

Mãos na massa: Packages

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) Em **user_dev**, crie a parte de especificação do *package*, com apenas os cabeçalhos das *procedures* ou *functions* a serem vistas no mundo externo:

```
CREATE OR REPLACE PACKAGE CLIENTE_PAC IS

    PROCEDURE ATUALIZAR_CLI_SEG_MERCADO
    (p_id in cliente.id%type,
      p_segmercado_id in cliente.segmercado_id%type);

    PROCEDURE ATUALIZAR_FATURAMENTO_PREVISTO
    (p_id in cliente.id%type,
      p_faturamento_previsto in cliente.faturamento_previsto%type);

    PROCEDURE EXCLUIR_CLIENTE
    (p_id in cliente.id%type);

    PROCEDURE INCLUIR_CLIENTE
    (p_id in cliente.id%type,
      p_razao_social in cliente.razao_social%type,
      p_CNPJ in cliente.CNPJ%type ,
      p_segmercado_id in cliente.segmercado_id%type,
      p_faturamento_previsto in cliente.faturamento_previsto%type);

END;
```

2) Novamente em **user_dev**, crie o corpo do *package*:

```
CREATE OR REPLACE PACKAGE BODY cliente_pac IS

    PROCEDURE atualizar_cli_seg_mercado (
        p_id          IN cliente.id%TYPE,
        p_segmercado_id IN cliente.segmercado_id%TYPE
    ) IS
        e_fk EXCEPTION;
        PRAGMA exception_init ( e_fk,-2291 );
        e_no_update EXCEPTION;
    BEGIN
        UPDATE cliente
            SET
                segmercado_id = p_segmercado_id
        WHERE
            id = p_id;

        IF
            SQL%notfound
        THEN
            RAISE e_no_update;
        END IF;
```

```
        COMMIT;
    EXCEPTION
        WHEN e_fk THEN
            raise_application_error(-20001,'Segmento de Mercado Inexistente');
        WHEN e_no_update THEN
            raise_application_error(-20002,'Cliente Inexistente');
END;
```

```
PROCEDURE atualizar_faturamento_previsto (
    p_id                IN cliente.id%TYPE,
    p_faturamento_previsto IN cliente.faturamento_previsto%TYPE
) IS
    v_categoria    cliente.categoria%TYPE;
    e_error_id EXCEPTION;
BEGIN
    v_categoria := obter_categoria_cliente(p_faturamento_previsto);
    UPDATE cliente
        SET
            categoria = v_categoria,
            faturamento_previsto = p_faturamento_previsto
    WHERE
        id = p_id;

    IF
        SQL%notfound
    THEN
        RAISE e_error_id;
    END IF;
    COMMIT;
EXCEPTION
    WHEN e_error_id THEN
        raise_application_error(-20010,'Cliente inexistente');
END;
```

```
PROCEDURE excluir_cliente (
    p_id IN cliente.id%TYPE
) IS
    e_error_id EXCEPTION;
BEGIN
    DELETE FROM cliente WHERE
        id = p_id;

    IF
        SQL%notfound
    THEN
        RAISE e_error_id;
    END IF;
    COMMIT;
EXCEPTION
    WHEN e_error_id THEN
        raise_application_error(-20010,'Cliente inexistente');
END;
```

```
PROCEDURE incluir_cliente (
    p_id                IN cliente.id%TYPE,
    p_razao_social      IN cliente.razao_social%TYPE,
    p_cnpj              IN cliente.cnpj%TYPE,
    p_segmercado_id     IN cliente.segmercado_id%TYPE,
```

```

    p_faturamento_previsto    IN cliente.faturamento_previsto%TYPE
) IS
    v_categoria                cliente.categoria%TYPE;
    v_cnpj                     cliente.cnpj%TYPE := p_cnpj;
    v_codigo_erro              NUMBER(5);
    v_mensagem_erro            VARCHAR2(200);
    v_dummy                    NUMBER;
    v_verifica_segmento        BOOLEAN;
    e_segmento                  EXCEPTION;
BEGIN
    v_verifica_segmento := verifica_segmento_mercado(p_segmercado_id);
    IF
        v_verifica_segmento = false
    THEN
        RAISE e_segmento;
    END IF;
    v_categoria := obter_categoria_cliente(p_faturamento_previsto);
    format_cnpj(v_cnpj);
    INSERT INTO cliente VALUES (
        p_id,
        upper(p_razao_social),
        v_cnpj,
        p_segmercado_id,
        SYSDATE,
        p_faturamento_previsto,
        v_categoria
    );

    COMMIT;
EXCEPTION
    WHEN dup_val_on_index THEN
        raise_application_error(-20010,'Cliente já cadastrado');
    WHEN e_segmento THEN
        raise_application_error(-20011,'Segmento de mercado inexistente');
    WHEN OTHERS THEN
        v_codigo_erro := sqlcode;
        v_mensagem_erro := sqlerrm;
        raise_application_error(-20000,TO_CHAR(v_codigo_erro)
        || v_mensagem_erro);
END;

END;
```

3) Ainda em **user_dev**, dê permissão de execução do pacote ao usuário **user_app**:

```
GRANT EXECUTE ON CLIENTE_PAC TO user_app;
```

4) Em **user_app**, insira um cliente utilizando o pacote:

```
EXEC user_dev.cliente_pac.incluir_cliente(10 , 'INCLUIDO POR USER_APP VIA PACKAGE', '24890', 2, 100000);
```

5) Volte a **user_dev** e crie um sinônimo para **CLIENTE_PAC** :

```
CREATE PUBLIC SYNONYM CLIENTE_PAC FOR user_dev.CLIENTE_PAC;
```

6) Em **user_app**, insira um cliente, dessa vez utilizando o sinônimo do pacote:

```
EXEC cliente_pac.incluir_cliente(11 , 'SEGUNDO INCLUIDO POR USER_APP VIA PACKAGE' , '24890' , 2 , 100000);
```



7) Por fim, ainda em **user_app**, exclua o cliente de ID 10:

```
EXEC cliente_pac.excluir_cliente(10)
```