FIAP

FONTE VIVA

GLOBAL SOLUTION

Francesco Di Benedetto: RM557313

Nathalia Gomes da Silva: RM554945

Gustavo Goulart Bretas: RM555708

Sumário

[Modelos lógico e relacional 3](#_Toc200213747)

[Create Tables 4](#_Toc200213748)

[Procedures 16](#_Toc200213749)

[Funções 32](#_Toc200213750)

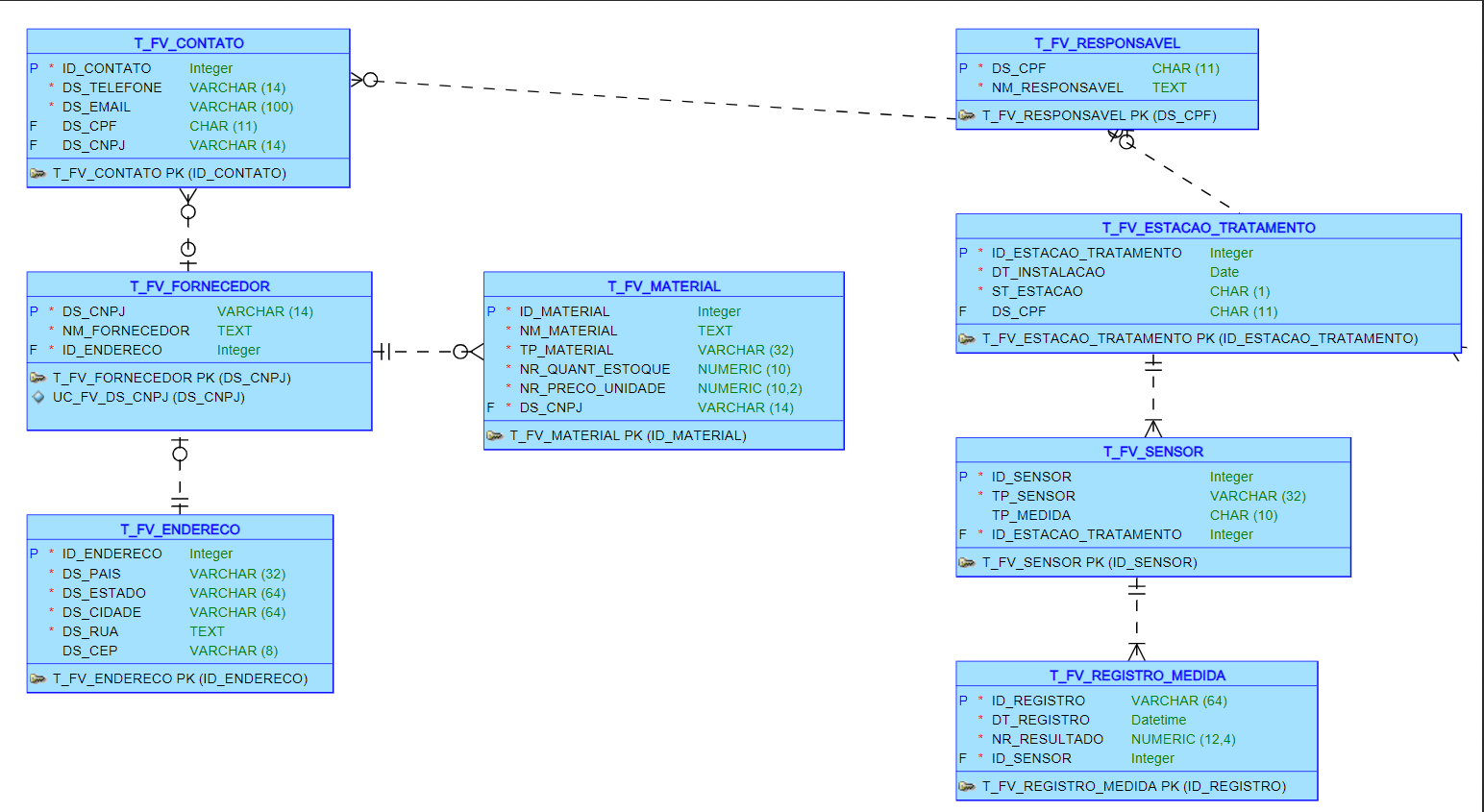
[Bloco anônimo com consultas complexas 34](#_Toc200213751)

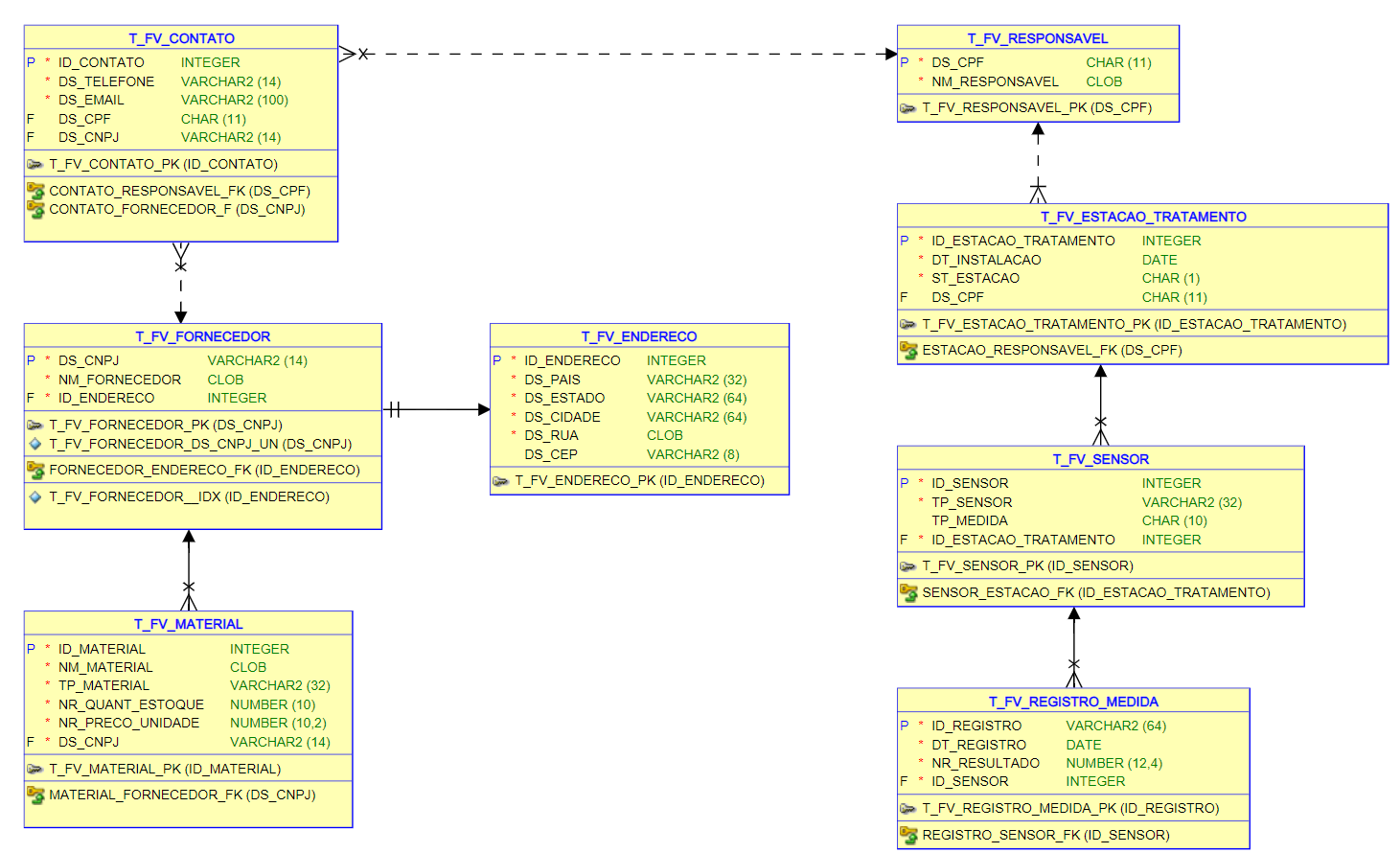
[Cursores Explícitos 40](#_Toc200213752)

[Consulta Complexa 43](#_Toc200213753)

[Integração com outras matérias 47](#_Toc200213754)

## Modelos lógico e relacional





## Create Tables

-- Gerado por Oracle SQL Developer Data Modeler 24.3.1.351.0831

-- em: 2025-05-29 17:41:17 BRT

-- site: Oracle Database 11g

-- tipo: Oracle Database 11g

DROP TABLE T\_FV\_CONTATO CASCADE CONSTRAINTS

;

DROP TABLE T\_FV\_ENDERECO CASCADE CONSTRAINTS

;

DROP TABLE T\_FV\_ESTACAO\_TRATAMENTO CASCADE CONSTRAINTS

;

DROP TABLE T\_FV\_FORNECEDOR CASCADE CONSTRAINTS

;

DROP TABLE T\_FV\_MATERIAL CASCADE CONSTRAINTS

;

DROP TABLE T\_FV\_REGISTRO\_MEDIDA CASCADE CONSTRAINTS

;

DROP TABLE T\_FV\_RESPONSAVEL CASCADE CONSTRAINTS

;

DROP TABLE T\_FV\_SENSOR CASCADE CONSTRAINTS

;

DROP SEQUENCE SEQ\_ID\_SENSOR;

DROP SEQUENCE SEQ\_ID\_ESTACAO;

DROP SEQUENCE SEQ\_ID\_ENDERECO;

DROP SEQUENCE SEQ\_ID\_CONTATO;

DROP SEQUENCE SEQ\_ID\_MATERIAL;

CREATE SEQUENCE SEQ\_ID\_SENSOR START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE SEQ\_ID\_ESTACAO START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE SEQ\_ID\_ENDERECO START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE SEQ\_ID\_CONTATO START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE SEQ\_ID\_MATERIAL START WITH 1 INCREMENT BY 1;

-- predefined type, no DDL - MDSYS.SDO\_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE T\_FV\_CONTATO

(

ID\_CONTATO INTEGER DEFAULT SEQ\_ID\_CONTATO.NEXTVAL ,

DS\_TELEFONE VARCHAR2 (14) NOT NULL ,

DS\_EMAIL VARCHAR2 (100) NOT NULL ,

DS\_CPF CHAR (11) ,

DS\_CNPJ VARCHAR2 (14)

)

;

COMMENT ON COLUMN T\_FV\_CONTATO.ID\_CONTATO IS 'ID DO CONTATO'

;

COMMENT ON COLUMN T\_FV\_CONTATO.DS\_TELEFONE IS 'NUMERO DE TELEFONE PARA CONTATO'

;

COMMENT ON COLUMN T\_FV\_CONTATO.DS\_EMAIL IS 'EMAIL PARA CONTATO'

;

ALTER TABLE T\_FV\_CONTATO

ADD CONSTRAINT T\_FV\_CONTATO\_PK PRIMARY KEY ( ID\_CONTATO ) ;

CREATE TABLE T\_FV\_ENDERECO

(

ID\_ENDERECO INTEGER DEFAULT SEQ\_ID\_ENDERECO.NEXTVAL ,

DS\_PAIS VARCHAR2 (32) NOT NULL ,

DS\_ESTADO VARCHAR2 (64) NOT NULL ,

DS\_CIDADE VARCHAR2 (64) NOT NULL ,

DS\_RUA VARCHAR2 (100) NOT NULL ,

DS\_CEP VARCHAR2 (8)

)

;

COMMENT ON COLUMN T\_FV\_ENDERECO.ID\_ENDERECO IS 'ID DO ENDERECO'

;

COMMENT ON COLUMN T\_FV\_ENDERECO.DS\_PAIS IS 'PAIS ONDE ESTA LOCALIZADO O FORNECEDOR'

;

COMMENT ON COLUMN T\_FV\_ENDERECO.DS\_ESTADO IS 'ESTADO ONDE ESTA LOCALIZADO O FORNECEDOR'

;

COMMENT ON COLUMN T\_FV\_ENDERECO.DS\_CIDADE IS 'CIDADE ONDE ESTA LOCALIZADO O FORNECEDOR'

;

COMMENT ON COLUMN T\_FV\_ENDERECO.DS\_RUA IS 'RUA ONDE ESTA LOCALIZADO O FORNECEDOR'

;

COMMENT ON COLUMN T\_FV\_ENDERECO.DS\_CEP IS 'CEP DO FORNECEDOR'

;

ALTER TABLE T\_FV\_ENDERECO

ADD CONSTRAINT T\_FV\_ENDERECO\_PK PRIMARY KEY ( ID\_ENDERECO ) ;

CREATE TABLE T\_FV\_ESTACAO\_TRATAMENTO

(

ID\_ESTACAO\_TRATAMENTO INTEGER DEFAULT SEQ\_ID\_ESTACAO.NEXTVAL ,

DT\_INSTALACAO DATE NOT NULL ,

ST\_ESTACAO CHAR (1) DEFAULT 'I' NOT NULL ,

DS\_CPF CHAR (11)

)

;

ALTER TABLE T\_FV\_ESTACAO\_TRATAMENTO

ADD CONSTRAINT C\_FV\_ST\_ESTACAO

CHECK (ST\_ESTACAO IN ('A', 'I', 'M'))

;

COMMENT ON COLUMN T\_FV\_ESTACAO\_TRATAMENTO.ID\_ESTACAO\_TRATAMENTO IS 'ID DA ESTAÇÃO DE TRATAMENTO'

;

COMMENT ON COLUMN T\_FV\_ESTACAO\_TRATAMENTO.ST\_ESTACAO IS 'STATUS DAS ESTACAO: ''A'' - ATIVA, ''M'' - MANUTENCAO, ''I'' - INATIVA.'

;

ALTER TABLE T\_FV\_ESTACAO\_TRATAMENTO

ADD CONSTRAINT T\_FV\_ESTACAO\_TRATAMENTO\_PK PRIMARY KEY ( ID\_ESTACAO\_TRATAMENTO ) ;

CREATE TABLE T\_FV\_FORNECEDOR

(

DS\_CNPJ VARCHAR2 (14) NOT NULL ,

NM\_FORNECEDOR VARCHAR2 (100) NOT NULL ,

ID\_ENDERECO INTEGER NOT NULL

)

;

CREATE UNIQUE INDEX T\_FV\_FORNECEDOR\_\_IDX ON T\_FV\_FORNECEDOR

(

ID\_ENDERECO ASC

)

;

ALTER TABLE T\_FV\_FORNECEDOR

ADD CONSTRAINT T\_FV\_FORNECEDOR\_PK PRIMARY KEY ( DS\_CNPJ ) ;

CREATE TABLE T\_FV\_MATERIAL

(

ID\_MATERIAL INTEGER DEFAULT SEQ\_ID\_MATERIAL.NEXTVAL ,

NM\_MATERIAL VARCHAR2 (100) NOT NULL ,

TP\_MATERIAL VARCHAR2 (32) NOT NULL ,

NR\_QUANT\_ESTOQUE NUMBER (10) NOT NULL ,

NR\_PRECO\_UNIDADE NUMBER (10,2) NOT NULL ,

DS\_CNPJ VARCHAR2 (14) NOT NULL

)

;

ALTER TABLE T\_FV\_MATERIAL

ADD CONSTRAINT C\_FV\_TP\_MATERIAL

CHECK (TP\_MATERIAL IN ('ELETRONICO', 'FILTRO', 'RESERVATORIO', 'SENSOR'))

;

COMMENT ON COLUMN T\_FV\_MATERIAL.ID\_MATERIAL IS 'ID DO MATERIAL'

;

COMMENT ON COLUMN T\_FV\_MATERIAL.NM\_MATERIAL IS 'NOME COMPLETO DO MATERIAL'

;

COMMENT ON COLUMN T\_FV\_MATERIAL.TP\_MATERIAL IS 'TIPO GENERICO DO MATERIAL'

;

COMMENT ON COLUMN T\_FV\_MATERIAL.NR\_QUANT\_ESTOQUE IS 'QUANTIDADE EM ESTOQUE'

;

COMMENT ON COLUMN T\_FV\_MATERIAL.NR\_PRECO\_UNIDADE IS 'PRECO DOS MATERIAIS'

;

ALTER TABLE T\_FV\_MATERIAL

ADD CONSTRAINT T\_FV\_MATERIAL\_PK PRIMARY KEY ( ID\_MATERIAL ) ;

CREATE TABLE T\_FV\_REGISTRO\_MEDIDA

(

ID\_REGISTRO VARCHAR2 (64) NOT NULL ,

DT\_REGISTRO DATE NOT NULL ,

NR\_RESULTADO NUMBER (12,4) NOT NULL ,

ID\_SENSOR INTEGER NOT NULL

)

;

COMMENT ON COLUMN T\_FV\_REGISTRO\_MEDIDA.ID\_REGISTRO IS 'IDENTIFICADOR DO REGISTRO'

;

ALTER TABLE T\_FV\_REGISTRO\_MEDIDA

ADD CONSTRAINT T\_FV\_REGISTRO\_MEDIDA\_PK PRIMARY KEY ( ID\_REGISTRO ) ;

CREATE TABLE T\_FV\_RESPONSAVEL

(

DS\_CPF CHAR (11) NOT NULL ,

NM\_RESPONSAVEL VARCHAR2 (100) NOT NULL

)

;

COMMENT ON COLUMN T\_FV\_RESPONSAVEL.DS\_CPF IS 'CPF DO RESPONSAVEL'

;

COMMENT ON COLUMN T\_FV\_RESPONSAVEL.NM\_RESPONSAVEL IS 'NOME DO RESPONSAVEL'

;

ALTER TABLE T\_FV\_RESPONSAVEL

ADD CONSTRAINT T\_FV\_RESPONSAVEL\_PK PRIMARY KEY ( DS\_CPF ) ;

CREATE TABLE T\_FV\_SENSOR

(

ID\_SENSOR INTEGER DEFAULT SEQ\_ID\_SENSOR.NEXTVAL,

TP\_SENSOR VARCHAR2 (32) NOT NULL ,

TP\_MEDIDA CHAR (10) ,

ID\_ESTACAO\_TRATAMENTO INTEGER NOT NULL

)

;

COMMENT ON COLUMN T\_FV\_SENSOR.ID\_SENSOR IS 'ID DO SENSOR'

;

COMMENT ON COLUMN T\_FV\_SENSOR.TP\_SENSOR IS 'TIPO DE SENSOR'

;

COMMENT ON COLUMN T\_FV\_SENSOR.TP\_MEDIDA IS 'TIPO DA MEDIDA DO SENSOR: PH, TEMPERATURA, NTU'

;

ALTER TABLE T\_FV\_SENSOR

ADD CONSTRAINT T\_FV\_SENSOR\_PK PRIMARY KEY ( ID\_SENSOR ) ;

ALTER TABLE T\_FV\_CONTATO

ADD CONSTRAINT CONTATO\_FORNECEDOR\_F FOREIGN KEY

(

DS\_CNPJ

)

REFERENCES T\_FV\_FORNECEDOR

(

DS\_CNPJ

)

ON DELETE CASCADE

;

ALTER TABLE T\_FV\_CONTATO

ADD CONSTRAINT CONTATO\_RESPONSAVEL\_FK FOREIGN KEY

(

DS\_CPF

)

REFERENCES T\_FV\_RESPONSAVEL

(

DS\_CPF

)

ON DELETE CASCADE

;

ALTER TABLE T\_FV\_ESTACAO\_TRATAMENTO

ADD CONSTRAINT ESTACAO\_RESPONSAVEL\_FK FOREIGN KEY

(

DS\_CPF

)

REFERENCES T\_FV\_RESPONSAVEL

(

DS\_CPF

)

;

ALTER TABLE T\_FV\_FORNECEDOR

ADD CONSTRAINT FORNECEDOR\_ENDERECO\_FK FOREIGN KEY

(

ID\_ENDERECO

)

REFERENCES T\_FV\_ENDERECO

(

ID\_ENDERECO

)

;

ALTER TABLE T\_FV\_MATERIAL

ADD CONSTRAINT MATERIAL\_FORNECEDOR\_FK FOREIGN KEY

(

DS\_CNPJ

)

REFERENCES T\_FV\_FORNECEDOR

(

DS\_CNPJ

)

ON DELETE CASCADE

;

ALTER TABLE T\_FV\_REGISTRO\_MEDIDA

ADD CONSTRAINT REGISTRO\_SENSOR\_FK FOREIGN KEY

(

ID\_SENSOR

)

REFERENCES T\_FV\_SENSOR

(

ID\_SENSOR

)

ON DELETE CASCADE

;

ALTER TABLE T\_FV\_SENSOR

ADD CONSTRAINT SENSOR\_ESTACAO\_FK FOREIGN KEY

(

ID\_ESTACAO\_TRATAMENTO

)

REFERENCES T\_FV\_ESTACAO\_TRATAMENTO

(

ID\_ESTACAO\_TRATAMENTO

)

ON DELETE CASCADE

;

## Procedures

-- ENDERECO

CREATE OR REPLACE PROCEDURE SP\_INSERE\_ENDERECO (

V\_PAIS IN VARCHAR2,

V\_ESTADO IN VARCHAR2,

V\_CIDADE IN VARCHAR2,

V\_RUA IN VARCHAR2

) AS

BEGIN

INSERT INTO T\_FV\_ENDERECO (

ID\_ENDERECO, DS\_PAIS, DS\_ESTADO, DS\_CIDADE, DS\_RUA

) VALUES (

SEQ\_ID\_ENDERECO.NEXTVAL, V\_PAIS, V\_ESTADO, V\_CIDADE, V\_RUA

);

END;

/

CREATE OR REPLACE PROCEDURE SP\_ATUALIZA\_ENDERECO (

V\_ID\_ENDERECO IN NUMBER,

V\_RUA IN VARCHAR2

) AS

BEGIN

UPDATE T\_FV\_ENDERECO

SET DS\_RUA = V\_RUA

WHERE ID\_ENDERECO = V\_ID\_ENDERECO;

END;

/

CREATE OR REPLACE PROCEDURE SP\_EXCLUI\_ENDERECO (

V\_ID\_ENDERECO IN NUMBER

) AS

BEGIN

DELETE FROM T\_FV\_ENDERECO

WHERE ID\_ENDERECO = V\_ID\_ENDERECO;

END;

/

-- FORNECEDOR

CREATE OR REPLACE PROCEDURE SP\_INSERE\_FORNECEDOR (

V\_CNPJ IN VARCHAR2,

V\_NOME IN VARCHAR2,

V\_ID\_ENDERECO IN NUMBER

) AS

BEGIN

INSERT INTO T\_FV\_FORNECEDOR (

DS\_CNPJ, NM\_FORNECEDOR, ID\_ENDERECO

) VALUES (

V\_CNPJ, V\_NOME, V\_ID\_ENDERECO

);

END;

/

CREATE OR REPLACE PROCEDURE SP\_ATUALIZA\_FORNECEDOR (

V\_CNPJ IN VARCHAR2,

V\_NOME IN VARCHAR2,

V\_ID\_ENDERECO IN NUMBER

) AS

BEGIN

UPDATE T\_FV\_FORNECEDOR

SET NM\_FORNECEDOR = V\_NOME,

ID\_ENDERECO = V\_ID\_ENDERECO

WHERE DS\_CNPJ = V\_CNPJ;

END;

/

CREATE OR REPLACE PROCEDURE SP\_EXCLUI\_FORNECEDOR (

V\_CNPJ IN VARCHAR2

) AS

BEGIN

DELETE FROM T\_FV\_FORNECEDOR

WHERE DS\_CNPJ = V\_CNPJ;

END;

/

-- CONTATO

CREATE OR REPLACE PROCEDURE SP\_INSERE\_CONTATO (

V\_TELEFONE IN VARCHAR2,

V\_EMAIL IN VARCHAR2,

V\_DOC IN VARCHAR2 DEFAULT NULL

) AS

BEGIN

IF LENGTH(TRIM(V\_DOC)) = 11 THEN

INSERT INTO T\_FV\_CONTATO (

ID\_CONTATO, DS\_TELEFONE, DS\_EMAIL, DS\_CPF

) VALUES (

SEQ\_ID\_CONTATO.NEXTVAL, V\_TELEFONE, V\_EMAIL, V\_DOC

);

ELSIF LENGTH(TRIM(V\_DOC)) = 14 THEN

INSERT INTO T\_FV\_CONTATO (

ID\_CONTATO, DS\_TELEFONE, DS\_EMAIL, DS\_CNPJ

) VALUES (

SEQ\_ID\_CONTATO.NEXTVAL, V\_TELEFONE, V\_EMAIL, V\_DOC

);

ELSE

INSERT INTO T\_FV\_CONTATO (

ID\_CONTATO, DS\_TELEFONE, DS\_EMAIL

) VALUES (

SEQ\_ID\_CONTATO.NEXTVAL, V\_TELEFONE, V\_EMAIL

);

END IF;

EXCEPTION

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Erro ao inserir contato: ' || SQLERRM);

END;

/

CREATE OR REPLACE PROCEDURE SP\_ATUALIZA\_CONTATO (

V\_ID\_CONTATO IN NUMBER,

V\_TELEFONE IN VARCHAR2,

V\_EMAIL IN VARCHAR2

) AS

BEGIN

UPDATE T\_FV\_CONTATO

SET DS\_TELEFONE = V\_TELEFONE,

DS\_EMAIL = V\_EMAIL

WHERE ID\_CONTATO = V\_ID\_CONTATO;

END;

/

CREATE OR REPLACE PROCEDURE SP\_EXCLUI\_CONTATO (

V\_ID\_CONTATO IN NUMBER

) AS

BEGIN

DELETE FROM T\_FV\_CONTATO

WHERE ID\_CONTATO = V\_ID\_CONTATO;

END;

/

-- MATERIAL

CREATE OR REPLACE PROCEDURE SP\_INSERE\_MATERIAL (

V\_NOME IN T\_FV\_MATERIAL.NM\_MATERIAL%TYPE,

V\_TIPO IN T\_FV\_MATERIAL.TP\_MATERIAL%TYPE,

V\_ESTOQUE IN T\_FV\_MATERIAL.NR\_QUANT\_ESTOQUE%TYPE,

V\_PRECO IN T\_FV\_MATERIAL.NR\_PRECO\_UNIDADE%TYPE,

V\_CNPJ IN VARCHAR2

)

AS

BEGIN

INSERT INTO T\_FV\_MATERIAL (

ID\_MATERIAL,

NM\_MATERIAL,

TP\_MATERIAL,

NR\_QUANT\_ESTOQUE,

NR\_PRECO\_UNIDADE,

DS\_CNPJ

) VALUES (

SEQ\_ID\_MATERIAL.NEXTVAL,

V\_NOME,

V\_TIPO,

V\_ESTOQUE,

V\_PRECO,

V\_CNPJ

);

END;

/

CREATE OR REPLACE PROCEDURE SP\_ATUALIZA\_MATERIAL (

P\_ID\_MATERIAL IN T\_FV\_MATERIAL.ID\_MATERIAL%TYPE,

P\_NM\_MATERIAL IN T\_FV\_MATERIAL.NM\_MATERIAL%TYPE,

P\_TP\_MATERIAL IN T\_FV\_MATERIAL.TP\_MATERIAL%TYPE,

P\_NR\_QUANT\_ESTOQUE IN T\_FV\_MATERIAL.NR\_QUANT\_ESTOQUE%TYPE,

P\_NR\_PRECO\_UNIDADE IN T\_FV\_MATERIAL.NR\_PRECO\_UNIDADE%TYPE

) AS

BEGIN

UPDATE T\_FV\_MATERIAL

SET NM\_MATERIAL = P\_NM\_MATERIAL,

TP\_MATERIAL = P\_TP\_MATERIAL,

NR\_QUANT\_ESTOQUE = P\_NR\_QUANT\_ESTOQUE,

NR\_PRECO\_UNIDADE = P\_NR\_PRECO\_UNIDADE

WHERE ID\_MATERIAL = P\_ID\_MATERIAL;

END;

/

CREATE OR REPLACE PROCEDURE SP\_DELETA\_MATERIAL (

V\_ID\_MATERIAL IN NUMBER

) AS

BEGIN

DELETE FROM T\_FV\_MATERIAL

WHERE ID\_MATERIAL = V\_ID\_MATERIAL;

END;

/

-- RESPONSAVEL

CREATE OR REPLACE PROCEDURE SP\_INSERE\_RESPONSAVEL (

V\_CPF IN T\_FV\_RESPONSAVEL.DS\_CPF%TYPE,

V\_NOME IN T\_FV\_RESPONSAVEL.NM\_RESPONSAVEL%TYPE

) AS

BEGIN

INSERT INTO T\_FV\_RESPONSAVEL (DS\_CPF, NM\_RESPONSAVEL)

VALUES (V\_CPF, V\_NOME);

END;

/

CREATE OR REPLACE PROCEDURE SP\_ATUALIZA\_RESPONSAVEL (

V\_CPF IN T\_FV\_RESPONSAVEL.DS\_CPF%TYPE,

V\_NOME IN T\_FV\_RESPONSAVEL.NM\_RESPONSAVEL%TYPE

) AS

BEGIN

UPDATE T\_FV\_RESPONSAVEL

SET NM\_RESPONSAVEL = V\_NOME

WHERE DS\_CPF = V\_CPF;

END;

/

CREATE OR REPLACE PROCEDURE SP\_DELETA\_RESPONSAVEL (

V\_CPF IN VARCHAR2

) AS

BEGIN

DELETE FROM T\_FV\_RESPONSAVEL

WHERE DS\_CPF = V\_CPF;

END;

/

-- ESTACAO DE TRATAMENTO

CREATE OR REPLACE PROCEDURE SP\_INSERE\_ESTACAO (

V\_DT\_INSTALACAO IN DATE,

V\_STATUS IN CHAR,

V\_CPF IN VARCHAR2 DEFAULT NULL

) AS

BEGIN

IF V\_CPF IS NULL THEN

INSERT INTO T\_FV\_ESTACAO\_TRATAMENTO (

ID\_ESTACAO\_TRATAMENTO,

DT\_INSTALACAO,

ST\_ESTACAO

)

VALUES (

SEQ\_ID\_ESTACAO.NEXTVAL,

V\_DT\_INSTALACAO,

UPPER(V\_STATUS)

);

ELSE

INSERT INTO T\_FV\_ESTACAO\_TRATAMENTO (

ID\_ESTACAO\_TRATAMENTO,

DT\_INSTALACAO,

ST\_ESTACAO,

DS\_CPF

)

VALUES (

SEQ\_ID\_ESTACAO.NEXTVAL,

V\_DT\_INSTALACAO,

UPPER(V\_STATUS),

V\_CPF

);

END IF;

END;

/

CREATE OR REPLACE PROCEDURE SP\_ATUALIZA\_ESTACAO (

V\_ID\_ESTACAO IN NUMBER,

V\_DT\_INSTALACAO IN DATE,

V\_STATUS IN CHAR,

V\_CPF IN VARCHAR2 DEFAULT NULL

) AS

BEGIN

IF V\_CPF IS NULL THEN

UPDATE T\_FV\_ESTACAO\_TRATAMENTO

SET DT\_INSTALACAO = V\_DT\_INSTALACAO,

ST\_ESTACAO = UPPER(V\_STATUS)

WHERE ID\_ESTACAO\_TRATAMENTO = V\_ID\_ESTACAO;

ELSE

UPDATE T\_FV\_ESTACAO\_TRATAMENTO

SET DT\_INSTALACAO = V\_DT\_INSTALACAO,

ST\_ESTACAO = UPPER(V\_STATUS),

DS\_CPF = V\_CPF

WHERE ID\_ESTACAO\_TRATAMENTO = V\_ID\_ESTACAO;

END IF;

END;

/

CREATE OR REPLACE PROCEDURE SP\_DELETA\_ESTACAO (

V\_ID\_ESTACAO IN NUMBER

) AS

BEGIN

DELETE FROM T\_FV\_ESTACAO\_TRATAMENTO

WHERE ID\_ESTACAO\_TRATAMENTO = V\_ID\_ESTACAO;

END;

/

-- SENSOR

CREATE OR REPLACE PROCEDURE SP\_INSERE\_SENSOR (

V\_TP\_SENSOR IN VARCHAR2,

V\_TP\_MEDIDA IN VARCHAR2,

V\_ID\_ESTACAO IN NUMBER

) AS

BEGIN

INSERT INTO T\_FV\_SENSOR (

ID\_SENSOR,

TP\_SENSOR,

TP\_MEDIDA,

ID\_ESTACAO\_TRATAMENTO

)

VALUES (

SEQ\_ID\_SENSOR.NEXTVAL,

V\_TP\_SENSOR,

V\_TP\_MEDIDA,

V\_ID\_ESTACAO

);

END;

/

CREATE OR REPLACE PROCEDURE SP\_ATUALIZA\_SENSOR (

V\_ID\_SENSOR IN NUMBER,

V\_TP\_SENSOR IN VARCHAR2,

V\_TP\_MEDIDA IN VARCHAR2,

V\_ID\_ESTACAO IN NUMBER

) AS

BEGIN

UPDATE T\_FV\_SENSOR

SET TP\_SENSOR = V\_TP\_SENSOR,

TP\_MEDIDA = V\_TP\_MEDIDA,

ID\_ESTACAO\_TRATAMENTO = V\_ID\_ESTACAO

WHERE ID\_SENSOR = V\_ID\_SENSOR;

END;

/

CREATE OR REPLACE PROCEDURE SP\_DELETA\_SENSOR (

V\_ID\_SENSOR IN NUMBER

) AS

BEGIN

DELETE FROM T\_FV\_SENSOR

WHERE ID\_SENSOR = V\_ID\_SENSOR;

END;

/

-- REGISTRO MEDIDA

CREATE OR REPLACE PROCEDURE SP\_INSERE\_REGISTRO (

V\_ID\_REGISTRO IN VARCHAR2,

V\_VALOR IN NUMBER,

V\_DT\_REGISTRO IN DATE,

V\_ID\_SENSOR IN NUMBER

) AS

BEGIN

INSERT INTO T\_FV\_REGISTRO\_MEDIDA (

ID\_REGISTRO,

NR\_RESULTADO,

DT\_REGISTRO,

ID\_SENSOR

)

VALUES (

v\_ID\_REGISTRO,

V\_VALOR,

V\_DT\_REGISTRO,

V\_ID\_SENSOR

);

END;

/

CREATE OR REPLACE PROCEDURE SP\_DELETA\_REGISTRO (

V\_ID\_REGISTRO IN VARCHAR2

) AS

BEGIN

DELETE FROM T\_FV\_REGISTRO\_MEDIDA

WHERE ID\_REGISTRO = V\_ID\_REGISTRO;

END;

/

DECLARE

V\_ID\_ENDERECO INTEGER;

BEGIN

SP\_INSERE\_ENDERECO('BRASIL', 'SP', 'São Paulo', 'Rua das Águas');

SELECT MAX(ID\_ENDERECO) INTO V\_ID\_ENDERECO FROM T\_FV\_ENDERECO;

SP\_INSERE\_FORNECEDOR('15432919000101', 'HidroTec Ambiental LTDA', V\_ID\_ENDERECO);

SP\_INSERE\_ENDERECO('BRASIL', 'RJ', 'Niterói', 'Avenida Solar 123');

SELECT MAX(ID\_ENDERECO) INTO V\_ID\_ENDERECO FROM T\_FV\_ENDERECO;

SP\_INSERE\_FORNECEDOR('25432919000102', 'SolarFix Energia Renovável', V\_ID\_ENDERECO);

SP\_INSERE\_ENDERECO('BRASIL', 'MG', 'Belo Horizonte', 'Travessa Hidro 56');

SELECT MAX(ID\_ENDERECO) INTO V\_ID\_ENDERECO FROM T\_FV\_ENDERECO;

SP\_INSERE\_FORNECEDOR('35432919000103', 'BioÁgua Soluções Sustentáveis', V\_ID\_ENDERECO);

SP\_INSERE\_ENDERECO('BRASIL', 'BA', 'Salvador', 'Rua do Reservatório');

SELECT MAX(ID\_ENDERECO) INTO V\_ID\_ENDERECO FROM T\_FV\_ENDERECO;

SP\_INSERE\_FORNECEDOR('45432919000104', 'ÁguaPura Equipamentos', V\_ID\_ENDERECO);

SP\_INSERE\_ENDERECO('BRASIL', 'AM', 'Manaus', 'Estrada do Filtro 88');

SELECT MAX(ID\_ENDERECO) INTO V\_ID\_ENDERECO FROM T\_FV\_ENDERECO;

SP\_INSERE\_FORNECEDOR('55432919000105', 'EcoReservas Brasil', V\_ID\_ENDERECO);

END;

/

BEGIN

SP\_INSERE\_CONTATO('11 9999-1234', 'hidrotec@contato.com', '15432919000101');

SP\_INSERE\_CONTATO('21 3333-5678', 'solarfix@energia.com', '25432919000102');

SP\_INSERE\_CONTATO('31 8888-2222', 'bioagua@solucoes.com', '35432919000103');

SP\_INSERE\_CONTATO('92 7777-4444', 'aguapura@equipamentos.com', '45432919000104');

SP\_INSERE\_CONTATO('85 6666-0000', 'ecoreservas@brasil.com', '55432919000105');

END;

/

BEGIN

SP\_INSERE\_MATERIAL('Filtro de Areia Camada Grossa', 'FILTRO', 50, 75, '15432919000101');

SP\_INSERE\_MATERIAL('Painel Solar 10W', 'ELETRONICO', 30, 140, '25432919000102');

SP\_INSERE\_MATERIAL('Caixa de Água 60L', 'RESERVATORIO', 20, 110, '35432919000103');

SP\_INSERE\_MATERIAL('Filtro UV Padrão', 'FILTRO', 25, 25, '45432919000104');

SP\_INSERE\_MATERIAL('Tanque Modular 100L', 'RESERVATORIO', 15, 130, '55432919000105');

END;

/

BEGIN

SP\_INSERE\_RESPONSAVEL('98765432100', 'Mariana Silva');

SP\_INSERE\_RESPONSAVEL('65432198700', 'Carlos Souza');

SP\_INSERE\_RESPONSAVEL('32165498700', 'Fernanda Costa');

SP\_INSERE\_RESPONSAVEL('78915432600', 'Ricardo Lima');

SP\_INSERE\_RESPONSAVEL('15432919900', 'Gabriela Oliveira');

SP\_INSERE\_CONTATO('85 9000-0001', 'gabriela@estacao.com', '15432919900');

END;

/

BEGIN

SP\_INSERE\_ESTACAO(TO\_DATE('2025-05-01', 'YYYY-MM-DD'), 'i');

SP\_INSERE\_ESTACAO(TO\_DATE('2025-05-02', 'YYYY-MM-DD'), 'i');

SP\_INSERE\_ESTACAO(TO\_DATE('2025-05-03', 'YYYY-MM-DD'), 'i');

SP\_INSERE\_ESTACAO(TO\_DATE('2025-05-04', 'YYYY-MM-DD'), 'i', '15432919900');

SP\_INSERE\_ESTACAO(TO\_DATE('2025-06-01', 'YYYY-MM-DD'), 'A', '15432919900');

END;

/

DECLARE

V\_ID\_ESTACAO INTEGER;

BEGIN

SELECT MAX(ID\_ESTACAO\_TRATAMENTO) INTO V\_ID\_ESTACAO FROM T\_FV\_ESTACAO\_TRATAMENTO;

SP\_INSERE\_SENSOR('PH', 'PH', V\_ID\_ESTACAO);

SP\_INSERE\_SENSOR('TURBIDEZ', 'NTU', V\_ID\_ESTACAO);

SP\_INSERE\_SENSOR('TEMPERATURA', 'CELSIUS', V\_ID\_ESTACAO);

SP\_INSERE\_SENSOR('NIVEL', 'CM', V\_ID\_ESTACAO);

SP\_INSERE\_SENSOR('VAZAO', 'LPM', V\_ID\_ESTACAO);

END;

/

## Funções

CREATE OR REPLACE FUNCTION FUNC\_MEDIA\_VALOR\_POR\_SENSOR (

V\_TIPO\_SENSOR IN VARCHAR2

) RETURN NUMBER IS

V\_MEDIA NUMBER;

BEGIN

SELECT AVG(NR\_RESULTADO)

INTO V\_MEDIA

FROM T\_FV\_REGISTRO\_MEDIDA R

JOIN T\_FV\_SENSOR S ON R.ID\_SENSOR = S.ID\_SENSOR

WHERE S.TP\_SENSOR = V\_TIPO\_SENSOR;

RETURN ROUND(V\_MEDIA, 4);

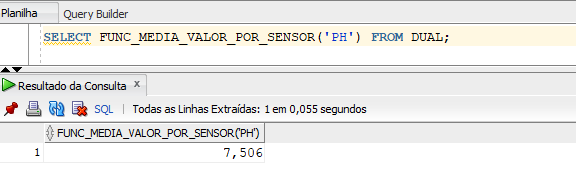
EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN NULL;

END;

/



CREATE OR REPLACE FUNCTION FUNC\_TOTAL\_POR\_FORNECEDOR(

V\_CNPJ IN CHAR

) RETURN NUMBER IS

V\_TOTAL NUMBER := 0;

BEGIN

SELECT SUM(NR\_QUANT\_ESTOQUE \* NR\_PRECO\_UNIDADE) INTO V\_TOTAL

FROM T\_FV\_MATERIAL

WHERE DS\_CNPJ = V\_CNPJ;

RETURN ROUND(V\_TOTAL, 2);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

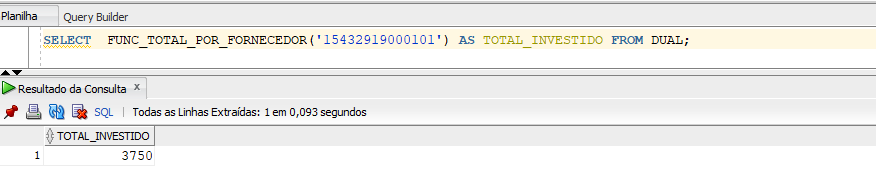
RETURN NULL;

WHEN OTHERS THEN

RETURN NULL;

END;

/



## Bloco anônimo com consultas complexas

-- monitoramento de pH por responsável

DECLARE

CURSOR c\_ph IS

SELECT

r.NR\_RESULTADO AS VALOR\_PH,

rs.NM\_RESPONSAVEL,

r.DT\_REGISTRO

FROM T\_FV\_REGISTRO\_MEDIDA r

JOIN T\_FV\_SENSOR s ON s.ID\_SENSOR = r.ID\_SENSOR

JOIN T\_FV\_ESTACAO\_TRATAMENTO e ON e.ID\_ESTACAO\_TRATAMENTO = s.ID\_ESTACAO\_TRATAMENTO

JOIN T\_FV\_RESPONSAVEL rs ON rs.DS\_CPF = e.DS\_CPF

WHERE s.TP\_SENSOR = 'PH'

ORDER BY r.DT\_REGISTRO;

v\_valor NUMBER;

v\_nome T\_FV\_RESPONSAVEL.NM\_RESPONSAVEL%TYPE;

v\_data DATE;

v\_classificacao VARCHAR2(10);

BEGIN

OPEN c\_ph;

LOOP

FETCH c\_ph INTO v\_valor, v\_nome, v\_data;

EXIT WHEN c\_ph%NOTFOUND;

IF v\_valor > 7.5 THEN

v\_classificacao := 'BASICO';

ELSIF v\_valor < 6.5 THEN

v\_classificacao := 'ACIDO';

ELSE

v\_classificacao := 'NEUTRO';

END IF;

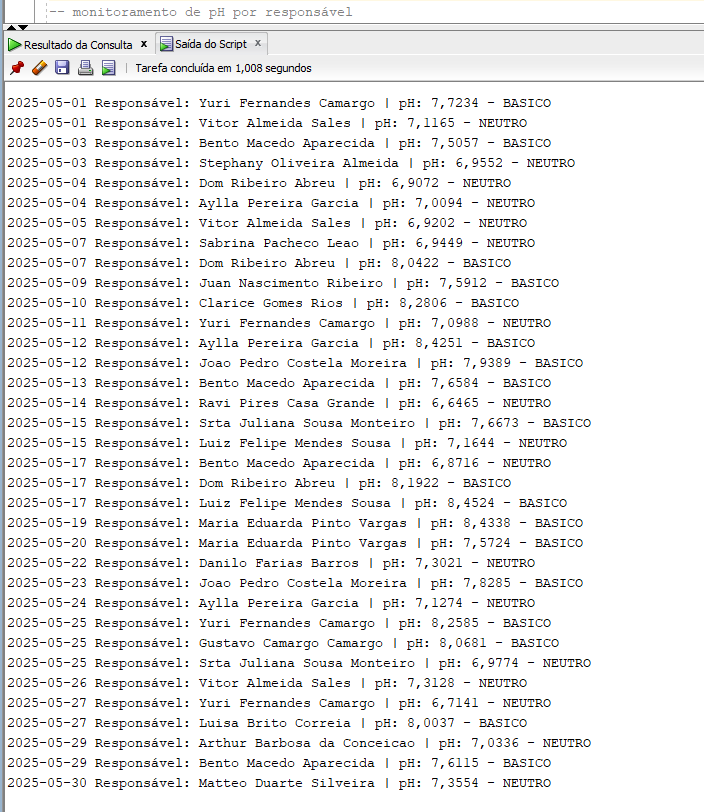
DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(v\_data, 'YYYY-MM-DD') || ' Responsável: ' || v\_nome || ' | pH: ' || v\_valor || ' - ' || v\_classificacao);

END LOOP;

CLOSE c\_ph;

END;

/



-- Monitoramento geral de sensores por estação

DECLARE

CURSOR c\_estacoes IS

SELECT e.ID\_ESTACAO\_TRATAMENTO, COUNT(s.ID\_SENSOR) AS QTD\_SENSORES

FROM T\_FV\_ESTACAO\_TRATAMENTO e

LEFT JOIN T\_FV\_SENSOR s ON s.ID\_ESTACAO\_TRATAMENTO = e.ID\_ESTACAO\_TRATAMENTO

WHERE e.ST\_ESTACAO = 'A'

GROUP BY e.ID\_ESTACAO\_TRATAMENTO;

v\_id\_estacao T\_FV\_ESTACAO\_TRATAMENTO.ID\_ESTACAO\_TRATAMENTO%TYPE;

v\_qtd\_sensores NUMBER;

v\_status VARCHAR2(15);

BEGIN

OPEN c\_estacoes;

LOOP

FETCH c\_estacoes INTO v\_id\_estacao, v\_qtd\_sensores;

EXIT WHEN c\_estacoes%NOTFOUND;

IF v\_qtd\_sensores = 5 THEN

v\_status := 'COMPLETA';

ELSIF v\_qtd\_sensores > 5 THEN

v\_status := 'EXCESSO';

ELSE

v\_status := 'INCOMPLETA';

END IF;

DBMS\_OUTPUT.PUT\_LINE(

'Estação: ' || v\_id\_estacao ||

' | Sensores: ' || v\_qtd\_sensores ||

' | Status: ' || v\_status

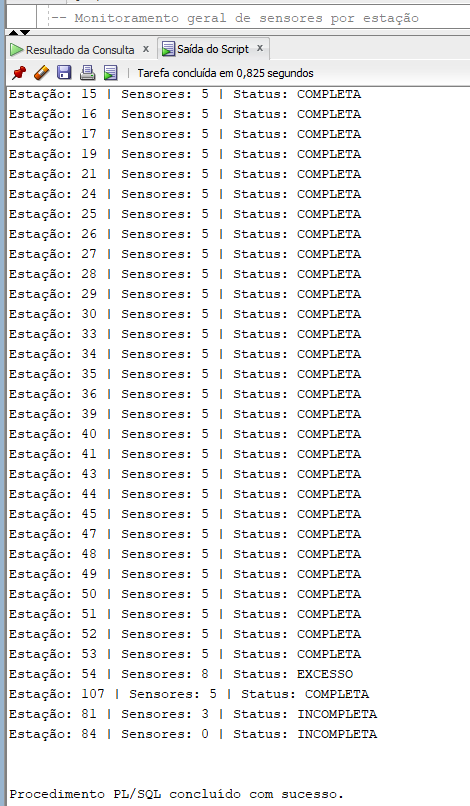
);

END LOOP;

CLOSE c\_estacoes;

END;

/



## Cursores Explícitos

-- Estoque médio por tipo de material

DECLARE

CURSOR c\_materiais IS

SELECT TP\_MATERIAL, ROUND(AVG(NR\_QUANT\_ESTOQUE)) AS MEDIA\_ESTOQUE

FROM T\_FV\_MATERIAL

GROUP BY TP\_MATERIAL;

v\_tipo T\_FV\_MATERIAL.TP\_MATERIAL%TYPE;

v\_media NUMBER;

BEGIN

OPEN c\_materiais;

LOOP

FETCH c\_materiais INTO v\_tipo, v\_media;

EXIT WHEN c\_materiais%NOTFOUND;

IF v\_media < 20 THEN

DBMS\_OUTPUT.PUT\_LINE('Tipo: ' || v\_tipo || ' | MÉDIA BAIXA: ' || ROUND(v\_media, 2));

ELSE

DBMS\_OUTPUT.PUT\_LINE('Tipo: ' || v\_tipo || ' | Média ok: ' || ROUND(v\_media, 2));

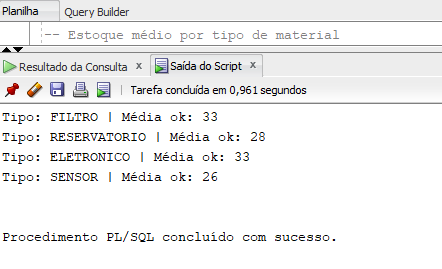
END IF;

END LOOP;

CLOSE c\_materiais;

END;

/



-- Última leitura por tipo de sensor

DECLARE

CURSOR c\_tipos IS

SELECT DISTINCT TP\_SENSOR FROM T\_FV\_SENSOR;

v\_tipo\_sensor T\_FV\_SENSOR.TP\_SENSOR%TYPE;

v\_id\_sensor T\_FV\_SENSOR.ID\_SENSOR%TYPE;

v\_valor T\_FV\_REGISTRO\_MEDIDA.NR\_RESULTADO%TYPE;

v\_data T\_FV\_REGISTRO\_MEDIDA.DT\_REGISTRO%TYPE;

BEGIN

OPEN c\_tipos;

LOOP

FETCH c\_tipos INTO v\_tipo\_sensor;

EXIT WHEN c\_tipos%NOTFOUND;

SELECT r.ID\_SENSOR, r.NR\_RESULTADO, r.DT\_REGISTRO

INTO v\_id\_sensor, v\_valor, v\_data

FROM T\_FV\_REGISTRO\_MEDIDA r

JOIN T\_FV\_SENSOR s ON s.ID\_SENSOR = r.ID\_SENSOR

WHERE s.TP\_SENSOR = v\_tipo\_sensor

AND r.DT\_REGISTRO = (

SELECT MAX(DT\_REGISTRO)

FROM T\_FV\_REGISTRO\_MEDIDA r2

JOIN T\_FV\_SENSOR s2 ON s2.ID\_SENSOR = r2.ID\_SENSOR

WHERE s2.TP\_SENSOR = v\_tipo\_sensor

)

FETCH FIRST 1 ROWS ONLY;

DBMS\_OUTPUT.PUT\_LINE(

'Sensor: ' || v\_tipo\_sensor ||

' | Última leitura: ' || v\_valor ||

' em ' || TO\_CHAR(v\_data, 'YYYY-MM-DD') ||

' (ID Sensor: ' || v\_id\_sensor || ')'

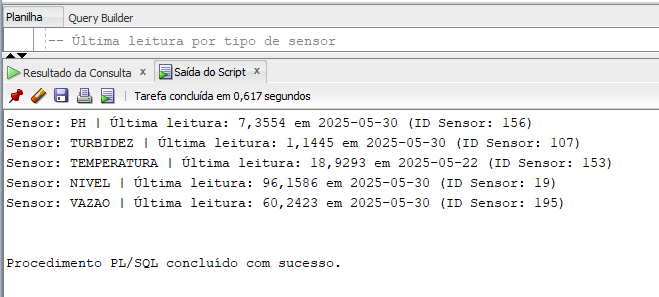
);

END LOOP;

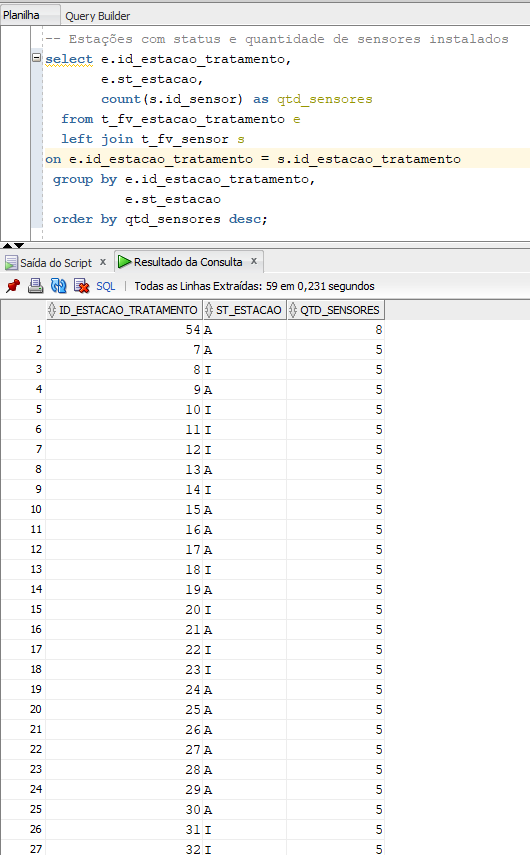
CLOSE c\_tipos;

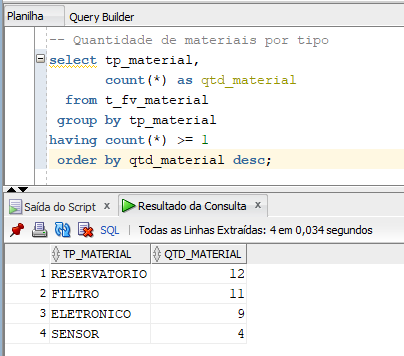
END;

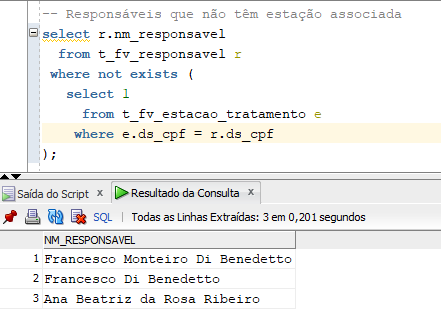
/

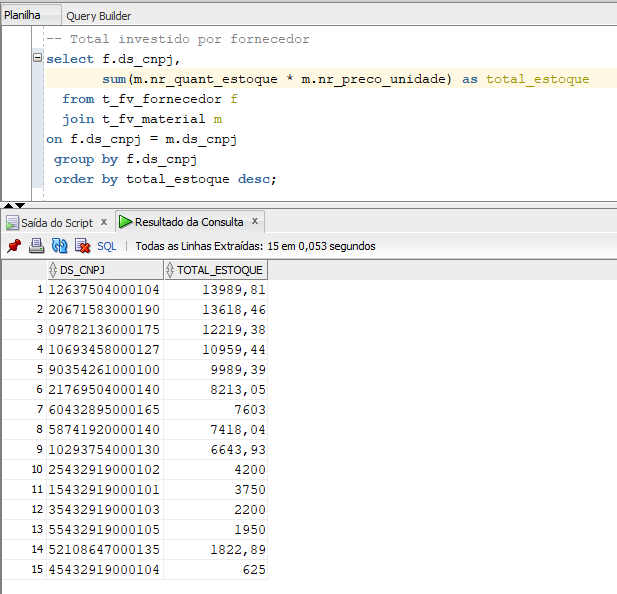
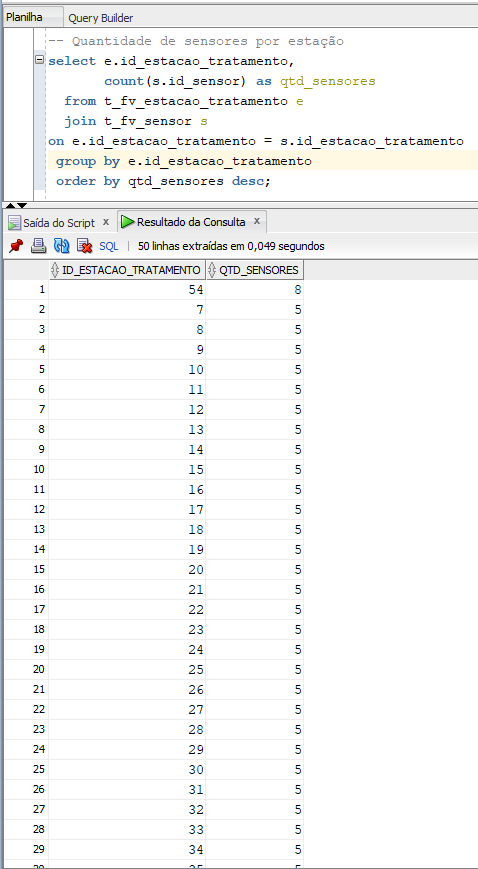


## Consulta Complexa









## Integração com outras matérias

O banco de dados Oracle foi utilizado em diferentes frentes do projeto, servindo como base para a persistência dos dados em diversas disciplinas:

- Disruptive Architectures – IoT, IOB & IA Generativa: os dados dos sensores (como pH, turbidez, nível, temperatura e vazão) simulados por dispositivos IoT foram enviados e armazenados em tabelas do banco Oracle via integração com o Node-RED e Mosquitto (MQTT).

-\*DevOps Tools & Cloud Computing: o banco foi hospedado em uma VM Linux, com acesso remoto configurado, permitindo testes, deploys e integração contínua com outras camadas do sistema.

- Framework Application (.NET): uma aplicação de interface para gestão de fornecedores, sensores e estações foi conectada diretamente ao banco Oracle, utilizando bibliotecas de conexão e manipulando os dados através de procedures e funções criadas na disciplina de Banco de Dados.

## Vídeos

PITCH - https://www.youtube.com/watch?v=pGPg7w3ZHkE

Aplicação - https://www.youtube.com/watch?v=04W2IT3F3AA

## Repositório GIT

https://github.com/2TDSPV-GS-01/MASTERING-RELATIONAL-AND-NON-RELATIONAL-DATABASE