

FIAP

FONTE VIVA

GLOBAL SOLUTION

Francesco Di Benedetto: RM557313

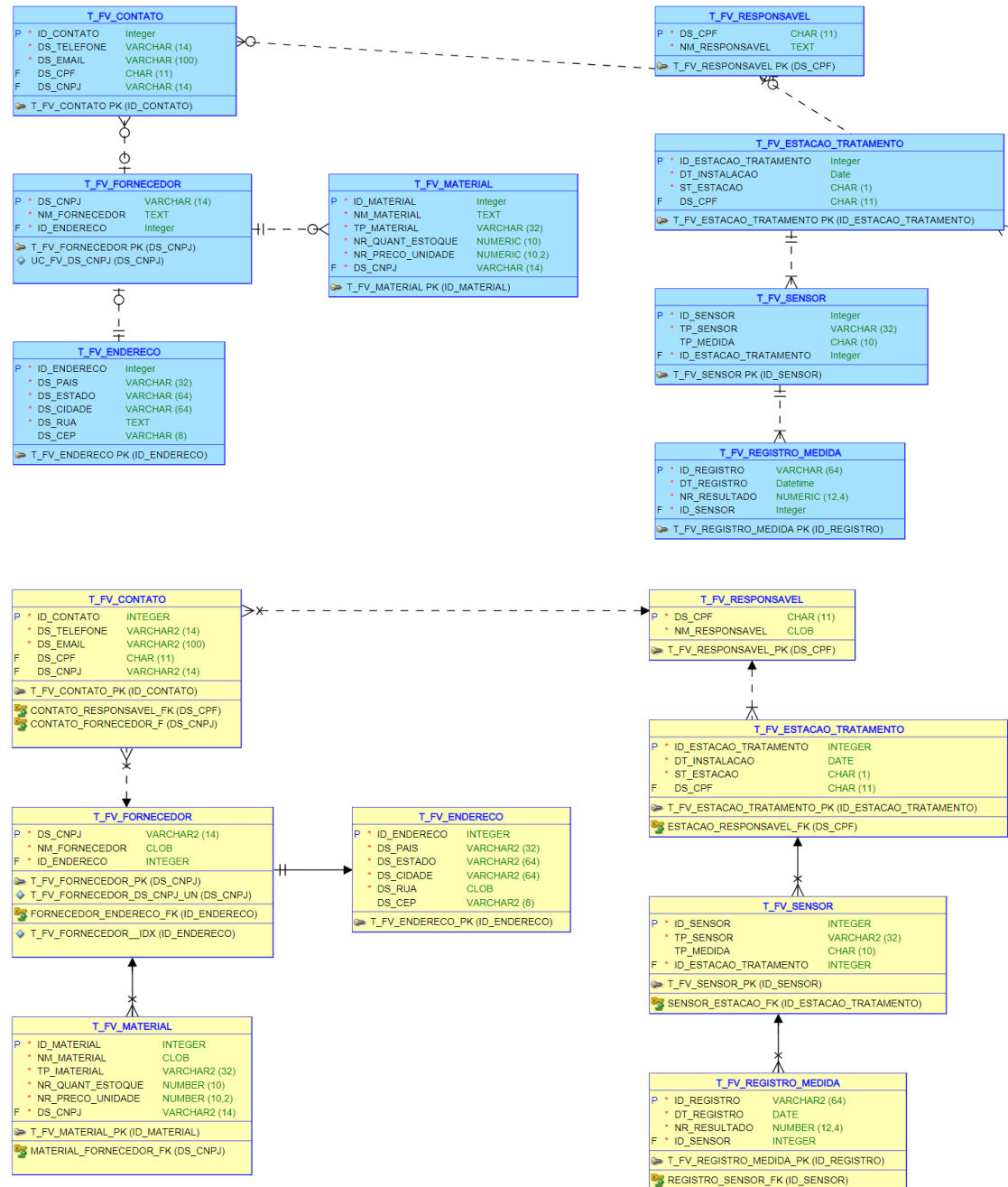
Nathalia Gomes da Silva: RM554945

Gustavo Goulart Bretas: RM555708

Sumário

Modelos lógico e relacional	3
Create Tables	4
Procedures.....	16
Funções.....	32
Bloco anônimo com consultas complexas	34
Cursors Explícitos	40
Consulta Complexa	43
Integração com outras matérias	47

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_RESPONSABEL('98765432100', 'Mariana Silva');
```

```
SP_INSERE_RESPONSABEL('65432198700', 'Carlos Souza');
```

```
SP_INSERE_RESPONSABEL('32165498700', 'Fernanda Costa');
```

```
SP_INSERE_RESPONSABEL('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;  
  
/
```

The screenshot shows a database query tool interface. At the top, there's a 'Planilha' tab and a 'Query Builder' tab. The 'Query Builder' tab is active, displaying the SQL query: `SELECT FUNC_MEDIA_VALOR_POR_SENSOR('PH') FROM DUAL;`. Below the query, there's a 'Resultado da Consulta' tab. The status bar indicates 'Todas as Linhas Extraídas: 1 em 0,055 segundos'. The query result is displayed in a table with one row and one column.

	FUNC_MEDIA_VALOR_POR_SENSOR('PH')
1	7,506


```

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;
/

```

Planilha		Query Builder	
		SELECT FUNC_TOTAL_POR_FORNECEDOR('15432919000101') AS TOTAL_INVESTIDO FROM DUAL;	
		Resultado da Consulta x	
		SQL Todas as Linhas Extraídas: 1 em 0,093 segundos	
		TOTAL_INVESTIDO	
		1	3750

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 de pH por responsável

Resultado da Consulta x Saída do Script x

Tarefa concluída em 1,008 segundos

2025-05-01 Responsável: Yuri Fernandes Camargo | pH: 7,7234 - BASICO
2025-05-01 Responsável: Vitor Almeida Sales | pH: 7,1165 - NEUTRO
2025-05-03 Responsável: Bento Macedo Aparecida | pH: 7,5057 - BASICO
2025-05-03 Responsável: Stephany Oliveira Almeida | pH: 6,9552 - NEUTRO
2025-05-04 Responsável: Dom Ribeiro Abreu | pH: 6,9072 - NEUTRO
2025-05-04 Responsável: Aylla Pereira Garcia | pH: 7,0094 - NEUTRO
2025-05-05 Responsável: Vitor Almeida Sales | pH: 6,9202 - NEUTRO
2025-05-07 Responsável: Sabrina Pacheco Leao | pH: 6,9449 - NEUTRO
2025-05-07 Responsável: Dom Ribeiro Abreu | pH: 8,0422 - BASICO
2025-05-09 Responsável: Juan Nascimento Ribeiro | pH: 7,5912 - BASICO
2025-05-10 Responsável: Clarice Gomes Rios | pH: 8,2806 - BASICO
2025-05-11 Responsável: Yuri Fernandes Camargo | pH: 7,0988 - NEUTRO
2025-05-12 Responsável: Aylla Pereira Garcia | pH: 8,4251 - BASICO
2025-05-12 Responsável: Joao Pedro Costela Moreira | pH: 7,9389 - BASICO
2025-05-13 Responsável: Bento Macedo Aparecida | pH: 7,6584 - BASICO
2025-05-14 Responsável: Ravi Pires Casa Grande | pH: 6,6465 - NEUTRO
2025-05-15 Responsável: Srta Juliana Sousa Monteiro | pH: 7,6673 - BASICO
2025-05-15 Responsável: Luiz Felipe Mendes Sousa | pH: 7,1644 - NEUTRO
2025-05-17 Responsável: Bento Macedo Aparecida | pH: 6,8716 - NEUTRO
2025-05-17 Responsável: Dom Ribeiro Abreu | pH: 8,1922 - BASICO
2025-05-17 Responsável: Luiz Felipe Mendes Sousa | pH: 8,4524 - BASICO
2025-05-19 Responsável: Maria Eduarda Pinto Vargas | pH: 8,4338 - BASICO
2025-05-20 Responsável: Maria Eduarda Pinto Vargas | pH: 7,5724 - BASICO
2025-05-22 Responsável: Danilo Farias Barros | pH: 7,3021 - NEUTRO
2025-05-23 Responsável: Joao Pedro Costela Moreira | pH: 7,8285 - BASICO
2025-05-24 Responsável: Aylla Pereira Garcia | pH: 7,1274 - NEUTRO
2025-05-25 Responsável: Yuri Fernandes Camargo | pH: 8,2585 - BASICO
2025-05-25 Responsável: Gustavo Camargo Camargo | pH: 8,0681 - BASICO
2025-05-25 Responsável: Srta Juliana Sousa Monteiro | pH: 6,9774 - NEUTRO
2025-05-26 Responsável: Vitor Almeida Sales | pH: 7,3128 - NEUTRO
2025-05-27 Responsável: Yuri Fernandes Camargo | pH: 6,7141 - NEUTRO
2025-05-27 Responsável: Luisa Brito Correia | pH: 8,0037 - BASICO
2025-05-29 Responsável: Arthur Barbosa da Conceicao | pH: 7,0336 - NEUTRO
2025-05-29 Responsável: Bento Macedo Aparecida | pH: 7,6115 - BASICO
2025-05-30 Responsável: Matteo Duarte Silveira | pH: 7,3554 - NEUTRO

-- 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;  
/
```

```
-- Monitoramento geral de sensores por estação

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,825 segundos

Estação: 15 | Sensores: 5 | Status: COMPLETA
Estação: 16 | Sensores: 5 | Status: COMPLETA
Estação: 17 | Sensores: 5 | Status: COMPLETA
Estação: 19 | Sensores: 5 | Status: COMPLETA
Estação: 21 | Sensores: 5 | Status: COMPLETA
Estação: 24 | Sensores: 5 | Status: COMPLETA
Estação: 25 | Sensores: 5 | Status: COMPLETA
Estação: 26 | Sensores: 5 | Status: COMPLETA
Estação: 27 | Sensores: 5 | Status: COMPLETA
Estação: 28 | Sensores: 5 | Status: COMPLETA
Estação: 29 | Sensores: 5 | Status: COMPLETA
Estação: 30 | Sensores: 5 | Status: COMPLETA
Estação: 33 | Sensores: 5 | Status: COMPLETA
Estação: 34 | Sensores: 5 | Status: COMPLETA
Estação: 35 | Sensores: 5 | Status: COMPLETA
Estação: 36 | Sensores: 5 | Status: COMPLETA
Estação: 39 | Sensores: 5 | Status: COMPLETA
Estação: 40 | Sensores: 5 | Status: COMPLETA
Estação: 41 | Sensores: 5 | Status: COMPLETA
Estação: 43 | Sensores: 5 | Status: COMPLETA
Estação: 44 | Sensores: 5 | Status: COMPLETA
Estação: 45 | Sensores: 5 | Status: COMPLETA
Estação: 47 | Sensores: 5 | Status: COMPLETA
Estação: 48 | Sensores: 5 | Status: COMPLETA
Estação: 49 | Sensores: 5 | Status: COMPLETA
Estação: 50 | Sensores: 5 | Status: COMPLETA
Estação: 51 | Sensores: 5 | Status: COMPLETA
Estação: 52 | Sensores: 5 | Status: COMPLETA
Estação: 53 | Sensores: 5 | Status: COMPLETA
Estação: 54 | Sensores: 8 | Status: EXCESSO
Estação: 107 | Sensores: 5 | Status: COMPLETA
Estação: 81 | Sensores: 3 | Status: INCOMPLETA
Estação: 84 | Sensores: 0 | Status: INCOMPLETA

Procedimento PL/SQL concluído com sucesso.
```

Cursors 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));

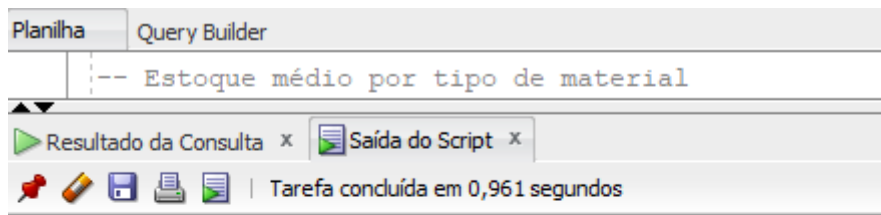
END IF;

END LOOP;

CLOSE c_materiais;

END;

/



```
Tipo: FILTRO | Média ok: 33
Tipo: RESERVATORIO | Média ok: 28
Tipo: ELETRONICO | Média ok: 33
Tipo: SENSOR | Média ok: 26
```

Procedimento PL/SQL concluído com sucesso.

-- 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 || ')'

);

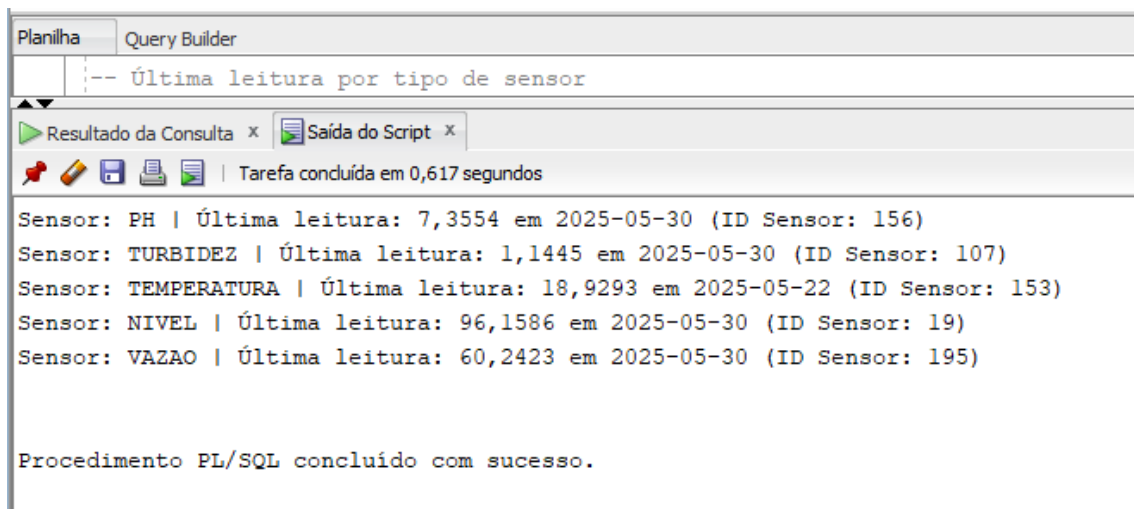
END LOOP;

CLOSE c_tipos;

END;

/

```



Consulta Complexa

Planilha

Query Builder

-- Estações com status e quantidade de sensores instalados

select

e.id_estacao_tratamento,

e.st_estacao,

count(s.id_sensor) as qtd_sensores

from t_fv_estacao_tratamento e

left join t_fv_sensor s

on e.id_estacao_tratamento = s.id_estacao_tratamento

group by e.id_estacao_tratamento,

e.st_estacao

order by qtd_sensores desc;

Saída do Script x

Resultado da Consulta x

Todas as Linhas Extraídas: 59 em 0,231 segundos

	ID_ESTACAO_TRATAMENTO	ST_ESTACAO	QTD_SENSORES
1	54	A	8
2	7	A	5
3	8	I	5
4	9	A	5
5	10	I	5
6	11	I	5
7	12	I	5
8	13	A	5
9	14	I	5
10	15	A	5
11	16	A	5
12	17	A	5
13	18	I	5
14	19	A	5
15	20	I	5
16	21	A	5
17	22	I	5
18	23	I	5
19	24	A	5
20	25	A	5
21	26	A	5
22	27	A	5
23	28	A	5
24	29	A	5
25	30	A	5
26	31	I	5
27	32	I	5

Planilha Query Builder

```
-- Quantidade de materiais por tipo
select tp_material,
       count(*) as qtd_material
  from t_fv_material
 group by tp_material
having count(*) >= 1
 order by qtd_material desc;
```

Saída do Script x
 Resultado da Consulta x

SQL | Todas as Linhas Extraídas: 4 em 0,034 segundos

	TP_MATERIAL	QTD_MATERIAL
1	RESERVATORIO	12
2	FILTRO	11
3	ELETRONICO	9
4	SENSOR	4

```
-- Responsáveis que não têm estação associada
select r.nm_responsavel
  from t_fv_responsavel r
 where not exists (
    select 1
      from t_fv_estacao_tratamento e
     where e.ds_cpf = r.ds_cpf
  );
```

Saída do Script x
 Resultado da Consulta x

SQL | Todas as Linhas Extraídas: 3 em 0,201 segundos

	NM_RESPONSAVEL
1	Francesco Monteiro Di Benedetto
2	Francesco Di Benedetto
3	Ana Beatriz da Rosa Ribeiro

Planilha		Query Builder
		<pre>-- Quantidade de sensores por estação select e.id_estacao_tratamento, count(s.id_sensor) as qtd_sensores from t_fv_estacao_tratamento e join t_fv_sensor s on e.id_estacao_tratamento = s.id_estacao_tratamento group by e.id_estacao_tratamento order by qtd_sensores desc;</pre>
		<div> <div>Saída do Script x</div> <div>Resultado da Consulta x</div> </div> <div> SQL 50 linhas extraídas em 0,049 segundos </div>
ID_ESTACAO_TRATAMENTO	QTD_SENSORES	
1	54	8
2	7	5
3	8	5
4	9	5
5	10	5
6	11	5
7	12	5
8	13	5
9	14	5
10	15	5
11	16	5
12	17	5
13	18	5
14	19	5
15	20	5
16	21	5
17	22	5
18	23	5
19	24	5
20	25	5
21	26	5
22	27	5
23	28	5
24	29	5
25	30	5
26	31	5
27	32	5
28	33	5
29	34	5
30	35	5

Planilha		Query Builder																																																
		<pre>-- Total investido por fornecedor select f.ds_cnpj, sum(m.nr_quant_estoque * m.nr_preco_unidade) as total_estoque from t_fv_fornecedor f join t_fv_material m on f.ds_cnpj = m.ds_cnpj group by f.ds_cnpj order by total_estoque desc;</pre>																																																
		<div> <div>Saída do Script x</div> <div>Resultado da Consulta x</div> </div> <div> SQL Todas as Linhas Extraídas: 15 em 0,053 segundos </div> <table> <thead> <tr> <th></th><th>DS_CNPJ</th><th>TOTAL_ESTOQUE</th></tr> </thead> <tbody> <tr><td>1</td><td>12637504000104</td><td>13989,81</td></tr> <tr><td>2</td><td>20671583000190</td><td>13618,46</td></tr> <tr><td>3</td><td>09782136000175</td><td>12219,38</td></tr> <tr><td>4</td><td>10693458000127</td><td>10959,44</td></tr> <tr><td>5</td><td>90354261000100</td><td>9989,39</td></tr> <tr><td>6</td><td>21769504000140</td><td>8213,05</td></tr> <tr><td>7</td><td>60432895000165</td><td>7603</td></tr> <tr><td>8</td><td>58741920000140</td><td>7418,04</td></tr> <tr><td>9</td><td>10293754000130</td><td>6643,93</td></tr> <tr><td>10</td><td>25432919000102</td><td>4200</td></tr> <tr><td>11</td><td>15432919000101</td><td>3750</td></tr> <tr><td>12</td><td>35432919000103</td><td>2200</td></tr> <tr><td>13</td><td>55432919000105</td><td>1950</td></tr> <tr><td>14</td><td>52108647000135</td><td>1822,89</td></tr> <tr><td>15</td><td>45432919000104</td><td>625</td></tr> </tbody> </table>		DS_CNPJ	TOTAL_ESTOQUE	1	12637504000104	13989,81	2	20671583000190	13618,46	3	09782136000175	12219,38	4	10693458000127	10959,44	5	90354261000100	9989,39	6	21769504000140	8213,05	7	60432895000165	7603	8	58741920000140	7418,04	9	10293754000130	6643,93	10	25432919000102	4200	11	15432919000101	3750	12	35432919000103	2200	13	55432919000105	1950	14	52108647000135	1822,89	15	45432919000104	625
	DS_CNPJ	TOTAL_ESTOQUE																																																
1	12637504000104	13989,81																																																
2	20671583000190	13618,46																																																
3	09782136000175	12219,38																																																
4	10693458000127	10959,44																																																
5	90354261000100	9989,39																																																
6	21769504000140	8213,05																																																
7	60432895000165	7603																																																
8	58741920000140	7418,04																																																
9	10293754000130	6643,93																																																
10	25432919000102	4200																																																
11	15432919000101	3750																																																
12	35432919000103	2200																																																
13	55432919000105	1950																																																
14	52108647000135	1822,89																																																
15	45432919000104	625																																																

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>