

FIAP GRADUAÇÃO

Database Application Development

Prof. Gustavo Calixto

AULA DE HOJE

- Stored Procedures
 - Tipos de chamadas para procedures
 - IN
 - OUT
 - INOUT
 - Compilação da Procedure no Banco de Dados
 - Exemplos de procedures com atribuições, estruturas de condição e laços.

Stored Procedures

STORED PROCEDURES

- São subprogramas que podem ser executados utilizando os recursos computacionais de um banco de dados.
- Vantagem
 - Repassa o consumo de processamento dos dispositivos de visualização.
 - Modificações na manipulação dos dados não fica vinculado aos dispositivos.
- Desvantagem
 - Excesso de procedures pode onerar o esforço computacional do BD para outras operações.

STORED PROCEDURES

```
CREATE OR REPLACE PROCEDURE nome_da_procedure
    (<lista de parâmetros>) IS

V1    TIPO_DE_V1;      }
V2    TIPO_DE_V2;      } declaração de variáveis
...    }
Vn    TIPO_DE_Vn;      }

BEGIN
    <comandos>
END nome_da_procedure;
```

Fonte: DevMedia

STORED PROCEDURES

- Procedures podem ser compiladas utilizando o Oracle SQL Developer.
- Depois de compilada, a mesma pode ser invocada de qualquer aplicação ou internamente ao banco de dados.

```
create or replace procedure HELLO_WORLD IS  
BEGIN  
    DBMS_OUTPUT.PUT_LINE('Minha primeira aula de Procedure');  
END;
```

STORED PROCEDURES

- Procedure com parâmetro de entrada (IN)

```
create or replace PROCEDURE HELLO_WORLD_IN(param1 IN VARCHAR2)
IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Minha primeira aula de ' || param1);
END;
```


STORED PROCEDURES

- Procedure com parâmetro de saída (OUT)

```
create or replace PROCEDURE hello_world_out(outParam1 OUT VARCHAR2)
IS
BEGIN
    outParam1 := 'Minha saída da procedure';
END;
```

STORED PROCEDURES

- Procedure com parâmetro de entrada e saída (INOUT)

```
create or replace PROCEDURE hello_world_inout(entrada IN OUT VARCHAR2)
IS
BEGIN
    entrada := 'Estou dizendo ao mundo minha saída com a entrada' || entrada;
END;
```

STORED PROCEDURES

- Procedure com uso de estrutura condicional (IF)

```
create or replace PROCEDURE PAR_IMPAR (NUMERO IN NUMBER) AS
BEGIN
    --ESTRUTURA CONDICIONAL
    IF numero MOD 2 = 0 THEN
        DBMS_OUTPUT.PUT_LINE('PAR');
    ELSE
        DBMS_OUTPUT.PUT_LINE('IMPAR');
    END IF;
END PAR_IMPAR;
```

STORED PROCEDURES

- Procedure com uso de laço while,

```
create or replace PROCEDURE EXECUTA_WHILE (REPETIC IN NUMBER) AS
BEGIN
    DECLARE
        CONT NUMBER(2) := 0;
    BEGIN
        WHILE CONT <= REPETIC LOOP
            DBMS_OUTPUT.PUT_LINE (CONT);
            CONT := CONT + 1;
        END LOOP;
    END;
END EXECUTA_WHILE;
```

STORED PROCEDURES

- Procedure com uso de laço FOR

```
create or replace PROCEDURE EXECUTA_FOR (REPETIC IN NUMBER) AS
BEGIN
    DECLARE
        CONT NUMBER(2) := REPETIC;
    BEGIN
        FOR I IN 1..CONT
        LOOP
            DBMS_OUTPUT.PUT_LINE (I);
        END LOOP;

        FOR I IN REVERSE 1..CONT
        LOOP
            DBMS_OUTPUT.PUT_LINE (I);
        END LOOP;

    END;
END EXECUTA_FOR;
```

STORED PROCEDURES

- Exercícios
 - Uma procedure que informa a idade de uma pessoa a partir do ano de nascimento.
 - Uma procedure que informa a faixa de IMC de um individuo.

Copyright © 2019 Prof. Gustavo Moreira Calixto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).