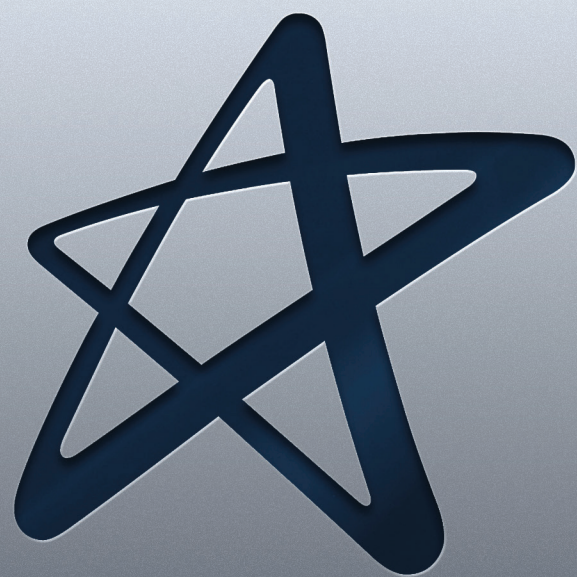


Banco de Dados



Cruzeiro do Sul Virtual
Educação a distância

Material Teórico



Cursores e procedures

Responsável pelo Conteúdo:

Prof. Ms. Alexander Gobbato Albuquerque

Revisão Textual:

Profa. Esp. Vera Lúcia de Sá Cicarone

UNIDADE

Cursors e procedures



- Introdução

- Conceito

- PL/SQL

- Cursors

- Stored procedure



Objetivo de APRENDIZADO

Program Language Sql é a linguagem de programação da Oracle que objetiva processar informações do banco de dados. Os cursores são áreas compostas de linhas e colunas em memória que servem para armazenar o resultado de uma seleção que retorna 0 ou mais linhas.

O Stored Procedure é um subprograma que é compilado e armazenado no servidor. Ele pode ser chamado a partir de um comando SQL qualquer.

Para nos tornarmos um profissional completo e resolvermos problemas ou executarmos tarefas complexas no banco de dados como por exemplo, realizar o cruzamentos de várias informações para geração de um relatório, com a utilização de cursores e procedures, essa tarefa poderia ser resolvida facilmente e a solução seria ficaria muito profissional.



Atenção

Para um bom aproveitamento do curso, leia o material teórico atentamente antes de realizar as atividades. É importante também respeitar os prazos estabelecidos no cronograma.

Contextualização

Program Language Sql é a linguagem de programação da Oracle que objetiva processar informações do banco de dados.



A imagem acima ilustra a inserção de comandos de programação (if, while, for) com SQL gerando o PL/SQL.

Os cursores são áreas compostas de linhas e colunas em memória que servem para armazenar o resultado de uma seleção que retorna 0 ou mais linhas.

O Stored Procedure é um subprograma que é compilado e armazenado no servidor. Ele pode ser chamado a partir de um comando SQL qualquer.

Introdução



PL/Sql

- Criar variáveis herdando o tipo e tamanho de outras variáveis ou objetos do banco de dados tais como tabelas;
- Criar cursores para tratamento de resultado de queries;
- Criar registros para armazenar resultado de cursores;
- Trata erros;
- Utiliza comandos de repetição e comparação;

Cursores

- Você usa um cursor para buscar linhas retornadas por uma consulta, você busca as linhas do cursor uma por vez.

Stored Procedure

- É um subprograma que é compilado e armazenado no servidor. Ele pode ser chamado a partir de um comando SQL qualquer

Conceito



Program Language Sql é a linguagem de programação da Oracle que objetiva processar informações do banco de dados.

Vantagens

- Portabilidade: qualquer máquina que rode o banco Oracle pode executar a mesma aplicação PL/SQL;
- Integração com o gerenciador de banco: visto que as variáveis podem ser definidas a partir da definição de colunas no banco, as alterações nele realizadas podem automaticamente (ou seja, sem manutenções), refletir nos blocos PL/SQL;
- Capacidade procedural: comandos de repetição, controle de fluxo e tratamento de erros;
- DECLARE (opcional)
 - Variáveis, cursores, constantes, tabelas, estruturas, exceptions

- BEGIN (obrigatório)
 - SQL statements
 - PL/SQL statements
- EXCEPTION (opcional)
 - Ações que deverão ser executadas quando ocorrer erros
- END; (obrigatório)

PL/SQL



Exemplo de declaração de variáveis

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
  Myname VARCHAR2(20);
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('My name is: ' || Myname);
```

```
  Myname := 'John';
```

```
  DBMS_OUTPUT.PUT_LINE('My name is: ' || Myname);
```

```
END;
```

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
  Myname VARCHAR2(20):= 'John';
```

```
BEGIN
```

```
  Myname := 'Steven';
```

```
  DBMS_OUTPUT.PUT_LINE('My name is: ' || Myname);
```

```
END;
```

ACCEPT: armazena o retorno do prompt;

PROMPT: exibe uma mensagem na tela;

ACCEPT empno PROMPT 'Digite o código do empregado '

SET serveroutput ON

DECLARE

v_empno NUMBER(6):= &empno;

v_emp_salary number (8,2);

BEGIN

SELECT salary INTO v_emp_salary FROM employees WHERE employee_id = v_empno;

dbms_output.put_line('O salário do funcionário ' || v_empno || ' é de ' || v_emp_salary);

END;

%TYPE – Herança de tipo e tamanho

- É usado para declarar uma variável com base em:
 - Um campo de uma tabela;
 - Outra variável já declarada

set serveroutput on

DECLARE

v_emp_hiredate employees.hire_date%TYPE;

v_emp_salary employees.salary%TYPE;

BEGIN

SELECT hire_date, salary

INTO v_emp_hiredate, v_emp_salary

FROM employees

WHERE employee_id = 100;

dbms_output.put_line(v_emp_salary || ' ' || v_emp_hiredate);

END;

As variáveis
herdam o tipo de
dato da tabela

Exemplo1: Incluindo dados na tabela de empregados.

BEGIN

INSERT INTO employees

(employee_id, first_name, last_name, email, hire_date, job_id, salary)

VALUES

(employees_seq.NEXTVAL, 'Ruth', 'Cores', 'RCORES',sysdate, 'AD_ASST', 4000);

END;

Exemplo2: Update: Atualizando o salário para 800 para todos que possuem o cargo = ST_CLERK

```
DECLARE
    sal_increase employees.salary%TYPE := 800;
BEGIN
    UPDATE employees
    SET      salary = salary + sal_increase
    WHERE   job_id = 'ST_CLERK';
END;
```

Exemplo3: Delete: Apaga todos os funcionários do departamento 10.

```
DECLARE
    deptno employees.department_id%TYPE := 10;
BEGIN
    DELETE FROM employees
    WHERE department_id = deptno;
END;
```

Exemplo4: If/else IF

```
DECLARE
    myage number:=31;
BEGIN
    IF myage < 11
    THEN
        DBMS_OUTPUT.PUT_LINE(' Sou criança');
    ELSIF myage < 20
    THEN
        DBMS_OUTPUT.PUT_LINE(' Sou jovem');
    ELSIF myage < 30
    THEN
        DBMS_OUTPUT.PUT_LINE(' Estou na casa dos vinte..');
```

```
ELSIF myage < 40
  THEN
    DBMS_OUTPUT.PUT_LINE(' Estou na casa dos trinta');
ELSE
  DBMS_OUTPUT.PUT_LINE(' Serei sempre jovem...');
END IF;
END;
```

Cursors



Você usa um cursor para buscar linhas retornadas por uma consulta;

Você busca as linhas do cursor uma por vez.

Temos 2 tipos de cursors:

- Implícitos = > declarado e gerenciado pelo oracle
- Explícitos = > declarado e gerenciado pelo programador

A sequencia de um bloco PL SQL deve conter

Declare

Onde criamos uma área SQL

Na sequência

Open Onde identificamos o grupo ativo.

Na sequência Fetch para carregar a linha corrente nas variáveis.

Na sequência caso esteja vazio fechamos o grupo ativo caso contrário voltamos em Fetch.

Declarando cursor

DECLARE

CURSOR emp_cursor IS

SELECT employee_id, last_name FROM employees

WHERE department_id =30;

Abrindo o cursor

DECLARE

CURSOR emp_cursor IS

SELECT employee_id, last_name FROM employees

WHERE department_id =30;

...

BEGIN

OPEN emp_cursor;

```
SET SERVEROUTPUT ON
DECLARE
CURSOR emp_cursor IS
SELECT employee_id, last_name FROM employees
WHERE department_id =30
v_empno employees.employee_id%TYPE;
v_l_name employees.Last_name%type;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO v_empno, v_lane;
    EXIT WHEN emp_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_empno || ' ' || v_lane)
    END LOOP;
    CLOSE emp_cursor;
END;
```

Stored procedure



- São Blocos PL/SQL armazenados no banco de dados de forma compilada e que podem ser chamados no SQL *PLUS;
- Podem ou não receber parâmetros. Os parâmetros passados para os subprogramas podem ser de 3 tipos:
 - IN (padrão) passa um valor do ambiente chamador para o subprograma e esse valor não pode ser alterado dentro do subprograma.
 - OUT – passa um valor do subprograma para o ambiente chamador
 - IN OUT passa um valor do ambiente chamador para o subprograma. Esse valor pode ser alterado dentro do subprograma e retornado com o valor atualizado para o ambiente chamador.

Exemplo:

```
CREATE OR REPLACE PROCEDURE Nome_Proc
```

```
  (Argumento1 modo Tipo_de_Dados,
```

```
  Argumento2 modo Tipo_de_Dados,
```

```
  Argumento3 modo Tipo_de_Dados)
```

```
IS ou AS
```

```
  Variáveis locais, constantes,...
```

```
Begin
```

```
  Bloco PL/SQL
```

```
  Exception
```

```
End nome_da_procedure;
```

Exemplo de passagem de parâmetros em procedimentos

```
Set serveroutput on
```

```
Create or replace procedure Nr_Fone (V_fone in varchar2)
```

```
IS
```

```
  NV_Fone varchar2(10);
```

Begin

NV_Fone:= Substr(V_fone,1,4)||'-'||Substr(V_fone,5,4);

Dbms_Output.Put_Line (NV_Fone);

End Nr_Fone;

Executando a procedure

exec nr_fone(12345678);

Exemplo de passagem de parâmetros em procedimentos

Create or replace procedure Testa_Parametros

(X IN number,

Y OUT number,

Z IN OUT number)

IS

Begin

Y:=X*2;

Y:= Y+Z;

Z:=Y+X+Z;

End Testa_Parametros;

Executando a procedure

Set serveroutput on

DECLARE

Var1 number;

Var2 number;

Var3 number;

Begin

Var1:=30;

Var2:=40;

Var3:=50;

Dbms_Output.Put_Line(Var1||'-'||Var2||'-'||Var3);

Testa_Parametros(Var1,Var2,Var3);

Dbms_Output.Put_Line(Var1||'-'||Var2||'-'||Var3);

End;

Material Complementar



Para fixar o conteúdo de procedures e function, um ótimo artigo disponível no site:

- http://www.linhadecodigo.com.br/artigo/335/pl_sql-procedures-e-funcoes.aspx

Verifique o link:

- http://www.linhadecodigo.com.br/artigo/335/pl_sql-procedures-e-funcoes.aspx

Referências

FANDERUFF, D. **Dominando o Oracle 9i**: modelagem e desenvolvimento. São Paulo: Pearson Education do Brasil, 2003.

MATERIAL OFICIAL da Oracle no curso. **Oracle Database 10 G**: SQL Fundamentals I.



Cruzeiro do Sul
Educatonal