

BANCO DE DADOS

JOÃO VITOR DE SANTANA DOS SANTOS – RM 554328

SARA INGRID DA SILVA PEREIRA – RM 554021

WILLIAN DANIEL OLIVEIRA DANTAS – RM 552671

EVOTECHS SOLUTIONS – CHALLENGE 2024

Sumário

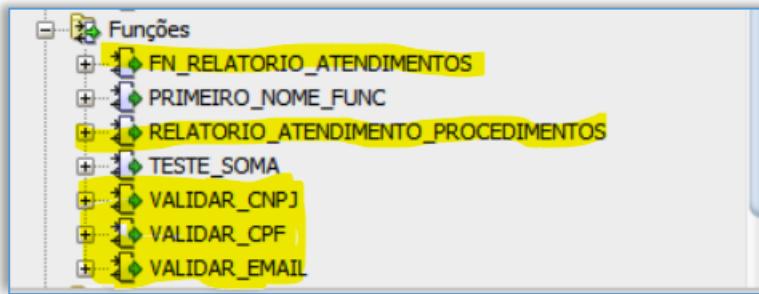
Link do Vídeo	3
Funções de validação, Regra de negócio e Relatório formatado	3
Função para Validar CNPJ.....	3
Função para Validar CPF	4
Função para Validar E-MAIL	4
FUNÇÃO com cursor e Joins relatório formatado.....	5
FUNÇÃO relatório com regra de negócio	6
Lista de Procedures compiladas para todas as tabelas do projeto.....	8
PROCEDURES	9
Inserts nas tabelas	9
TB_PLANO	9
TB_ENDERECO	10
TB_PROCEDIMENTO.....	10
TB_CLIENTE.....	11
TB_REDE_CREDENCIADA	11
TB_ATENDIMENTO.....	12
TB_TRATAMENTO.....	12
TB_ASSOC_ATEND_PROC.....	13
Atualização das tabelas	13
TB_PLANO	13
TB_CLIENTE.....	14
TB_ATENDIMENTO.....	15
TB_ENDEREÇO	15
TB_PROCEDIMENTO.....	16
TB_TRATAMENTO.....	16
Deletar dados	17

TB_CLIENTE	17
TB_PLANO	17
TB_REDE_CREDENCIADA	18
TB_ENDEREÇO	18
TB_ATENDIMENTO.....	19
TB_PROCEDIMENTO.....	19
TB_TRATAMENTO.....	20
TB_ASSOC_ATEND_PROC.....	20

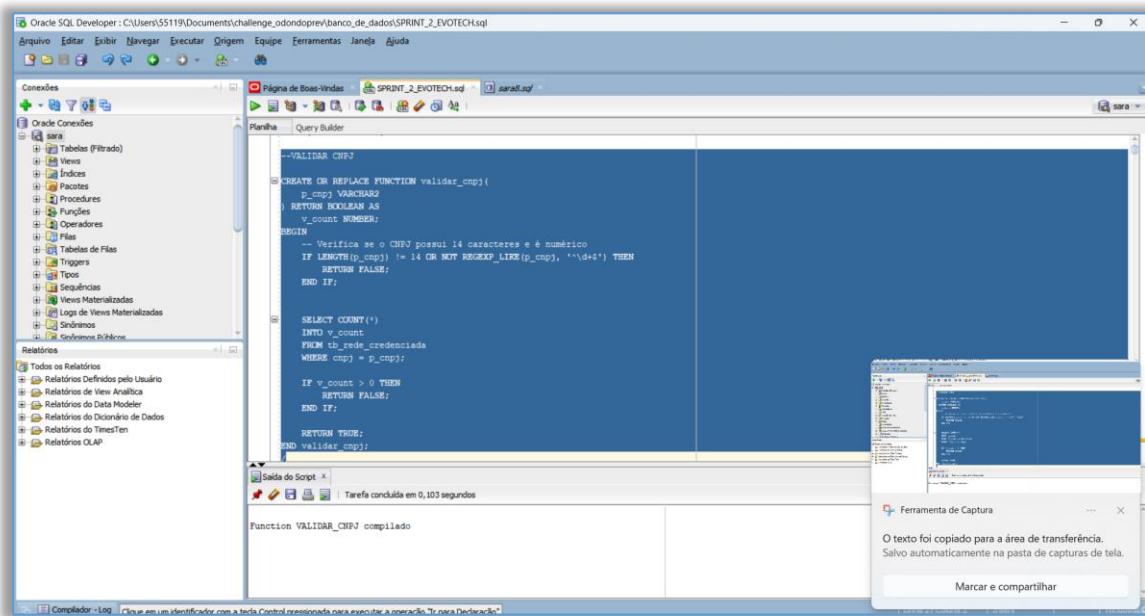
Link do Vídeo

<https://youtu.be/AIb5LStLf1w>

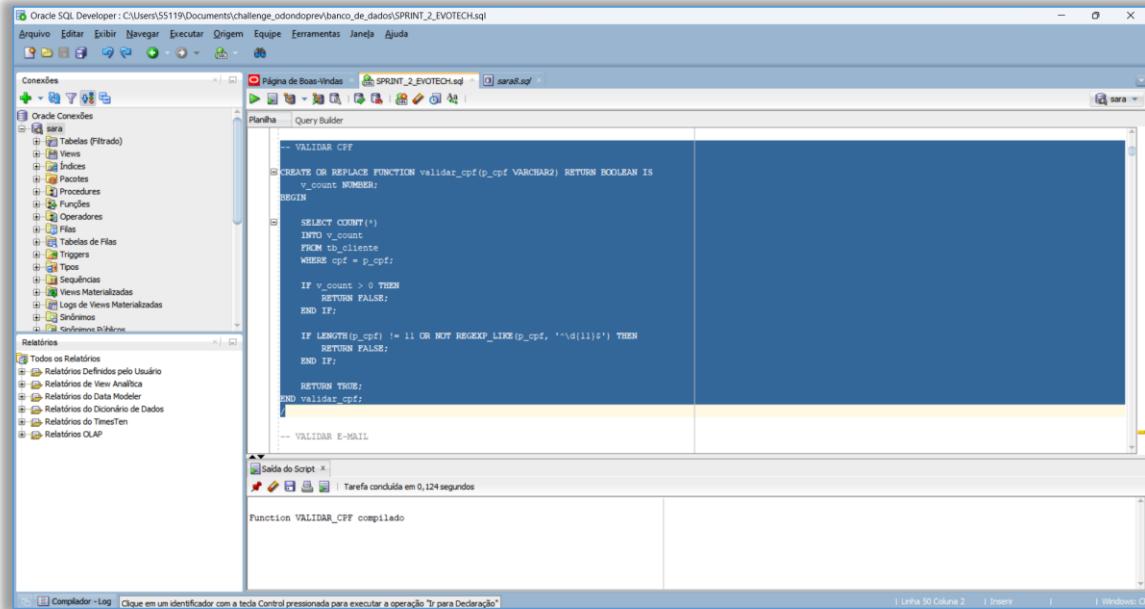
Funções de validação, Regra de negócio e Relatório formatado



Função para Validar CNPJ



Função para Validar CPF



The screenshot shows the Oracle SQL Developer interface with the following details:

- File Path:** Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
- Connections:** Oracle Conexões (sara)
- Relatórios:** Todos os Relatórios, Relatórios Definidos pelo Usuário, Relatórios de View Analítica, Relatórios do Data Modeler, Relatórios do Dicionário de Dados, Relatórios do TimesTen, Relatórios OLAP.
- Editor:** Página de Boas-Vindas (SPRINT_2_EVOTECH.sql) - sara8.sql
- Code:**

```
-- VALIDAR CPF
CREATE OR REPLACE FUNCTION validar_cpf(p_cpf VARCHAR2) RETURN BOOLEAN IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_count
    FROM tb_cliente
    WHERE cpf = p_cpf;

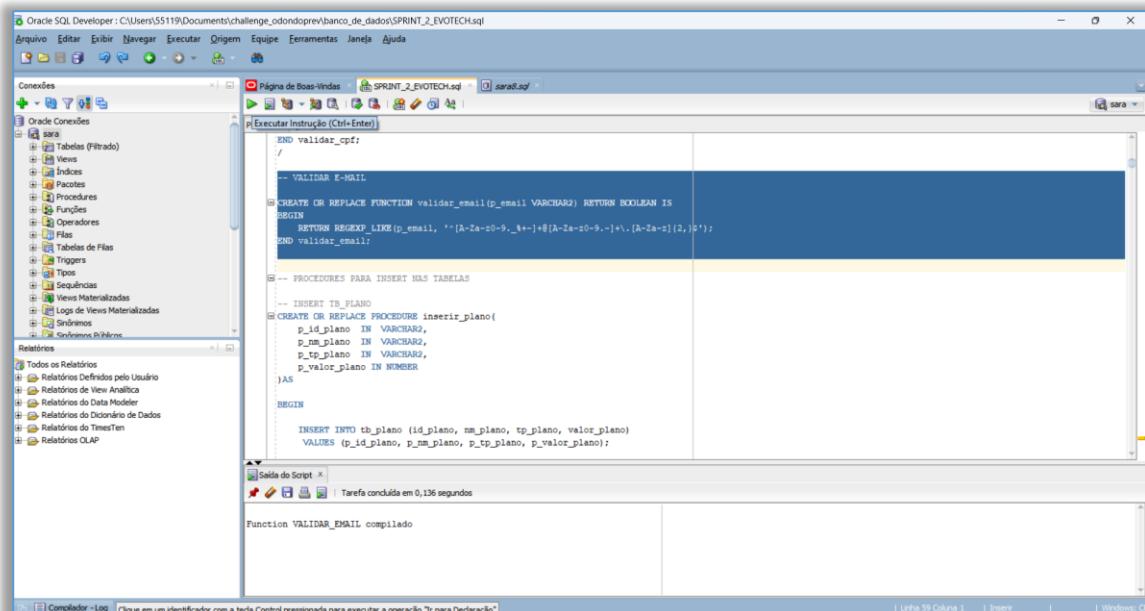
    IF v_count > 0 THEN
        RETURN FALSE;
    END IF;

    IF LENGTH(p_cpf) != 11 OR NOT REGEXP_LIKE(p_cpf, '^\\d{11}$') THEN
        RETURN FALSE;
    END IF;

    RETURN TRUE;
END validar_cpf;

--- VALIDAR E-MAIL
```
- Output:** Saída do Script (Tarefa concluída em 0,124 segundos)
- Compiler Log:** Clique em um identificador com a tecla Control pressionada para executar a operação "Ir para Declaração".

Função para Validar E-MAIL



The screenshot shows the Oracle SQL Developer interface with the following details:

- File Path:** Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
- Connections:** Oracle Conexões (sara)
- Relatórios:** Todos os Relatórios, Relatórios Definidos pelo Usuário, Relatórios de View Analítica, Relatórios do Data Modeler, Relatórios do Dicionário de Dados, Relatórios do TimesTen, Relatórios OLAP.
- Editor:** Página de Boas-Vindas (SPRINT_2_EVOTECH.sql) - sara8.sql
- Code:**

```
END validar_cpf;
/

-- VALIDAR E-MAIL
CREATE OR REPLACE FUNCTION validar_email(p_email VARCHAR2) RETURN BOOLEAN IS
BEGIN
    RETURN REGEXP_LIKE(p_email, '^([A-Za-z0-9_.+-]+@[A-Za-z0-9.-]+.[A-Za-z]{2,9})$');
END validar_email;

-- PROCEDURES PARA INSERT NAS TABELAS
-- INSERT TB_PLANO
CREATE OR REPLACE PROCEDURE inserir_plano(
    p_id_plano IN VARCHAR2,
    p_nm_plano IN VARCHAR2,
    p_tp_plano IN VARCHAR2,
    p_valor_plano IN NUMBER
) AS
BEGIN
    INSERT INTO tb_plano (id_plano, nm_plano, tp_plano, valor_plano)
    VALUES (p_id_plano, p_nm_plano, p_tp_plano, p_valor_plano);

```
- Output:** Saída do Script (Tarefa concluída em 0,136 segundos)
- Compiler Log:** Clique em um identificador com a tecla Control pressionada para executar a operação "Ir para Declaração".

FUNÇÃO com cursor e Joins relatório formatado

PRIMEIRA PARTE:

Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

```
-->--FUNÇÃO COM JOINS E CURSOR
CREATE OR REPLACE TYPE relatorio_associacao_tipo AS OBJECT (
    atendimento_id VARCHAR2(20),
    procedimento_id VARCHAR2(6),
    diagnostico VARCHAR2(300),
    nm_procedimento VARCHAR2(200),
    complexidade VARCHAR2(20)
);

-- RELATÓRIO DO TIPO TABLE
CREATE OR REPLACE TYPE relatorio_associacao_tab AS TABLE OF relatorio_associacao_tipo;
/
```

-->--proc
CREATE OR REPLACE FUNCTION relatorio_atendimento_procedimentos
RETURNS relatorio_associacao_tab PIPELINED IS
CURSOR c_relatorio IS
SELECT
 ap.tb_atendimento_id_atendimento AS atendimento_id,
 ap.tb_procedimento_id_proc AS procedimento_id,
 a.diagnostico,
 p_nm_proc AS nm_procedimento,
 n.complexidade
 FROM tb_assoc_atend_proc ap
 JOIN tb_atendimento a ON ap.tb_atendimento_id_atendimento = a.id_atendimento
 JOIN tb_procedimento p ON ap.tb_procedimento_id_proc = p.id_proc;

Type RELATORIO_ASSOCIAÇÃO_TIPO compilado

Type RELATORIO_ASSOCIAÇÃO_TAB compilado

SEGUNDA PARTE:

Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

```
-->--FUNÇÃO
CREATE OR REPLACE FUNCTION relatorio_atendimento_procedimentos
RETURNS relatorio_associacao_tab PIPELINED IS
CURSOR c_relatorio IS
SELECT
    ap.tb_atendimento_id_atendimento AS atendimento_id,
    ap.tb_procedimento_id_proc AS procedimento_id,
    a.diagnostico,
    p_nm_proc AS nm_procedimento,
    p.complexidade
    FROM tb_assoc_atend_proc ap
    JOIN tb_atendimento a ON ap.tb_atendimento_id_atendimento = a.id_atendimento
    JOIN tb_procedimento p ON ap.tb_procedimento_id_proc = p.id_proc;
```

BEGIN
FOR reg IN c_relatorio LOOP
PIPE ROW(relatorio_associacao_tipo(
 reg.atendimento_id,
 reg.procedimento_id,
 reg.diagnostico,
 reg.nm_procedimento,
 reg.complexidade
));
END LOOP;
RETURN;
END relatorio_atendimento_procedimentos;

/-->-- SELECT DO RELATÓRIO GERADO
SELECT * FROM TABLE(relatorio_atendimento_procedimentos());

/esse não funciona

Function RELATORIO_ATENDIMENTO_PROCEDIMENTOS compilado

SELECT do resultado:

```

    Oracle SQL Developer : C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
    Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda

    Conexões
    Relatórios
    Planilha Query Builder
    Página de Boas-Vindas SPRINT_2_EVOTECH.sql sara8.sql
    Resultado da Consulta
    SQL | Todas as Linhas Extraiidas: 10 em 0,036 segundos
    1 ATENDIMENTO_ID PROCEDIMENTO_ID DIAGNOSTICO NM_PROCEDIMENTO COMPLEXIDADE
    1 1 1 Atendimento inicial para exame clinico e avaliação. Limpeza Simples Baixa
    2 2 2 Limpeza dentária simples. Limpeza Profunda Baixa
    3 3 3 Extração de dente do siso. Extração Dente Posição V23 Média
    4 4 4 Consulta para colocação de aparelho ortodontico. Colocação de Aparelho Ortodontico Alta
    5 5 5 Avaliação para colocação de coroa dentária. Colocação de Coroa Posição V30 Média
    6 6 6 Tratamento de canal e colocação de coroa. Anestesia Local Alta
    7 7 7 Consulta para restauração parcial do dente. Restauração Parcial Média
    8 8 8 Aplicação de anestesia local e restauração total. Restauração Total Alta
    9 10 10 Tratamento de canal e colocação de aparelho. Exame Raio X Baixa
    10 9 9 Exame de raio-X da mandíbula. Tratamento Canal Raiz Posição V24 Alta
  
```

FUNÇÃO relatório com regra de negócio

Primeira parte:

```

    Oracle SQL Developer : C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
    Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda

    Conexões
    Relatórios
    Planilha Query Builder
    Página de Boas-Vindas SPRINT_2_EVOTECH.sql sara8.sql
    Resultado da Consulta
    SQL | Tarefa concluída em 0,097 segundos
    Type ATENDIMENTO_RELATORIO_LINHA compilado
    Type ATENDIMENTO_RELATORIO_TAB compilado
  
```

Segunda parte:

The screenshot shows the Oracle SQL Developer interface with the following details:

- File Path:** Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
- Toolbar:** Arquivo, Editar, Exibir, Navegar, Executar, Origem, Equipe, Ferramentas, Janela, Ajuda.
- Left Panel (Conexões):** Shows various database objects like Views, Índices, Pacotes, Procedures, Operadores, etc.
- Left Panel (Relatórios):** Shows reports like Relatórios Definidos pelo Usuário, Relatórios de View Analítica, Relatórios do Data Modeler, Relatórios do Dicionário de Dados, Relatórios do TimesTen, and Relatórios OLAP.
- Central Area (Planilha - Query Builder):**

```

CREATE OR REPLACE TYPE atendimento_relatorio_tab AS TABLE OF atendimento_relatorio_linha;
/
CREATE OR REPLACE FUNCTION fn_relatorio_atendimentos (
    p_data_inicio IN DATE,
    p_data_fim IN DATE
) RETURN atendimento_relatorio_tab PIPABLENDED
IS
BEGIN
    FOR rec IN (
        SELECT
            cl.id_cliente AS cliente_id,
            cl_nm_cliente AS cliente_nome,
            COUNT(a.id_atendimento) AS total_atendimentos,
            COUNT(DISTINCT a.diagnostico) AS total_diagnosticos,
            ro.id_empresa AS rede_id,
            a.dt_atendimento AS data_atendimento
        FROM
            atendimento a
        INNER JOIN tb_cliente cl ON a.tb_cliente_id_cliente = cl.id_cliente
        INNER JOIN tb_rede_credenciais rc ON a.tb_rede_credenciais_id_empresa = rc.id_empresa
        WHERE
            a.dt_atendimento BETWEEN p_data_inicio AND p_data_fim
        GROUP BY
            cl.id_cliente, cl_nm_cliente, ro.id_empresa, a.dt_atendimento
        ORDER BY
            a.dt_atendimento DESC, total_atendimentos DESC
    ) LOOP
        PIPE ROW (atendimento_relatorio_linha(
            cliente_id,
            cliente_nome,
            total_atendimentos,
            total_diagnosticos,
            rede_id,
            data_atendimento
        ));
    END LOOP;
    RETURN;
END fn_relatorio_atendimentos;
/

```
- Bottom Panels:**
 - Saída do Script: A
 - Resultado da Consulta: X
 - Tarefa concluída em 0,103 segundos
- Status Bar:** Function FN_RELATORIO_ATENDIMENTOS compilado

Resultado com SELECT:

The screenshot shows the Oracle SQL Developer interface with the following details:

- File Path:** Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
- Toolbar:** Arquivo, Editar, Exibir, Navegar, Executar, Origem, Equipe, Ferramentas, Janela, Ajuda.
- Left Panel (Conexões):** Shows various database objects like Views, Índices, Pacotes, Procedures, Operadores, etc.
- Left Panel (Relatórios):** Shows reports like Relatórios Definidos pelo Usuário, Relatórios de View Analítica, Relatórios do Data Modeler, Relatórios do Dicionário de Dados, Relatórios do TimesTen, and Relatórios OLAP.
- Central Area (Planilha - Query Builder):**

```

        );
    END LOOP;
    RETURN;
END fn_relatorio_atendimentos;
/

```

Below the code, the query is executed:

```

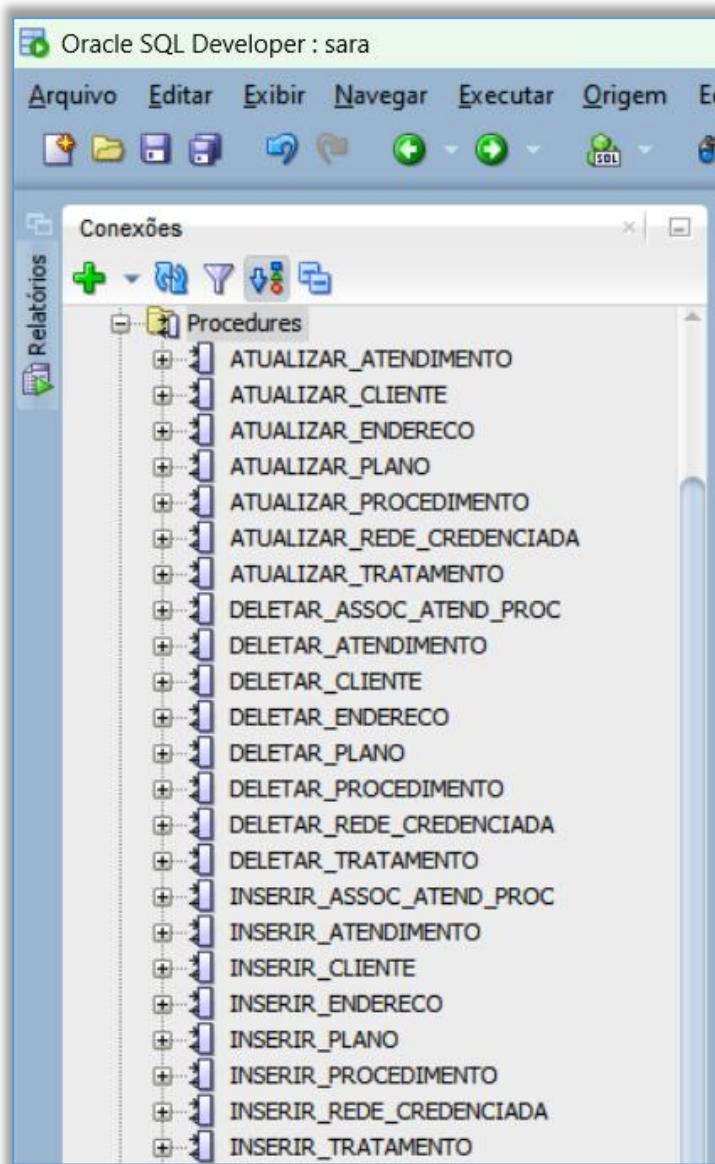
SELECT *
FROM TABLE(fn_relatorio_atendimentos(DATE '2023-01-01', DATE '2023-12-31'));

```

- Bottom Panels:**
 - Saída do Script: X
 - Resultado da Consulta: X
 - Resultado da Consulta 1: X
- Table Results:**

CLIENTE_ID	CLIENTE_NOME	TOTAL_ATENDIMENTOS	TOTAL_DIAGNOSTICOS	REDE_NOME	DATA_ATENDIMENTO
1 6	Aline Mendes Simoes	1	1	Consultorio Marques Nunes ltda	21/09/23
2 2	Caroline Andrade Correia	1	1	Consultorio Apagao Moreira ME	14/09/23
3 8	Loren Pratsinski Kurt	1	1	Consultorio dr Ney Santos me	30/08/23
4 1	Fernando Vilares Silva	1	1	Consultorio Meneses Rosario ME	12/08/23
5 9	João Victor Veronezi Matarazzo dos Santos	1	1	Almirante Exames Medicologicos Ltda	12/08/23
6 10	Ronilda Pedroso Gomes	1	1	Consultorio Fernando Muniz ME	18/07/23
7 3	Kamilla Raquelly Santos da Silva	1	1	Clinica Sorrisal Ltda	10/07/23
8 5	Jerônimo Nunes Pereira	1	1	Consultorio Rosimar e Conrado ME	11/06/23
9 4	Ronaldo Souza Goncalves	1	1	Clinice Dentalmais Ltda	09/06/23
10 7	Kleber Monteiro Souza	1	1	Dentistas no flow me	01/05/23
- Status Bar:** Salvo: C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql | Linha 901 Coluna 77 | Inserir

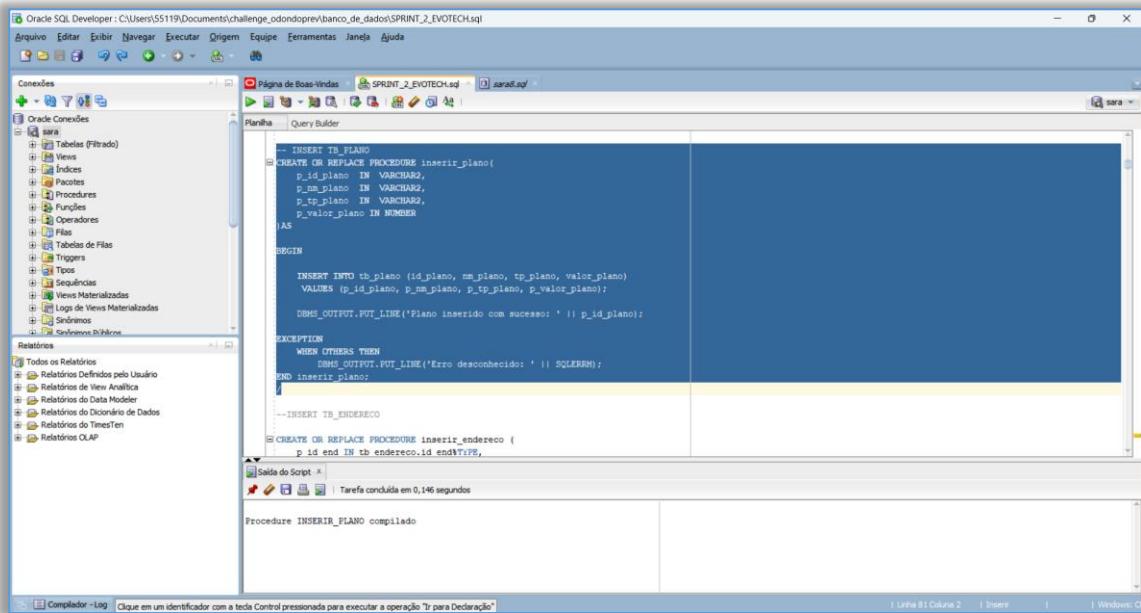
Lista de Procedures compiladas para todas as tabelas do projeto



PROCEDURES

Inserts nas tabelas

TB_PLANO



The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar:** Arquivo, Editar, Exibir, Navegar, Executar, Origem, Equipe, Ferramentas, Janela, Ajuda.
- Central Area (Panha):** Displays the PL/SQL code for the `inserir_plano` procedure.

```
-- INSERT TB_PLANO
CREATE OR REPLACE PROCEDURE inserir_plano(
    p_id_plano IN VARCHAR2,
    p_nm_plano IN VARCHAR2,
    p_tp_plano IN VARCHAR2,
    p_valor_plano IN NUMBER
) AS
BEGIN
    INSERT INTO tb_plano (id_plano, nm_plano, tp_plano, valor_plano)
    VALUES (p_id_plano, p_nm_plano, p_tp_plano, p_valor_plano);
    DBMS_OUTPUT.PUT_LINE('Plano inserido com sucesso: ' || p_id_plano);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro desconhecido: ' || SQLERRM);
END inserir_plano;
--INSERT TB_ENDERECHO
CREATE OR REPLACE PROCEDURE inserir_endereco (
    p_id_end IN tb_endereco.id_end%TYPE,
    p_endereco IN tb_endereco.endereco%TYPE
);
```
- Left Sidebar:** Shows the database schema with various objects like tables, views, procedures, and triggers.
- Bottom Status Bar:** Shows the message "Procedure INSERIR_PLANO compilado".

TB_ENDERECHO

```
--INSERT TB_ENDERECHO
CREATE OR REPLACE PROCEDURE inserir_endereco (
    p_id_end IN tb_endereco.id_end%TYPE,
    p_cep IN tb_endereco.cep%TYPE,
    p_logradouro IN tb_endereco.logradouro%TYPE,
    p_num_end IN tb_endereco.num_end%TYPE,
    p_compl_end IN tb_endereco.compl_end%TYPE,
    p_bairro IN tb_endereco.bairro%TYPE,
    p_cidade IN tb_endereco.cidade%TYPE,
    p_uf IN tb_endereco.uf%TYPE
) IS
BEGIN
    INSERT INTO tb_endereco (id_end, cep, logradouro, num_end, compl_end, bairro,
    cidade, uf)
    VALUES (p_id_end,p_cep,p_logradouro,p_num_end,p_compl_end,p_bairro,p_cidade,
    p_uf);

    DBMS_OUTPUT.PUT_LINE('Endereço inserido com sucesso: ' || p_id_end);

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao inserir endereço: ' || SQLERRM);
END inserir_endereco;
```

Procedure INserir_Endereco compiled

TB_PROCEDIMENTO

```
--INSERT TB_PROCEDIMENTO
CREATE OR REPLACE PROCEDURE inserir_procedimento (
    p_id_proc IN tb_procedimento.id_proc%TYPE,
    p_nm_proc IN tb_procedimento.nm_proc%TYPE,
    p_tp_proc IN tb_procedimento.tp_proc%TYPE,
    p_custo_medio IN tb_procedimento.custo_medio%TYPE,
    p_complexidade IN tb_procedimento.complexidade%TYPE
) IS
BEGIN
    -- Inserir o registro na tabela tb_procedimento
    INSERT INTO tb_procedimento (id_proc, nm_proc, tp_proc, custo_medio,
    complexidade)
    VALUES (p_id_proc, p_nm_proc, p_tp_proc, p_custo_medio, p_complexidade);

    DBMS_OUTPUT.PUT_LINE('Procedimento inserido com sucesso: ' || p_id_proc);

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao inserir procedimento: ' || SQLERRM);
END inserir_procedimento;
```

Procedure INserir_Procedimento compiled

TB_CLIENTE

```

Oracle SQL Developer : C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda

Conexões
  Oracle Conexões
    sara
      Tabelas (Filtrado)
      Views
      Índices
      Pacotes
      Procedures
      Funções
      Operadores
      Físicas
      Tabelas de Filas
      Triggers
      Tipos
      Sequências
      Views Materializadas
      Logs de Views Materializadas
      Síndromos
      Síndromos Rôbótico

Relatórios
  Todos os Relatórios
  Relatórios Definidos pelo Usuário
  Relatórios do Data Modeler
  Relatórios do Dicionário de Dados
  Relatórios do TimesTen
  Relatórios OLAP

Página de Boas-Vindas  SPRINT_2_EVOTECH.sql  sara8.sql
Planilha Query Builder

p_genero          IN CBAR,
p_telefone        IN VARCHAR2,
p_nm_endereco    IN VARCHAR2,
p_tb_endereco_id IN VARCHAR2,
p_tb_plano_id     IN VARCHAR2
) AS
BEGIN
  IF NOT validar_cpf(p_cpf) THEN
    RAISE_APPLICATION_ERROR(-20001, 'CPF inválido ou já existente.');
  END IF;

  IF NOT validar_email(p_email) THEN
    RAISE_APPLICATION_ERROR(-20002, 'E-mail inválido.');
  END IF;

  INSERT INTO tb_cliente (
    id_clie, cpf, dt_cadastro, nm_clie, dt_nasc, genero, telefone, email, tb_endereco_id_end, tb_plano_id_plano
  ) VALUES (
    p_id_clie, p_cpf, p_dt_cadastro, p_nm_clie, p_dt_nasc, p_genero, p_telefone, p_email, p_tb_endereco_id, p_tb_plano_id
  );

  DBMS_OUTPUT.PUT_LINE('Cliente inserido com sucesso: ' || p_id_clie);
EXCEPTION
  WHEN DUP_VAL_ON_INDEX THEN
    DBMS_OUTPUT.PUT_LINE('Erro: CPF, telefone ou email já cadastrados.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro desconhecido: ' || SQLEERR);
END inserir_cliente;

```

Procedura INSERIR_CLIENTE compilado

Linha 171 Coluna 2 | Inserir | Modificado | Windows: C

TB_REDE_CREDENCIADA

```

Oracle SQL Developer : C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda

Conexões
  Oracle Conexões
    sara
      Tabelas (Filtrado)
      Views
      Índices
      Pacotes
      Procedures
      Funções
      Operadores
      Físicas
      Tabelas de Filas
      Triggers
      Tipos
      Sequências
      Views Materializadas
      Logs de Views Materializadas
      Síndromos
      Síndromos Rôbótico

Relatórios
  Todos os Relatórios
  Relatórios Definidos pelo Usuário
  Relatórios do New Analytics
  Relatórios do Data Modeler
  Relatórios do Dicionário de Dados
  Relatórios do TimesTen
  Relatórios OLAP

Página de Boas-Vindas  SPRINT_2_EVOTECH.sql  sara8.sql
Planilha Query Builder

  INSERT INTO tb_rede_credenciada (
    id_empresa, cnpj, dt_cadastro, nm_empresa, especialidade, telefone, email, tb_endereco_id_end
  ) VALUES (
    p_id_empresa, p_cnpj, p_dt_cadastro, p_nm_empresa, p_especialidade, p_telefone, p_email, p_tb_endereco_id
  );

  DBMS_OUTPUT.PUT_LINE('Rede Credenciada inserida com sucesso: ' || p_id_empresa);
EXCEPTION
  WHEN DUP_VAL_ON_INDEX THEN
    DBMS_OUTPUT.PUT_LINE('Erro: CNPJ, telefone ou email já cadastrados.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro desconhecido: ' || SQLEERR);
END inserir_rede_credenciada;

```

--INSERT TB_ATENDIMENTO

```

  CREATE OR REPLACE PROCEDURE inserir_atendimento (
    p_id_atendimento IN tb_atendimento.id_atendimento%TYPE,
    p_dt_atendimento IN tb_atendimento.dt_atendimento%TYPE,
    p_diagnóstico IN tb_atendimento.diagnóstico%TYPE,
    p_tb_cliente_id_clie IN tb_atendimento.tb_cliente_id_clie%TYPE,
    p_tb_rede_credenciada_id_empresa IN tb_atendimento.tb_rede_credenciada_id_empresa%TYPE
  ) IS
    v_cliente_existe NUMBER;
    v_empresa_existe NUMBER;
  BEGIN
    -- Verificar se o cliente existe
    SELECT COUNT() INTO v_cliente_existe
    FROM tb_cliente
    WHERE id_clie = p_tb_cliente_id_clie;
    IF v_cliente_existe = 0 THEN
      RAISE_APPLICATION_ERROR(-20001, 'Cliente não encontrado.');
    END IF;
    -- Verificar se a empresa existe
    SELECT COUNT() INTO v_empresa_existe
    FROM tb_rede_credenciada
    WHERE id_empresa = p_tb_rede_credenciada_id_empresa;
    IF v_empresa_existe = 0 THEN
      RAISE_APPLICATION_ERROR(-20002, 'Rede credenciada não encontrada.');
    END IF;
    -- Inserir novo atendimento
    INSERT INTO tb_atendimento (
      id_atendimento, dt_atendimento, diagnóstico, tb_cliente_id_clie, tb_rede_credenciada_id_empresa
    ) VALUES (
      p_id_atendimento, p_dt_atendimento, p_diagnóstico, p_tb_cliente_id_clie, p_tb_rede_credenciada_id_empresa
    );
  END;

```

Procedure INSERIR_REDE_CREDENCIADA compilado

Linha 205 Coluna 2 | Inserir | Modificado | Windows: C

TB_ATENDIMENTO

The screenshot shows the Oracle SQL Developer interface with the following details:

- Connexões:** Oracle Conexões - sara
- Relatórios:** Todos os Relatórios
- Planilha / Query Builder:**

```
--INSERT TB_ATENDIMENTO
CREATE OR REPLACE PROCEDURE inserir_atendimento (
    p_id_atendimento IN tb_atendimento.id_atendimento%TYPE,
    p_dt_atendimento IN tb_atendimento.dt_atendimento%TYPE,
    p_diagnostico IN tb_atendimento.diagnostico%TYPE,
    p_tb_cliente_id_clie IN tb_atendimento.tb_cliente_id_clie%TYPE,
    p_tb_rede_credenciada_id_empresa IN tb_atendimento.tb_rede_credenciada_id_empresa%TYPE
) IS
    v_cliente_exists NUMBER;
    v_empresa_exists NUMBER;
BEGIN
    -- Verificar se o cliente existe
    SELECT COUNT(1)
    INTO v_cliente_exists
    FROM tb_cliente
    WHERE id_clie = p_tb_cliente_id_clie;

    IF v_cliente_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Cliente não encontrado: ' || p_tb_cliente_id_clie);
    END IF;

    -- Verificar se a rede credenciada existe
    SELECT COUNT(1)
    INTO v_empresa_exists
    FROM tb_rede_credenciada
    WHERE id_empresa = p_tb_rede_credenciada_id_empresa;

    IF v_empresa_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Rede credenciada não encontrada: ' || p_tb_rede_credenciada_id_empresa);
    END IF;
END;
```
- Saída do Script:** Tarefa concluída em 0,149 segundos
- Compositor - Log:** Clique em um identificador com a tecla Control pressionada para executar a operação "Tr para Declaração"

TB_TRATAMENTO

The screenshot shows the Oracle SQL Developer interface with the following details:

- Connexões:** Oracle Conexões - sara
- Relatórios:** Todos os Relatórios
- Planilha / Query Builder:**

```
--TB TRATAMENTO
CREATE OR REPLACE PROCEDURE inserir_tratamento (
    p_id_tratamento IN tb_tratamento.id_tratamento%TYPE,
    p_desc_tratamento IN tb_tratamento.desc_tratamento%TYPE,
    p_nivel_urgencia IN tb_tratamento.nivel_urgencia%TYPE,
    p_tb_atendimento_id_atendimento IN tb_tratamento.tb_atendimento_id_atendimento%TYPE
) IS
    v_atendimento_exists NUMBER;
BEGIN
    SELECT COUNT(1)
    INTO v_atendimento_exists
    FROM tb_atendimento
    WHERE id_atendimento = p_tb_atendimento_id_atendimento;

    IF v_atendimento_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Atendimento não encontrado: ' || p_tb_atendimento_id_atendimento);
    END IF;

    INSERT INTO tb_tratamento (id_tratamento, desc_tratamento, nivel_urgencia,
    tb_atendimento_id_atendimento)
    VALUES (p_id_tratamento, p_desc_tratamento, p_nivel_urgencia,
    p_tb_atendimento_id_atendimento);

    DBMS_OUTPUT.PUT_LINE('Tratamento inserido com sucesso: ' || p_id_tratamento);

    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao inserir tratamento: ' || SQLERRM);
    END inserir_tratamento;
```
- Saída do Script:** Tarefa concluída em 0,215 segundos
- Compositor - Log:** Salvo: C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

TB_ASSOC_ATEND_PROC

```

CREATE OR REPLACE PROCEDURE inserir_assoc_atend_proc(
    p_tb_atendimento_id_atendimento IN VARCHAR2,
    p_tb_procedimento_id_proc      IN VARCHAR2
) IS
    v_atendimento_exists NUMBER;
    v_procedimento_exists NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_atendimento_exists
    FROM tb_atendimento
    WHERE id_atendimento = p_tb_atendimento_id_atendimento;

    SELECT COUNT(*)
    INTO v_procedimento_exists
    FROM tb_procedimento
    WHERE id_proc = p_tb_procedimento_id_proc;

    IF v_atendimento_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Erro: Atendimento não encontrado.');
    ELSIF v_procedimento_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Erro: Procedimento não encontrado.');
    END IF;

    INSERT INTO tb_assoc_atend_proc (
        tb_atendimento_id_atendimento, tb_procedimento_id_proc
    ) VALUES (
        p_tb_atendimento_id_atendimento, p_tb_procedimento_id_proc
    );
END;

```

Procedure INserir_ASSOC_ATEND_PROC compilado

Atualização das tabelas

TB_PLANO

```

--UPDATE ATUALIZAR_PLANO
CREATE OR REPLACE PROCEDURE atualizar_plano(
    p_id_plano IN VARCHAR2,
    p_nm_plano IN VARCHAR2,
    p_tp_plano IN VARCHAR2,
    p_valor_plano IN NUMBER
) AS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_count
    FROM tb_plano
    WHERE id_plano = p_id_plano;

    IF v_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20003, 'Plano não encontrado.');
    END IF;

    UPDATE tb_plano SET
        nm_plano = p_nm_plano,
        tp_plano = p_tp_plano,
        valor_plano = p_valor_plano
    WHERE id_plano = p_id_plano;

    DBMS_OUTPUT.PUT_LINE('Plano atualizado com sucesso: ' || p_id_plano);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Plano não encontrado.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro: ' || SQLERRM);
END;

```

Procedure ATUALIZAR_PLANO compilado

TB_CLIENTE

Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

```
--> UPDATE TB_CLIENTE
--> CREATE OR REPLACE PROCEDURE atualizar_cliente(
-->   p_id_clie          IN VARCHAR2,
-->   p_cpf              IN VARCHAR2,
-->   p_dt_cadastro      IN DATE,
-->   p_nm_clie          IN VARCHAR2,
-->   p_dt_nasc           IN DATE,
-->   p_endereco          IN CHAR,
-->   p_telefone          IN VARCHAR2,
-->   p_email             IN VARCHAR2,
-->   p_tb_endereco_id   IN VARCHAR2,
-->   p_tb_plano_id       IN VARCHAR2
--> ) AS
-->   v_count NUMBER;
--> BEGIN
-->   SELECT COUNT(*)
-->     INTO v_count
-->   FROM tb_cliente
-->   WHERE id_clie = p_id_clie;
-->
-->   IF v_count = 0 THEN
-->     RAISE_APPLICATION_ERROR(-20003, 'Cliente não encontrado.');
-->   END IF;
-->
-->   IF NOT valida_cpf(p_cpf) THEN
-->     RAISE_APPLICATION_ERROR(-20001, 'CPF inválido ou já existente.');
-->   END IF;
-->
-->   UPDATE TB_CLIENTE
-->   SET cpf = p_cpf,
-->       dt_cadastro = p_dt_cadastro,
-->       nm_clie = p_nm_clie,
-->       dt_nasc = p_dt_nasc,
-->       endereco = p_endereco,
-->       telefone = p_telefone,
-->       email = p_email,
-->       tb_endereco_id = p_tb_endereco_id,
-->       tb_plano_id = p_tb_plano_id
-->   WHERE id_clie = p_id_clie;
-->
-->   v_count := 0;
-->   SELECT COUNT(*)
-->     INTO v_count
-->   FROM TB_CLIENTE
-->   WHERE id_clie = p_id_clie;
-->
-->   IF v_count > 1 THEN
-->     RAISE_APPLICATION_ERROR(-20002, 'Cliente já existe.');
-->   END IF;
-->
-->   COMMIT;
-->
-->   RAISE_APPLICATION_ERROR(-20000, 'Operação realizada com sucesso.');
-->
--> END;
-->
--> /
```

Salvo: C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

TB_REDE_CREDENCIADA

Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

```
--> UPDATE TB_REDE_CREDENCIADA
--> CREATE OR REPLACE PROCEDURE atualizar_rede_credenciada(
-->   p_id_empresa        IN VARCHAR2,
-->   p_cnpj              IN VARCHAR2,
-->   p_dt_cadastro       IN DATE,
-->   p_nm_empresa         IN VARCHAR2,
-->   p_especialidade     IN VARCHAR2,
-->   p_endereco           IN CHAR,
-->   p_email              IN VARCHAR2,
-->   p_tb_endereco_id    IN VARCHAR2
--> ) AS
-->   v_count NUMBER;
--> BEGIN
-->   SELECT COUNT(*)
-->     INTO v_count
-->   FROM tb_rede_credenciada
-->   WHERE id_empresa = p_id_empresa;
-->
-->   IF v_count = 0 THEN
-->     RAISE_APPLICATION_ERROR(-20003, 'Rede Credenciada não encontrada.');
-->   END IF;
-->
-->   IF NOT valida_cnpj(p_cnpj) THEN
-->     RAISE_APPLICATION_ERROR(-20001, 'CNPJ inválido ou já existente.');
-->   END IF;
-->
-->   UPDATE TB_REDE_CREDENCIADA
-->   SET cnpj = p_cnpj,
-->       dt_cadastro = p_dt_cadastro,
-->       nm_empresa = p_nm_empresa,
-->       especialidade = p_especialidade,
-->       endereco = p_endereco,
-->       email = p_email,
-->       tb_endereco_id = p_tb_endereco_id
-->   WHERE id_empresa = p_id_empresa;
-->
-->   v_count := 0;
-->   SELECT COUNT(*)
-->     INTO v_count
-->   FROM TB_REDE_CREDENCIADA
-->   WHERE id_empresa = p_id_empresa;
-->
-->   IF v_count > 1 THEN
-->     RAISE_APPLICATION_ERROR(-20002, 'Rede Credenciada já existe.');
-->   END IF;
-->
-->   COMMIT;
-->
-->   RAISE_APPLICATION_ERROR(-20000, 'Operação realizada com sucesso.');
-->
--> END;
-->
--> /
```

Salvo: C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

TB_ATENDIMENTO

Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda

Coneções Oracle Conexões sara Tabelas (Filtrado) Views Índices Pacotes Proceduras Funções Operadores Físicas Tabelas de Fila Triggers Tipos Sequências Views Materializadas Logs de Views Materializadas Síndromos Síndromos Búlhos Relatórios Todos os Relatórios Relatórios Definidos pelo Usuário Relatórios do Data Modeler Relatórios do Dicionário de Dados Relatórios do TimesTen Relatórios OLAP

Página de Boas-Vindas SPRINT_2_EVOTECH.sql sara8.sql Planilha Query Builder

```
--UPDATE TB_ATENDIMENTO
CREATE OR REPLACE PROCEDURE atualizar_atendimento (
    p_id_atendimento IN tb_atendimento.id_atendimento%TYPE,
    p_dt_atendimento IN tb_atendimento.dt_atendimento%TYPE,
    p_diagnostico IN tb_atendimento.diagnostico%TYPE,
    p_tb_cliente_id_clie IN tb_atendimento.tb_cliente_id_clie%TYPE,
    p_tb_rede_credenciada_id_empresa IN tb_atendimento.tb_rede_credenciada_id_empresa%TYPE
)
IS
    v_cliente_exists NUMBER;
    v_empresa_exists NUMBER;
BEGIN
    SELECT COUNT()
    INTO v_cliente_exists
    FROM tb_cliente
    WHERE id_clie = p_tb_cliente_id_clie;

    IF v_cliente_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-2001, 'Cliente não encontrado: ' || p_tb_cliente_id_clie);
    END IF;

    SELECT COUNT()
    INTO v_empresa_exists
    FROM tb_rede_credenciada
    WHERE id_empresa = p_tb_rede_credenciada_id_empresa;

    IF v_empresa_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-2002, 'Rede credenciada não encontrada: ' || p_tb_rede_credenciada_id_empresa);
    END IF;
END;
```

Saída do Script Tarefa concluída em 0,154 segundos

Procedura ATUALIZAR_ATENDIMENTO compilado

Linha 525 Coluna 2 | Inserir | Modificado | Windows | O

TB_ENDEREÇO

Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda

Coneções Oracle Conexões sara Tabelas (Filtrado) Views Índices Pacotes Proceduras Funções Operadores Físicas Tabelas de Fila Triggers Tipos Sequências Views Materializadas Logs de Views Materializadas Síndromos Síndromos Búlhos Relatórios Todos os Relatórios Relatórios Definidos pelo Usuário Relatórios do Data Modeler Relatórios do Dicionário de Dados Relatórios do TimesTen Relatórios OLAP

Página de Boas-Vindas SPRINT_2_EVOTECH.sql sara8.sql Planilha Query Builder

```
--UPDATE TB_ENDERECO
CREATE OR REPLACE PROCEDURE atualizar_endereco (
    p_id_end IN tb_endereco.id_end%TYPE,
    p_cpf IN tb_endereco.cpf%TYPE,
    p_logradouro IN tb_endereco.logradouro%TYPE,
    p_num_end IN tb_endereco.num_end%TYPE,
    p_compl_end IN tb_endereco.compl_end%TYPE,
    p_bairro IN tb_endereco.bairro%TYPE,
    p_cidade IN tb_endereco.cidade%TYPE,
    p_uf IN tb_endereco.uf%TYPE
)
IS
BEGIN
    -- Atualizar o registro na tabela tb_endereco
    UPDATE tb_endereco
    SET cep = p_cep,
        logradouro = p_logradouro,
        num_end = p_num_end,
        compl_end = p_compl_end,
        bairro = p_bairro,
        cidade = p_cidade,
        uf = p_uf
    WHERE id_end = p_id_end;

    IF SQLORCOUNT = 0 THEN
        RAISE_APPLICATION_ERROR(-2001, 'Endereço não encontrado: ' || p_id_end);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Endereço atualizado com sucesso: ' || p_id_end);
    END IF;
END;
```

Saída do Script Tarefa concluída em 0,124 segundos

Procedura ATUALIZAR_ENDERECO compilado

Linha 560 Coluna 2 | Inserir | Modificado | Windows | O

TB_PROCEDIMENTO

The screenshot shows the Oracle SQL Developer interface with the following details:

- Conexões:** Oracle Conexões - sara
- Página de Boas-Vindas:** SPRINT_2_EVOTECH.sql - sara8.sql
- Planilha:** Query Builder
- Código:**

```
-->CREATE OR REPLACE PROCEDURE ATUALIZAR_PROCEDIMENTO (
    p_id_proc IN tb_procedimento.id_procedureTYPE,
    p_nm_proc IN tb_procedimento.nm_procedureTYPE,
    p_tp_proc IN tb_procedimento.tp_procedureTYPE,
    p_custo_medio IN tb_procedimento.custo_medioTYPE,
    p_complexidade IN tb_procedimento.complexidadeTYPE
) IS
BEGIN
    -- Atualizar o registro na tabela tb_procedimento
    UPDATE tb_procedimento
    SET nm_proc = p_nm_proc,
        tp_proc = p_tp_proc,
        custo_medio = p_custo_medio,
        complexidade = p_complexidade
    WHERE id_proc = p_id_proc;

    IF SQAROWCOUNT = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Procedimento não encontrado: ' || p_id_proc);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Procedimento atualizado com sucesso: ' || p_id_proc);
    END IF;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao atualizar procedimento: ' || SQLERRM);
END ATUALIZAR_PROCEDIMENTO;
```

- Saída do Script:** Tarefa concluída em 0,123 segundos
- Comprador - Log:** Salvo: C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

TB_TRATAMENTO

The screenshot shows the Oracle SQL Developer interface with the following details:

- Conexões:** Oracle Conexões - sara
- Página de Boas-Vindas:** SPRINT_2_EVOTECH.sql - sara8.sql
- Planilha:** Query Builder
- Código:**

```
-->CREATE OR REPLACE PROCEDURE ATUALIZAR_TRATAMENTO (
    p_id_tratamento IN tb_tratamento.id_tratamentoTYPE,
    p_desc_tratamento IN tb_tratamento.desc_tratamentoTYPE,
    p_nivel_urgencia IN tb_tratamento.nivel_urgenciaTYPE,
    p_tb_atendimento_id_atendimento IN tb_tratamento.tb_atendimento_id_atendimentoTYPE
) IS
    v_atendimento_exists NUMBER;
BEGIN
    SELECT COUNT()
    INTO v_atendimento_exists
    FROM tb_atendimento
    WHERE id_atendimento = p_tb_atendimento_id_atendimento;

    IF v_atendimento_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Atendimento não encontrado: ' || p_tb_atendimento_id_atendimento);
    END IF;

    UPDATE tb_tratamento
    SET desc_tratamento = p_desc_tratamento,
        nivel_urgencia = p_nivel_urgencia,
        tb_atendimento_id_atendimento = p_tb_atendimento_id_atendimento
    WHERE id_tratamento = p_id_tratamento;

    IF SQAROWCOUNT = 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Tratamento não encontrado: ' || p_id_tratamento);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Tratamento atualizado com sucesso: ' || p_id_tratamento);
    END IF;
```

- Saída do Script:** Tarefa concluída em 0,124 segundos
- Comprador - Log:** Salvo: C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

Deletar dados

TB_CLIENTE

The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar:** Arquivo, Editar, Exibir, Navegar, Executar, Origem, Equipe, Ferramentas, Janela, Ajuda.
- Connections:** Shows various database objects like Views, Indices, Pacotes, Procedures, Funções, Operadores, Tabelas de Fitas, Triggers, Tipos, Sequências, Views Materializadas, Logs de Views Materializadas, Símbolos, Símbolos Públicos, Links de Banco de Dados, Links de Bancos de Dados Públicos, and Diretórios.
- Relatórios:** Shows various report types defined by the user.
- Painel:** Query Builder window containing the following PL/SQL code:

```
END atualizar_tratamento;
/
--PROCEDURES PARA DELETAR DADOS
--- DELETAR TB_CLIENTE
CREATE OR REPLACE PROCEDURE deletar_cliente(
    p_id_clie IN VARCHAR2
) IS
BEGIN
    -- Excluir o cliente
    DELETE FROM tb_cliente WHERE id_clie = p_id_clie;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Cliente não encontrado.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Cliente excluído com sucesso.');
    END IF;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao excluir cliente: ' || SQLERRM);
END deletar_cliente;
/
--DELETE TB_PLANO
```

- Saída do Script:** Shows the message "Procedure DELETAR_CLIENTE compilado".
- Bottom Status Bar:** Compilador - Log, Salvo: C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql, Linha 652 Coluna 2, Windows: 0.

TB_PLANO

The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar:** Arquivo, Editar, Exibir, Navegar, Executar, Origem, Equipe, Ferramentas, Janela, Ajuda.
- Connections:** Shows various database objects like Views, Indices, Pacotes, Procedures, Funções, Operadores, Tabelas de Fitas, Triggers, Tipos, Sequências, Views Materializadas, Logs de Views Materializadas, Símbolos, Símbolos Públicos, Links de Banco de Dados, Links de Bancos de Dados Públicos, and Diretórios.
- Relatórios:** Shows various report types defined by the user.
- Painel:** Query Builder window containing the following PL/SQL code:

```
--DELETE TB_PLANO
CREATE OR REPLACE PROCEDURE deletar_plano(
    p_id_plano IN VARCHAR2
) IS
BEGIN
    -- Excluir o plano
    DELETE FROM tb_plano WHERE id_plano = p_id_plano;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Plano não encontrado.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Plano excluído com sucesso.');
    END IF;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao excluir plano: ' || SQLERRM);
END deletar_plano;
/
--DELETE TB_REDE_CRED
CREATE OR REPLACE PROCEDURE deletar_rede_credenciada(
    p_id_empresa IN VARCHAR2
) IS
BEGIN
    DELETE FROM tb_rede_credenciada WHERE id_empresa = p_id_empresa;
```

- Saída do Script:** Shows the message "Procedure DELETAR_PLANO compilado".
- Bottom Status Bar:** Compilador - Log, Salvo: C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql, Linha 674 Coluna 1, Windows: 0.

TB_REDE_CREDENCIADA

```
--DELETE TB_REDE_CRED
CREATE OR REPLACE PROCEDURE deletar_rede_credenciada(
    p_id_empresa IN VARCHAR2
) IS
BEGIN
    DELETE FROM tb_rede_credenciada WHERE id_empresa = p_id_empresa;
    IF SQLEROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Empresa não encontrada.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('empresa excluída com sucesso.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao excluir empresa: ' || SQLEERRM);
END deletar_rede_credenciada;

--DELETE ENDERECO
CREATE OR REPLACE PROCEDURE deletar_endereco(
    p_id_endereco IN VARCHAR2
) IS
BEGIN
    DELETE FROM tb_endereco WHERE id_end= p_id_endereco;

```

Salvo: C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

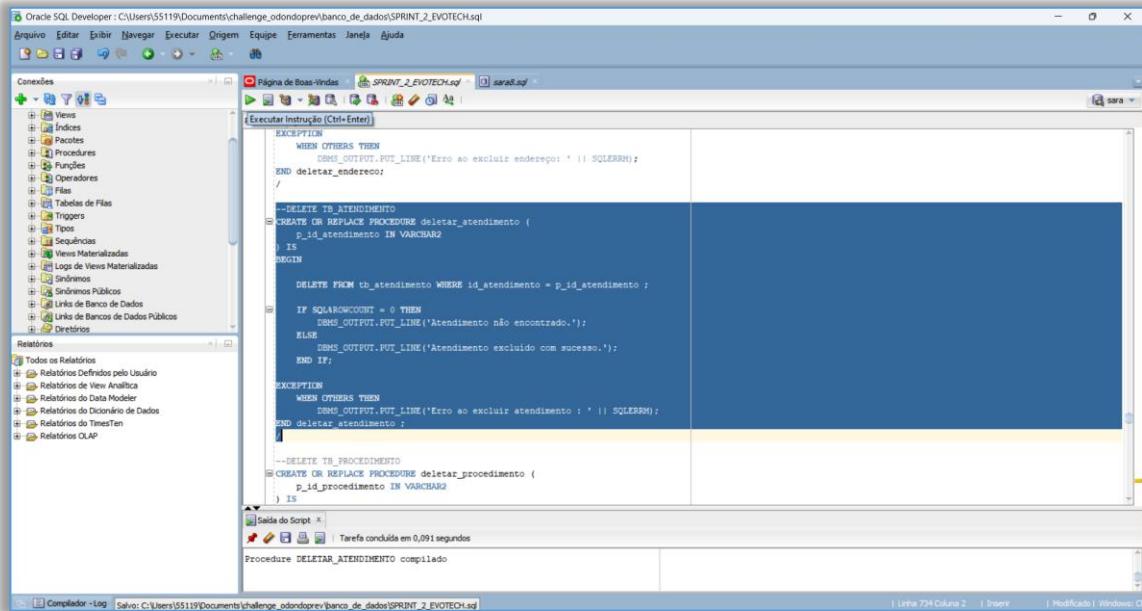
TB_ENDEREÇO

```
END deletar_rede_credenciada;
/
--DELETE ENDERECO
CREATE OR REPLACE PROCEDURE deletar_endereco(
    p_id_endereco IN VARCHAR2
) IS
BEGIN
    DELETE FROM tb_endereco WHERE id_end= p_id_endereco;
    IF SQLEROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Endereço não encontrado.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Endereço excluído com sucesso.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao excluir endereço: ' || SQLEERRM);
END deletar_endereco;
/
--DELETE TB_ATENDIMENTO
CREATE OR REPLACE PROCEDURE deletar_atendimento (
    p_id_atendimento IN VARCHAR2
) IS
BEGIN
    DELETE FROM tb_atendimento WHERE id_atendimento = p_id_atendimento ;

```

Salvo: C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

TB_ATENDIMENTO



```

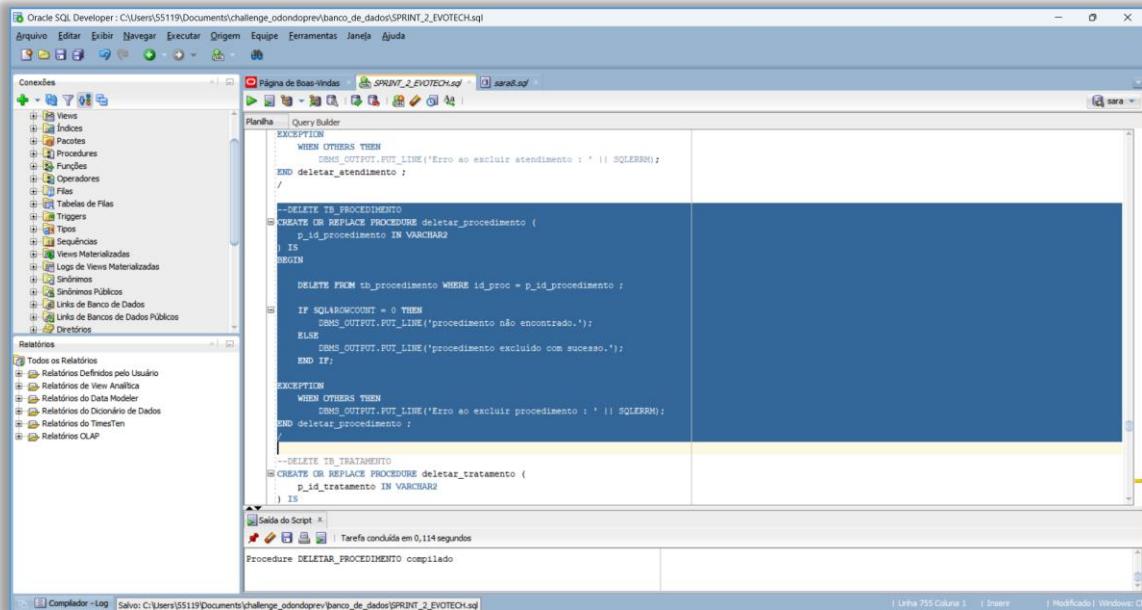
Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda
Conexões Página de Boas-Vindas SPRINT_2_EVOTECH.sql sara&sql
Executar Instrução (Ctrl+Enter)
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro ao excluir endereço : ' || SQLEERR);
END deletar_endereco;
/
--DELETE TB_ATENDIMENTO
CREATE OR REPLACE PROCEDURE deletar_atendimento (
  p_id_atendimento IN VARCHAR2
) IS
BEGIN
  DELETE FROM tb_atendimento WHERE id_atendimento = p_id_atendimento ;
  IF SQLROWCOUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Atendimento não encontrado.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Atendimento excluído com sucesso.');
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro ao excluir atendimento : ' || SQLEERR);
END deletar_atendimento;
/
--DELETE TB_PROCEDIMENTO
CREATE OR REPLACE PROCEDURE deletar_procedimento (
  p_id_procedimento IN VARCHAR2
) IS
BEGIN
  DELETE FROM tb_procedimento WHERE id_proc = p_id_procedimento ;
  IF SQLROWCOUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE('procedimento não encontrado.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('procedimento excluído com sucesso.');
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro ao excluir procedimento : ' || SQLEERR);
END deletar_procedimento;
/
--DELETE TB_TRAITEMENTO
CREATE OR REPLACE PROCEDURE deletar_tratamento (
  p_id_tratamento IN VARCHAR2
) IS
BEGIN
  DELETE FROM tb_tratamento WHERE id_tratamento = p_id_tratamento ;
  IF SQLROWCOUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE('tratamento não encontrado.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('tratamento excluído com sucesso.');
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro ao excluir tratamento : ' || SQLEERR);
END deletar_tratamento;
/

```

Saida do Script : Tarefa concluída em 0,091 segundos

Procedure DELETAR_ATENDIMENTO compilado

TB_PROCEDIMENTO



```

Oracle SQL Developer : C:\Users\55119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda
Conexões Página de Boas-Vindas SPRINT_2_EVOTECH.sql sara&sql
Panha Query Builder
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro ao excluir atendimento : ' || SQLEERR);
END deletar_atendimento;
/
--DELETE TB_PROCEDIMENTO
CREATE OR REPLACE PROCEDURE deletar_procedimento (
  p_id_procedimento IN VARCHAR2
) IS
BEGIN
  DELETE FROM tb_procedimento WHERE id_proc = p_id_procedimento ;
  IF SQLROWCOUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE('procedimento não encontrado.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('procedimento excluído com sucesso.');
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro ao excluir procedimento : ' || SQLEERR);
END deletar_procedimento;
/
--DELETE TB_TRAITEMENTO
CREATE OR REPLACE PROCEDURE deletar_tratamento (
  p_id_tratamento IN VARCHAR2
) IS
BEGIN
  DELETE FROM tb_tratamento WHERE id_tratamento = p_id_tratamento ;
  IF SQLROWCOUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE('tratamento não encontrado.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('tratamento excluído com sucesso.');
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro ao excluir tratamento : ' || SQLEERR);
END deletar_tratamento;
/

```

Saida do Script : Tarefa concluída em 0,114 segundos

Procedure DELETAR_PROCEDIMENTO compilado

TB_TRAATAMENTO

```

Oracle SQL Developer : C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda
Conexões Página de Boas-Vindas SPRINT_2_EVOTECH.sql sara8.sql
Panha Query Builder
--DBMS_OUTPUT.PUT_LINE('Erro ao excluir procedimento : ' || SQLERRM);
END deletar_procedimento ;
/
--DELETE TB_TRAATAMENTO
CREATE OR REPLACE PROCEDURE deletar_tratamento (
    p_id_tratamento IN VARCHAR2
) IS
BEGIN
    DELETE FROM tb_tratamento WHERE id_tratamento = p_id_tratamento ;
    IF SQLROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('tratamento não encontrado.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('tratamento excluído com sucesso.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao excluir tratamento : ' || SQLERRM);
END deletar_tratamento ;
/
--DELETE TB_ASSOCIATIVA
CREATE OR REPLACE PROCEDURE deletar_assoc_atend_proc(
    p_tb_atendimento_id_atendimento IN VARCHAR2,
    p_tb_procedimento_id_proc      IN VARCHAR2
)
/

```

Salvo: C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

Compilador - Log | Linha 77; Coluna 1 | Inserir | Modificado | Windows |

TB_ASSOC_ATEND_PROC

```

Oracle SQL Developer : C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql
Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda
Conexões Página de Boas-Vindas SPRINT_2_EVOTECH.sql sara8.sql
Panha Executar Instrução (Ctrl+Enter)
--DBMS_OUTPUT.PUT_LINE('Erro ao excluir tratamento : ' || SQLERRM);
END deletar_tratamento ;
/
--DELETE TB_ASSOCIATIVA
CREATE OR REPLACE PROCEDURE deletar_assoc_atend_proc(
    p_tb_atendimento_id_atendimento IN VARCHAR2,
    p_tb_procedimento_id_proc      IN VARCHAR2
) IS
BEGIN
    DELETE FROM tb_assoc_atend_proc
    WHERE tb_atendimento_id_atendimento = p_tb_atendimento_id_atendimento
        AND tb_procedimento_id_proc = p_tb_procedimento_id_proc;
    IF SQLROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Associação não encontrada.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Associação deletada com sucesso.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao deletar associação: ' || SQLERRM);
END deletar_assoc_atend_proc ;
/

```

Salvo: C:\Users\SS119\Documents\challenge_odontoprev\banco_de_dados\SPRINT_2_EVOTECH.sql

Compilador - Log | Linha 799; Coluna 2 | Inserir | Modificado | Windows |