



Faculdade de Informática e Administração Paulista

## DATABASE APPLICATION & DATA SCIENCE

### **Challenge – Sprint 3**

<b>RM (SOMENTE NÚMEROS)</b>	<b>NOME COMPLEMENTO (SEM ABREVIAR)</b>
<b>93613</b>	<b>Vinicius de Oliveira</b>
<b>96320</b>	<b>NATHÁLIA MAIA</b>
<b>94972</b>	<b>RAFAELA DA SILVA</b>
<b>94311</b>	<b>MARCOS BILOBRAM</b>



**SUMÁRIO**

1 – DESCRIÇÃO DO PROJETO E REGRAS DE NEGÓCIO .....	5
2 – Dicionário de Dados.....	6
3 – Projeto Lógico do Banco de Dados.....	7
4 – Projeto Físico do Banco de Dados .....	8
5 – Data Definition Language – DDL .....	9
6 – Data Manipulation Language – DML (INSERT) .....	16
7 – DATA MANIPULATION LANGUAGE – DML (UPDATE / DELETE).....	20
8 – DATA QUERY LANGUAGE – DQL (SELECT) .....	21
8.1 – Relatório simples contendo apenas uma tabela com dados ordenados. .....	<b>Erro! Indicador não definido.</b>

## **1 – Descrição do Projeto e Regras de Negócio**

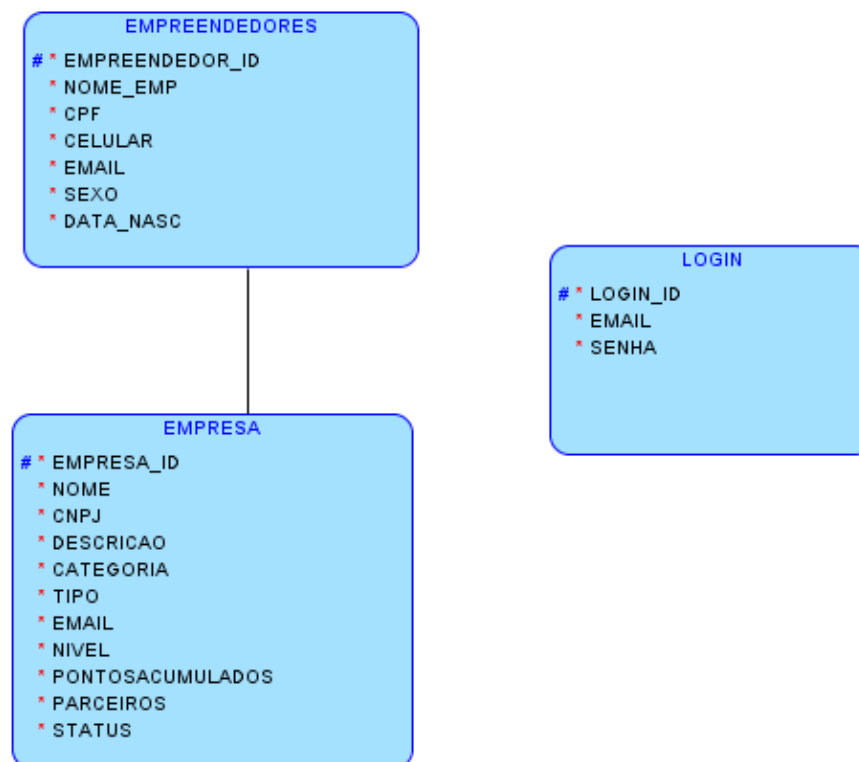
A inovação e a transformação tecnológica podem acabar trazendo diversos impactos práticos aos empreendedores brasileiros. Alguns desses impactos seriam a redução de tarefas manuais, como gestão de tarefas, redução de mão de obra, diminuição de desperdícios, além da melhoria na qualidade dos produtos, e outros. Desse modo, o investimento em processos de automação e em tecnologia transformam a realidade de diversas indústrias, desde a matéria-prima até o consumidor final.

A AgroSync coleta e analisa as informações do ambiente de produção de agricultura brasileira e tem como objetivo principal auxiliar os agentes do agro nacional a realizar uma gestão automatizada e personalizada de suas plantações. Ademais, essa plataforma auxilia no monitoramento remoto de plantações, na previsão de safras e condições climáticas, disponibilizando aos seus clientes uma orientação personalizada para otimização da produção e uma comunicação direta com especialistas em agricultura, se necessário. Por fim, o AgroSync analisa dados fornecidos à ele, para que possua uma tomada de decisões mais precisa.

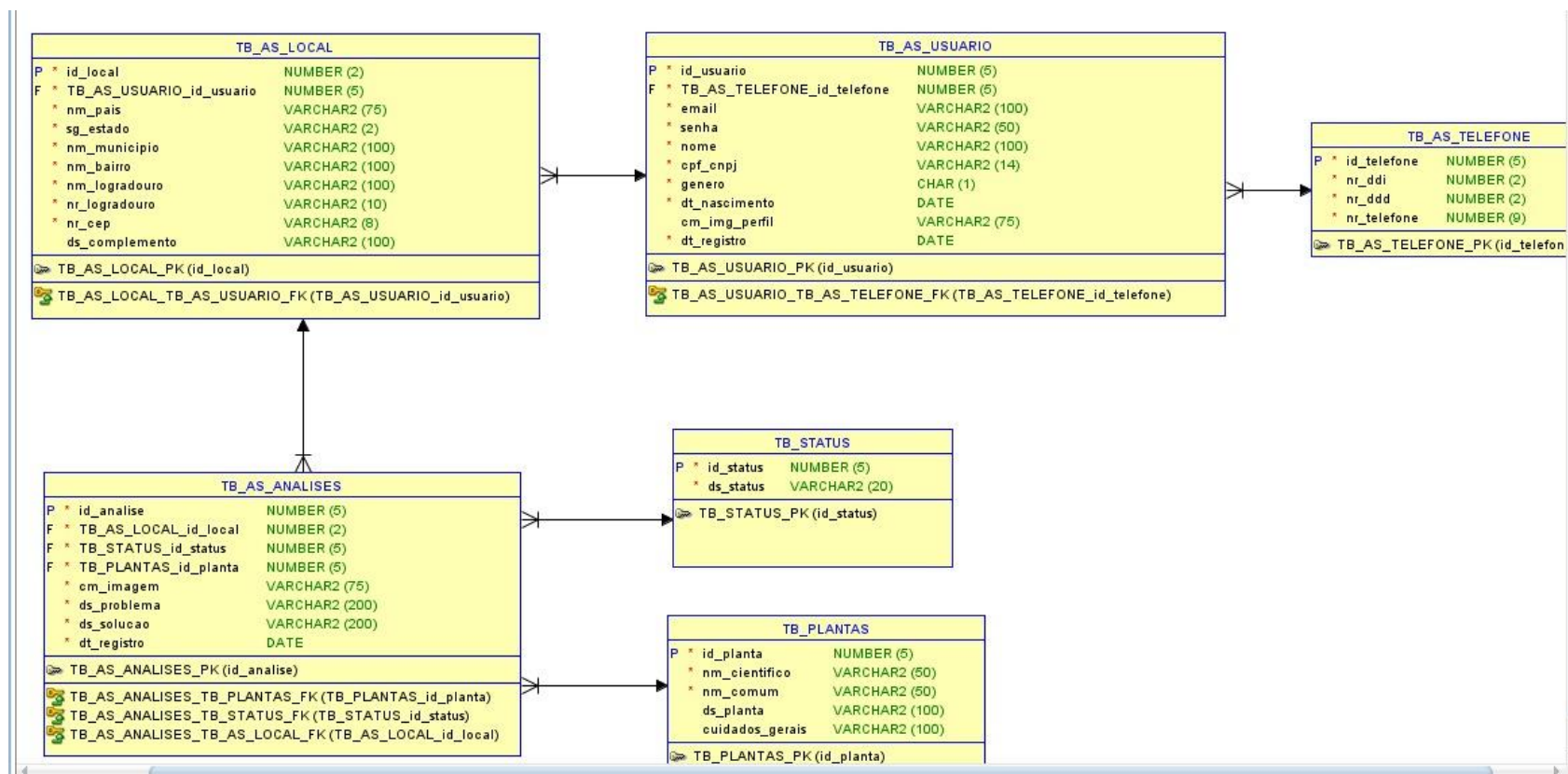
## 2 – Dicionário de Dados

Tabela	EMPRESA			
Descrição	Tabela de Empresas. Relaciona-se com a tabela EMPREENDEDOR			
Coluna	Tipo de Dados	Tamanho	Constraint	Descrição
empresa_id	NUMBER	7	PK	Identificador único da empresa
nome	VARCHAR	50	NN	nome da empresa
cnpj	VARCHAR	50	NN	CNPJ da empresa
descricao	VARCHAR	100	NN	Tipo da empresa
categoria	VARCHAR	20	NN	Categoria da empresa
tipo	VARCHAR	20	NN	Tipo da empresa
email	VARCHAR	200	NN	E-mail da empresa
nivel	NUMBER	7	NN	Nível da empresa
pontosacumulados	NUMBER	7	NN	Pontos da empresa
parceiros	NUMBER	7	NN	Parceiros da empresa
status	NUMBER	7	NN	Status da conta da empresa (1 = Ativo, 0 = Inativo)

### 3 – Projeto Lógico do Banco de Dados



## 4 – Projeto Físico do Banco de Dados





## 5 – Data Definition Language – DDL

```
-- Removendo tabelas e sequences

DROP TABLE tb_as_analise CASCADE CONSTRAINTS;
DROP TABLE tb_as_local CASCADE CONSTRAINTS;
DROP TABLE tb_as_telefone CASCADE CONSTRAINTS;
DROP TABLE tb_as_usuario CASCADE CONSTRAINTS;
DROP TABLE tb_as_planta CASCADE CONSTRAINTS;
DROP TABLE tb_as_status CASCADE CONSTRAINTS;
DROP TABLE tb_as_auditoria CASCADE CONSTRAINTS;
DROP TABLE TB_AS_ERRO;

DROP SEQUENCE seq_id_analise;
DROP SEQUENCE seq_id_local;
DROP SEQUENCE seq_id_telefone;
DROP SEQUENCE seq_id_usuario;
DROP SEQUENCE seq_id_planta;
DROP SEQUENCE seq_id_status;
DROP SEQUENCE seq_id_registro;
DROP SEQUENCE seq_id_erro;

-- Criação de tabelas e sequences

-- Gerado por Oracle SQL Developer Data Modeler 22.2.0.165.1149
-- em:          2023-08-19 12:32:53 BRT
-- site:        Oracle Database 11g
-- tipo:        Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
-- predefined type, no DDL - XMLTYPE

CREATE TABLE tb_as_analise (
    id_analise    NUMBER(5) NOT NULL,
    id_local      NUMBER(2) NOT NULL,
    id_status     NUMBER(5) NOT NULL,
    id_planta     NUMBER(5) NOT NULL,
    cm_imagem     VARCHAR2(75) NOT NULL,
    ds_problema   VARCHAR2(200) NOT NULL,
    ds_solucao    VARCHAR2(200),
    dt_registro   DATE NOT NULL
);

COMMENT ON COLUMN tb_as_analise.id_analise IS
```

```
'Este campo sera para armazenar o id da analise. Ele e
obrigatorio e chave primaria.';

COMMENT ON COLUMN tb_as_analise.id_local IS
    'Este campo sera para armazenar o id do local. Ele e
obrigatorio e chave estrangeira.';

COMMENT ON COLUMN tb_as_analise.id_status IS
    'Este campo sera para armazenar o id do status. Ele e
obrigatorio e chave estrangeira.';

COMMENT ON COLUMN tb_as_analise.id_planta IS
    'Este campo sera para armazenar o id da planta. Ele e
obrigatorio e chave estrangeira.';

COMMENT ON COLUMN tb_as_analise.cm_imagem IS
    'Este campo sera para armazenar o caminho da imagem da
analise. Ele e obrigatorio.';

COMMENT ON COLUMN tb_as_analise.ds_problema IS
    'Este campo sera para armazenar a descrição da analise.
Ele e obrigatorio.';

COMMENT ON COLUMN tb_as_analise.ds_solucão IS
    'Este campo sera para armazenar a descrição da solucao.
Ele e obrigatorio.';

COMMENT ON COLUMN tb_as_analise.dt_registro IS
    'Este campo sera para armazenar a data de registro da
solucao. Ele e obrigatorio.';

ALTER TABLE tb_as_analise ADD CONSTRAINT tb_as_analises_pk
PRIMARY KEY ( id_analise );

CREATE TABLE tb_as_local (
    id_local          NUMBER(2) NOT NULL,
    id_usuario        NUMBER(5) NOT NULL,
    nm_pais            VARCHAR2(75) NOT NULL,
    sg_estado          VARCHAR2(2) NOT NULL,
    nm_municipio       VARCHAR2(100) NOT NULL,
    nm_bairro          VARCHAR2(100) NOT NULL,
    nm_logradouro       VARCHAR2(100) NOT NULL,
    nr_logradouro       VARCHAR2(10) NOT NULL,
    nr_cep              VARCHAR2(8) NOT NULL,
    ds_complemento     VARCHAR2(100)
);

COMMENT ON COLUMN tb_as_local.id_local IS
    'Este campo sera para armazenar o id do local. Ele e
obrigatorio e chave primaria.';
```

```
COMMENT ON COLUMN tb_as_local.id_usuario IS
    'Este campo será para armazenar o id do usuario. Ele é
obrigatorio e chave estrangeira.';

COMMENT ON COLUMN tb_as_local.nm_pais IS
    'Este campo será para armazenar o nome do pais do local.
Ele é obrigatorio.';

COMMENT ON COLUMN tb_as_local.sg_estado IS
    'Este campo será para armazenar a sigla do estado do local.
Ele é obrigatorio.';

COMMENT ON COLUMN tb_as_local.nm_municipio IS
    'Este campo será para armazenar o nome do municipio do
local. Ele é obrigatorio.';

COMMENT ON COLUMN tb_as_local.nm_bairro IS
    'Este campo será para armazenar o nome do bairro do local.
Ele é obrigatorio.';

COMMENT ON COLUMN tb_as_local.nm_logradouro IS
    'Este campo será para armazenar o nome do logradouro do
local. Ele é obrigatorio.';

COMMENT ON COLUMN tb_as_local.nr_logradouro IS
    'Este campo será para armazenar o numero do logradouro do
local. Ele é obrigatorio.';

COMMENT ON COLUMN tb_as_local.nr_cep IS
    'Este campo será para armazenar o cep do local. Ele é
obrigatorio.';

COMMENT ON COLUMN tb_as_local.ds_complemento IS
    'Este campo será para armazenar o complemento local. Ele é
obrigatorio.';

ALTER TABLE tb_as_local ADD CONSTRAINT tb_as_local_pk PRIMARY
KEY ( id_local );

CREATE TABLE tb_as_telefone (
    id_telefone NUMBER(5) NOT NULL,
    nr_ddi      NUMBER(2) NOT NULL,
    nr_ddd      NUMBER(2) NOT NULL,
    nr_telefone NUMBER(9) NOT NULL
);

COMMENT ON COLUMN tb_as_telefone.id_telefone IS
    'Este campo será para armazenar o id do telefone. Ele é
obrigatorio e chave primaria.';
```

```
COMMENT ON COLUMN tb_as_telefone.nr_ddi IS
    'Este campo sera para armazenar o ddi do telefone. Ele e
    obrigatorio';

COMMENT ON COLUMN tb_as_telefone.nr_ddd IS
    'Este campo sera para armazenar o ddd do telefone. Ele e
    obrigatorio.';

COMMENT ON COLUMN tb_as_telefone.nr_telefone IS
    'Este campo sera para armazenar o número do telefone. Ele
    e obrigatorio.';

ALTER TABLE tb_as_telefone ADD CONSTRAINT tb_as_telefone_pk
PRIMARY KEY ( id_telefone );

CREATE TABLE tb_as_usuario (
    id_usuario      NUMBER(5) NOT NULL,
    id_telefone     NUMBER(5) NOT NULL,
    email           VARCHAR2(100) NOT NULL,
    senha           VARCHAR2(50) NOT NULL,
    nome            VARCHAR2(100) NOT NULL,
    cpf_cnpj        VARCHAR2(14) NOT NULL,
    genero          CHAR(1) NOT NULL,
    dt_nascimento   DATE NOT NULL,
    cm_img_perfil   VARCHAR2(75),
    dt_registro     DATE NOT NULL
);

COMMENT ON COLUMN tb_as_usuario.id_usuario IS
    'Este campo sera para armazenar o id do usuário. Ele e
    obrigatorio e chave primaria.';

COMMENT ON COLUMN tb_as_usuario.id_telefone IS
    'Este campo sera para armazenar o id do telefone do
    usuario. Ele e obrigatorio e chave estrangeira.';

COMMENT ON COLUMN tb_as_usuario.email IS
    'Este campo sera para armazenar o email do usuario e sera
    usado como login. Ele e obrigatorio.';

COMMENT ON COLUMN tb_as_usuario.senha IS
    'Este campo sera para armazenar a senha do usuario e sera
    usado como senha para o login. Ele e obrigatorio.';

COMMENT ON COLUMN tb_as_usuario.nome IS
    'Este campo sera para armazenar o nome do usuario. Ele e
    obrigatorio.';

COMMENT ON COLUMN tb_as_usuario.cpf_cnpj IS
```

```
'Este campo sera para armazenar o cpd ou cnpj do usuario.  
Ele e obrigatorio.';
```

```
COMMENT ON COLUMN tb_as_usuario.genero IS  
'Este campo sera para armazenar o genero do usuario. Ele e  
obrigatorio.';
```

```
COMMENT ON COLUMN tb_as_usuario.dt_nascimento IS  
'Este campo sera para armazenar a data de nascimento do  
usuario. Ele e obrigatorio.';
```

```
COMMENT ON COLUMN tb_as_usuario.cm_img_perfil IS  
'Este campo sera para armazenar o caminho da foto de perfil  
do usuario.';
```

```
COMMENT ON COLUMN tb_as_usuario.dt_registro IS  
'Este campo sera para armazenar a data de registro do  
usuario. Ele e obrigatorio.';
```

```
ALTER TABLE tb_as_usuario ADD CONSTRAINT tb_as_usuario_pk  
PRIMARY KEY ( id_usuario );
```

```
CREATE TABLE tb_as_planta (  
    id_planta          NUMBER(5) NOT NULL,  
    nm_comum           VARCHAR2(50) NOT NULL,  
    nm_cientifico       VARCHAR2(50),  
    ds_planta          VARCHAR2(100),  
    cuidados_gerais     VARCHAR2(100)  
);
```

```
COMMENT ON COLUMN tb_as_planta.id_planta IS  
'Este campo sera para armazenar o id da planta. Ele e  
obrigatorio e chave primaria.';
```

```
COMMENT ON COLUMN tb_as_planta.nm_comum IS  
'Este campo sera para armazenar o nome comum da planta.  
Ele e obrigatorio';
```

```
COMMENT ON COLUMN tb_as_planta.nm_cientifico IS  
'Este campo sera para armazenar o nome cientifico da  
planta.';
```

```
COMMENT ON COLUMN tb_as_planta.ds_planta IS  
'Este campo sera para armazenar a descricao da planta.';
```

```
COMMENT ON COLUMN tb_as_planta.cuidados_gerais IS  
'Este campo sera para armazenar os cuidados gerais da  
planta.';
```

```
ALTER TABLE tb_as_planta ADD CONSTRAINT tb_as_plantas_pk
PRIMARY KEY ( id_planta );

CREATE TABLE tb_as_status (
    id_status NUMBER(5) NOT NULL,
    ds_status VARCHAR2(20) NOT NULL
);

COMMENT ON COLUMN tb_as_status.id_status IS
    'Este campo sera para armazenar o id do status. Ele e
    obrigatorio e chave primaria.';

COMMENT ON COLUMN tb_as_status.ds_status IS
    'Este campo sera para armazenar a descricao do status. Ele
    e obrigatorio.';

ALTER TABLE tb_as_status ADD CONSTRAINT tb_as_status_pk
PRIMARY KEY ( id_status );

ALTER TABLE tb_as_analise
    ADD CONSTRAINT tb_as_analises_tb_as_local_fk FOREIGN KEY (
id_local )
    REFERENCES tb_as_local ( id_local );

ALTER TABLE tb_as_analise
    ADD CONSTRAINT tb_as_analises_tb_as_plantas_fk FOREIGN KEY
( id_planta )
    REFERENCES tb_as_planta ( id_planta );

ALTER TABLE tb_as_analise
    ADD CONSTRAINT tb_as_analises_tb_as_status_fk FOREIGN KEY
( id_status )
    REFERENCES tb_as_status ( id_status );

ALTER TABLE tb_as_local
    ADD CONSTRAINT tb_as_local_tb_as_usuario_fk FOREIGN KEY (
id_usuario )
    REFERENCES tb_as_usuario ( id_usuario );

ALTER TABLE tb_as_usuario
    ADD CONSTRAINT tb_usuario_telefone_fk FOREIGN KEY (
id_telefone )
    REFERENCES tb_as_telefone ( id_telefone );

CREATE TABLE tb_as_auditoria (
    id_registro NUMBER(5),
    nm_usuario VARCHAR2(30),
    dt_hora DATE,
    acao_realizada VARCHAR2(10),
    tabela_afetada VARCHAR2(50)
```

```
);

CREATE TABLE TB_AS_ERRO (
    ID_ERRO NUMBER(5) NOT NULL,
    CD_ERRO NUMBER(10) NOT NULL,
    NM_ERRO VARCHAR2(300) NOT NULL,
    DT_REGISTRO DATE NOT NULL,
    USUARIO VARCHAR2(100) NOT NULL,
    PROCEDIMENTO VARCHAR2(50) NOT NULL
);

CREATE SEQUENCE seq_id_analise
    START WITH 1
    INCREMENT BY 1;

CREATE SEQUENCE seq_id_local
    START WITH 1
    INCREMENT BY 1;

CREATE SEQUENCE seq_id_telefone
    START WITH 1
    INCREMENT BY 1;

CREATE SEQUENCE seq_id_usuario
    START WITH 1
    INCREMENT BY 1;

CREATE SEQUENCE seq_id_planta
    START WITH 1
    INCREMENT BY 1;

CREATE SEQUENCE seq_id_status
    START WITH 1
    INCREMENT BY 1;

CREATE SEQUENCE seq_id_registro
    START WITH 1
    INCREMENT BY 1;

CREATE SEQUENCE seq_id_erro
    START WITH 1
    INCREMENT BY 1;
```

Exemplo - DDL do banco

## 6 – Data Manipulation Language – DML (INSERT)

```
-- Inserindo dados iniciais

INSERT INTO tb_as_telefone (ID_TELEFONE, NR_DDI, NR_DDD,
NR_TELEFONE)
VALUES (seq_id_telefone.NEXTVAL, 55, 11, 935684267);
INSERT INTO tb_as_telefone (ID_TELEFONE, NR_DDI, NR_DDD,
NR_TELEFONE)
VALUES (seq_id_telefone.NEXTVAL, 55, 21, 932689367);
INSERT INTO tb_as_telefone (ID_TELEFONE, NR_DDI, NR_DDD,
NR_TELEFONE)
VALUES (seq_id_telefone.NEXTVAL, 55, 31, 932974268);
INSERT INTO tb_as_telefone (ID_TELEFONE, NR_DDI, NR_DDD,
NR_TELEFONE)
VALUES (seq_id_telefone.NEXTVAL, 55, 41, 939286832);
INSERT INTO tb_as_telefone (ID_TELEFONE, NR_DDI, NR_DDD,
NR_TELEFONE)
VALUES (seq_id_telefone.NEXTVAL, 55, 41, 949382859);

INSERT INTO tb_as_usuario (ID_USUARIO, ID_TELEFONE, EMAIL,
SENHA, NOME, CPF_CNPJ, GENERO, DT_NASCIMENTO, DT_REGISTRO)
VALUES (seq_id_usuario.NEXTVAL, 1, 'vinicius@email.com',
'Vincius123', 'Vinicius de Oliveira', '03954810000135', 'M',
TO_DATE('15/05/2003', 'DD-MM-YYYY'), SYSDATE);
INSERT INTO tb_as_usuario (ID_USUARIO, ID_TELEFONE, EMAIL,
SENHA, NOME, CPF_CNPJ, GENERO, DT_NASCIMENTO, DT_REGISTRO)
VALUES (seq_id_usuario.NEXTVAL, 2, 'maria@email.com',
'Marial23', 'Maria da Lima Silva', '26942875369', 'F',
TO_DATE('21/03/1930', 'DD-MM-YYYY'), SYSDATE);
INSERT INTO tb_as_usuario (ID_USUARIO, ID_TELEFONE, EMAIL,
SENHA, NOME, CPF_CNPJ, GENERO, DT_NASCIMENTO, DT_REGISTRO)
VALUES (seq_id_usuario.NEXTVAL, 3, 'jose@email.com',
'Jose123', 'Jose Carlos Souza', '36256938000123', 'M',
TO_DATE('30/09/1999', 'DD-MM-YYYY'), SYSDATE);
INSERT INTO tb_as_usuario (ID_USUARIO, ID_TELEFONE, EMAIL,
SENHA, NOME, CPF_CNPJ, GENERO, DT_NASCIMENTO, DT_REGISTRO)
VALUES (seq_id_usuario.NEXTVAL, 4, 'camila@email.com',
'Camila123', 'Camila da Silva de Souza', '39658423698', 'F',
TO_DATE('29/07/2004', 'DD-MM-YYYY'), SYSDATE);
INSERT INTO tb_as_usuario (ID_USUARIO, ID_TELEFONE, EMAIL,
SENHA, NOME, CPF_CNPJ, GENERO, DT_NASCIMENTO, DT_REGISTRO)
VALUES (seq_id_usuario.NEXTVAL, 5, 'ricardo@email.com',
'Ricardo123', 'Ricardo dos Santos Cruz', '03954810000135',
'M', TO_DATE('01/03/1960', 'DD-MM-YYYY'), SYSDATE);
```



```
INSERT INTO tb_as_status (ID_STATUS, DS_STATUS)
VALUES (seq_id_status.NEXTVAL, 'Morta');
INSERT INTO tb_as_status (ID_STATUS, DS_STATUS)
VALUES (seq_id_status.NEXTVAL, 'Em progresso');
INSERT INTO tb_as_status (ID_STATUS, DS_STATUS)
VALUES (seq_id_status.NEXTVAL, 'Em tratamento');
INSERT INTO tb_as_status (ID_STATUS, DS_STATUS)
VALUES (seq_id_status.NEXTVAL, 'Analisando');
INSERT INTO tb_as_status (ID_STATUS, DS_STATUS)
VALUES (seq_id_status.NEXTVAL, 'Curada');

INSERT INTO tb_as_planta (ID_PLANTA, NM_COMUM, NM_CIENTIFICO,
DS_PLANTA, CUIDADOS_GERAIS)
VALUES (seq_id_planta.NEXTVAL, 'Rosa', 'Rosa spp.', 'A
rosa é uma planta conhecida por suas flores delicadas e
perfumadas.', 'A rosa requer rega regular e luz solar direta
para florescer bem.');
```

```
INSERT INTO tb_as_planta (ID_PLANTA, NM_COMUM, NM_CIENTIFICO,
DS_PLANTA, CUIDADOS_GERAIS)
VALUES (seq_id_planta.NEXTVAL, 'Lírio', 'Lilium spp.', 'O
lírio é uma planta de flores vistosas e variadas cores.', 'Os
lírios preferem solo bem drenado e devem ser regados
regularmente.');
```

```
INSERT INTO tb_as_planta (ID_PLANTA, NM_COMUM, NM_CIENTIFICO,
DS_PLANTA, CUIDADOS_GERAIS)
VALUES (seq_id_planta.NEXTVAL, 'Cacto', 'Cactaceae', 'Os
cactos são plantas suculentas conhecidas por sua resistência à
seca.', 'Os cactos precisam de pouca água e luz solar
direta.');
```

```
INSERT INTO tb_as_planta (ID_PLANTA, NM_COMUM, NM_CIENTIFICO,
DS_PLANTA, CUIDADOS_GERAIS)
VALUES (seq_id_planta.NEXTVAL, 'Orquídea', 'Orchidaceae',
'As orquídeas são conhecidas por suas flores elegantes e
variadas.', 'As orquídeas necessitam de umidade constante e
luz indireta.');
```

```
INSERT INTO tb_as_planta (ID_PLANTA, NM_COMUM, NM_CIENTIFICO,
DS_PLANTA, CUIDADOS_GERAIS)
VALUES (seq_id_planta.NEXTVAL, 'Samambaia', 'Nephrolepis
exaltata', 'A samambaia é uma planta que gosta de ambientes
úmidos e sombreados.', 'Mantenha o solo da samambaia sempre
úmido e evite luz solar direta.');
```

```
INSERT INTO tb_as_local (ID_LOCAL, ID_USUARIO, NM_PAIS,
SG_ESTADO, NM_MUNICIPIO, NM_BAIRRO, NM_LOGRADOURO,
NR_LOGRADOURO, NR_CEP, DS_COMPLEMENTO)
VALUES (seq_id_local.NEXTVAL, 1, 'Brasil', 'SP', 'São
Paulo', 'Centro', 'Avenida Paulista', '123', '01234567', 'Apto
101');
```

```
INSERT INTO tb_as_local (ID_LOCAL, ID_USUARIO, NM_PAIS,
SG_ESTADO, NM_MUNICIPIO, NM_BAIRRO, NM_LOGRADOURO,
NR_LOGRADOURO, NR_CEP, DS_COMPLEMENTO)
VALUES (seq_id_local.NEXTVAL, 2, 'Brasil', 'RJ', 'Rio de
Janeiro', 'Copacabana', 'Rua Nossa Senhora de Copacabana',
'456', '22334455', 'Casa 2A');
INSERT INTO tb_as_local (ID_LOCAL, ID_USUARIO, NM_PAIS,
SG_ESTADO, NM_MUNICIPIO, NM_BAIRRO, NM_LOGRADOURO,
NR_LOGRADOURO, NR_CEP, DS_COMPLEMENTO)
VALUES (seq_id_local.NEXTVAL, 3, 'Brasil', 'MG', 'Belo
Horizonte', 'Savassi', 'Avenida Getúlio Vargas', '789',
'30123456', NULL);
INSERT INTO tb_as_local (ID_LOCAL, ID_USUARIO, NM_PAIS,
SG_ESTADO, NM_MUNICIPIO, NM_BAIRRO, NM_LOGRADOURO,
NR_LOGRADOURO, NR_CEP, DS_COMPLEMENTO)
VALUES (seq_id_local.NEXTVAL, 4, 'Brasil', 'RS', 'Porto
Alegre', 'Moinhos de Vento', 'Rua Padre Chagas', '101',
'90450123', 'Sala 3');
INSERT INTO tb_as_local (ID_LOCAL, ID_USUARIO, NM_PAIS,
SG_ESTADO, NM_MUNICIPIO, NM_BAIRRO, NM_LOGRADOURO,
NR_LOGRADOURO, NR_CEP, DS_COMPLEMENTO)
VALUES (seq_id_local.NEXTVAL, 5, 'Brasil', 'DF',
'Brasília', 'Asa Norte', 'Quadra 102', 'Apto 501', '70000123',
'Bloco B');

INSERT INTO tb_as_analise (ID_ANALISE, ID_LOCAL, ID_STATUS,
ID_PLANTA, CM_IMAGEM, DS_PROBLEMA, DS_SOLUCAO, DT_REGISTRO)
VALUES (seq_id_analise.NEXTVAL, 1,
ROUND(DBMS_RANDOM.VALUE(1, 5)), ROUND(DBMS_RANDOM.VALUE(1,
5)), 'planta1.jpg', 'A planta apresenta folhas murchas.',
'Aumentar a rega e verificar a exposição solar.', SYSDATE);
INSERT INTO tb_as_analise (ID_ANALISE, ID_LOCAL, ID_STATUS,
ID_PLANTA, CM_IMAGEM, DS_PROBLEMA, DS_SOLUCAO, DT_REGISTRO)
VALUES (seq_id_analise.NEXTVAL, 2,
ROUND(DBMS_RANDOM.VALUE(1, 5)), ROUND(DBMS_RANDOM.VALUE(1,
5)), 'planta2.png', 'A planta está com folhas amareladas.',
'Adicionar fertilizante e ajustar a rega.', SYSDATE);
INSERT INTO tb_as_analise (ID_ANALISE, ID_LOCAL, ID_STATUS,
ID_PLANTA, CM_IMAGEM, DS_PROBLEMA, DS_SOLUCAO, DT_REGISTRO)
VALUES (seq_id_analise.NEXTVAL, 3,
ROUND(DBMS_RANDOM.VALUE(1, 5)), ROUND(DBMS_RANDOM.VALUE(1,
5)), 'planta3.jpg', 'A planta possui pragas nas folhas.', 'Usar
inseticida e remover as folhas afetadas.', SYSDATE);
INSERT INTO tb_as_analise (ID_ANALISE, ID_LOCAL, ID_STATUS,
ID_PLANTA, CM_IMAGEM, DS_PROBLEMA, DS_SOLUCAO, DT_REGISTRO)
VALUES (seq_id_analise.NEXTVAL, 4,
ROUND(DBMS_RANDOM.VALUE(1, 5)), ROUND(DBMS_RANDOM.VALUE(1,
5)), 'planta4.jpg', 'A planta está com excesso de água.',
'Reduzir a rega e melhorar a drenagem do solo.', SYSDATE);
```

```
INSERT INTO tb_as_analise (ID_ANALISE, ID_LOCAL, ID_STATUS,  
ID_PLANTA, CM_IMAGEM, DS_PROBLEMA, DS_SOLUCAO, DT_REGISTRO)  
VALUES (seq_id_analise.NEXTVAL, 5,  
ROUND(DBMS_RANDOM.VALUE(1, 5)), ROUND(DBMS_RANDOM.VALUE(1,  
5)), 'planta5.png', 'A planta está com manchas nas folhas.',  
'Identificar a causa das manchas e tratá-las.', SYSDATE);  
  
COMMIT;
```

Exemplo de Script em SQL para carga inicial do banco de dados no banco

## 7 – Data Manipulation Language – DML (UPDATE / DELETE)

```
UPDATE tb_as_analise  
    SET CM_IMAGEM = 'nova_imagem_planta.png'  
    WHERE ID_ANALISE = 3;
```

```
UPDATE tb_as_telefone  
    SET NR_TELEFONE = '966309946'  
    WHERE ID_TELEFONE = 2;
```

```
COMMIT;
```

```
DELETE FROM tb_as_analise  
WHERE ID_ANALISE = 4;
```

```
COMMIT;
```

Exemplo de Comandos de Atualização e de Deleção

## 8 – Data Query Language – DQL (SELECT)

```
select * from tb_as_usuario u
inner join tb_as_telefone t on (u.id_telefone = t.id_telefone);

SELECT ID_STATUS, COUNT(*) FROM tb_as_analise GROUP BY
ID_STATUS;

SELECT COUNT(*)
FROM TB_AS_ANALISE a
INNER JOIN TB_AS_STATUS s ON (a.id_status = s.id_status)
WHERE UPPER(s.DS_STATUS) = UPPER('ANALISANDO');

SELECT DS_SOLUCAO
FROM TB_AS_ANALISE
WHERE DS_PROBLEMA LIKE '%' || 'murchas' || '%';
```

Exemplo de Relatório

## 9 – Procedures

### 9.1 - Procedure para cadastrar um usuario junto com o telefone dele

```
CREATE OR REPLACE PROCEDURE PRC_CADASTRO_USUARIO (  
    p_email IN VARCHAR2,  
    p_senha IN VARCHAR2,  
    p_nome IN VARCHAR2,  
    p_cpf_cnpj IN VARCHAR2,  
    p_genero IN CHAR,  
    p_dt_nascimento IN DATE,  
    p_cm_img_perfil IN VARCHAR2,  
    p_nr_ddi IN NUMBER,  
    p_nr_ddd IN NUMBER,  
    p_nr_telefone IN NUMBER  
) IS
```

```
    idTelefone NUMBER;
```

```
    v_codigo_erro NUMBER;
```

```
    v_mensagem_erro VARCHAR2(250);
```

```
BEGIN
```

```
    IF INSTR(p_email, '@') = 0 THEN
```

*RAISE\_APPLICATION\_ERROR(-20001, 'O email deve conter o caracter @.');*

*END IF;*

*IF LENGTH(p\_cpf\_cnpj) != 11*

*AND LENGTH(p\_cpf\_cnpj) != 14 THEN*

*RAISE\_APPLICATION\_ERROR(-20002, 'O conteudo do CPF/CNPJ deve ter 11  
ou 14 caracteres.');*

*END IF;*

*IF p\_genero != 'M'*

*AND p\_genero != 'F' THEN*

*RAISE\_APPLICATION\_ERROR(-20003, 'O valor de genero deve ser M  
(Masculino) ou F (Feminino).');*

*END IF;*

*IF INSTR(UPPER(p\_cm\_img\_perfil), '.PNG') = 0*

*AND INSTR(UPPER(p\_cm\_img\_perfil), '.JPEG') = 0 THEN*

*RAISE\_APPLICATION\_ERROR(-20004, 'Os formatos de imagem permitos sao:  
png ou jpeg.');*

*END IF;*

*idTelefone := seq\_id\_telefone.NEXTVAL;*

*INSERT INTO tb\_as\_telefone (ID\_TELEFONE, NR\_DDI, NR\_DDD,  
NR\_TELEFONE)*

```
VALUES (idTelefone, p_nr_ddi, p_nr_ddd, p_nr_telefone);
```

```
INSERT INTO tb_as_usuario (ID_USUARIO, ID_TELEFONE, EMAIL, SENHA,  
NOME, CPF_CNPJ, GENERO, DT_NASCIMENTO, CM_IMG_PERFIL,  
DT_REGISTRO)
```

```
VALUES (seq_id_usuario.NEXTVAL, idTelefone, p_email, p_senha, p_nome,  
p_cpf_cnpj, p_genero, TO_DATE(p_dt_nascimento, 'DD-MM-YYYY'),  
p_cm_img_perfil, SYSDATE);
```

```
EXCEPTION
```

```
WHEN DUP_VAL_ON_INDEX THEN
```

```
DBMS_OUTPUT.PUT_LINE('Erro: Valor duplicado em coluna unica.');
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)
```

```
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'PRC_CADASTRO_USUARIO');
```

```
WHEN NO_DATA_FOUND THEN
```

```
DBMS_OUTPUT.PUT_LINE('Erro: Chave estrangeira nao encontrada.');
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```



```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)  
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'PRC_CADASTRO_USUARIO');
```

```
WHEN VALUE_ERROR THEN
```

```
DBMS_OUTPUT.PUT_LINE('Erro: Violou a restricao de tamanho.');
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)  
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'PRC_CADASTRO_USUARIO');
```

```
WHEN OTHERS THEN
```

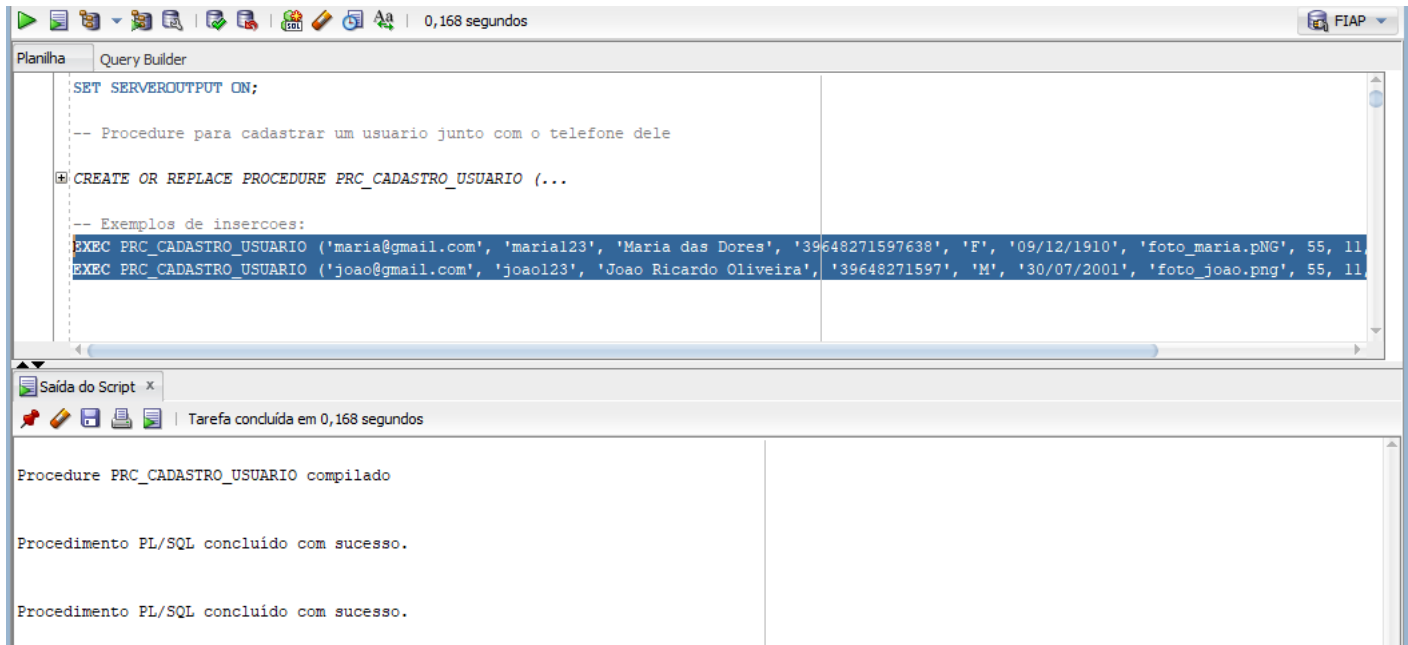
```
DBMS_OUTPUT.PUT_LINE('Ocorreu um erro: ' || SQLERRM);
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)  
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'PRC_CADASTRO_USUARIO');
```

END;



## 9.2 - Procedure para cadastrar um local

```
CREATE OR REPLACE PROCEDURE PRC_CADASTRO_LOCAL (  
    p_id_usuario IN NUMBER,  
    p_nm_pais IN VARCHAR2,  
    p_sg_estado IN VARCHAR2,  
    p_nm_municipio IN VARCHAR2,  
    p_nm_bairro IN VARCHAR2,  
    p_nm_logradouro IN VARCHAR2,  
    p_nr_logradouro IN VARCHAR2,  
    p_nr_cep IN VARCHAR2,  
    p_ds_complemento IN VARCHAR2  
) IS
```

*v\_verificaUsuario* NUMBER;

*v\_codigo\_erro* NUMBER;

*v\_mensagem\_erro* VARCHAR2(250);

*BEGIN*

*SELECT* COUNT(\*)

*INTO* *v\_verificaUsuario*

*FROM* *tb\_as\_usuario*

*WHERE* *ID\_USUARIO* = *p\_id\_usuario*;

*IF* *v\_verificaUsuario* = 0 *THEN*

*RAISE\_APPLICATION\_ERROR*(-20005, 'Usuario nao encontrado.');

*END IF*;

*INSERT INTO* *tb\_as\_local* (*ID\_LOCAL*, *ID\_USUARIO*, *NM\_PAIS*, *SG\_ESTADO*,  
*NM\_MUNICIPIO*, *NM\_BAIRRO*, *NM\_LOGRADOURO*, *NR\_LOGRADOURO*,  
*NR\_CEP*, *DS\_COMPLEMENTO*)

*VALUES* (*seq\_id\_local.NEXTVAL*, *p\_id\_usuario*, *p\_nm\_pais*, *p\_sg\_estado*,  
*p\_nm\_municipio*, *p\_nm\_bairro*, *p\_nm\_logradouro*, *p\_nr\_logradouro*, *p\_nr\_cep*,  
*p\_ds\_complemento*);

*EXCEPTION*

*WHEN DUP\_VAL\_ON\_INDEX THEN*

*DBMS\_OUTPUT.PUT\_LINE('Erro: Valor duplicado em coluna unica.');*

*v\_codigo\_erro := SQLCODE;*

*v\_mensagem\_erro := SQLERRM;*

*INSERT INTO TB\_AS\_ERRO (ID\_ERRO, CD\_ERRO, NM\_ERRO,  
DT\_REGISTRO, USUARIO, PROCEDIMENTO)*

*VALUES (seq\_id\_erro.NEXTVAL, v\_codigo\_erro, v\_mensagem\_erro,  
SYSDATE, USER, 'PRC\_CADASTRO\_LOCAL');*

*WHEN NO\_DATA\_FOUND THEN*

*DBMS\_OUTPUT.PUT\_LINE('Erro: Chave estrangeira nao encontrada.');*

*v\_codigo\_erro := SQLCODE;*

*v\_mensagem\_erro := SQLERRM;*

*INSERT INTO TB\_AS\_ERRO (ID\_ERRO, CD\_ERRO, NM\_ERRO,  
DT\_REGISTRO, USUARIO, PROCEDIMENTO)*

*VALUES (seq\_id\_erro.NEXTVAL, v\_codigo\_erro, v\_mensagem\_erro,  
SYSDATE, USER, 'PRC\_CADASTRO\_LOCAL');*

*WHEN VALUE\_ERROR THEN*

*DBMS\_OUTPUT.PUT\_LINE('Erro: Violou a restricao de tamanho.');*

*v\_codigo\_erro := SQLCODE;*

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)
```

```
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'PRC_CADASTRO_LOCAL');
```

```
WHEN OTHERS THEN
```

```
DBMS_OUTPUT.PUT_LINE('Ocorreu um erro: ' || SQLERRM);
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)
```

```
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'PRC_CADASTRO_LOCAL');
```

```
END;
```

The screenshot displays a SQL development environment with two main windows. The top window, titled 'Planilha Query Builder', contains a PL/SQL procedure definition for 'PRC\_CADASTRO\_LOCAL'. The procedure includes comments in Portuguese, a 'CREATE OR REPLACE' statement, and two example execution calls using the 'EXEC' command. The bottom window, titled 'Saída do Script', shows the execution results, indicating that the procedure was compiled successfully and executed without errors.

```
-- Procedure para cadastrar um local  
CREATE OR REPLACE PROCEDURE PRC_CADASTRO_LOCAL (...  
-- Exemplo  
EXEC PRC_CADASTRO_LOCAL(5, 'Brasil', 'RJ', 'Rio de Janeiro', 'Barra da Tijuca', 'Rua das rosas', '89', '3612597008', null);  
EXEC PRC_CADASTRO_LOCAL(5, 'Brasil', 'RJ', 'Rio de Janeiro', 'Barra da Tijuca', 'Rua das rosas', '89', '36125978', null);
```

Tarefa concluída em 0,113 segundos

Procedure PRC\_CADASTRO\_LOCAL compilado

Procedimento PL/SQL concluído com sucesso.

Procedimento PL/SQL concluído com sucesso.

**9.3 - Procedure para cadastrar uma analise**

```
CREATE OR REPLACE PROCEDURE PRC_CADASTRO_ANALISE (
```

```
    p_id_local IN NUMBER,
```

```
    p_id_status IN NUMBER,
```

```
    p_id_planta IN NUMBER,
```

```
    p_cm_imagem IN VARCHAR2,
```

```
    p_ds_problema IN VARCHAR2,
```

```
    p_ds_solucão IN VARCHAR2
```

```
) IS
```

```
    v_verificaLocal NUMBER;
```

```
    v_verificaStatus NUMBER;
```

```
    v_verificaPlanta NUMBER;
```

```
    v_codigo_erro NUMBER;
```

```
    v_mensagem_erro VARCHAR2(250);
```

```
BEGIN
```

```
    SELECT COUNT(*)
```

```
    INTO v_verificaLocal
```

```
    FROM tb_as_local
```

```
    WHERE ID_LOCAL = p_id_local;
```

```
    IF v_verificaLocal = 0 THEN
```

```
        RAISE_APPLICATION_ERROR(-20006, 'Local nao encontrado.');
```

```
    END IF;
```

```
SELECT COUNT(*)
  INTO v_verificaStatus
  FROM tb_as_status
 WHERE ID_STATUS = p_id_status;
```

```
IF v_verificaStatus = 0 THEN
    RAISE_APPLICATION_ERROR(-20007, 'Status nao encontrado.');
```

```
END IF;
```

```
SELECT COUNT(*)
  INTO v_verificaPlanta
  FROM tb_as_planta
 WHERE ID_PLANTA = p_id_planta;
```

```
IF v_verificaPlanta = 0 THEN
    RAISE_APPLICATION_ERROR(-20008, 'Planta nao encontrado.');
```

```
END IF;
```

```
IF INSTR(UPPER(p_cm_imagem), '.PNG') = 0
AND INSTR(UPPER(p_cm_imagem), '.JPEG') = 0 THEN
```

```
RAISE_APPLICATION_ERROR(-20009, 'Os formatos de imagem permitidos são:
png ou jpeg.');
```

```
END IF;
```

```
INSERT INTO tb_as_analise (ID_ANALISE, ID_LOCAL, ID_STATUS, ID_PLANTA,
CM_IMAGEM, DS_PROBLEMA, DS_SOLUCAO, DT_REGISTRO)
```

```
VALUES (seq_id_analise.NEXTVAL, p_id_local, p_id_status, p_id_planta,
p_cm_imagem, p_ds_problema,
p_ds_solucao, SYSDATE);
```

```
EXCEPTION
```

```
WHEN DUP_VAL_ON_INDEX THEN
```

```
DBMS_OUTPUT.PUT_LINE('Erro: Valor duplicado em coluna unica.');
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,
DT_REGISTRO, USUARIO, PROCEDIMENTO)
```

```
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,
SYSDATE, USER, 'PRC_CADASTRO_ANALISE');
```

```
WHEN NO_DATA_FOUND THEN
```



```
DBMS_OUTPUT.PUT_LINE('Erro: Chave estrangeira nao encontrada.');
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)
```

```
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'PRC_CADASTRO_ANALISE');
```

```
WHEN VALUE_ERROR THEN
```

```
DBMS_OUTPUT.PUT_LINE('Erro: Violou a restricao de tamanho.');
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)
```

```
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'PRC_CADASTRO_ANALISE');
```

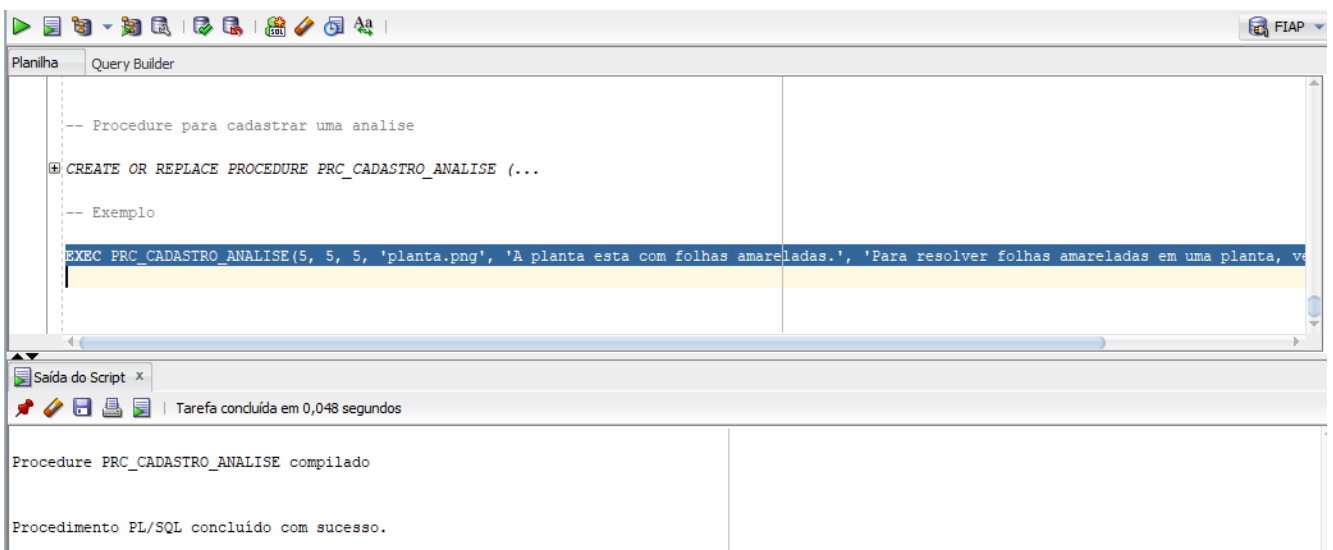
```
WHEN OTHERS THEN
```

```
DBMS_OUTPUT.PUT_LINE('Ocorreu um erro: ' || SQLERRM);
```

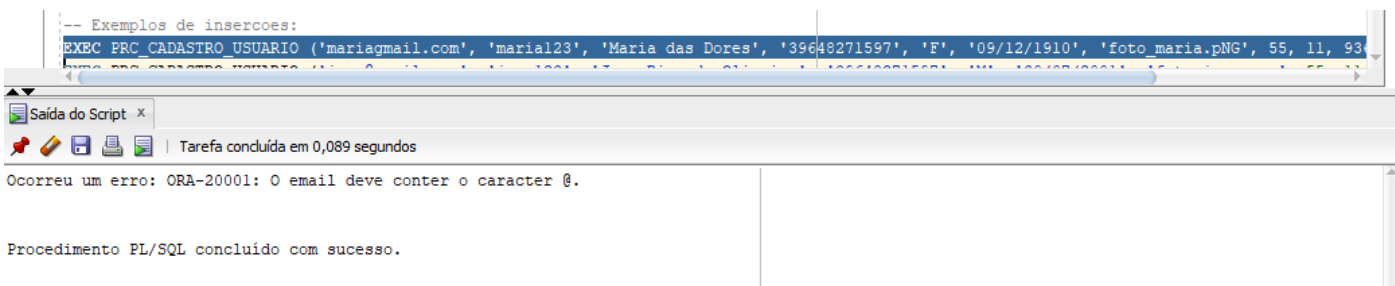
```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)  
  
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'PRC_CADASTRO_ANALISE');  
  
END;
```



## 9.4 – Exemplo de exception



## 10 – Functions

**10.1 – Função para retornar o percentual de análises com o valor do parametro passado referente ao total de análises.**

```
CREATE OR REPLACE FUNCTION FNC_PORCENT_ANALISES(  
    p_status IN VARCHAR  
)  
RETURN NUMBER  
IS  
  
    v_total NUMBER;  
    v_porStatus NUMBER;  
    v_resultado NUMBER;  
  
    v_codigo_erro NUMBER;  
    v_mensagem_erro VARCHAR2(250);  
  
BEGIN  
  
    IF UPPER(p_status) IN ('MORTA', 'EM PROGRESSO', 'EM TRATAMENTO',  
        'ANALISANDO', 'CURADA') THEN  
  
        SELECT COUNT(*)  
        INTO v_total  
        FROM TB_AS_ANALISE;
```

```
SELECT COUNT(*)
```

```
INTO v_porStatus
```

```
FROM TB_AS_ANALISE a
```

```
INNER JOIN TB_AS_STATUS s ON (a.id_status = s.id_status)
```

```
WHERE UPPER(s.DS_STATUS) = UPPER(p_status);
```

```
v_resultado := (v_porStatus / v_total) * 100;
```

```
RETURN v_resultado;
```

```
ELSE
```

```
RAISE_APPLICATION_ERROR(-20001, 'O valor deve ser um valor entre esses:  
Morta, Em progresso, Em tratamento, Analisando e Curada');
```

```
end if;
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN
```

```
DBMS_OUTPUT.PUT_LINE('Nenhuma solucao encontrada para o problema: ' ||  
p_status);
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)
```

```
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'FNC_PORCENT_ANALISES');
```

```
RETURN 0;
```

```
WHEN OTHERS THEN
```

```
DBMS_OUTPUT.PUT_LINE('Ocorreu um erro: ' || SQLERRM);
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)
```

```
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'FNC_PORCENT_ANALISES');
```

```
RETURN NULL;
```

```
END;
```

The screenshot shows a SQL Query Builder window with a tab labeled 'Query Builder'. The main area contains a PL/SQL function definition for `FNC_PORCENT_ANALISES`. The function is a demonstration that takes a status as input and returns a percentage. The output window, titled 'Saída do Script', shows the function was compiled successfully and the output message: 'A porcentagem de analises com o status ANALISANDO e: 8,33%'. The overall procedure completed successfully in 0.066 seconds.

```

CREATE OR REPLACE FUNCTION FNC_PORCENT_ANALISES(...)
-- Demonstracao

DECLARE
    v_resultado NUMBER;
    v_status VARCHAR2(50) := 'ANALISANDO';

BEGIN
    v_resultado := FNC_PORCENT_ANALISES(v_status);
    DBMS_OUTPUT.PUT_LINE('A porcentagem de analises com o status ' || v_status || ' e: ' || ROUND(v_resultado, 2) || '%');

END;
```

Saída do Script x

Tarefa concluída em 0,066 segundos

Function FNC\_PORCENT\_ANALISES compilado

A porcentagem de analises com o status ANALISANDO e: 8,33%

Procedimento PL/SQL concluído com sucesso.

**10.2 – Funcao para retornar uma solucao para o problema passado como parametro com base em palavra ou pequenas frases chaves.**

```

CREATE OR REPLACE FUNCTION FNC_RETORNA_SOLUCAO(
    p_problema IN VARCHAR2
)
RETURN VARCHAR2
IS
```

```
    v_retorno VARCHAR2(500);
```

```
    v_codigo_erro NUMBER;
```

```
    v_mensagem_erro VARCHAR2(250);
```

```
BEGIN
```

```
SELECT DS_SOLUCAO
INTO v_retorno
FROM TB_AS_ANALISE
WHERE UPPER(DS_PROBLEMA) LIKE '%' || UPPER(p_problema) || '%'
AND DT_REGISTRO = (SELECT MAX(DT_REGISTRO) FROM TB_AS_ANALISE)
AND ROWNUM = 1;

RETURN v_retorno;

EXCEPTION

WHEN NO_DATA_FOUND THEN

    DBMS_OUTPUT.PUT_LINE('Nenhuma solucao encontrada para o problema: ' ||
p_problema);

    v_codigo_erro := SQLCODE;
    v_mensagem_erro := SQLERRM;

    INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,
DT_REGISTRO, USUARIO, PROCEDIMENTO)
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,
SYSDATE, USER, 'FNC_RETORNA_SOLUCAO');

RETURN NULL;
```

*WHEN OTHERS THEN*

*DBMS\_OUTPUT.PUT\_LINE('Ocorreu um erro: ' || SQLERRM);*

*v\_codigo\_erro := SQLCODE;*

*v\_mensagem\_erro := SQLERRM;*

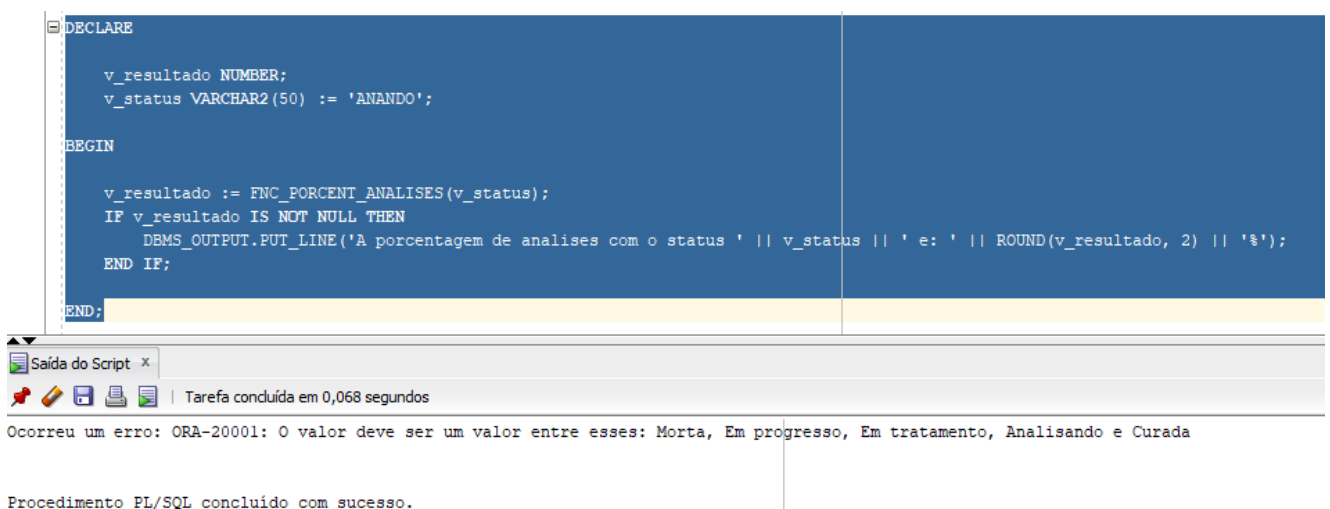
*INSERT INTO TB\_AS\_ERRO (ID\_ERRO, CD\_ERRO, NM\_ERRO,  
DT\_REGISTRO, USUARIO, PROCEDIMENTO)*

*VALUES (seq\_id\_erro.NEXTVAL, v\_codigo\_erro, v\_mensagem\_erro,  
SYSDATE, USER, 'FNC\_RETORNA\_SOLUCAO');*

*RETURN NULL;*

*END;*

### 10.3 – Exemplo de exception



```
DECLARE
    v_resultado NUMBER;
    v_status VARCHAR2(50) := 'ANANDO';

BEGIN
    v_resultado := FNC_PORCENT_ANALISES(v_status);
    IF v_resultado IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('A porcentagem de analises com o status ' || v_status || ' e: ' || ROUND(v_resultado, 2) || '%');
    END IF;
END;
```

Saída do Script x

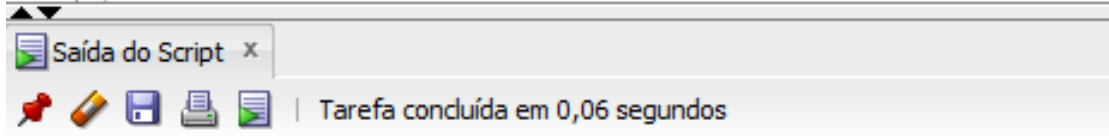
Tarefa concluída em 0,068 segundos

Ocorreu um erro: ORA-20001: O valor deve ser um valor entre esses: Morta, Em progresso, Em tratamento, Analisando e Curada

Procedimento PL/SQL concluído com sucesso.



```
+ CREATE OR REPLACE FUNCTION FNC_RETORNA_SOLUCAO(...  
-- Demonstracao  
- DECLARE  
    v_resultado VARCHAR2(500);  
  
BEGIN  
    v_resultado := FNC_RETORNA_SOLUCAO('folha');  
    DBMS_OUTPUT.PUT_LINE(v_resultado);  
END;
```



Function FNC\_RETORNA\_SOLUCAO compilado

Para resolver folhas amareladas em uma planta, verifique a rega.

Procedimento PL/SQL concluído com sucesso.

## 11 – Triggers (Uma trigger para cada principal tabela do banco)

```
SET SERVEROUTPUT ON;
```

```
-- Uma trigger para as principais tabelas do banco
```

```
CREATE OR REPLACE TRIGGER TRG_AUDITORIA_ANALISE
```

```
AFTER INSERT OR UPDATE OR DELETE ON tb_as_analise
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    v_comando VARCHAR2(50);
```

```
    v_codigo_erro NUMBER;
```

```
    v_mensagem_erro VARCHAR2(250);
```

```
BEGIN
```

```
    IF INSERTING THEN
```

```
        v_comando := 'INSERT';
```

```
    ELSIF UPDATING THEN
```

```
        v_comando := 'UPDATE';
```

```
    ELSIF DELETING THEN
```

```
        v_comando := 'DELETE';
```

```
    END IF;
```

```
    INSERT INTO tb_as_auditoria (ID_REGISTRO, NM_USUARIO, DT_HORA,  
    ACAO_REALIZADA, TABELA_AFETADA)
```

```
VALUES (seq_id_registro.NEXTVAL, USER, SYSDATE, v_comando,  
'TB_AS_ANALISE');
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
DBMS_OUTPUT.PUT_LINE('Ocorreu um erro: ' || SQLERRM);
```

```
v_codigo_erro := SQLCODE;
```

```
v_mensagem_erro := SQLERRM;
```

```
INSERT INTO TB_AS_ERRO (ID_ERRO, CD_ERRO, NM_ERRO,  
DT_REGISTRO, USUARIO, PROCEDIMENTO)
```

```
VALUES (seq_id_erro.NEXTVAL, v_codigo_erro, v_mensagem_erro,  
SYSDATE, USER, 'TRG_AUDITORIA_ANALISE');
```

```
END;
```

```
CREATE OR REPLACE TRIGGER TRG_AUDITORIA_LOCAL
```

```
AFTER INSERT OR UPDATE OR DELETE ON tb_as_local
```

```
FOR EACH ROW
```

```
DECLARE
```

```
v_comando VARCHAR2(50);
```

```
v_codigo_erro NUMBER;
```

```
v_mensagem_erro VARCHAR2(250);
```

```
BEGIN
```

*IF INSERTING THEN*

*v\_comando := 'INSERT';*

*ELSIF UPDATING THEN*

*v\_comando := 'UPDATE';*

*ELSIF DELETING THEN*

*v\_comando := 'DELETE';*

*END IF;*

*INSERT INTO tb\_as\_auditoria (ID\_REGISTRO, NM\_USUARIO, DT\_HORA,  
ACAO\_REALIZADA, TABELA\_AFETADA)*

*VALUES (seq\_id\_registro.NEXTVAL, USER, SYSDATE, v\_comando,  
'TB\_AS\_ANALISE');*

*EXCEPTION*

*WHEN OTHERS THEN*

*DBMS\_OUTPUT.PUT\_LINE('Ocorreu um erro: ' || SQLERRM);*

*v\_codigo\_erro := SQLCODE;*

*v\_mensagem\_erro := SQLERRM;*

*INSERT INTO TB\_AS\_ERRO (ID\_ERRO, CD\_ERRO, NM\_ERRO,  
DT\_REGISTRO, USUARIO, PROCEDIMENTO)*

*VALUES (seq\_id\_erro.NEXTVAL, v\_codigo\_erro, v\_mensagem\_erro,  
SYSDATE, USER, 'TRG\_AUDITORIA\_LOCAL');*

END;

CREATE OR REPLACE TRIGGER TRG\_AUDITORIA\_USUARIO

AFTER INSERT OR UPDATE OR DELETE ON tb\_as\_usuario

FOR EACH ROW

DECLARE

v\_comando VARCHAR2(50);

v\_codigo\_erro NUMBER;

v\_mensagem\_erro VARCHAR2(250);

BEGIN

IF INSERTING THEN

v\_comando := 'INSERT';

ELSIF UPDATING THEN

v\_comando := 'UPDATE';

ELSIF DELETING THEN

v\_comando := 'DELETE';

END IF;

INSERT INTO tb\_as\_auditoria (ID\_REGISTRO, NM\_USUARIO, DT\_HORA,  
ACAO\_REALIZADA, TABELA\_AFETADA)

VALUES (seq\_id\_registro.NEXTVAL, USER, SYSDATE, v\_comando,  
'TB\_AS\_ANALISE');

*EXCEPTION*

*WHEN OTHERS THEN*

*DBMS\_OUTPUT.PUT\_LINE('Ocorreu um erro: ' || SQLERRM);*

*v\_codigo\_erro := SQLCODE;*

*v\_mensagem\_erro := SQLERRM;*

*INSERT INTO TB\_AS\_ERRO (ID\_ERRO, CD\_ERRO, NM\_ERRO,  
DT\_REGISTRO, USUARIO, PROCEDIMENTO)*

*VALUES (seq\_id\_erro.NEXTVAL, v\_codigo\_erro, v\_mensagem\_erro,  
SYSDATE, USER, 'TRG\_AUDITORIA\_LOCAL');*

*END;*