhos (i.e., interface)  
PROCEDURE nome_procedimento (parâmetros);  
FUNCTION nome_função (parâmetros) RETURN tipo;  
  
-- declaração de variáveis, constantes, exceções e  
-- cursores públicos  
  
END nome_pacote;
```

# Corpo

```
CREATE OR REPLACE PACKAGE BODY nome_pacote IS
-- declaração de variáveis, constantes, exceções e
-- cursores privados

PROCEDURE nome_procedimento (parâmetros) IS
BEGIN ... END;

FUNCTION nome_função (parâmetros) RETURN tipo IS
BEGIN ... RETURN ...END;

END;
```



# SHOW ERRORS PACKAGE;

- Mostra erros de compilação
  - executado após a tentativa de compilação de um pacote
- Opções
  - SHOW ERRORS PACKAGE nome\_pacote;
  - SHOW ERRORS PACKAGE BODY nome\_pacote;

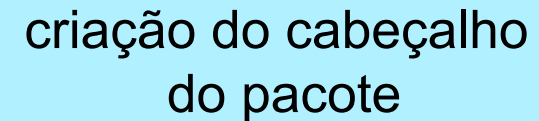
# Execução das Estruturas Públicas do Pacote

- Necessidade de se especificar o nome do pacote que define a estrutura pública
- Exemplos
  - EXECUTE `nome_pacote.nome_procedimento` (parâmetros);
  - `variável := nome_pacote.nome_função` (parâmetros)

# Exemplo (1/4)

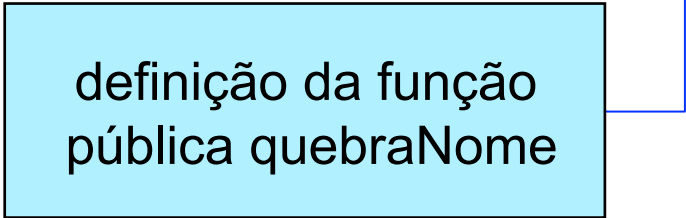
```
CREATE OR REPLACE PACKAGE ManipulaNomes  
IS
```

criação do cabeçalho  
do pacote



```
    FUNCTION quebraNome  
        (nomeCompleto IN VARCHAR2)  
        RETURN VARCHAR2;
```

definição da função  
pública quebraNome



```
END ManipulaNomes;
```

# Exemplo (2/4)

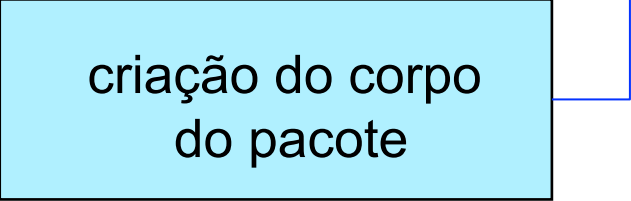
```
CREATE OR REPLACE PACKAGE BODY ManipulaNomes IS
```

```
FUNCTION quebraNome  
(nomeCompleto IN VARCHAR2)  
RETURN VARCHAR2  
IS
```

```
    primeiroNome VARCHAR2 (30);  
    achou BOOLEAN;  
    contador NUMBER (2);  
    caractere CHAR(1);
```

```
BEGIN
```

```
    primeiroNome := "";  
    achou := false;  
    contador := 1;
```



criação do corpo  
do pacote

# Exemplo (3/4)

WHILE not achou

LOOP

caractere := substr (NomeCompleto, contador, 1);

primeiroNome := primeiroNome || caractere;

IF caractere = ''

THEN achou := true;

ELSE contador := contador + 1;

END IF;

END LOOP;

RETURN primeiroNome;

END quebraNome;

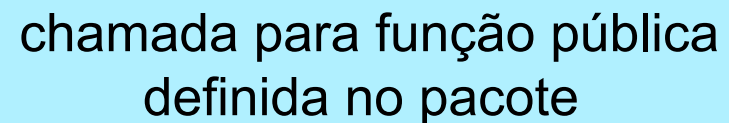
END;



finalização do pacote

# Exemplo (4/4)

```
set serveroutput on;
DECLARE
  CURSOR nomeCompleto IS SELECT dep_nome FROM dependente;
  variavelNomeCompleto nomeCompleto%RowType;
BEGIN
  OPEN nomeCompleto;
  LOOP
    FETCH nomeCompleto INTO variavelNomeCompleto;
    EXIT WHEN nomeCompleto%NotFound;
    dbms_output.put_line ('primeiro: ' || ManipulaNomes.quebraNome
                          (variavelNomeCompleto.dep_nome) );
  END LOOP;
  CLOSE nomeCompleto;
END;
```



chamada para função pública  
definida no pacote

# Remoção

- DROP PACKAGE nome\_pacote
  - elimina a especificação e o corpo
- DROP PACKAGE BODY nome\_pacote
  - elimina o corpo