



Faculdade de Informática e Administração Paulista

DATABASE APPLICATION & DATA SCIENCE

1º CheckPoint – 2º Semestre

RM 93613 – VINICIUS DE OLIVEIRA

RM: 94311 - MARCOS BILOBRAM

RM: 96320 - NATHÁLIA MAIA

RM: 94972 - RAFAELA DA SILVA

Sumário

1 - O código fonte atualizado do projeto (Python)	3
2 - Os códigos fontes dos procedimentos criados.....	10
3 - O código fonte da Package e Package Body criadas	13
4 - O Código fonte do Trigger criado	16
5 - Prints da evidência do funcionamento de cada uma das ações propostas no script do Python	19

Link do projeto no GitHub: <https://github.com/ViniOlr/CheckPoint1-2Semestre-DBA>

1 - O código fonte atualizado do projeto (Python)

```
#
=====
=== #
#
#
# Antes de executar o Script Python, instale as Bibliotecas
necessárias #
# Execute no
Terminal: #
#
#
# pip install
oracledb #
# pip install
pandas #
#
#
#
=====
=== #
#
# Importação dos módulos
#
import os
import platform
import oracledb
import pandas as pd

# Try para tentativa de Conexão com o Banco de Dados Oracle
try:

    # Conectar ao banco de dados

    #
=====
===== #
    # CUIDADO!!! ==>>> COM 3 TENTATIVAS ERRADAS SUA CONTA FICA EM LOCK
    (Solicitar ao Help Desk via Whats para desbloquear)
    #
=====
===== #

    conn = oracledb.connect(user="rm93613", password="150503",
dsn="oracle.fiap.com.br:1521/orcl")
```

```

# Cria as instruções para cada módulo
inst_cadastro = conn.cursor()
inst_consulta = conn.cursor()
inst_alteracao = conn.cursor()
inst_exclusao = conn.cursor()

except Exception as e:
    # Informa o erro
    print("Erro: ", e)
    # Flag para não executar a Aplicação
    conexao = False
else:
    # Flag para executar a Aplicação
    conexao = True

margem = ' ' * 4 # Define uma margem para a exibição da aplicação

# Recupera o Sistema Operacional do Cliente
sist_oper = platform.system()

# Enquanto o flag conexao estiver apontado com True
while conexao:

    # Limpa a tela via SO - Windows: cls   Mac / Linux: clear
    if sist_oper == 'Windows':
        os.system('cls')
    else:
        os.system('clear')

    # Apresenta o menu
    print("----- CRUD - PETSHOP -----")
    print("""
1 - Cadastrar Pet
2 - Listar Pets por ID
3 - Listar Todos os Pets
4 - Alterar Pet
5 - Excluir Pet
6 - EXCLUIR TODOS OS PETS
7 - SAIR
""")

    # Captura a escolha do usuário
    escolha = input(margem + "Escolha -> ")

    # Verifica se o número digitado é um valor numérico
    if escolha.isdigit():
        escolha = int(escolha)
    else:
        escolha = 7

```

```

        print("Digite um número.\nReinicie a Aplicação!")

# Limpa a tela via SO
if sist_oper == 'Windows':
    os.system('cls')
else:
    os.system('clear')

# VERIFICA QUAL A ESCOLHA DO USUÁRIO

# CADASTRAR UM PET
if escolha == 1:

    print("----- CADASTRAR PET ----- \n")

    # Recebe os valores para cadastro
    tipo = input(margem + "Digite o tipo....: ")
    nome = input(margem + "Digite o nome....: ")
    idade = int(input(margem + "Digite a idade...: "))

    # Chama a procedure para inserir um novo Pet - SEM SE PREOCUPAR
    # COM A ORDEM DOS PARÂMETROS

    # inst_cadastro.callproc('CP1_CADASTRO_PET',
    # keywordParameters={"p_tipoPet": tipo, "p_nomePet": nome,
    # "p_nomePet": idade})

    inst_cadastro.callproc('CP1_CADASTRO_PET',
        parameters=[tipo, nome, idade])

    conn.commit()

    #
    # Para realizar um Roolback, caso necessário:
    #
    #conn.rollback()

    # Caso haja sucesso na gravação
    print("\n##### Dados GRAVADOS #####")

# LISTAR PETS POR ID
elif escolha == 2:

    print("----- LISTAR PET POR ID----- \n")

    # Permite o usuário escolher um Pet pelo id
    pet_id = int(input(margem + "Escolha um Id: "))

```

```

        # Chama a Função para Recuperar um Pet (Uma linha somente)

        lista = inst_consulta.callfunc('CP1_GET_PET',
                                       str, ## int para retorno com NUMBER na
Função. Em nosso caso retorna uma String (VARCHAR2)
                                       [pet_id])

        print(lista)

        print("\n##### LISTADO! #####")

# LISTAR TODOS OS PETS
elif escolha == 3:

    # Lista para a captura de dados do Banco
    lista_dados = []

    # Monta a instrução SQL de seleção de todos os registros da
tabela
    inst_consulta.execute('SELECT * FROM petshop')
    # Captura todos os registros da tabela e armazena no objeto data
    data = inst_consulta.fetchall()

    # Insere os valores da tabela na Lista
    for dt in data:
        lista_dados.append(dt)

    # ordena a lista
    lista_dados = sorted(lista_dados)

    # Gera um DataFrame com os dados da lista utilizando o Pandas
    dados_df = pd.DataFrame.from_records(lista_dados, columns=['Id',
'Tipo', 'Nome', 'Idade'], index='Id')

    # Verifica se não há registro através do dataframe
    if dados_df.empty:
        print(f"Não há um Pets cadastrados!")
    else:
        print(dados_df) # Exibe os dados selecionados da tabela

    print("\n##### LISTADOS! #####")

# ALTERAR OS DADOS DE UM REGISTRO
elif escolha == 4:

    # ALTERANDO UM REGISTRO
    print("----- ALTERAR DADOS DO PET -----\\n")

```

```

# Lista para a captura de dados da tabela
lista_dados = []

# Permite o usuário escolher um Pet pelo id
pet_id = int(input(margem + "Escolha um Id: "))

# Constroi a instrução de consulta para verificar a existencia ou
não do id
consulta = f""" SELECT * FROM petshop WHERE id = {pet_id}"""
inst_consulta.execute(consulta)
data = inst_consulta.fetchall()

# Preenche a lista com o registro encontrado (ou não)
for dt in data:
    lista_dados.append(dt)

# analisa se foi encontrado algo
if len(lista_dados) == 0: # se não há o id
    print(f"Não há um pet cadastrado com o ID = {pet_id}")
    input("\nPressione ENTER")
else:
    # Captura os novos dados
    novo_tipo = input(margem + "Digite um novo tipo: ")
    novo_nome = input(margem + "Digite um novo nome: ")
    nova_idade = input(margem + "Digite uma nova idade: ")

    # Chama a procedure para atualizar um novo Pet - SEM SE
    PREOCUPAR COM A ORDEM DOS PARÂMETROS
    # inst_alteracao.callproc('nomeProcedure',
    # keywordParameters={"nomeParam1_naProc": param1,
    "nomeParam2_naProc": param2, "nomeParam3_naProc": param3,
    "nomeParam4_naProc": param4})

    inst_cadastro.callproc('CP1_ALTERA_PET',
        parameters=[pet_id, novo_tipo, novo_nome, nova_idade])

    conn.commit()

    if len(lista_dados) == 0:
        print("\n##### ID NÃO ENCONTRADO! #####")
    else:
        print("\n##### Dados ATUALIZADOS! #####")

# EXCLUIR UM REGISTRO
elif escolha == 5:

    print("----- EXCLUIR PET -----\\n")

```

```

# Lista para a captura de dados da tabela
lista_dados = []

# Permite o usuário escolher um Pet pelo ID
pet_id = input(margem + "Escolha um Id: ")

if pet_id.isdigit():
    pet_id = int(pet_id)
    consulta = f"SELECT * FROM petshop WHERE id = {pet_id}"
    inst_consulta.execute(consulta)
    data = inst_consulta.fetchall()

    # Insere os valores da tabela na lista
    for dt in data:
        lista_dados.append(dt)

    # Verifica se o registro está cadastrado
    if len(lista_dados) == 0:
        print(f"Não há um pet cadastrado com o ID = {pet_id}")
    else:

        # Chama a procedure para deletar um novo Pet - SEM SE
        # PREOCUPAR COM A ORDEM DOS PARÂMETROS
        inst_exclusao.callproc('CP1_EXCLUI_PET',
parameters=[pet_id] )

        conn.commit()
        print("\n##### Pet APAGADO! #####") # Exibe mensagem
        caso haja sucesso
    else:
        print("O Id não é numérico!")

# EXCLUIR TODOS OS REGISTROS
elif escolha == 6:

    print("\n!!!! EXCLUI TODOS OS DADOS TABELA !!!!!\n")

    confirma = input(margem + "CONFIRMA A EXCLUSÃO DE TODOS OS PETS?
[S]im ou [N]ÃO?")

    if confirma.upper() == "S":

        # Chama a procedure para limpar a tabela
        inst_exclusao.callproc('CP1_EXCLUI_TODOS') ## A Procedure não
        recebe parâmetros

        conn.commit()

```


2 - Os códigos fontes dos procedimentos criados

```
SET SERVEROUTPUT ON

-- Uma Procedure para Cadastrar um Pet

CREATE OR REPLACE PROCEDURE CP1_CADASTRO_PET (
    p_tipoPet IN VARCHAR,
    p_nomePet IN VARCHAR,
    p_idade IN NUMBER
) AS
BEGIN

    INSERT INTO PETSHOP (TIPO_PET, NOME_PET, IDADE)
    VALUES (p_tipoPet, p_nomePet, p_idade);

END;

-- Uma Função para recuperar dados de um Pet com base no parâmetro ID.

CREATE OR REPLACE FUNCTION CP1_GET_PET(id_pet NUMBER)
RETURN VARCHAR2
IS
    dados_pet VARCHAR2(4000);
BEGIN

    SELECT
        'ID: ' || ID || CHR(10) ||
        'Nome: ' || NOME_PET || CHR(10) ||
        'Tipo: ' || TIPO_PET || CHR(10) ||
        'Idade: ' || IDADE
    INTO dados_pet
    FROM PETSHOP
    WHERE ID = id_pet;

    IF dados_pet IS NOT NULL THEN
        RETURN dados_pet;
    ELSE
        RETURN 'Pet não encontrado para o ID ' || id_pet;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 'Pet não encontrado para o ID ' || id_pet;
END;
```

```

-- Uma Procedure para alterar dados de um Pet com base no parâmetro ID

CREATE OR REPLACE PROCEDURE CP1_ALTERA_PET (
    p_id IN NUMBER,
    p_tipoPet IN VARCHAR,
    p_nomePet IN VARCHAR,
    p_idade IN NUMBER
)
AS
BEGIN

    UPDATE PETSHOP
    SET TIPO_PET = p_tipoPet,
    NOME_PET = p_nomePet,
    IDADE = p_idade
    WHERE ID = p_id;

END;

-- Uma Procedure para excluir um Pet com base no parâmetro ID

CREATE OR REPLACE PROCEDURE CP1_EXCLUI_PET (
    p_id IN NUMBER
)
AS
BEGIN

    DELETE PETSHOP
    WHERE ID = p_id;

END;

-- Uma Procedure para excluir todos as linhas da Tabela

CREATE OR REPLACE PROCEDURE CP1_EXCLUI_TODOS
AS
BEGIN

    DELETE PETSHOP;

END;

```

Evidência:

The screenshot shows a software interface with a top toolbar containing icons for file operations, execution, and formatting. Below the toolbar is a tabbed interface with two tabs: 'Planilha' and 'Query Builder'. The 'Query Builder' tab is active, displaying a SQL script. The script includes several SQL statements for creating and replacing procedures and functions, with comments in Portuguese. The last statement, 'CREATE OR REPLACE PROCEDURE CP1_EXCLUI_TODOS...', is highlighted in yellow. Below the script editor is a 'Saída do Script' (Script Output) window. It shows a status bar with icons and the text 'Tarefa concluída em 0,093 segundos'. The output area lists the compilation status for each object: 'Procedure CP1_CADASTRO_PET compilado', 'Function CP1_GET_PET compilado', 'Procedure CP1_ALTERA_PET compilado', 'Procedure CP1_EXCLUI_PET compilado', and 'Procedure CP1_EXCLUI_TODOS compilado'.

```
SET SERVEROUTPUT ON

-- Uma Procedure para Cadastrar um Pet

+ CREATE OR REPLACE PROCEDURE CP1_CADASTRO_PET (...

-- Uma Função para recuperar dados de um Pet com base no parâmetro ID.

+ CREATE OR REPLACE FUNCTION CP1_GET_PET(id_pet NUMBER)...

-- Uma Procedure para alterar dados de um Pet com base no parâmetro ID

+ CREATE OR REPLACE PROCEDURE CP1_ALTERA_PET (...

-- Uma Procedure para excluir um Pet com base no parâmetro ID

+ CREATE OR REPLACE PROCEDURE CP1_EXCLUI_PET (...

-- Uma Procedure para excluir todos as linhas da Tabela

+ CREATE OR REPLACE PROCEDURE CP1_EXCLUI_TODOS...
```

Saída do Script x

Tarefa concluída em 0,093 segundos

Procedure CP1_CADASTRO_PET compilado

Function CP1_GET_PET compilado

Procedure CP1_ALTERA_PET compilado

Procedure CP1_EXCLUI_PET compilado

Procedure CP1_EXCLUI_TODOS compilado

3 - O código fonte da Package e Package Body criadas

```

CREATE OR REPLACE PACKAGE RM93613_PKG_PETS AS

    PROCEDURE CP1_CADASTRO_PET(p_tipoPet VARCHAR, p_nomePet VARCHAR,
p_idade NUMBER);
    FUNCTION CP1_GET_PET(p_id NUMBER) RETURN VARCHAR2;
    PROCEDURE CP1_ALTERA_PET(p_id NUMBER, p_tipoPet VARCHAR, p_nomePet
VARCHAR, p_idade NUMBER);
    PROCEDURE CP1_EXCLUI_PET(p_id NUMBER);
    PROCEDURE CP1_EXCLUI_TODOS;

END RM93613_PKG_PETS;

CREATE OR REPLACE PACKAGE BODY RM93613_PKG_PETS AS

    -- Uma Procedure para Cadastrar um Pet

    PROCEDURE CP1_CADASTRO_PET (
        p_tipoPet IN VARCHAR,
        p_nomePet IN VARCHAR,
        p_idade IN NUMBER
    ) IS
    BEGIN

        INSERT INTO PETSHOP (TIPO_PET, NOME_PET, IDADE)
        VALUES (p_tipoPet, p_nomePet, p_idade);

        commit;

    END CP1_CADASTRO_PET;

    -- Uma Função para recuperar dados de um Pet com base no parâmetro
ID.

    FUNCTION CP1_GET_PET(p_id NUMBER)
    RETURN VARCHAR2
    IS
        dados_pet VARCHAR2(4000);
    BEGIN

        SELECT
            'ID: ' || ID || CHR(10) ||
            'Nome: ' || NOME_PET || CHR(10) ||
            'Tipo: ' || TIPO_PET || CHR(10) ||
            'Idade: ' || IDADE

```

```

        INTO dados_pet
        FROM PETSHOP
        WHERE ID = p_id;

        IF dados_pet IS NOT NULL THEN
            RETURN dados_pet;
        ELSE
            RETURN 'Pet não encontrado para o ID ' || p_id;
        END IF;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RETURN 'Pet não encontrado para o ID ' || p_id;
    END CP1_GET_PET;

```

-- Uma Procedure para alterar dados de um Pet com base no parâmetro ID

```

PROCEDURE CP1_ALTERA_PET (
    p_id IN NUMBER,
    p_tipoPet IN VARCHAR,
    p_nomePet IN VARCHAR,
    p_idade IN NUMBER
)
IS
BEGIN

    UPDATE PETSHOP
    SET TIPO_PET = p_tipoPet,
        NOME_PET = p_nomePet,
        IDADE = p_idade
    WHERE ID = p_id;

    commit;

END CP1_ALTERA_PET;

```

-- Uma Procedure para excluir um Pet com base no parâmetro ID

```

PROCEDURE CP1_EXCLUI_PET (
    p_id IN NUMBER
)
IS
BEGIN

    DELETE PETSHOP
    WHERE ID = p_id;

```

```
        commit;

    END CP1_EXCLUI_PET;

    -- Uma Procedure para excluir todos as linhas da Tabela

    PROCEDURE CP1_EXCLUI_TODOS
    IS
    BEGIN

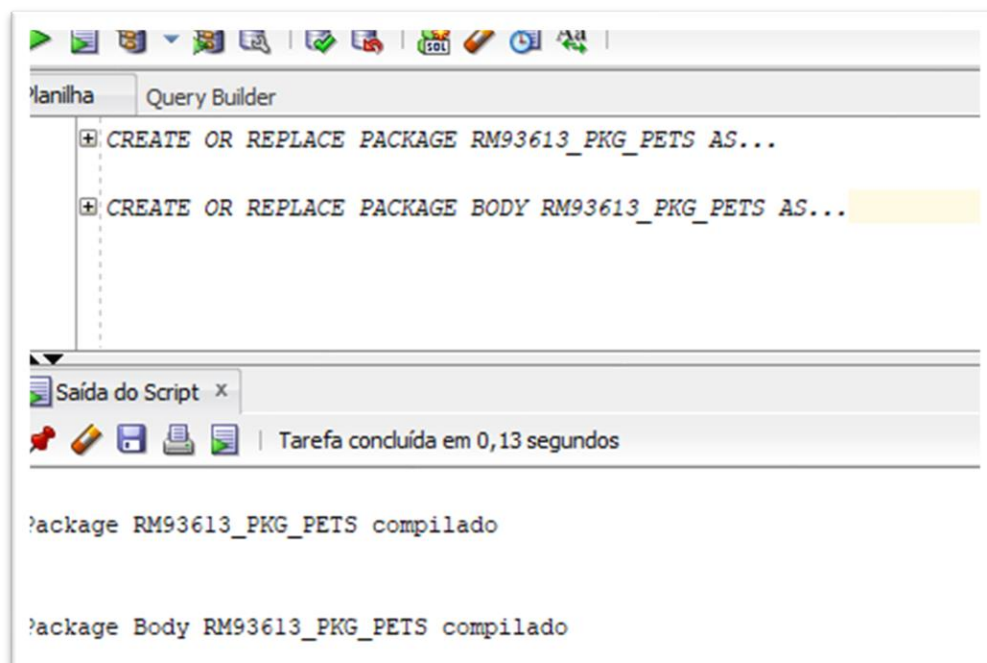
        DELETE FROM PETSHOP;

        commit;

    END CP1_EXCLUI_TODOS;

END RM93613_PKG_PETS;
```

Evidência:



4 - O Código fonte do Trigger criado

```
CREATE TABLE audit_petshop (  
  id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  usuario VARCHAR2(30),  
  data_registro DATE,  
  comando VARCHAR2(6),  
  valor_tipo_antigo VARCHAR2(30),  
  valor_nome_antigo VARCHAR2(30),  
  valor_idade_antigo NUMBER,  
  valor_tipo_novo VARCHAR2(30),  
  valor_nome_novo VARCHAR2(30),  
  valor_idade_novo NUMBER  
)
```

```
CREATE OR REPLACE TRIGGER RM93613_trg_audit_petshop  
AFTER INSERT OR UPDATE OR DELETE ON petshop  
FOR EACH ROW  
DECLARE  
  
  v_comando VARCHAR2(6);  
  
  v_valorTipoAntigo VARCHAR2(30);  
  v_valorNomeAntigo VARCHAR2(30);  
  v_valorIdadeAntigo NUMBER;  
  
  v_valorTipoNovo VARCHAR2(30);  
  v_valorNomeNovo VARCHAR2(30);  
  v_valorIdadeNovo NUMBER;  
  
BEGIN  
  
  IF INSERTING THEN  
    v_comando := 'INSERT';  
  
    v_valorTipoAntigo := null;  
    v_valorNomeAntigo := null;  
    v_valorIdadeAntigo := null;  
  
    v_valorTipoNovo := :new.tipo_pet;  
    v_valorNomeNovo := :new.nome_pet;  
    v_valorIdadeNovo := :new.idade;  
  
  ELSIF UPDATING THEN  
    v_comando := 'UPDATE';
```



```

        v_valorTipoAntigo := :old.tipo_pet;
        v_valorNomeAntigo := :old.nome_pet;
        v_valorIdadeAntigo := :old.idade;

        v_valorTipoNovo := :new.tipo_pet;
        v_valorNomeNovo := :new.nome_pet;
        v_valorIdadeNovo := :new.idade;

    ELSIF DELETING THEN
        v_comando := 'DELETE';

        v_valorTipoAntigo := :old.tipo_pet;
        v_valorNomeAntigo := :old.nome_pet;
        v_valorIdadeAntigo := :old.idade;

        v_valorTipoNovo := null;
        v_valorNomeNovo := null;
        v_valorIdadeNovo := null;

    END IF;

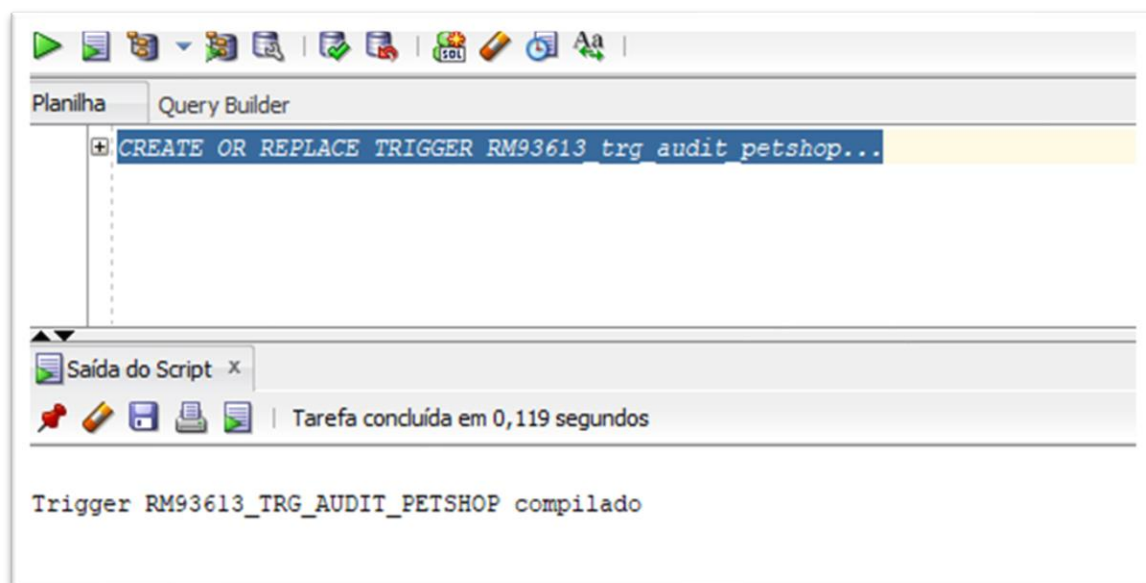
    INSERT INTO audit_petshop(usuario, data_registro, comando,
valor_tipo_antigo, valor_nome_antigo, valor_idade_antigo,
valor_tipo_novo, valor_nome_novo, valor_idade_novo)
    VALUES (USER, SYSDATE, v_comando, v_valorTipoAntigo,
v_valorNomeAntigo, v_valorIdadeAntigo, v_valorTipoNovo, v_valorNomeNovo,
v_valorIdadeNovo);

EXCEPTION

    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Ocorreu um erro: ' || SQLERRM);

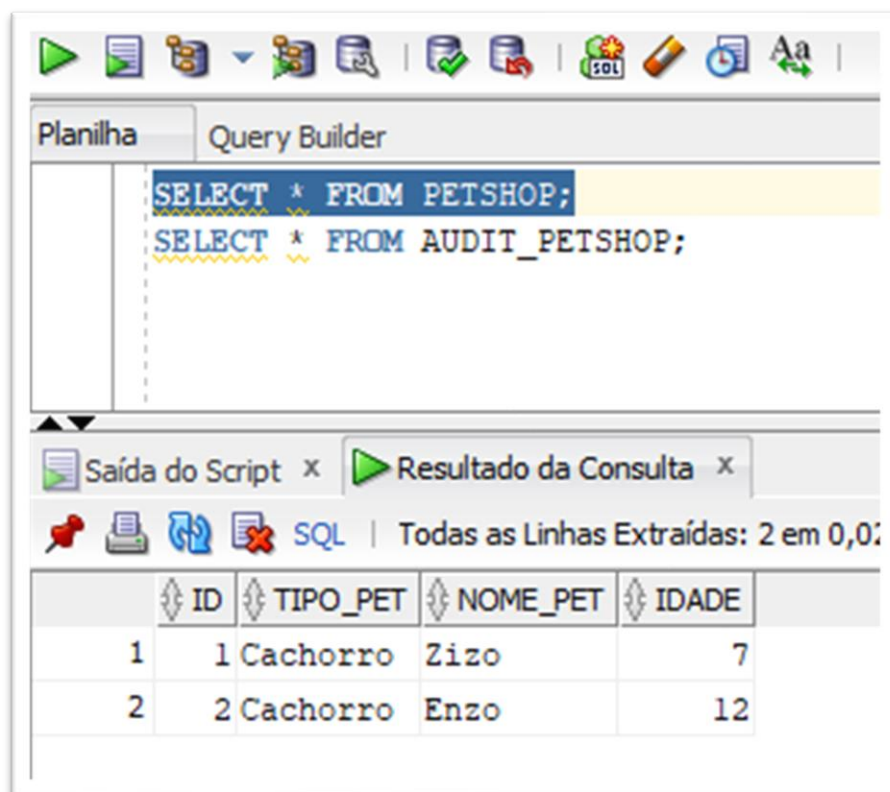
END;
```

Evidência:



5 - Prints da evidência do funcionamento de cada uma das ações propostas no script do Python

Tabela PETSHOP antes das ações:



Planilha Query Builder

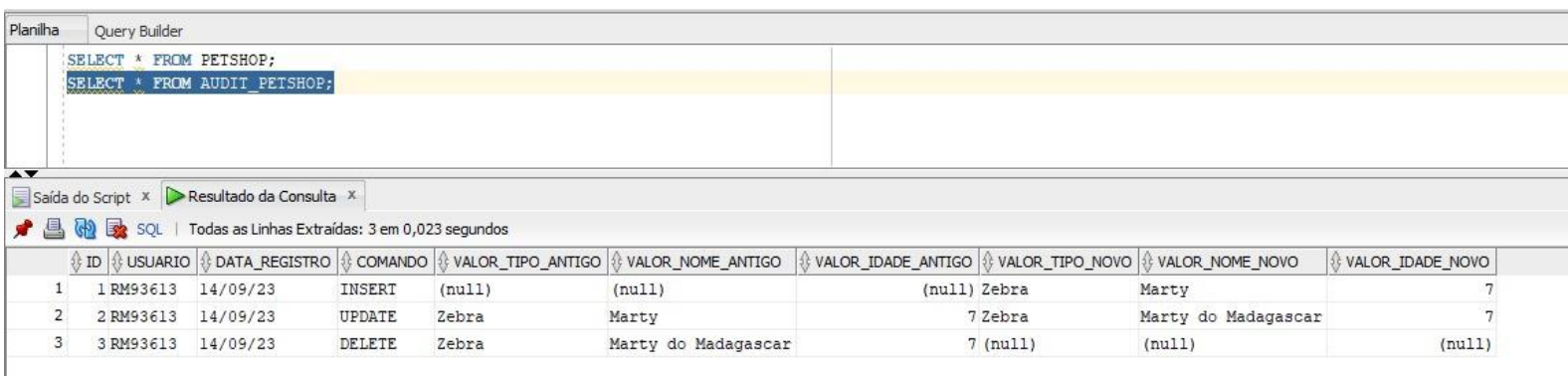
```
SELECT * FROM PETSHOP;
SELECT * FROM AUDIT_PETSHOP;
```

Saída do Script x Resultado da Consulta x

SQL | Todas as Linhas Extraídas: 2 em 0,0%

	ID	TIPO_PET	NOME_PET	IDADE
1	1	Cachorro	Zizo	7
2	2	Cachorro	Enzo	12

Tabela AUDIT_PETSHOP antes das ações:



Planilha Query Builder


```
SELECT * FROM PETSHOP;
SELECT * FROM AUDIT_PETSHOP;
```

Saída do Script x Resultado da Consulta x

SQL | Todas as Linhas Extraídas: 3 em 0,023 segundos

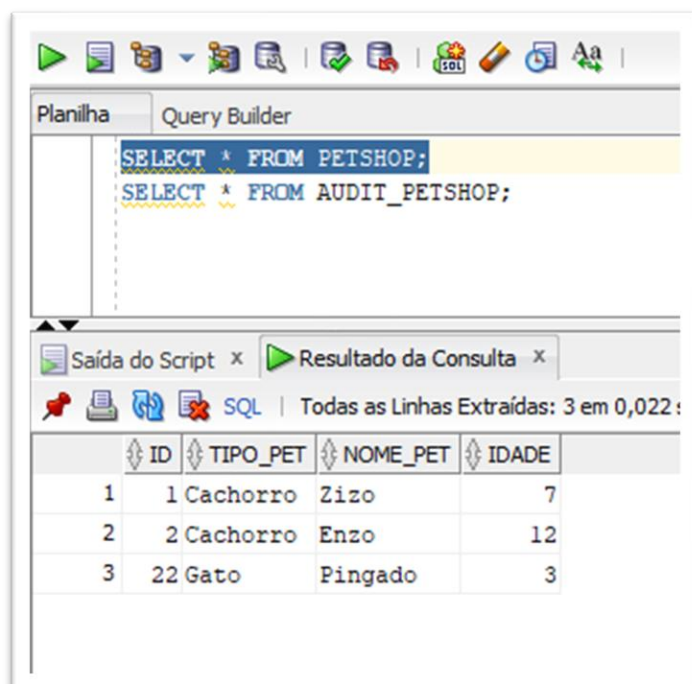
	ID	USUARIO	DATA_REGISTRO	COMANDO	VALOR_TIPO_ANTIGO	VALOR_NOME_ANTIGO	VALOR_IDADE_ANTIGO	VALOR_TIPO_NOVO	VALOR_NOME_NOVO	VALOR_IDADE_NOVO
1	1	RM93613	14/09/23	INSERT	(null)	(null)	(null)	Zebra	Marty	7
2	2	RM93613	14/09/23	UPDATE	Zebra	Marty	7	Zebra	Marty do Madagascar	7
3	3	RM93613	14/09/23	DELETE	Zebra	Marty do Madagascar	7 (null)	(null)	(null)	(null)

1. Cadastrar Pet:



```
----- CADAstrar PET -----  
  
  Digite o tipo....: Gato  
  Digite o nome....: Pingado  
  Digite a idade...: 3  
  
#### Dados GRAVADOS ####  
  Pressione ENTER
```

Evidência de funcionamento:



Planilha Query Builder

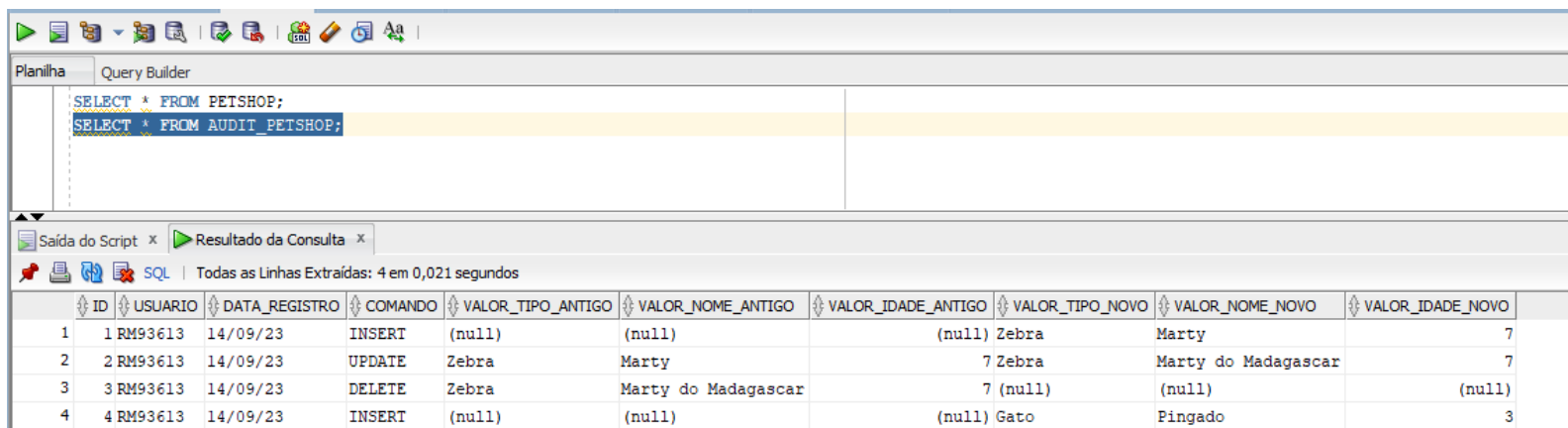
```
SELECT * FROM PETSHOP;  
SELECT * FROM AUDIT_PETSHOP;
```

Saída do Script x Resultado da Consulta x

SQL | Todas as Linhas Extraídas: 3 em 0,022 s

	ID	TIPO_PET	NOME_PET	IDADE
1	1	Cachorro	Zizo	7
2	2	Cachorro	Enzo	12
3	22	Gato	Pingado	3

Evidência do trigger:



Planilha Query Builder

SELECT * FROM PETSHOP;
SELECT * FROM AUDIT_PETSHOP;

Saída do Script x Resultado da Consulta x

Todas as Linhas Extraídas: 4 em 0,021 segundos

ID	USUARIO	DATA_REGISTRO	COMANDO	VALOR_TIPO_ANTIGO	VALOR_NOME_ANTIGO	VALOR_IDADE_ANTIGO	VALOR_TIPO_NOVO	VALOR_NOME_NOVO	VALOR_IDADE_NOVO
1	1RM93613	14/09/23	INSERT	(null)	(null)	(null)	Zebra	Marty	7
2	2RM93613	14/09/23	UPDATE	Zebra	Marty	7	Zebra	Marty do Madagascar	7
3	3RM93613	14/09/23	DELETE	Zebra	Marty do Madagascar	7	(null)	(null)	(null)
4	4RM93613	14/09/23	INSERT	(null)	(null)	(null)	Gato	Pingado	3

2. Listar Pet por ID:



```

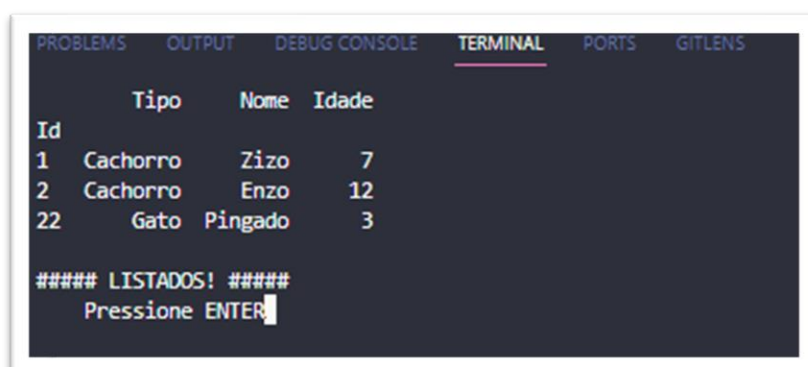
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

----- LISTAR PET POR ID-----

      Escolha um Id: 22
ID: 22
Nome: Pingado
Tipo: Gato
Idade: 3

##### LISTADO! #####
      Pressione ENTER
  
```

3. Listar todos os Pets:



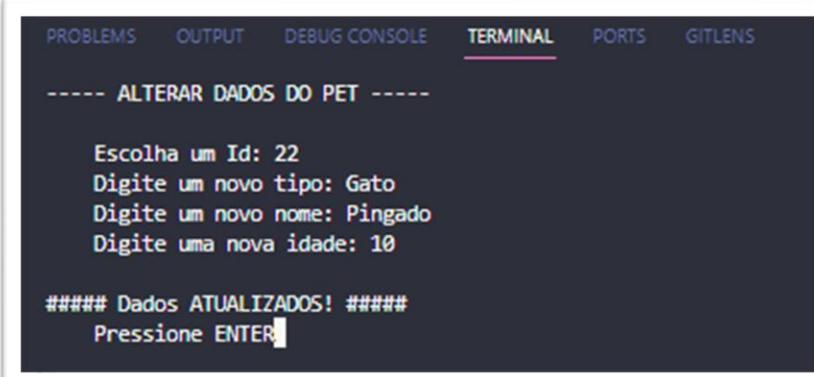
```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

      Tipo      Nome  Idade
Id
1  Cachorro    Zizo   7
2  Cachorro    Enzo  12
22 Gato      Pingado 3

##### LISTADOS! #####
      Pressione ENTER
  
```

4. Alterar Pet:



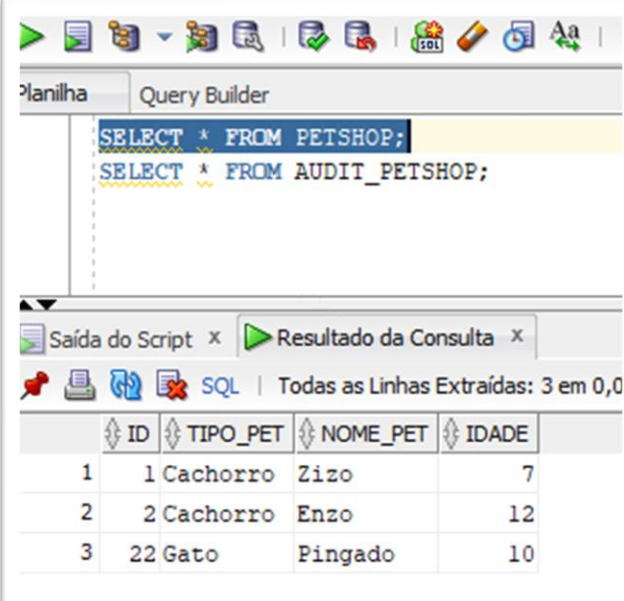
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

----- ALTERAR DADOS DO PET -----

Escolha um Id: 22
Digite um novo tipo: Gato
Digite um novo nome: Pingado
Digite uma nova idade: 10

##### Dados ATUALIZADOS! #####
Pressione ENTER
```

Evidência do funcionamento:



The screenshot shows a database query tool with a 'Query Builder' tab. Two SQL queries are listed:

```
SELECT * FROM PETSHOP;
SELECT * FROM AUDIT_PETSHOP;
```

Below the queries, there is a 'Resultado da Consulta' (Query Result) tab showing the results of the first query. The results are displayed in a table with columns: ID, TIPO_PET, NOME_PET, and IDADE. The table contains three rows of data.

ID	TIPO_PET	NOME_PET	IDADE
1	1 Cachorro	Zizo	7
2	2 Cachorro	Enzo	12
3	22 Gato	Pingado	10

Evidência do trigger:

Planilha Query Builder

```
SELECT * FROM PETSHOP;
SELECT * FROM AUDIT_PETSHOP;
```

Saída do Script x Resultado da Consulta x

Todas as Linhas Extraídas: 5 em 0,022 segundos

	ID	USUARIO	DATA_REGISTRO	COMANDO	VALOR_TIPO_ANTIGO	VALOR_NOME_ANTIGO	VALOR_IDADE_ANTIGO	VALOR_TIPO_NOVO	VALOR_NOME_NOVO	VALOR_IDADE_NOVO
1	1	RM93613	14/09/23	INSERT	(null)	(null)	(null)	Zebra	Marty	7
2	2	RM93613	14/09/23	UPDATE	Zebra	Marty	7	Zebra	Marty do Madagascar	7
3	3	RM93613	14/09/23	DELETE	Zebra	Marty do Madagascar	7	(null)	(null)	(null)
4	4	RM93613	14/09/23	INSERT	(null)	(null)	(null)	Gato	Pingado	3
5	5	RM93613	14/09/23	UPDATE	Gato	Pingado	3	Gato	Pingado	10

5. Excluir Pet:

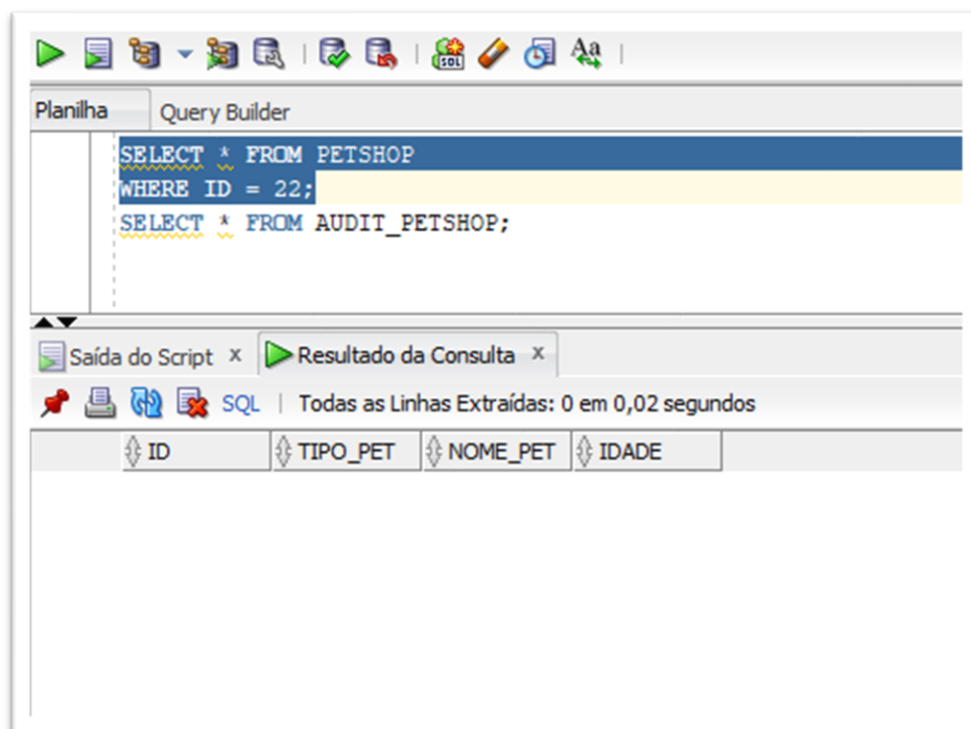
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

----- EXCLUIR PET -----

Escolha um Id: 22

##### Pet APAGADO! #####
Pressione ENTER
```

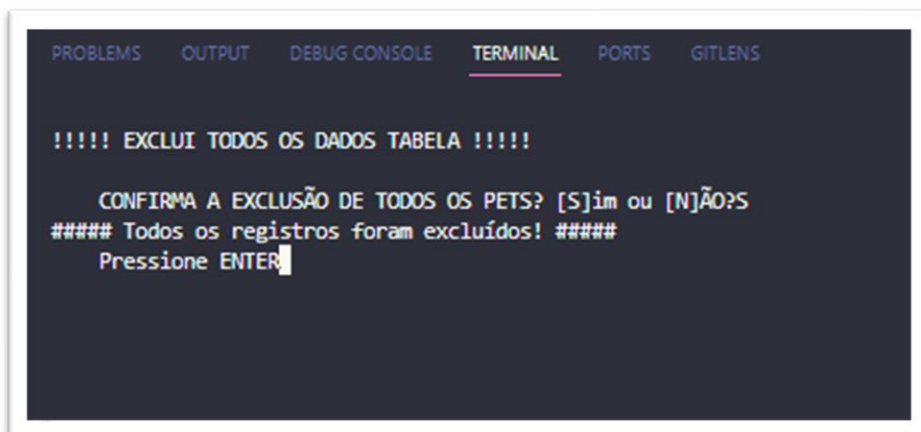
Evidência do funcionamento:



Evidência do trigger:

ID	USUARIO	DATA_REGISTRO	COMANDO	VALOR_TIPO_ANTIGO	VALOR_NOME_ANTIGO	VALOR_IDADE_ANTIGO	VALOR_TIPO_NOVO	VALOR_NOME_NOVO	VALOR_IDADE_NOVO
1	1 RM93613	14/09/23	INSERT	(null)	(null)	(null)	Zebra	Marty	7
2	2 RM93613	14/09/23	UPDATE	Zebra	Marty	7	Zebra	Marty do Madagascar	7
3	3 RM93613	14/09/23	DELETE	Zebra	Marty do Madagascar	7 (null)	(null)	(null)	(null)
4	4 RM93613	14/09/23	INSERT	(null)	(null)	(null)	Gato	Pingado	3
5	5 RM93613	14/09/23	UPDATE	Gato	Pingado	3	Gato	Pingado	10
6	6 RM93613	14/09/23	DELETE	Gato	Pingado	10 (null)	(null)	(null)	(null)

6. Excluir todos os Pets:

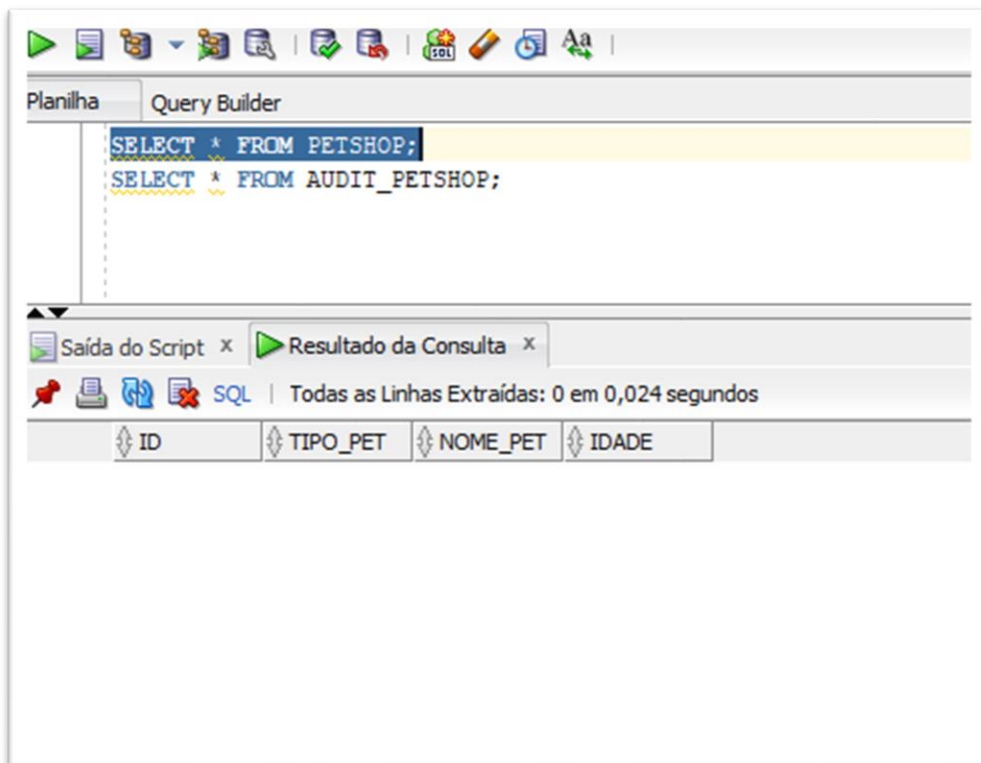


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

!!!!! EXCLUI TODOS OS DADOS TABELA !!!!!

CONFIRMA A EXCLUSÃO DE TODOS OS PETS? [S]im ou [N]ÃO?S
##### Todos os registros foram excluídos! #####
Pressione ENTER
```

Evidência do funcionamento:



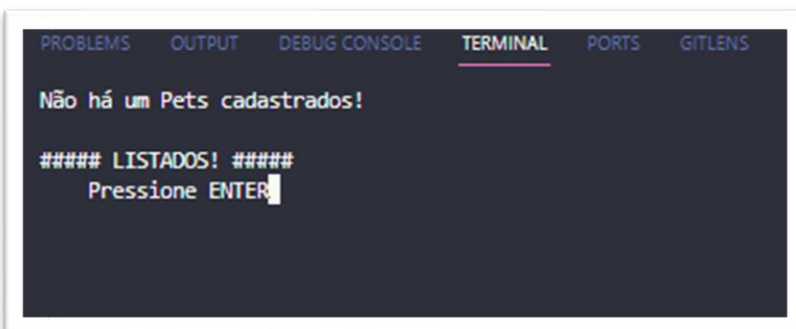
Planilha Query Builder

```
SELECT * FROM PETSHOP;
SELECT * FROM AUDIT_PETSHOP;
```

Saída do Script x Resultado da Consulta x

Todas as Linhas Extraídas: 0 em 0,024 segundos

ID	TIPO_PET	NOME_PET	IDADE
----	----------	----------	-------



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

Não há um Pets cadastrados!

##### LISTADOS! #####
Pressione ENTER
```

Evidência do trigger:

Planilha

Query Builder

```
SELECT * FROM PETSHOP;
SELECT * FROM AUDIT_PETSHOP;
```

Saída do Script

Resultado da Consulta

SQL

Todas as Linhas Extraídas: 8 em 0,026 segundos

	ID	USUARIO	DATA_REGI...	COMANDO	VALOR_TIPO_ANTIGO	VALOR_NOME_ANTIGO	VALOR_IDADE_ANTIGO	VALOR_TIPO_NOVO	VALOR_NOME_NOVO	VALOR_IDADE_NOVO
1	1	RM93613	14/09/23	INSERT	(null)	(null)	(null)	Zebra	Marty	7
2	2	RM93613	14/09/23	UPDATE	Zebra	Marty	7	Zebra	Marty do Madagascar	7
3	3	RM93613	14/09/23	DELETE	Zebra	Marty do Madagascar	7	(null)	(null)	(null)
4	4	RM93613	14/09/23	INSERT	(null)	(null)	(null)	Gato	Pingado	3
5	5	RM93613	14/09/23	UPDATE	Gato	Pingado	3	Gato	Pingado	10
6	6	RM93613	14/09/23	DELETE	Gato	Pingado	10	(null)	(null)	(null)
7	7	RM93613	14/09/23	DELETE	Cachorro	Zizo	7	(null)	(null)	(null)
8	8	RM93613	14/09/23	DELETE	Cachorro	Enzo	12	(null)	(null)	(null)