

Entrega | Semana 4 | Fabiana Vilela de Lima Moura

1. Criar rocedure Inseere_projeto:

- Crie o arquivo **sql/plsql.sql**;
- Crie a *procedure* **brh.inseere_projeto** para cadastrar um novo projeto na base de dados:
 - **Parâmetros da procedure:**
 - *Nome do projeto*: *varchar* com nome do novo projeto.
 - *Responsável do projeto*: *varchar* com a matrícula do colaborador responsável.
- Faça *commit* do arquivo.

Resposta:

```
+ show user;

+ grant alter user to brh;

+ ALTER USER BRH QUOTA UNLIMITED ON USERS;

+ GRANT connect, resource TO brh;

+ alter user brh account lock;

+ SELECT * FROM BRH.PROJETO;

+ CREATE OR REPLACE PROCEDURE brh.inseere_projeto (p_NOME
BRH.PROJETO.NOME%type, p_RESPONSAVEL BRH.PROJETO.RESPONSAVEL%type)

+ IS

+ BEGIN

+ INSERT INTO BRH.PROJETO(NOME,RESPONSAVEL,INICIO)

+ VALUES(p_NOME, p_RESPONSAVEL,sysdate);

+ COMMIT;

+ END;

+ EXECUTE brh.inseere_projeto('NOV2', 'T123');
```

2. Criar função calcula_idade :

- Crie a *function* **brh.calcula_idade**, que informa a idade a partir de uma data:
 - **Parâmetros da *function*:**
 - *Data: date* com a data de referência para calcular a idade.
 - **Retorno da *function*:**
 - Deve retornar um número inteiro com a idade.
- Adicione o código no arquivo **sql/plsql.sql**;
- Faça *commit* do arquivo.

```
+ CREATE OR REPLACE FUNCTION brh.calcula_idade(p_data_nascimento DATE)
+ RETURN NUMBER
+ IS
+   v_idade NUMBER;
+ BEGIN
+   SELECT TRUNC(months_between(sysdate, p_data_nascimento) / 12)
+   INTO v_idade
+   FROM DUAL;
+   RETURN v_idade;
+ END;
+ SELECT brh.calcula_idade(TO_DATE('10-12-1970', 'DD-MM-YYYY')) AS IDADE
+ FROM DUAL;
+ END;
+ EXECUTE brh.insere_projeto('NOV2', 'T123');
+ -- idade
+ CREATE OR REPLACE FUNCTION brh.calcula_idade(p_data_nascimento DATE)
+ RETURN NUMBER
+ SELECT brh.calcula_idade(TO_DATE('10-12-1970', 'DD-MM-YYYY')) AS IDADE
FROM DUAL;
```

3. Criar função finaliza_projeto:

- Crie a *function* **brh.finaliza_projeto** para registrar o término da execução de um projeto:
 - **Parâmetros da *function*:**
 - *ID do projeto: number* com identificador do projeto a ser finalizado.
 - **Retorno da *function*:**
 - Deve retornar a data de finalização atribuída ao projeto.
- Adicione o código no arquivo **sql/plsql.sql**;
- Faça *commit* do arquivo.

```
+ SELECT * FROM BRH.PROJETO;

+ -- finaliza projeto

+ CREATE OR REPLACE FUNCTION brh.finaliza_projeto(p_id_projeto
brh.projeto.id%type)
+ RETURN brh.projeto.fim%type
+ IS
+ v_data_fim brh.projeto.fim%type;
+ BEGIN
+ UPDATE brh.projeto
+ SET brh.projeto.fim = SYSDATE
+ WHERE id = p_id_projeto AND brh.projeto.fim IS NULL;
+ v_data_fim := SYSDATE;
+ COMMIT;
+ RETURN v_data_fim;
+ END;

+ BEGIN
+ brh.finaliza_projeto(47);
+ END;
```

4. Validar novo projeto:

- Altere a *procedure* **brh.inserir_projeto** para não permitir cadastrar projetos inválidos;
- Adicione o código no arquivo **sql/plsql.sql**;
- Faça *commit* do arquivo.

```
+ CRIAR OU SUBSTITUIR PROCEDIMENTO brh . inserir_projeto
+ (p_NOME BRH . PROJETO .NOME%tipo, p_RESPONSAVEL BRH . PROJETO
  .RESPONSAVEL%tipo)
+ É
+ COMEÇAR
+ SE COMPRIMENTO(p_NOME) <= 2 ENTÃO
+   SGBD_OUTPUT . PUT_LINE ( ' O nome do projeto deve ter mais de 2
    caracteres. ' );
+ OUTRO
+ INSERIR NO BRH . PROJETO (NOME, RESPONSÁVEL, INICIO)
+ VALORES (p_NOME, p_RESPONSAVEL, SYSDATE );
+ COMEÇAR ;
+ SGBD_OUTPUT . PUT_LINE ( ' Projeto cadastrado com sucesso. ' );
+ FIM SE;
+ FIM;
+ EXECUTAR, br . inserir_projeto ( ' teste ' , ' T123 ' );
```