

### Procedure para limpar e remover tabelas existentes:

```
--limpa tabelas(apaga o conteúdo existente)
CREATE OR REPLACE PROCEDURE limpa_tabelas IS
BEGIN
    DELETE FROM RECEITA;
    DELETE FROM CONSULTA;
    DELETE FROM MEDICO;
    DELETE FROM USUARIO;
END;

create or replace PROCEDURE dropa_tabelas
IS
BEGIN
    runddl('DROP TABLE RECEITA');
    runddl('DROP TABLE CONSULTA');
    runddl('DROP TABLE MEDICO');
    runddl('DROP TABLE USUARIO');
END;
```

### Procedure para para criação de tabelas:

```
--procedure para criar as tabelas
CREATE OR REPLACE PROCEDURE cria_tabelas IS
    usuario VARCHAR2(500) := '
    CREATE TABLE USUARIO (
    ID_USUARIO INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    NOME VARCHAR2(100) NOT NULL,
    CPF VARCHAR2(12) NOT NULL
    )';

    medico VARCHAR2(500) := '
    CREATE TABLE MEDICO (
    ID_MEDICO INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    NOME          VARCHAR2(100) NOT NULL,
    CPF   VARCHAR2(12) NOT NULL,
    CRM   VARCHAR2(10) NOT NULL,
    ESPECIALIDADE VARCHAR2(50) NOT NULL
    )';

    consulta VARCHAR2(500) := '
    CREATE TABLE CONSULTA (
    ID_CONSULTA INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    DATA_HORA  TIMESTAMP NOT NULL,
    DESCRICAO  VARCHAR2(100) NOT NULL,
    ID_MEDICO   NUMBER(10) NOT NULL,
```

```

ID_USUARIO NUMBER(10) NOT NULL,
FOREIGN KEY (ID_MEDICO) REFERENCES MEDICO (ID_MEDICO),
FOREIGN KEY (ID_USUARIO) REFERENCES USUARIO (ID_USUARIO)
);

```

```

receita VARCHAR2(500) := '
CREATE TABLE RECEITA (
ID_RECEITA INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
DATA_EMISSAO TIMESTAMP NOT NULL,
PRESCRICAO VARCHAR2(100) NOT NULL,
ID_CONSULTA NUMBER(10) NOT NULL,
FOREIGN KEY (ID_CONSULTA) REFERENCES CONSULTA (ID_CONSULTA)
);

```

```

BEGIN
--EXECUTE IMMEDIATE usuario;
--EXECUTE IMMEDIATE medico;
--EXECUTE IMMEDIATE consulta;
--EXECUTE IMMEDIATE receita;
END;

```

#### Procedure para população das tabelas:

```

CREATE OR REPLACE PROCEDURE popular_tabelas IS
BEGIN
-- Populando tabela USUARIO (10 USUARIOS)
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário0', '1700545535');
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário1', '1234567890');
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário2', '0987654321');
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário3', '2345789245');
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário4', '9523486578');
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário5', '1954283486');
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário6', '1029384681');
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário7', '0293857694');
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário8', '5934593821');
INSERT INTO USUARIO (NOME, CPF) VALUES ('Usuário9', '5434534667');

-- Populando tabela MEDICO (10 MEDICOS)
INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico0',
'1700545535', '12345678', 'Clínico Geral');
INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico1',
'1234567890', '23523456', 'Cardiologia');
INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico2',
'0987654321', '63645788', 'Imunologia');

```

```

INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico3',
'2345789245', '06456234', 'Dermatologia');
INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico4',
'9523486578', '57567867', 'Infectologia');
INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico5',
'1954283486', '53465689', 'Neurologia');
INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico6',
'1029384681', '54564568', 'Oftalmologia');
INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico7',
'0293857694', '34563456', 'Pediatria');
INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico8',
'5934593821', '45674570', 'Pneumologia');
INSERT INTO MEDICO (NOME, CPF, CRM, ESPECIALIDADE) VALUES ('Medico9',
'5434534667', '23424567', 'Patologia');

```

-- Populando tabela CONSULTA (10 CONSULTAS)

```

INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('11-Jan-23 10.00', 'consulta0', 1, 1);
INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('15-Feb-23 11.00', 'consulta1', 2, 2);
INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('03-Mar-23 12.00', 'consulta2', 3, 3);
INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('14-Apr-23 13.00', 'consulta3', 4, 4);
INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('02-May-23 14.00', 'consulta4', 5, 5);
INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('25-Jun-24 15.00', 'consulta5', 6, 6);
INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('09-Jul-25 16.00', 'consulta6', 7, 7);
INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('06-Aug-18 17.00', 'consulta7', 8, 8);
INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('30-Sep-25 18.00', 'consulta8', 9, 9);
INSERT INTO CONSULTA (DATA_HORA, DESCRICAO, ID_MEDICO, ID_USUARIO)
VALUES ('17-Oct-23 19.00', 'consulta9', 10, 10);

```

-- Populando tabela RECEITA (10 RECEITAS)

```

INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('11-Jan-23 10.00', 'Repousar uma semana', 1);
INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('15-Feb-23 11.00', 'Tomar 2 litros de coca cola', 2);
INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('03-Mar-23 12.00', 'não usar drogas', 3);
INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('14-Apr-23 13.00', 'andar de bicicleta', 4);

```

```

INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('02-May-23 14.00', 'fumar menos', 5);
INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('25-Jun-24 15.00', 'dipirona todos os dias', 6);
INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('09-Jul-25 16.00', 'rivotril 5ml/dia', 7);
INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('06-Aug-18 17.00', 'nimesulida 1 comp a cada 8 horas', 8);
INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('30-Sep-25 18.00', 'advil a cada 4 horas 1 comp', 9);
INSERT INTO RECEITA (DATA_EMISSAO, PRESCRICAO, ID_CONSULTA) VALUES
('17-Oct-23 19.00', 'dormir', 10);

COMMIT;
END;
/

```

Procedure que procura a disponibilidade médica pela data e especialidade.

```

CREATE OR REPLACE PROCEDURE prBuscaMedicosPorEspecialidadeEData
(dataBusca DATE, especialidadeMedico VARCHAR2)
IS

```

```

    eDataErro                EXCEPTION;
    eCursorErro              EXCEPTION;

    vHorarioDisponivel       DATE;
    vNumeroCrm               varchar(250);
    vMedicoEspecialidade     VARCHAR2(250);
    vNomeMedico              VARCHAR2(250);

    vDataBusca               DATE := trunc(dataBusca);
    vContaLinha              NUMBER := 0;

```

```

CURSOR curDadosConsulta IS

```

```

    SELECT
        m.nome,
        m.CRM,
        c1.horario,
        m.especialidade
    FROM
        medico m
    CROSS JOIN (
        SELECT
            vDataBusca + 8 / 24 + (LEVEL-1)/ 24 AS horario

```

```

        FROM
            dual
        CONNECT BY
            vDataBusca + 8 / 24 + (LEVEL-1)/ 24 < vDataBusca + 18 / 24
        ) c1
        WHERE
            m.especialidade = 'Cardiologia'
            AND NOT EXISTS (
                SELECT 1
                FROM consulta c2
                WHERE c2.medico_id = m.id
                AND c1.horario = c2.DT_CONSULTA
            )
        ORDER BY
            c1.horario;

BEGIN
    IF curDadosConsulta%ISOPEN THEN
        CLOSE curDadosConsulta;
    END IF;
    --
    IF vDataBusca < TRUNC(sysdate) THEN
        RAISE eDataErro;
    END IF;
    --
    OPEN curDadosConsulta;
    LOOP
        FETCH curDadosConsulta INTO vNomeMedico, vNumeroCrm, vHorarioDisponivel,
vMedicoEspecialidade;
        EXIT WHEN curDadosConsulta%NOTFOUND;
        dbms_output.put_line('Médico: ' || vNomeMedico
                                || ' CRM: ' || vNumeroCrm
                                || ' Especialidade: ' || vMedicoEspecialidade
                                || ' Data: ' || TO_CHAR(vHorarioDisponivel,
'dd/mm/yyyy')
                                || ' Horário: ' || TO_CHAR(vHorarioDisponivel,
'hh24:mi'));
        vContaLinha := 1;
    END LOOP;
    CLOSE curDadosConsulta;
    --
    IF vContaLinha = 0 THEN
        RAISE eCursorErro;
    END IF;
    --
    EXCEPTION
    WHEN eDataErro THEN
        dbms_output.put_line('Data de pesquisa não deve ser menor que dia atual!');
    WHEN eCursorErro THEN

```

```
dbms_output.put_line('Sem horários disponíveis!');  
END;
```

Procedure que pega histórico de consultas do paciente no intervalo de tempo determinado.

```
CREATE OR REPLACE PROCEDURE prConsultaHistoricoPaciente  
(pacienteID NUMBER, dataBuscaInicial DATE, dataBuscaFinal DATE)  
IS  
    eDataErro          EXCEPTION;  
    eCursorErro        EXCEPTION;  
  
    vDataConsulta      DATE;  
    vDescricaoReceita  varchar(250);  
    vMedicoEspecialidade varchar(250);  
    vNumeroCrm         varchar(250);  
    vNomeMedico        varchar(200);  
  
    vContaLinha        NUMBER := 0;  
    vdataBuscaInicial  DATE := TRUNC(dataBuscaInicial);  
    vdataBuscaFinal    DATE := TRUNC(dataBuscaFinal);  
  
    CURSOR curDadosConsulta IS  
        SELECT  
            m.NOME,  
            m.CRM,  
            m.ESPECIALIDADE,  
            r.DESCRICAO,  
            c.DT_CONSULTA  
        FROM  
            PACIENTE p  
            INNER JOIN CONSULTA c  
            ON p.ID = c.PACIENTE_ID  
            INNER JOIN MEDICO m  
            ON m.ID = c.MEDICO_ID  
            INNER JOIN RECEITA r  
            ON c.ID = r.CONSULTA_ID  
        WHERE  
            TRUNC(c.DT_CONSULTA) BETWEEN vdataBuscaInicial AND  
vdataBuscaFinal AND p.ID = pacienteID  
        ORDER BY  
            c.DT_CONSULTA ;  
  
BEGIN  
    IF curDadosConsulta%ISOPEN THEN  
        CLOSE curDadosConsulta;  
    END IF;  
    --
```

```

OPEN curDadosConsulta;
LOOP
  FETCH curDadosConsulta INTO vNomeMedico, vNumeroCrm,
vMedicoEspecialidade, vDescricaoReceita, vDataConsulta;
  EXIT WHEN curDadosConsulta%NOTFOUND;
  IF vContaLinha != 0 THEN
    dbms_output.put_line('-----');
  END IF;
  dbms_output.put_line('Médico: ' || vNomeMedico);
  dbms_output.put_line('CRM: ' || vNumeroCrm);
  dbms_output.put_line('Especialidade: ' || vMedicoEspecialidade);
  dbms_output.put_line('Receita: ' || vDescricaoReceita);
  dbms_output.put_line('Data: ' || TO_CHAR(vDataConsulta, 'dd/mm/yyyy'));
  dbms_output.put_line('Horario: ' || TO_CHAR(vDataConsulta, 'hh24:mi'));
  vContaLinha := 1;
END LOOP;
CLOSE curDadosConsulta;
--
IF vContaLinha = 0 THEN
  RAISE eCursorErro;
END IF;
EXCEPTION
WHEN eCursorErro THEN
  dbms_output.put_line('Histórico não encontrado nas datas fornecidas!');
END;

```