

**Faculdade de Informática e Administração Paulista**

**CHALLENGE ODONTOPREV – ODONTOFAST**

**Projeto:** Aplicação de Acompanhamento para Incentivo ao Tratamento Preventivo



**MASTERING RELATIONAL AND NON-RELATIONAL DATABASE**

**INTEGRANTES (2TDSPS)**

Felipe Amador RM: 553528

Leonardo de Oliveira RM: 554024

Sara Sousa RM: 552656

**São Paulo**

**Outubro/2024**

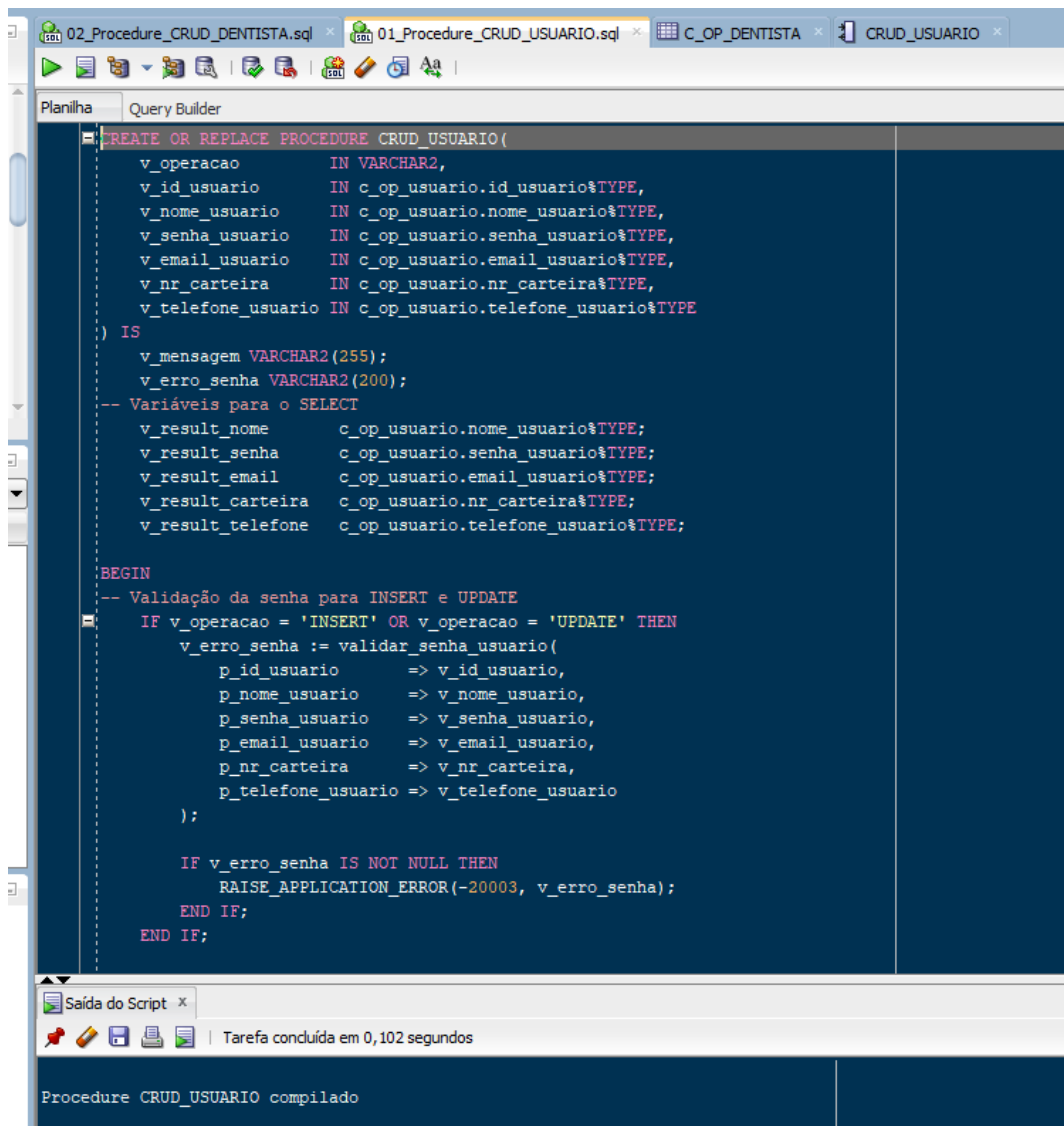
# Projeto de Acompanhamento Odontológico

Link vídeo (inserção de dados nas tabelas):

<https://youtu.be/S6WYpJgMUXs>

Prints da execução dos scripts:

1. Execução do CRUD, insert e exemplo de erro para a validação para a tabela **C\_OP\_USUARIO**:



```
CREATE OR REPLACE PROCEDURE CRUD_USUARIO(
    v_operacao      IN VARCHAR2,
    v_id_usuario    IN c_op_usuario.id_usuario%TYPE,
    v_nome_usuario  IN c_op_usuario.nome_usuario%TYPE,
    v_senha_usuario IN c_op_usuario.senha_usuario%TYPE,
    v_email_usuario IN c_op_usuario.email_usuario%TYPE,
    v_nr_carteira   IN c_op_usuario.nr_carteira%TYPE,
    v_telefone_usuario IN c_op_usuario.telefone_usuario%TYPE
) IS
    v_mensagem VARCHAR2(255);
    v_erro_senha VARCHAR2(200);
    -- Variáveis para o SELECT
    v_result_nome      c_op_usuario.nome_usuario%TYPE;
    v_result_senha     c_op_usuario.senha_usuario%TYPE;
    v_result_email     c_op_usuario.email_usuario%TYPE;
    v_result_carteira  c_op_usuario.nr_carteira%TYPE;
    v_result_telefone  c_op_usuario.telefone_usuario%TYPE;

BEGIN
    -- Validação da senha para INSERT e UPDATE
    IF v_operacao = 'INSERT' OR v_operacao = 'UPDATE' THEN
        v_erro_senha := validar_senha_usuario(
            p_id_usuario      => v_id_usuario,
            p_nome_usuario    => v_nome_usuario,
            p_senha_usuario   => v_senha_usuario,
            p_email_usuario   => v_email_usuario,
            p_nr_carteira     => v_nr_carteira,
            p_telefone_usuario => v_telefone_usuario
        );

        IF v_erro_senha IS NOT NULL THEN
            RAISE_APPLICATION_ERROR(-20003, v_erro_senha);
        END IF;
    END IF;
END IF;
```

Procedure CRUD\_USUARIO compilado

Funções\_validacoes.sql | Procedure\_CRUD\_usuario.sql | C\_OP\_USUARIO | CRUD\_ENDERECO\_CLIENTE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort: | Filter:

ID_USUARIO	NOME_USUARIO	SENHA_USUARIO	EMAIL_USUARIO	NR_CARTEIRA	TELEFONE_USUARIO
1	Ana Souza	senha@123	ana.souza@example.com	CARTEIRA1234	11999999999
2	Carlos Silva	senha@456	carlos.silva@example.com	CARTEIRA5678	11998888888
3	Maria Santos	senha@789	maria.santos@example.com	CARTEIRA9012	11977777777
4	João Oliveira	senha@234	joao.oliveira@example.com	CARTEIRA3456	11966666666
5	Laura Pereira	senha@567	laura.pereira@example.com	CARTEIRA7890	11955555555
6	Pedro Ferreira	senha@890	pedro.ferreira@example.com	CARTEIRA2345	11944444444
7	Camila Costa	senha@345	camila.costa@example.com	CARTEIRA6789	11933333333
8	Rafael Almeida	senha@678	rafael.almeida@example.com	CARTEIRA0123	11922222222
9	Julia Rodrigues	senha@901	julia.rodrigues@example.com	CARTEIRA4567	11911111111
10	Marcos Lima	senha@234	marcos.lima@example.com	CARTEIRA8901	11900000000
11	555 Raphael Amador	SenhaSegura4545	rapha.amador@email.com	12548844	11994151589

```

BEGIN
  CRUD_USUARIO(
    v_operacao => 'INSERT',
    v_id_usuario => 555,
    v_nome_usuario => 'Raphael Amador',
    v_senha_usuario => 'SenhaSegura4545',
    v_email_usuario => 'rapha.amador@email.com',
    v_nr_carteira => '12548844',
    v_telefone_usuario => 11994151589
  );
END;

```

Script Output x

Task completed in 0,023 seconds

PL/SQL procedure successfully completed.

Funções\_validacoes.sql | Procedure\_CRUD\_usuario.sql | Procedure\_CRUD\_dentista.sql | C\_OP\_DI

Worksheet | Query Builder

```

EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Erro ao realizar a operação: ' || SQLERRM);
END CRUD_USUARIO;

BEGIN
  CRUD_USUARIO(
    v_operacao => 'INSERT',
    v_id_usuario => 666,
    v_nome_usuario => 'Felipe Amador',
    v_senha_usuario => 'SenhaSegura4545/(4556//-----,,,',
    v_email_usuario => 'rapha.amador@email.com',
    v_nr_carteira => '12548844',
    v_telefone_usuario => 11994151589
  );
END;

```

Script Output x

Task completed in 0,023 seconds

Erro ao realizar a operação: ORA-20003: Senha deve ter 20 caracteres ou menos.

PL/SQL procedure successfully completed.

## 2. Execução do CRUD, insert e exemplo de erro para a validação para a tabela C\_OP\_DENTISTA:

```

CREATE OR REPLACE PROCEDURE CRUD_DENTISTA
(
  v_operacao IN VARCHAR2,
  v_id_dentista IN c_op_dentista.id_dentista%TYPE,
  v_nome_dentista IN c_op_dentista.nome_dentista%TYPE,
  v_senha_dentista IN c_op_dentista.senha_dentista%TYPE,
  v_cro IN c_op_dentista.cro%TYPE,
  v_telefone_dentista IN c_op_dentista.telefone_dentista%TYPE,
  v_email_dentista IN c_op_dentista.email_dentista%TYPE,
  v_id_especialidade IN c_op_dentista.id_especialidade%TYPE
)
IS
  v_erro_cro VARCHAR2(400);
  v_mensagem VARCHAR2(255);

  -- Variáveis para o SELECT
  v_result_nome c_op_dentista.nome_dentista%TYPE;
  v_result_senha c_op_dentista.senha_dentista%TYPE;
  v_result_cro c_op_dentista.cro%TYPE;
  v_result_telefone c_op_dentista.telefone_dentista%TYPE;
  v_result_email c_op_dentista.email_dentista%TYPE;
  v_result_especialidade c_op_dentista.id_especialidade%TYPE;

BEGIN
  -- Validação do CRO para INSERT e UPDATE
  IF v_operacao = 'INSERT' OR v_operacao = 'UPDATE' THEN
    v_erro_cro := validar_formato_cro(v_cro);

    IF v_erro_cro IS NOT NULL THEN
      RAISE_APPLICATION_ERROR(-20003, v_erro_cro);
    END IF;
  END IF;

  IF v_operacao = 'INSERT' THEN
    INSERT INTO c_op_dentista (
      id_dentista,

```

```

BEGIN
  CRUD_DENTISTA(
    v_operacao => 'INSERT',
    v_id_dentista => 5454,
    v_nome_dentista => 'Dr. Carlos Halilton',
    v_senha_dentista => 'Senha123',
    v_cro => '12345SP',
    v_telefone_dentista => '11987654321',
    v_email_dentista => 'carlos.halilton@exemplo.com',
    v_id_especialidade => 2
  );
END;

```

ID_DENTISTA	NOME_DENTISTA	SENHA_DENTISTA	CRO	TELEFONE_DENTISTA	EMAIL_DENTISTA	ID_ESPECIALIDADE
1	Dr. Roberto Andrade	senha123	CRO12345	11999999999	roberto.andrade@exemplo.com	1
2	Dra. Carla Mendes	senha86	CRO12346	11999999999	carla.mendes@exemplo.com	2
3	Dr. Fernando Santos	senha789	CRO14567	11977777777	fernando.santos@exemplo.com	3
4	Dra. Luciana Oliveira	senha234	CRO45678	11966666666	luciana.oliveira@exemplo.com	4
5	Dr. Marcelo Costa	senha567	CRO6789	11955555555	marcelo.costa@exemplo.com	5
6	Dra. Patricia Lima	senha90	CRO7890	11944444444	patricia.lima@exemplo.com	1
7	Dr. Ricardo Ferreira	senha45	CRO9012	11933333333	ricardo.ferreira@exemplo.com	2
8	Dra. Sandra Rodrigues	senha70	CRO9012	11922222222	sandra.rodrigues@exemplo.com	3
9	Dr. Thiago Almeida	senha901	CRO9123	11911111111	thiago.almeida@exemplo.com	4
10	Dra. Vanessa Pereira	senha34	CRO1234	11900000000	vanessa.pereira@exemplo.com	5
11	5454 Dr. Carlos Halilton	Senha123	12345SP	11987654321	carlos.halilton@exemplo.com	2

Script Output: Task completed in 0,027 seconds

Dentista inserido com sucesso.

PL/SQL procedure successfully completed.

```

BEGIN
  CRUD_DENTISTA(
    v_operacao => 'INSERT',
    v_id_dentista => 5454,
    v_nome_dentista => 'Dr. Carlos Halilton',
    v_senha_dentista => 'Senha123',
    v_cro => '12345SP45484',
    v_telefone_dentista => '11987654321',
    v_email_dentista => 'carlos.halilton@exemplo.com',
    v_id_especialidade => 2
  );
END;

```

Script Output: Task completed in 0,02 seconds

Erro ao realizar a operação: ORA-20003: CRO deve ter entre 4 e 8 caracteres. CRO deve conter apenas números seguidos de duas letras para a UF.

PL/SQL procedure successfully completed.

### 3. Execução do CRUD e insert para a tabela **C\_OP\_ESPECIALIDADE**:

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The main window displays the following SQL code:

```
-- CRUD tabela C_OP_ESPECIALIDADE

CREATE OR REPLACE PROCEDURE CRUD_ESPECIALIDADE (
    v_operacao          IN VARCHAR2,
    v_id_especialidade  IN c_op_especialidade.id_especialidade%TYPE,
    v_tipo_especialidade IN c_op_especialidade.tipo_especialidade%TYPE,
    v_descr_especialidade IN c_op_especialidade.descr_especialidade%TYPE
) IS
    v_mensagem VARCHAR2(255);
BEGIN
    IF v_operacao = 'INSERT' THEN
        INSERT INTO c_op_especialidade (
            id_especialidade,
            tipo_especialidade,
            descr_especialidade
        ) VALUES (
            v_id_especialidade,
            v_tipo_especialidade,
            v_descr_especialidade
        );
        v_mensagem := 'Especialidade inserida com sucesso.';

    ELSIF v_operacao = 'UPDATE' THEN
        UPDATE c_op_especialidade
```

The 'Script Output' tab at the bottom shows the message: "Task completed in 0,06 seconds". Below the script output, it states: "Procedure CRUD\_ESPECIALIDADE compiled".

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The main window displays the following SQL code:

```
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao realizar a operação: ' || SQLERRM);
END CRUD_ESPECIALIDADE;

BEGIN
    CRUD_ESPECIALIDADE(
        v_operacao => 'INSERT',
        v_id_especialidade => 6,
        v_tipo_especialidade => 'Buco-Maxilo',
        v_descr_especialidade => 'Especialidade focada em cirurgias da face, maxila e mandibula.'
    );
END;
```

The 'Script Output' tab at the bottom shows the message: "Task completed in 0,023 seconds". Below the script output, it states: "Especialidade inserida com sucesso." and "PL/SQL procedure successfully completed.".

The 'Data' tab on the right shows the following data in the C\_OP\_ESPECIALIDADE table:

ID_ESPECIALIDADE	TIPO_ESPECIALIDADE	DESCR_ESPECIALIDADE
1	Ortodontia	Especialidade que corrige a posição dos dentes e dos ossos maxilares
2	Endodontia	Tratamento de canal e outras doenças da polpa dentária
3	Periodontia	Tratamento das doenças das gengivas e ossos de suporte dos dentes
4	Implantodontia	Colocação de implantes dentários
5	Odontopediatria	Cuidados dentários especializados para crianças
6	Buco-Maxilo	Especialidade focada em cirurgias da face, maxila e mandibula.

#### 4. Execução do CRUD e insert para a tabela **C\_OP\_CHECKLIST**:

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The script being executed is for creating a procedure named 'CRUD\_CHECKLIST'. The procedure takes several input parameters: v\_operacao (VARCHAR2), v\_id\_checklist (IN c\_op\_checklist.id\_checklist%TYPE), v\_nivel (IN c\_op\_checklist.nivel%TYPE), v\_escovacao\_dentes (IN c\_op\_checklist.escovacao\_dentes%TYPE), v\_fio\_dental (IN c\_op\_checklist.fio\_dental%TYPE), v\_enxaguante\_bucal (IN c\_op\_checklist.enxaguante\_bucal%TYPE), and v\_id\_usuario (IN c\_op\_usuario.id\_usuario%TYPE). The procedure is designed to handle an 'INSERT' operation by inserting data into the 'c\_op\_checklist' table. It includes a message variable 'v\_mensagem' and a 'BEGIN' block with an 'IF' statement for the 'INSERT' operation. The 'INSERT' statement lists the columns: id\_checklist, nivel, escovacao\_dentes, fio\_dental, enxaguante\_bucal, and id\_usuario. The 'VALUES' clause provides the corresponding input values. After the insert, a message is set: 'v\_mensagem := 'Checklist inserido com sucesso.';'. The script is compiled successfully, as indicated by the 'Saída do Script' window showing 'Tarefa concluída em 0,099 segundos' and 'Procedure CRUD\_CHECKLIST compilado'.

```
-- CRUD tabela C_OP_CHECKLIST

CREATE OR REPLACE PROCEDURE CRUD_CHECKLIST (
    v_operacao      IN VARCHAR2,
    v_id_checklist  IN c_op_checklist.id_checklist%TYPE,
    v_nivel         IN c_op_checklist.nivel%TYPE,
    v_escovacao_dentes IN c_op_checklist.escovacao_dentes%TYPE,
    v_fio_dental    IN c_op_checklist.fio_dental%TYPE,
    v_enxaguante_bucal IN c_op_checklist.enxaguante_bucal%TYPE,
    v_id_usuario    IN c_op_usuario.id_usuario%TYPE
) IS
    v_mensagem VARCHAR2(255);
    -- Variáveis para SELECT
    v_result_nivel         c_op_checklist.nivel%TYPE;
    v_result_escovacao     c_op_checklist.escovacao_dentes%TYPE;
    v_result_fio_dental    c_op_checklist.fio_dental%TYPE;
    v_result_enxaguante     c_op_checklist.enxaguante_bucal%TYPE;
BEGIN
    IF v_operacao = 'INSERT' THEN
        INSERT INTO c_op_checklist (
            id_checklist,
            nivel,
            escovacao_dentes,
            fio_dental,
            enxaguante_bucal,
            id_usuario
        ) VALUES (
            v_id_checklist,
            v_nivel,
            v_escovacao_dentes,
            v_fio_dental,
            v_enxaguante_bucal,
            v_id_usuario
        );
        v_mensagem := 'Checklist inserido com sucesso.';
    END IF;
END;
```

The screenshot shows the SQL Developer interface with the 'Script Output' window open. The script being executed is the 'CRUD\_CHECKLIST' procedure. The output shows 'Checklist inserido com sucesso.' and 'PL/SQL procedure successfully completed.'.

The 'Query Builder' window shows the data in the 'C\_OP\_CHECKLIST' table. The table has columns: ID\_CHECKLIST, NIVEL, ESCOVACAO\_DENTES, FIO\_DENTAL, ENXAGUANTE\_BUCAL, and ID\_USUARIO. The data is as follows:

ID_CHECKLIST	NIVEL	ESCOVACAO_DENTES	FIO_DENTAL	ENXAGUANTE_BUCAL	ID_USUARIO
1	1	3	3	2	1
2	2	2	2	1	0
3	3	4	3	3	2
4	4	1	1	0	0
5	5	3	3	2	1
6	6	2	2	1	1
7	7	4	3	3	3
8	8	3	3	2	2
9	9	2	2	1	0
10	10	3	3	2	1
11	11	2	1	1	1

## 5. Execução do CRUD e insert para a tabela **C\_OP\_STATUS**:

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The main window displays the following PL/SQL code:

```

CREATE OR REPLACE PROCEDURE CRUD_STATUS (
    v_operacao      IN VARCHAR2,
    v_id_status     IN c_op_status.id_status%TYPE,
    v_tipo_status   IN c_op_status.tipo_status%TYPE,
    v_descr_status  IN c_op_status.descr_status%TYPE
) IS
    v_mensagem VARCHAR2(255);
    -- Variáveis para SELECT
    v_result_tipo  c_op_status.tipo_status%TYPE;
    v_result_descr c_op_status.descr_status%TYPE;
BEGIN
    IF v_operacao = 'INSERT' THEN
        INSERT INTO c_op_status (
            id_status,
            tipo_status,
            descr_status
        ) VALUES (
            v_id_status,
            v_tipo_status,
            v_descr_status
        );
        v_mensagem := 'Status inserido com sucesso.';

    ELSIF v_operacao = 'UPDATE' THEN
        UPDATE c_op_status
        SET tipo_status = v_tipo_status,
            descr_status = v_descr_status
        WHERE id_status = v_id_status;
        v_mensagem := 'Registro atualizado com sucesso.';

    ELSIF v_operacao = 'DELETE' THEN
        DELETE FROM c_op_status
        WHERE id_status = v_id_status;
    END IF;
END;
  
```

Below the code editor, the 'Saída do Script' (Script Output) window shows the execution results:

```

Tarefa concluída em 0,086 segundos

Procedure CRUD_STATUS compilado
  
```

The screenshot shows the SQL Developer interface with the 'Script' tab active. The main window displays the following PL/SQL code:

```

BEGIN
    CRUD_STATUS(
        v_operacao => 'INSERT',
        v_id_status => 11,
        v_tipo_status => 'Aguardando confirmação',
        v_descr_status => 'Esperando a confirmação do dentista ou clínica'
    );
END;
  
```

Below the code editor, the 'Saída do Script' (Script Output) window shows the execution results:

```

Tarefa concluída em 0,194 segundos

Procedure CRUD_STATUS compilado

Status inserido com sucesso.

Procedimento PL/SQL concluído com sucesso.
  
```

The screenshot shows the SQL Developer interface with the 'Table' tab active. The main window displays the following table data:

ID_STATUS	TIPO_STATUS	DESCR_STATUS
1	Agendado	Consulta marcada e confirmada
2	Em andamento	Tratamento iniciado e em progresso
3	Concluído	Tratamento finalizado com sucesso
4	Cancelado	Consulta ou tratamento cancelado
5	Reagendado	Consulta remarcada para outra data
6	Em espera	Paciente na lista de espera para atendimento
7	Não compareceu	Paciente faltou à consulta agendada
8	Em revisão	Paciente em fase de acompanhamento pós-tratamento
9	Pendente de pagamento	Aguardando pagamento do tratamento
10	Emergência	Atendimento de urgência
11	Aguardando confirmação	Esperando a confirmação do dentista ou clínica

## 6. Execução do CRUD e insert para a tabela **C\_OP\_TIPOPLANO**:

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The procedure being created is named 'CRUD\_TIPO\_PLANO'. It takes four input parameters: 'v\_operacao' (VARCHAR2), 'v\_id\_tipo\_plano' (C\_OP\_TIPO\_PLANO.ID\_TIPO\_PLANO%TYPE), 'v\_nome\_tipo\_plano' (C\_OP\_TIPO\_PLANO.NOME\_TIPO\_PLANO%TYPE), and 'v\_descr\_tipo\_plano' (C\_OP\_TIPO\_PLANO.DESCR\_TIPO\_PLANO%TYPE). The procedure logic includes an 'INSERT' branch, an 'UPDATE' branch, and a 'DELETE' branch, each with a corresponding success message. The 'INSERT' branch inserts a new record into the 'C\_OP\_TIPO\_PLANO' table. The 'UPDATE' branch updates the 'NOME\_TIPO\_PLANO' and 'DESCR\_TIPO\_PLANO' fields. The 'DELETE' branch deletes the record from the 'C\_OP\_TIPO\_PLANO' table. The procedure is compiled successfully, as indicated by the 'Saída do Script' window showing 'Tarefa concluída em 0,093 segundos' and 'Procedure CRUD\_TIPO\_PLANO compilado'.

```
--Set SERVEROUTPUT on;

CREATE OR REPLACE PROCEDURE CRUD_TIPO_PLANO (
    v_operacao      IN VARCHAR2,
    v_id_tipo_plano IN c_op_tipo_plano.id_tipo_plano%TYPE,
    v_nome_tipo_plano IN c_op_tipo_plano.nome_tipo_plano%TYPE,
    v_descr_tipo_plano IN c_op_tipo_plano.descr_tipo_plano%TYPE
) IS
    v_mensagem VARCHAR2(255);
    -- Variáveis para SELECT
    v_result_nome      c_op_tipo_plano.nome_tipo_plano%TYPE;
    v_result_descr     c_op_tipo_plano.descr_tipo_plano%TYPE;
BEGIN
    IF v_operacao = 'INSERT' THEN
        INSERT INTO c_op_tipo_plano (
            id_tipo_plano,
            nome_tipo_plano,
            descr_tipo_plano
        ) VALUES (
            v_id_tipo_plano,
            v_nome_tipo_plano,
            v_descr_tipo_plano
        );
        v_mensagem := 'Tipo de plano inserido com sucesso.';

    ELSIF v_operacao = 'UPDATE' THEN
        UPDATE c_op_tipo_plano
        SET nome_tipo_plano = v_nome_tipo_plano,
            descr_tipo_plano = v_descr_tipo_plano
        WHERE id_tipo_plano = v_id_tipo_plano;
        v_mensagem := 'Registro atualizado com sucesso.';

    ELSIF v_operacao = 'DELETE' THEN
        DELETE FROM c_op_tipo_plano

```

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The procedure being executed is named 'CRUD\_TIPO\_PLANO'. It takes four input parameters: 'v\_operacao' (VARCHAR2), 'v\_id\_tipo\_plano' (C\_OP\_TIPO\_PLANO.ID\_TIPO\_PLANO%TYPE), 'v\_nome\_tipo\_plano' (C\_OP\_TIPO\_PLANO.NOME\_TIPO\_PLANO%TYPE), and 'v\_descr\_tipo\_plano' (C\_OP\_TIPO\_PLANO.DESCR\_TIPO\_PLANO%TYPE). The procedure logic includes an 'INSERT' branch, an 'UPDATE' branch, and a 'DELETE' branch, each with a corresponding success message. The 'INSERT' branch inserts a new record into the 'C\_OP\_TIPO\_PLANO' table. The 'UPDATE' branch updates the 'NOME\_TIPO\_PLANO' and 'DESCR\_TIPO\_PLANO' fields. The 'DELETE' branch deletes the record from the 'C\_OP\_TIPO\_PLANO' table. The procedure is compiled successfully, as indicated by the 'Saída do Script' window showing 'Tarefa concluída em 0,051 segundos' and 'Tipo de plano inserido com sucesso.' and 'Procedimento PL/SQL concluído com sucesso.'

```
47 BEGIN
48   CRUD_TIPO_PLANO(
49     v_operacao => 'INSERT',
50     v_id_tipo_plano => 10,
51     v_nome_tipo_plano => 'Emergencial',
52     v_descr_tipo_plano => 'Para fins de emergencias inesperadas.'
53   );
54 END;
55
56
57
58
59
60
61
```

ID_TIPO_PLANO	NOME_TIPO_PLANO	DESCR_TIPO_PLANO
1	Básico	Cobertura para procedimentos preventivos e básicos
2	Intermediário	Cobertura básica mais alguns procedimentos especializados
3	Completo	Cobertura ampla incluindo a maioria dos procedimentos
4	Premium	Cobertura total com benefícios adicionais
5	Familiar	Plano com cobertura para toda a família
6	Empresarial	Plano corporativo para funcionários
7	Estudantil	Plano com descontos para estudantes
8	Sênior	Plano focado em necessidades de pacientes idosos
9	Ortodôntico	Plano especializado em tratamentos ortodônticos
10	Emergencial	Para fins de emergencias inesperadas.



## 7. Execução do CRUD e insert para a tabela **C\_OP\_PLANOSAUDE**:

```

set SERVEROUTPUT on;

CREATE OR REPLACE PROCEDURE CRUD_PLANO_SAUDE (
    v_operacao      IN VARCHAR2,
    v_id_plano      IN c_op_plano_de_saude.id_plano%TYPE DEFAULT NULL,
    v_nome_plano    IN c_op_plano_de_saude.nome_plano%TYPE DEFAULT NULL,
    v_telefone_plano IN c_op_plano_de_saude.telefone_plano%TYPE DEFAULT NULL,
    v_email_plano   IN c_op_plano_de_saude.email_plano%TYPE DEFAULT NULL,
    v_id_tipo_plano IN c_op_plano_de_saude.id_tipo_plano%TYPE DEFAULT NULL
) IS
    v_mensagem VARCHAR2(255);
    v_nome c_op_plano_de_saude.nome_plano%TYPE;
    v_telefone c_op_plano_de_saude.telefone_plano%TYPE;
    v_email c_op_plano_de_saude.email_plano%TYPE;
    v_tipo_plano c_op_plano_de_saude.id_tipo_plano%TYPE;
BEGIN
    IF v_operacao = 'INSERT' THEN
        INSERT INTO c_op_plano_de_saude (
            id_plano,
            nome_plano,
            telefone_plano,
            email_plano,
            id_tipo_plano
        ) VALUES (
            v_id_plano,
            v_nome_plano,
            v_telefone_plano,
            v_email_plano,
            v_id_tipo_plano
        );
    END IF;
END;

```

Script Output x

Task completed in 0,026 seconds

Procedure CRUD\_PLANO\_SAUDE compiled

```

BEGIN
    CRUD_PLANO_SAUDE(
        v_operacao => 'INSERT',
        v_id_plano => 12,
        v_nome_plano => 'Plano B',
        v_telefone_plano => 1234567890,
        v_email_plano => 'contato@planob.com',
        v_id_tipo_plano => 5
    );
END;

```

Script Output x

Task completed in 0,026 seconds

Plano de saúde inserido com sucesso.

PL/SQL procedure successfully completed.

ID_PLANO	NOME_PLANO	TELEFONE_PLANO	EMAIL_PLANO	ID_TIPO_PLANO
1	Alice	11987654321	contato@alice.com	1
2	Assim Saúde	11987654322	contato@assin-saude.com	2
3	Intermédica Notredame	11987654323	contato@intermedica-notredame.com	3
4	Klini Saúde	11987654324	contato@klini-saude.com	4
5	MedSênior	11987654325	contato@medsenior.com	5
6	Nosso Médico Intermédica	11987654326	contato@nosso-medico-intermedica.com	6
7	Portomed - Porto Seguro	11987654327	contato@portoseguro.com	7
8	SB Saúde	11987654328	contato@sbsaude.com	3
9	Brasil Dental	11987654329	contato@brasildental.com	5
10	Bradesco Saúde	11987654330	contato@bradescosaude.com	2
11	Plano A	1234567890	contato@planoa.com	2
12	Plano B	1234567890	contato@planob.com	5

## 8. Execução do CRUD e insert para a tabela **C\_OP\_NOTIIFICACAO**:

The screenshot shows the SQL Developer interface with the following tabs: 08\_Procedure\_CRUD\_NOTIIFICACAO.sql, CriacaoTabelas(SQL).sql, C\_OP\_PLANO\_DE\_SAUDE, and 07\_... The main window displays the SQL code for creating a procedure named CRUD\_NOTIIFICACAO. The procedure takes five input parameters: v\_operacao (VARCHAR2), v\_id\_notificacao (NUMBER), v\_mensagem (VARCHAR2), v\_data\_envio (DATE), and v\_leitura (NUMBER). It uses a conditional logic to perform INSERT or UPDATE operations on the C\_OP\_NOTIIFICACAO table. The Script Output window at the bottom shows the message "Task completed in 0,058 seconds" and "Procedure CRUD\_NOTIIFICACAO compiled".

```

CREATE OR REPLACE PROCEDURE CRUD_NOTIIFICACAO (
    v_operacao          IN VARCHAR2,
    v_id_notificacao    IN c_op_notificacao.id_notificacao%TYPE DEFAULT NULL,
    v_mensagem          IN c_op_notificacao.mensagem%TYPE DEFAULT NULL,
    v_data_envio        IN c_op_notificacao.data_envio%TYPE DEFAULT NULL,
    v_leitura           IN c_op_notificacao.leitura%TYPE DEFAULT NULL
) IS
    v_mensagem_resposta VARCHAR2(255);
    v_msg c_op_notificacao.mensagem%TYPE;
    v_data c_op_notificacao.data_envio%TYPE;
    v_leitura_status c_op_notificacao.leitura%TYPE;
BEGIN
    IF v_operacao = 'INSERT' THEN
        INSERT INTO c_op_notificacao (
            id_notificacao,
            mensagem,
            data_envio,
            leitura
        ) VALUES (
            v_id_notificacao,
            v_mensagem,
            v_data_envio,
            v_leitura
        );
        v_mensagem_resposta := 'Notificação inserida com sucesso.';

    ELSIF v_operacao = 'UPDATE' THEN
        UPDATE c_op_notificacao

```

The screenshot shows the SQL Developer interface with the following tabs: 08\_Procedure\_CRUD\_NOTIIFICACAO.sql, CriacaoTabelas(SQL).sql, and 07\_Procedure\_CRUD\_PLA... The main window displays the SQL code for calling the procedure. The Script Output window at the bottom shows the message "Task completed in 0,03 seconds" and "Procedure CRUD\_NOTIIFICACAO compiled". Below the script output, the text "Notificação inserida com sucesso." and "PL/SQL procedure successfully completed." are displayed. On the right, the Data tab of the C\_OP\_NOTIIFICACAO table is shown, displaying 11 rows of data.

```

BEGIN
    CRUD_NOTIIFICACAO(
        v_operacao => 'INSERT',
        v_id_notificacao => 11,
        v_mensagem => 'Notificação de teste.',
        v_data_envio => SYSDATE,
        v_leitura => 'N'
    );
END;

```

ID_NOTIFICACAO	MENSAGEM	DATA_ENVIO	LEITURA
1	Nova mensagem!	01-JAN-22	N
2	Alerta de sistema!	01-JAN-22	S
3	Escove os dentes!	01-JAN-22	N
4	Use o fio dental!	01-JAN-22	S
5	Use mais enxaguante bucal!	01-JAN-22	N
6	Você tem uma consulta amanhã!	01-JAN-22	S
7	Você tem uma consulta hoje!	01-JAN-22	N
8	Consulta reagendada!	01-JAN-22	S
9	Nova mensagem de usuário!	01-JAN-22	N
10	Sistema em manutenção!	01-JAN-22	S
11	Notificação de teste.	31-OCT-24	N

## 9. Execução do CRUD e insert para a tabela C\_OP\_TRATAMENTO:

The screenshot shows the SQL Developer interface with the following components:

- Worksheet:** Contains the PL/SQL code for the procedure `CRUD_TRATAMENTO`. The code defines input parameters for various fields and performs an insert operation into the `c_op_tratamento` table.
- Script Output:** Displays the message "Task completed in 0,074 seconds" and "Procedure CRUD\_TRATAMENTO compiled".

```

CREATE OR REPLACE PROCEDURE CRUD_TRATAMENTO (
    v_operacao          IN VARCHAR2,
    v_id_tratamento     IN c_op_tratamento.id_tratamento%TYPE DEFAULT NULL,
    v_tipo_tratamento  IN c_op_tratamento.tipo_tratamento%TYPE DEFAULT NULL,
    v_dt_inicio_tratamento IN c_op_tratamento.dt_inicio_tratamento%TYPE DEFAULT NULL,
    v_dt_termino_tratamento IN c_op_tratamento.dt_termino_tratamento%TYPE DEFAULT NULL,
    v_descr_tratamento IN c_op_tratamento.descr_tratamento%TYPE DEFAULT NULL,
    v_id_dentista        IN c_op_tratamento.id_dentista%TYPE DEFAULT NULL,
    v_id_usuario          IN c_op_tratamento.id_usuario%TYPE DEFAULT NULL,
    v_id_notificacao      IN c_op_tratamento.id_notificacao%TYPE DEFAULT NULL,
    v_id_status           IN c_op_tratamento.id_status%TYPE DEFAULT NULL,
    v_id_plano            IN c_op_tratamento.id_plano%TYPE DEFAULT NULL
) IS
    v_mensagem_resposta VARCHAR2(255);
    v_tipo               c_op_tratamento.tipo_tratamento%TYPE;
    v_data_inicio        c_op_tratamento.dt_inicio_tratamento%TYPE;
    v_data_termino       c_op_tratamento.dt_termino_tratamento%TYPE;
    v_descricao          c_op_tratamento.descr_tratamento%TYPE;
    v_id_dentista_out     c_op_tratamento.id_dentista%TYPE;
    v_id_usuario_out      c_op_tratamento.id_usuario%TYPE;
    v_id_notificacao_out  c_op_tratamento.id_notificacao%TYPE;
    v_id_status_out       c_op_tratamento.id_status%TYPE;
    v_id_plano_out        c_op_tratamento.id_plano%TYPE;
BEGIN
    IF v_operacao = 'INSERT' THEN
        INSERT INTO c_op_tratamento (
            id tratamento,

```

The screenshot shows the SQL Developer interface with the following components:

- Script Output:** Displays the message "Task completed in 0,022 seconds" and "Tratamento inserido com sucesso.".
- Table Data:** A table with 12 rows and 10 columns, showing the data inserted into the `C_OP_TRATAMENTO` table.

ID_TRATAMENTO	TIPO_TRATAMENTO	DT_INICIO_TRATAMENTO	DT_TERMINO_TRATAMENTO	DESCR_TRATAMENTO	ID_DENTISTA	ID_USUARIO	ID_NOTIFICACAO	ID_STATUS	ID_PLANO
1	1 Limpeza	22-DEC-24	05-APR-25	Limpeza completa	1	1	1	3	1
2	2 Canal	15-NOV-24	20-MAY-25	Tratamento de canal no molar	2	2	2	3	2
3	11 Canal	15-NOV-24	16-NOV-24	Limpeza	2	2	2	3	2
4	3 Clareamento	28-OCT-24	02-JUL-25	Clareamento dentário completo	3	3	3	3	3
5	4 Aparelho	10-JUN-24	16-AUG-25	Colocação de aparelho ortodôntico	4	4	4	2	4
6	5 Extração	24-JUN-24	29-SEP-25	Extração do siso	5	5	5	3	5
7	6 Restauração	01-JUL-02	13-OCT-25	Restauração de dente frontal	6	6	6	3	6
8	7 Implante	12-AUG-24	27-NOV-25	Implante dentário	7	7	7	3	7
9	8 Ortodontia	26-SEP-24	04-DEC-25	Ajuste ortodôntico mensal	8	8	8	2	8
10	9 Canal	03-OCT-24	18-JAN-25	Tratamento de canal no incisivo	9	9	9	3	9
11	10 Ponte Fixa	17-NOV-24	01-FEB-25	Colocação de ponte fixa	10	10	10	1	10
12	12 Limpeza	31-OCT-24	(null)	Limpeza dental padrão.	10	2	11	1	5

## 10. Execução do CRUD e insert para a tabela **C\_OP\_AGENDAMENTO**:

```

CREATE OR REPLACE PROCEDURE CRUD_AGENDAMENTO (
    v_operacao          IN VARCHAR2,
    v_id_agendamento    IN c_op_agendamento.id_agendamento%TYPE DEFAULT NULL,
    v_data_agendada     IN c_op_agendamento.data_agendada%TYPE DEFAULT NULL,
    v_horario_agendado  IN c_op_agendamento.horario_agendado%TYPE DEFAULT NULL,
    v_id_status         IN c_op_agendamento.id_status%TYPE DEFAULT NULL,
    v_id_tratamento    IN c_op_agendamento.id_tratamento%TYPE DEFAULT NULL
) IS
    v_mensagem_resposta VARCHAR2(255);
    v_data_agendada_out c_op_agendamento.data_agendada%TYPE;
    v_horario_agendado_out c_op_agendamento.horario_agendado%TYPE;
    v_id_status_out      c_op_agendamento.id_status%TYPE;
    v_id_tratamento_out c_op_agendamento.id_tratamento%TYPE;
BEGIN
    IF v_operacao = 'INSERT' THEN
        INSERT INTO c_op_agendamento (
            id_agendamento,
            data_agendada,
            horario_agendado,
            id_status,
            id_tratamento
        ) VALUES (
            v_id_agendamento,
            v_data_agendada,
            v_horario_agendado,
            v_id_status,
            v_id_tratamento
        );
    END IF;
END;

```

Script Output x

Task completed in 0,066 seconds

Procedure CRUD\_AGENDAMENTO compiled

```

BEGIN
    CRUD_AGENDAMENTO(
        v_operacao => 'INSERT',
        v_id_agendamento => 11,
        v_data_agendada => TO_DATE('2024-11-10', 'YYYY-MM-DD'),
        v_horario_agendado => TO_DATE('2024-11-10 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),
        v_id_status => 1,
        v_id_tratamento => 12
    );
END;

```

Script Output x

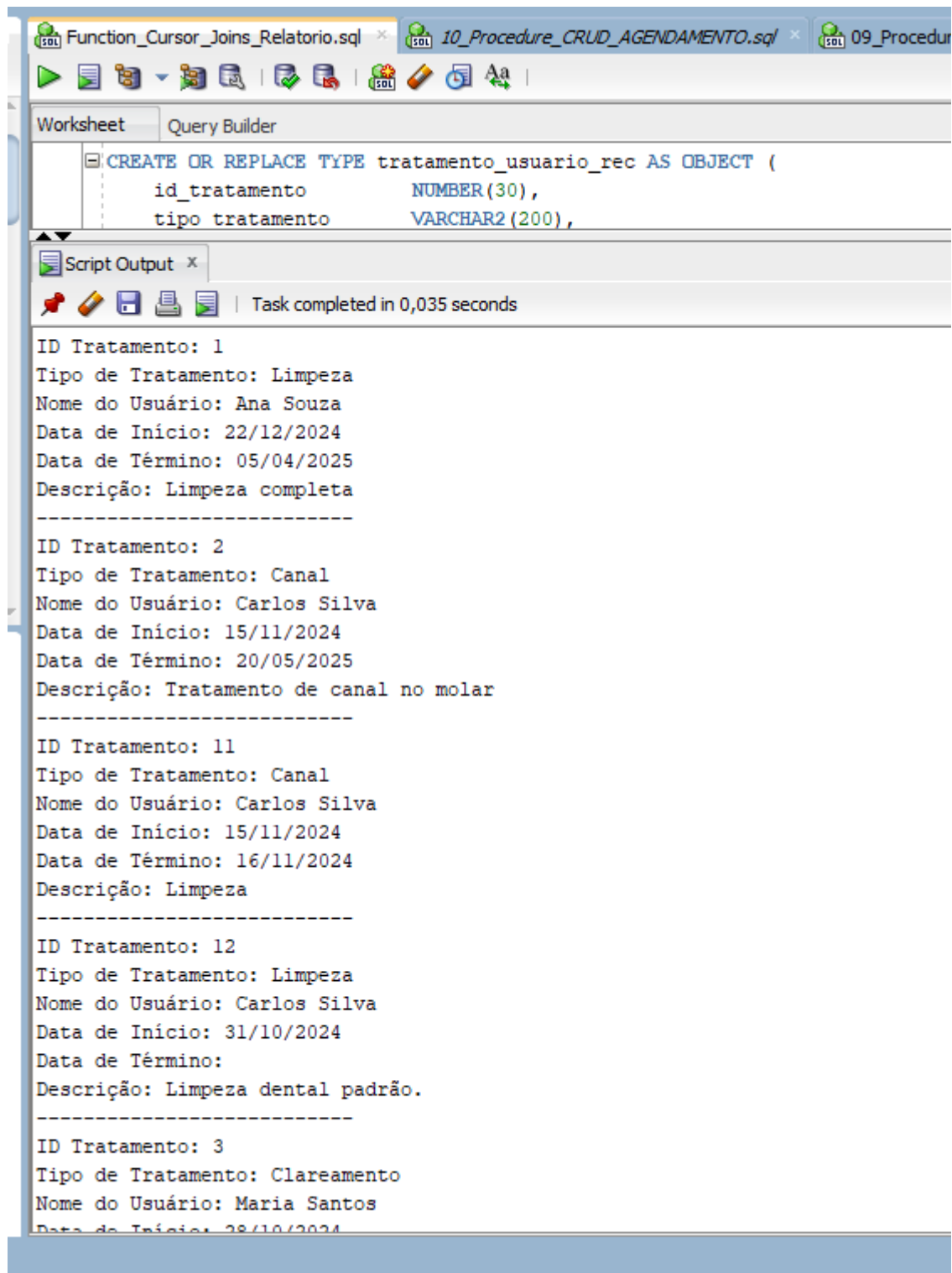
Task completed in 0,032 seconds

Agendamento inserido com sucesso.

PL/SQL procedure successfully completed.

ID_AGENDAMENTO	DATA_AGENDADA	HORARIO_AGENDADO	ID_STATUS	ID_TRATAMENTO
1	1 25-DEC-23	01-OCT-24	2	1
2	2 03-JAN-24	01-OCT-24	1	2
3	3 15-APR-23	01-OCT-24	3	3
4	4 28-FEB-24	01-OCT-24	2	4
5	5 10-JUL-23	01-OCT-24	1	5
6	6 22-NOV-23	01-OCT-24	3	6
7	7 05-SEP-24	01-OCT-24	2	7
8	8 18-MAR-24	01-OCT-24	1	8
9	9 20-APR-24	01-OCT-24	2	9
10	10 25-MAY-24	01-OCT-24	3	10
11	11 10-NOV-24	10-NOV-24	1	12

## 11. Execução da Função com Cursor e Joins para Relatório Formatado:



## 12. Execução da Função para Relatório com Regra de Negócio:

The screenshot displays a SQL IDE interface with three tabs: 'CriacaoTabelas(SQL).sql', 'Function\_Relatorio\_RegraDeNegocio.sql', and 'Function\_Cursor\_Joins\_Relatorio.sql'. The 'Function\_Relatorio\_RegraDeNegocio.sql' tab is active, showing a PL/SQL script. The script defines a function 'gerar\_relatorio\_agendamentos\_dia' that uses a cursor to fetch data from a table and returns a result set. The script is as follows:

```

c_op_dentista d ON a.id_tratamento = d.id_dentista
INNER JOIN
TABLE(v_tratamentos) t ON t.id_tratamento = a.id_tratamento
GROUP BY
a.data_agendada, d.nome_dentista, t.tipo_tratamento, t.descr_tratamento, t.nome_usuario
ORDER BY
a.data_agendada, d.nome_dentista
) LOOP
PIPE ROW (
relatorio_agendamentos_dia_extendido(
r.data_agendada,
r.dentista,
r.tratamentos_count,
r.tipo_tratamento,
r.descr_tratamento,
r.usuario_responsavel
)
);
END LOOP;

RETURN;
END gerar_relatorio_agendamentos_dia;

SELECT * FROM TABLE(gerar_relatorio_agendamentos_dia);

```

Below the script, the 'Resultado da Consulta' tab shows the output of the function. The results are displayed in a table with the following columns: DATA\_AGENDADA, DENTISTA, TRATAMENTOS\_COUNT, TIPO\_TRATAMENTO, DESCR\_TRATAMENTO, and USUARIO\_RESPONSAVEL. The table contains 10 rows of data.

DATA_AGENDADA	DENTISTA	TRATAMENTOS_COUNT	TIPO_TRATAMENTO	DESCR_TRATAMENTO	USUARIO_RESPONSAVEL
15/04/23	Dr. Fernando Santos	1	Clareamento	Clareamento dentário completo	Maria Santos
10/07/23	Dr. Marcelo Costa	1	Extração	Extração do siso	Laura Pereira
22/11/23	Dra. Patricia Lima	1	Restauração	Restauração de dente frontal	Pedro Ferreira
25/12/23	Dr. Roberto Andrade	1	Limpeza	Limpeza completa	Ana Souza
03/01/24	Dra. Carla Mendes	1	Canal	Tratamento de canal no molar	Carlos Silva
28/02/24	Dra. Luciana Oliveira	1	Aparelho	Colocação de aparelho ortodôntico	João Oliveira
18/03/24	Dra. Sandra Rodrigues	1	Ortodontia	Ajuste ortodôntico mensal	Rafael Almeida
20/04/24	Dr. Thiago Almeida	1	Canal	Tratamento de canal no incisivo	Julia Rodrigues
25/05/24	Dra. Vanessa Pereira	1	Ponte Fixa	Colocação de ponte fixa	Marcos Lima
05/09/24	Dr. Ricardo Ferreira	1	Implante	Implante dentário	Camila Costa