

DISCIPLINA

Database Application & Data Science

INTEGRANTES

RM (SOMENTE NÚMEROS)	NOME COMPLEMENTO (SEM ABREVIAR)
93038	Breno Massa Martins
93090	Gustavo Balero Cosse de Sousa
94280	Gustavo Henrique Moura
93567	Pedro Henrique Vidal da Silva
95224	Leonard Karic Klovza
93897	Marcelo Gimenes

Codificação PL/SQL

-- Package - Procedure de listar Transações dos usuários

CREATE OR REPLACE PACKAGE pkg_transacoes AS

PROCEDURE listar_transacoes_usuario(v_id_usuario NUMBER);

END pkg_transacoes;

/

-- Package Body - Procedure de listar Transações dos usuários

CREATE OR REPLACE PACKAGE BODY pkg_transacoes AS

PROCEDURE listar_transacoes_usuario(

v_id_usuario NUMBER)

IS

v_id_transacao NUMBER;

v_ds_titulo VARCHAR2(20);

BEGIN

v_id_transacao := NULL;

v_ds_titulo := NULL;

BEGIN

SELECT id_transacao, ds_titulo

INTO v_id_transacao, v_ds_titulo

FROM t_insightia_transacao

WHERE id_usuario = v_id_usuario;

DBMS_OUTPUT.PUT_LINE('ID Transação: ' || v_id_transacao || ', Título: ' || v_ds_titulo);

EXCEPTION

WHEN NO_DATA_FOUND THEN

DBMS_OUTPUT.PUT_LINE('Nenhuma informação encontrada para o usuário');

END;

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Erro inesperado: ' || SQLERRM);

END listar_transacoes_usuario;

END pkg_transacoes;

/

```
--1) Empacotando as Procedures

-- Package - Procedure de listar Transações dos usuários
CREATE OR REPLACE PACKAGE pkg_transacoes AS
    PROCEDURE listar_transacoes_usuario(v_id_usuario NUMBER);

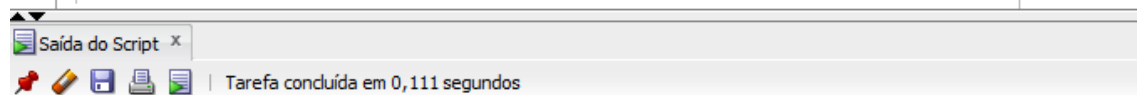
END pkg_transacoes;
/

-- Package Body - Procedure de listar Transações dos usuários
CREATE OR REPLACE PACKAGE BODY pkg_transacoes AS
    PROCEDURE listar_transacoes_usuario(
        v_id_usuario NUMBER)
    IS
        v_id_transacao NUMBER;
        v_ds_titulo VARCHAR2(20);
    BEGIN
        v_id_transacao := NULL;
        v_ds_titulo := NULL;

        BEGIN
            SELECT id_transacao, ds_titulo
            INTO v_id_transacao, v_ds_titulo
            FROM t_insightia_transacao
            WHERE id_usuario = v_id_usuario;

            DBMS_OUTPUT.PUT_LINE('ID Transação: ' || v_id_transacao || ', Título: ' || v_ds_titul

        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('Nenhuma informação encontrada para o usuário');
        END;
```



Package PKG_TRANSACOES compilado

Package Body PKG_TRANSACOES compilado

-- Package - Listar anúncios ativos

CREATE OR REPLACE PACKAGE pkg_anuncios AS

 PROCEDURE sp_relatorio_anuncios_ativos(p_id_usuario NUMBER);

END pkg_anuncios;

/

-- Package Body - Listar anúncios ativos

CREATE OR REPLACE PACKAGE BODY pkg_anuncios AS

 PROCEDURE sp_relatorio_anuncios_ativos(p_id_usuario NUMBER)

 IS

 v_id_anuncio NUMBER;

 v_ds_descricao VARCHAR2(100);

 BEGIN

 BEGIN

 SELECT id_anuncio, ds_descricao

 INTO v_id_anuncio, v_ds_descricao

 FROM t_insightia_anuncio

 WHERE id_usuario = p_id_usuario AND sg_ativo = 'S';

 DBMS_OUTPUT.PUT_LINE('ID do Anúncio: ' || v_id_anuncio);

 DBMS_OUTPUT.PUT_LINE('Descrição: ' || v_ds_descricao);

 EXCEPTION

 WHEN NO_DATA_FOUND THEN

 DBMS_OUTPUT.PUT_LINE('Nenhum anúncio ativo encontrado para o usuário');

 END;

 END sp_relatorio_anuncios_ativos;

END pkg_anuncios;

/

```

-- Package - Listar anúncios ativos
CREATE OR REPLACE PACKAGE pkg_anuncios AS
    PROCEDURE sp_relatorio_anuncios_ativos(p_id_usuario NUMBER);
END pkg_anuncios;
/

-- Package Body - Listar anúncios ativos
CREATE OR REPLACE PACKAGE BODY pkg_anuncios AS
    PROCEDURE sp_relatorio_anuncios_ativos(p_id_usuario NUMBER)
    IS
        v_id_anuncio NUMBER;
        v_ds_descricao VARCHAR2(100);
    BEGIN
        BEGIN
            SELECT id_anuncio, ds_descricao
            INTO v_id_anuncio, v_ds_descricao
            FROM t_insightia_anuncio
            WHERE id_usuario = p_id_usuario AND sg_ativo = 'S';

            DBMS_OUTPUT.PUT_LINE('ID do Anúncio: ' || v_id_anuncio);
            DBMS_OUTPUT.PUT_LINE('Descrição: ' || v_ds_descricao);
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('Nenhum anúncio ativo encontrado para o usuário');
            END;
        END sp_relatorio_anuncios_ativos;
    END pkg_anuncios;
/

```

Saída do Script x

Tarefa concluída em 0,085 segundos

Package PKG_ANUNCIOS compilado

Package Body PKG_ANUNCIOS compilado

-- Package - Funções relacionadas ao cálculo de bônus

CREATE OR REPLACE PACKAGE pkg_func_cancular_bonus AS

FUNCTION calcular_bonus(p_id_usuario NUMBER, p_valor_inserido NUMBER)

RETURN NUMBER;

END pkg_func_cancular_bonus;

/

CREATE OR REPLACE PACKAGE BODY pkg_func_cancular_bonus AS

FUNCTION calcular_bonus(

p_id_usuario NUMBER,

p_valor_inserido NUMBER

) RETURN NUMBER

IS

v_bonus NUMBER := 0.05; -- Bônus de 5%

v_valor_bonus NUMBER;

BEGIN

-- Verificar se o usuário existe

SELECT 1

INTO v_valor_bonus

FROM t_insightia_usuario

WHERE id_usuario = p_id_usuario;

-- Calcular o valor do bônus

v_valor_bonus := (p_valor_inserido) + (p_valor_inserido * v_bonus);

-- Atualizar o saldo do usuário com o bônus calculado

UPDATE t_insightia_usuario

SET nr_saldo = nr_saldo + v_valor_bonus

WHERE id_usuario = p_id_usuario;

```
-- Retornar o valor do bônus calculado

RETURN v_valor_bonus;

EXCEPTION

WHEN NO_DATA_FOUND THEN

    DBMS_OUTPUT.PUT_LINE('Usuário não encontrado.');
```

RETURN NULL; -- Retornar NULL em caso de usuário não encontrado

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Erro inesperado: ' || SQLERRM);

RETURN NULL; -- Retornar NULL em caso de erro

END calcular_bonus;

END pkg_func_cancelar_bonus;

/

```

-- Package - Funções relacionadas ao cálculo de bônus
CREATE OR REPLACE PACKAGE pkg_func_cancelar_bonus AS
    FUNCTION calcular_bonus(p_id_usuario NUMBER, p_valor_inserido NUMBER)
    RETURN NUMBER;
END pkg_func_cancelar_bonus;
/

CREATE OR REPLACE PACKAGE BODY pkg_func_cancelar_bonus AS
    FUNCTION calcular_bonus(
        p_id_usuario NUMBER,
        p_valor_inserido NUMBER
    ) RETURN NUMBER
    IS
        v_bonus NUMBER := 0.05; -- Bônus de 5%
        v_valor_bonus NUMBER;

    BEGIN
        -- Verificar se o usuário existe
        SELECT 1
        INTO v_valor_bonus
        FROM t_insightia_usuario
        WHERE id_usuario = p_id_usuario;

        -- Calcular o valor do bônus
        v_valor_bonus := (p_valor_inserido) + (p_valor_inserido * v_bonus);

        -- Atualizar o saldo do usuário com o bônus calculado
        UPDATE t_insightia_usuario
        SET nr_saldo = nr_saldo + v_valor_bonus
        WHERE id_usuario = p_id_usuario;

        -- Retornar o valor do bônus calculado
        RETURN v_valor_bonus;
    EXCEPTION

```

Saída do Script x

Tarefa concluída em 0,091 segundos

Package PKG_FUNC_CANCELAR_BONUS compilado

Package Body PKG_FUNC_CANCELAR_BONUS compilado

-- Package Body - Funções relacionadas à contagem de anúncios ativos

CREATE OR REPLACE PACKAGE pkg_func_cont_anuncios AS

 FUNCTION contar_anuncios_ativos RETURN NUMBER;

END pkg_func_cont_anuncios;

/

CREATE OR REPLACE PACKAGE BODY pkg_func_cont_anuncios AS

 FUNCTION contar_anuncios_ativos

 RETURN NUMBER

 IS

 v_count NUMBER;

 BEGIN

 SELECT COUNT(*) INTO v_count

 FROM t_insightia_anuncio

 WHERE sg_ativo = 'Y';

 RETURN v_count;

END contar_anuncios_ativos;

END pkg_func_cont_anuncios;


/

```
-- Package Body - Funções relacionadas à contagem de anúncios ativos
CREATE OR REPLACE PACKAGE pkg_func_cont_anuncios AS
    FUNCTION contar_anuncios_ativos RETURN NUMBER;
END pkg_func_cont_anuncios;
/

CREATE OR REPLACE PACKAGE BODY pkg_func_cont_anuncios AS
    FUNCTION contar_anuncios_ativos
    RETURN NUMBER
    IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count
        FROM t_insightia_anuncio
        WHERE sg_ativo = 'Y';

        RETURN v_count;
    END contar_anuncios_ativos;
END pkg_func_cont_anuncios;
/
```

Saída do Script x

 Tarefa concluída em 0,084 segundos

Package PKG_FUNC_CONT_ANUNCIOS compilado

Package Body PKG_FUNC_CONT_ANUNCIOS compilado

-- Gatilho para auditar operações de DELETE na tabela t_insightia_insight

CREATE OR REPLACE TRIGGER trg_auditoria_delete_insight

AFTER DELETE ON t_insightia_insight

FOR EACH ROW

BEGIN

INSERT INTO t_insightia_auditoria (id_auditoria, usuario_auditoria, date_auditoria,
comando_auditoria, vlr_antigo, vlr_novo, coluna_afetada, tabela_afetada)

VALUES (sq_insightia_auditoria.nextval, USER, SYSTIMESTAMP, 'DELETE', :old.ds_conteudo,
NULL, 'DS_CONTEUDO', 't_insightia_insight');

END trg_auditoria_delete_anuncio;

/

-- Gatilho para auditar operações de INSERT na tabela t_insightia_anuncio

CREATE OR REPLACE TRIGGER trg_auditoria_insert_anuncio

AFTER INSERT ON t_insightia_anuncio

FOR EACH ROW

BEGIN

INSERT INTO t_insightia_auditoria (id_auditoria, usuario_auditoria, date_auditoria,
comando_auditoria, vlr_antigo, vlr_novo, coluna_afetada, tabela_afetada)

VALUES (sq_insightia_auditoria.nextval, USER, SYSTIMESTAMP, 'INSERT', NULL,
:NEW.ds_descricao, 'DS_DESCRICAO', 't_insightia_anuncio');

END trg_auditoria_insert_comando;

/

```
-- Gatilho para auditar operações de DELETE na tabela t_insightia_insight
CREATE OR REPLACE TRIGGER trg_auditoria_delete_insight
AFTER DELETE ON t_insightia_insight
FOR EACH ROW
BEGIN
    INSERT INTO t_insightia_auditoria (id_auditoria, usuario_auditoria, date_auditoria, comando_auditoria, vlr_antigo, vlr_novo, coluna_afetada, tabela_afetada)
    VALUES (sq_insightia_auditoria.nextval, USER, SYSTIMESTAMP, 'DELETE', :old.ds_conteudo, NULL, 'DS_CONTEUDO', 't_insightia_insight');

END trg_auditoria_delete_anuncio;
/

-- Gatilho para auditar operações de INSERT na tabela t_insightia_anuncio
CREATE OR REPLACE TRIGGER trg_auditoria_insert_anuncio
AFTER INSERT ON t_insightia_anuncio
FOR EACH ROW
BEGIN
    INSERT INTO t_insightia_auditoria (id_auditoria, usuario_auditoria, date_auditoria, comando_auditoria, vlr_antigo, vlr_novo, coluna_afetada, tabela_afetada)
    VALUES (sq_insightia_auditoria.nextval, USER, SYSTIMESTAMP, 'INSERT', NULL, :NEW.ds_descricao, 'DS_DESCRICAO', 't_insightia_anuncio');

END trg_auditoria_insert_comando;
/

-- Exemplo de INSERT em t_insightia_anuncio
INSERT INTO t_insightia_anuncio (id_anuncio, id_usuario, sg_ativo, ds_descricao)
VALUES (SQ_INSIGHTIA_ANUNCIO.NEXTVAL, 1, 'S', 'Exemplo');

select * from t_insightia_anuncio;

-- Exemplo de INSERT em t_insightia_insight
INSERT INTO t_insightia_insight (id_insight, id_anuncio, ds_conteudo, ft_image)
```

Selecção do Script x

Tarefa concluída em 0,116 segundos

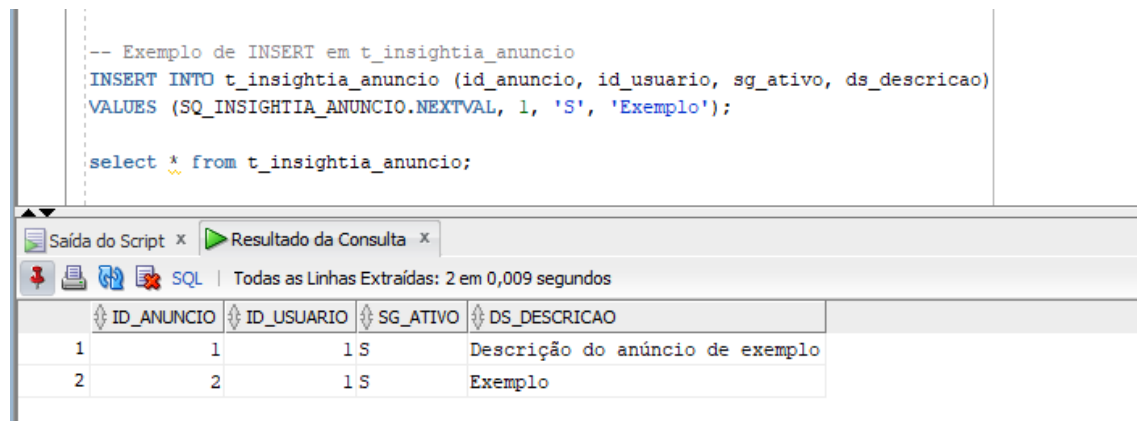
Trigger TRG_AUDITORIA_DELETE_INSIGHT compilado

Trigger TRG_AUDITORIA_INSERT_ANUNCIO compilado

-- Exemplo de INSERT em t_insightia_anuncio

```
INSERT INTO t_insightia_anuncio (id_anuncio, id_usuario, sg_ativo, ds_descricao)
VALUES (SQ_INSIGHTIA_ANUNCIO.NEXTVAL, 1, 'S', 'Exemplo');
```

select * from t_insightia_anuncio;



The screenshot shows a SQL script editor with the following code:

```
-- Exemplo de INSERT em t_insightia_anuncio
INSERT INTO t_insightia_anuncio (id_anuncio, id_usuario, sg_ativo, ds_descricao)
VALUES (SQ_INSIGHTIA_ANUNCIO.NEXTVAL, 1, 'S', 'Exemplo');

select * from t_insightia_anuncio;
```

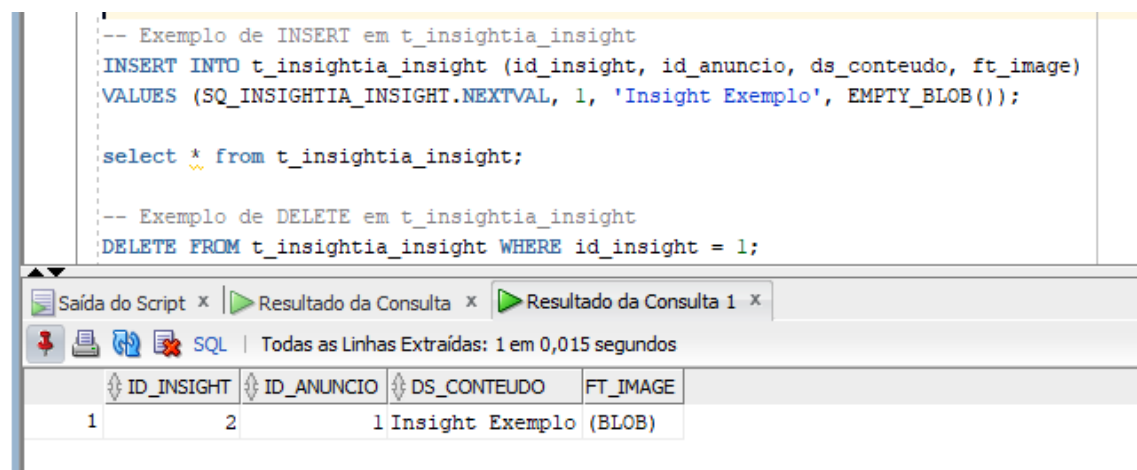
Below the script, the 'Resultado da Consulta' (Query Result) tab is active, displaying the following table:

ID_ANUNCIO	ID_USUARIO	SG_ATIVO	DS_DESCRICAO
1	1	S	Descrição do anúncio de exemplo
2	2	S	Exemplo

-- Exemplo de INSERT em t_insightia_insight

```
INSERT INTO t_insightia_insight (id_insight, id_anuncio, ds_conteudo, ft_image)
VALUES (SQ_INSIGHTIA_INSIGHT.NEXTVAL, 1, 'Insight Exemplo', EMPTY_BLOB());
```

select * from t_insightia_insight;



The screenshot shows a SQL script editor with the following code:

```
-- Exemplo de INSERT em t_insightia_insight
INSERT INTO t_insightia_insight (id_insight, id_anuncio, ds_conteudo, ft_image)
VALUES (SQ_INSIGHTIA_INSIGHT.NEXTVAL, 1, 'Insight Exemplo', EMPTY_BLOB());

select * from t_insightia_insight;

-- Exemplo de DELETE em t_insightia_insight
DELETE FROM t_insightia_insight WHERE id_insight = 1;
```

Below the script, the 'Resultado da Consulta' (Query Result) tab is active, displaying the following table:

ID_INSIGHT	ID_ANUNCIO	DS_CONTEUDO	FT_IMAGE
1	2	1 Insight Exemplo	(BLOB)

-- Exemplo de DELETE em t_insightia_insight

DELETE FROM t_insightia_insight WHERE id_insight = 2;

select * from t_insightia_insight;

```
-- Exemplo de DELETE em t_insightia_insight
DELETE FROM t_insightia_insight WHERE id_insight = 2;

select * from t_insightia_insight;
```

0 linhas excluído.

1 linha excluído.

```
select * from t_insightia_insight;
```

ID_INSIGHT	ID_ANUN...	DS_CONT...	FT_IMAGE
------------	------------	------------	----------

select * FROM t_insightia_auditoria;

ID_AUDITORIA	USUARIO_AUDIT...	DATE_AUDITORIA	COMANDO_AUDITORIA	VLR_ANTIGO	VLR_NOVO	COLUMNA_AFETADA	TABELA_AFETADA
1	2 RM93038	11/11/23 15:36:26,760000000	UPDATE	310	415	nr_saldo	t_insightia_transacao
2	3 RM93038	11/11/23 17:40:08,290000000	INSERT	(null)	Descrição do anúncio de exemplo	DS DESCRICAO	t_insightia_anuncio
3	4 RM93038	11/11/23 17:40:38,990000000	DELETE	Insight Exemplo (null)	(null)	DS CONTEUDO	t_insightia_insight
4	5 RM93038	11/11/23 18:15:57,290000000	INSERT	(null)	Exemplo	DS DESCRICAO	t_insightia_anuncio
5	6 RM93038	11/11/23 18:17:07,730000000	DELETE	Insight Exemplo (null)	(null)	DS CONTEUDO	t_insightia_insight
6	1 RM93038	11/09/23 23:09:15,600000000	UPDATE	205	310	nr_saldo	t_insightia_transacao