

Disciplina:

**Laboratório de
Banco de Dados
Avançado**

Unidade 5:

Packages - Consultas complexas

Consultas de recuperação SQL complexas.



Conceitos

- *Packages*, são programas armazenados, tendo como diferencial a possibilidade de funcionar como repositório para agrupar vários objetos do tipo *procedure* e *function*;
- Pode conter código PL/SQL ou servir de área para definições de:
 - variáveis,
 - cursores,
 - exceções,
 - procedimentos
 - funções.
- Sua maior utilização está no agrupamento de programas que possuem uma mesma finalidade

Estrutura

- Constituído por até duas partes, uma chamada especification e outra chamada body.
- Podemos ter um package specification sem package body, contudo, não podemos ter o contrário.

Podemos ter no *especification*:

- Especificação de *procedures* e *functions*;
- Declaração de variáveis;
- Declaração de constantes;
- Declaração de cursores;
- Declaração de *exceptions*;
- Declaração de *types*.

Podemos ter no *body*:

- Códigos PL/SQL;
- Códigos de *procedures* e *functions*;
- Declaração de variáveis e constantes;
- Declaração de cursores;
- Declaração de *exceptions*;
- Declaração de *types*.



Exemplo prático com a utilização de *packages*

Criação da tabela *departments*

```
create table DEPARTMENTS (  
  deptno      number,  
  name        varchar2(50) not null,  
  location    varchar2(50),  
  constraint pk_departments primary key (deptno)  
);
```


Especificação do *package*

```
create or replace package pkg_exemplo as
--
  PROCEDURE exibir_mensagem;
  PROCEDURE exibir_valor_constante;
  PROCEDURE remove_department (numeroDepto NUMBER);
  PROCEDURE insert_department(var_numero number,var_nome varchar2,var_localizacao varchar);

  --Declaração de constante
  CONST_AUMENTO CONSTANT NUMBER := 10;
--
end;
```

Body do *package*

```
create or replace package body pkg_exemplo as
--
  procedure exibir_mensagem as
  begin
    DBMS_OUTPUT.PUT_LINE('Hello World!');
  end;

  procedure exibir_valor_constante as
  begin
    DBMS_OUTPUT.PUT_LINE('Valor da constante aumento: ' || CONST_AUMENTO);
  end;

  --criação de uma procedure com a utilização de parâmetro
  PROCEDURE remove_department (numeroDepto NUMBER) AS
  BEGIN
    DELETE FROM departments
    WHERE deptno = numeroDepto;
  END;

  --criação de uma procedure para inserção de dados
  PROCEDURE insert_department(var_numero number,var_nome varchar2,var_localizacao varchar) as
  BEGIN
    --validações para inserção
    insert into departments values(var_numero,var_nome,var_localizacao);
  END;
--
end;
```

Execução do package

```
begin
    pkg_exemplo.exibir_mensagem;
    pkg_exemplo.exibir_valor_constante;
    pkg_exemplo.insert_department(1, 'RH', 'São Paulo');
    pkg_exemplo.insert_department(2, 'Financeiro', 'São Paulo');
    pkg_exemplo.remove_department(1);
    pkg_exemplo.remove_department(2);
end;
```

Exercícios

1. Crie os exemplos propostos.
2. Adicione uma procedure ao package que liste os departamentos.
3. Adicione uma procedure ao *package* da rotina de alteração por id.
4. Adicione uma procedure ao *package* com uma consulta entre datas.

Contato

E-mail: allan@cruzeirosul.edu.br



Universidade **Cruzeiro do Sul**

www.cruzeirodosul.edu.br