Projeto NeoMoto Data Base

Participantes:

Adel Mouhaidly RM557705

Afonso Correia Pereira RM557863

Tiago Augusto Desiderato RM558485

Faculdade de Informática e Administração Paulista (FIAP)

1. DEMONSTRAÇÃO DAS FUNÇÕES

1.1 Teste Função Senha

```
== Teste Funcao Senha ==
Senha "Abc123!@" => OK
Senha "abcdef"  => Senha inválida: mínimo 8
```

1.2 Execução do Trigger TR_AUDITORIA_RESERVA

```
USUARIO_BD           OPERACAO   DATA_OPERACAO
-------------------- ---------- -------------------
VALORES_OLD
--------------------------------------------------------------------------------------------------------------
VALORES_NEW
--------------------------------------------------------------------------------------------------------------
RM557863             UPDATE     2025-09-25 17:14:04
{"id_reserva":1,"data_reserva":"2024-06-10","data_devolucao":"2024-06-12","id_usuario":1,"id_moto":1}
{"id_reserva":1,"data_reserva":"2024-06-10","data_devolucao":"2024-06-13","id_usuario":1,"id_moto":1}

RM557863             DELETE     2025-09-25 17:14:04
{"id_reserva":5,"data_reserva":"2024-08-10","data_devolucao":"2024-08-12","id_usuario":5,"id_moto":5}


RM557863             INSERT     2025-09-25 17:14:04

{"id_reserva":6,"data_reserva":"2025-09-25","data_devolucao":"2025-09-27","id_usuario":1,"id_moto":6}
```

1.3 Função fn_motos_em_json - Execução Normal

```
================================================================================
1. DEMONSTRAÇÃO DAS FUNÇÕES
================================================================================

*** 1.1 FUNÇÃO fn_motos_em_json - EXECUÇÃO NORMAL ***


JSON_MOTOS
--------------------------------------------------------------------------------
[{"id_moto":1,"placa":"ABC1A23","modelo":"CG 160","ano":2022,"status":"Disponive
```

1.4 Função fn_valida_senha - Execução Normal

```
*** 1.2 FUNÇÃO fn_valida_senha - EXECUÇÃO NORMAL ***


SENHA_FORTE
----------------------------------------------------------------
OK


SENHA_FRACA
----------------------------------------------------------------
Senha fraca


SENHA_CURTA
----------------------------------------------------------------
Senha inválida: mínimo 8


SENHA_NULA
----------------------------------------------------------------
Senha inválida: vazia
```

1.5 Forçando exceções nas funções

```
*** 1.3 FORÇANDO EXCEÇÕES NAS FUNÇÕES ***

=== Testando fn_motos_em_json com erro (tabela inexistente) ===
EXCEÇÃO CAPTURADA na função: ORA-06508: PL/SQL: não foi localizada a unidade de programa que está sendo chamada
Tabela restaurada com sucesso.


Procedimento PL/SQL concluído com sucesso.
```

2. DEMONSTRAÇÃO DOS PROCEDIMENTOS

2.1 Procedimento pr_listar_motos_json - Execução Normaaal

```
*** 2.1 PROCEDIMENTO pr_listar_motos_json - EXECUÇÃO NORMAL ***

JSON de motos e localizacao:
[{"id_moto":1,"placa":"ABC1A23","modelo":"CG 160","ano":2022,
"status":"Disponivel","cidade":"São Paulo","estado":"SP"},{"id_moto":2,
"placa":"DEF4B56","modelo":"NMAX 160","ano":2023,"status":"Disponivel",
"cidade":"São Paulo","estado":"SP"},{"id_moto":3,"placa":"GHI7C89",
"modelo":"PCX 160","ano":2021,"status":"Manutencao","cidade":"Rio de Janeiro",
"estado":"RJ"},{"id_moto":4,"placa":"JKL0D12","modelo":"Fazer 250","ano":2020,
"status":"Disponivel","cidade":"Rio de Janeiro","estado":"RJ"},{"id_moto":5,
"placa":"MNO3E45","modelo":"XTZ 250","ano":2022,"status":"Disponivel",
"cidade":"Belo Horizonte","estado":"MG"},{"id_moto":6,"placa":"PQR6F78",
"modelo":"CB 300","ano":2024,"status":"Disponivel","cidade":"Curitiba",
"estado":"PR"},{"id_moto":7,"placa":"STU9G01","modelo":"ADV 160","ano":2023,
"status":"Disponivel","cidade":"Brasília","estado":"DF"}]


Procedimento PL/SQL concluído com sucesso.
```

2.2 Procedimento pr_saldos_manutencao_relatorio - Execução Normal

```
*** 2.2 PROCEDIMENTO pr_saldos_manutencao_relatorio - EXECUÇÃO NORMAL ***

Agencia   Conta   Saldo
----------------------------
1         1            370.00
1         2            830.00
Sub Total             1200.00
2         3            470.00
2         4            600.00
Sub Total             1070.00
3         5            350.00
Sub Total              350.00
4         6            220.00
Sub Total              220.00
5         7            420.00
Sub Total              420.00
Total Gera            3260.00


Procedimento PL/SQL concluído com sucesso.
```

2.3 Forçando exceções nos procedimentos

```
*** 2.3 FORÇANDO EXCEÇÕES NOS PROCEDIMENTOS ***

=== Testando pr_listar_motos_json com erro ===
Erro no teste do procedimento: ORA-06508: PL/SQL: não foi localizada a unidade de programa que está sendo chamada


Procedimento PL/SQL concluído com sucesso.

=== Testando pr_saldos_manutencao_relatorio com erro ===
Erro no teste do procedimento: ORA-06508: PL/SQL: não foi localizada a unidade de programa que está sendo chamada


Procedimento PL/SQL concluído com sucesso.
```

3. DEMONSTRAÇÃO DO TRIGGER DE AUDITORIA

3.1 Trigger tr_auditoria_reserva - Execução Normal

```
========================================================================
3. DEMONSTRAÇÃO DO TRIGGER DE AUDITORIA
========================================================================

*** 3.1 TRIGGER TR_AUDITORIA_RESERVA - EXECUÇÃO NORMAL ***


3 linhas excluído.

Commit concluído.
=== Testando INSERT (trigger vai capturar) ===

1 linha inserido.

Commit concluído.
=== Testando UPDATE (trigger vai capturar) ===

1 linha atualizado.

Commit concluído.
=== Testando DELETE (trigger vai capturar) ===

1 linha excluído.

Commit concluído.
```

3.2 Visualizando registros de auditoria

```
*** 3.2 VISUALIZANDO REGISTROS DE AUDITORIA ***


ID_AUDITORIA USUARIO_BD           OPERACAO   DATA_OPERACAO
------------ -------------------- ---------- -------------------
VALORES_OLD_RESUMO
--------------------------------------------------------------------------------
VALORES_NEW_RESUMO
--------------------------------------------------------------------------------
           4 RM557863             INSERT     25/09/2025 17:14:07
NULL
{"id_reserva":10,"data_reserva":"2025-01-15","data...

           5 RM557863             UPDATE     25/09/2025 17:14:07
{"id_reserva":10,"data_reserva":"2025-01-15","data...
{"id_reserva":10,"data_reserva":"2025-01-15","data...

           6 RM557863             DELETE     25/09/2025 17:14:07
{"id_reserva":10,"data_reserva":"2025-01-15","data...
NULL
```

3.3 Forçando erro no trigger

```
*** 3.3 FORÇANDO ERRO NO TRIGGER (mas não vai quebrar a transação) ***

=== Inserindo reserva mesmo com possível erro no trigger ===
Inserção realizada com sucesso - trigger funcionou ou falhou silenciosamente


Procedimento PL/SQL concluído com sucesso.


RESERVAS_INSERIDAS
------------------
                 1


1 linha excluído.

Commit concluído.
```

4. TESTE DE CENÁRIOS EXTREMOS E TRATAMENTO DE EXCEÇÕES

4.1 Testando função com dados extremos

```
*** 4.1 TESTANDO FUNÇÃO COM DADOS EXTREMOS ***


SENHA_MUITO_LONGA
--------------------------------------------
OK


SENHA_CARACTERES_ESPECIAIS
--------------------------------------------
Erro de valor na senha
```

4.2 Testando procedimentos com cenários de exceção

```
*** 4.2 TESTANDO PROCEDIMENTOS COM CENÁRIOS DE EXCEÇÃO ***

=== Testando função com consulta que não retorna dados ===
EXCEÇÃO NO_DATA_FOUND capturada conforme esperado
Função executada normalmente após exceção


Procedimento PL/SQL concluído com sucesso.

=== Testando procedimento com tratamento de exceções internas ===
=== Executando procedimento que trata exceções internamente ===
JSON de motos e localizacao:
[{"id_moto":1,"placa":"ABC1A23","modelo":"CG 160","ano":2022,
"status":"Disponivel","cidade":"São Paulo","estado":"SP"},{"id_moto":2,
"placa":"DEF4B56","modelo":"NMAX 160","ano":2023,"status":"Disponivel",
"cidade":"São Paulo","estado":"SP"},{"id_moto":3,"placa":"GHI7C89",
"modelo":"PCX 160","ano":2021,"status":"Manutencao","cidade":"Rio de Janeiro",
"estado":"RJ"},{"id_moto":4,"placa":"JKL0D12","modelo":"Fazer 250","ano":2020,
"status":"Disponivel","cidade":"Rio de Janeiro","estado":"RJ"},{"id_moto":5,
"placa":"MNO3E45","modelo":"XTZ 250","ano":2022,"status":"Disponivel",
"cidade":"Belo Horizonte","estado":"MG"},{"id_moto":6,"placa":"PQR6F78",
"modelo":"CB 300","ano":2024,"status":"Disponivel","cidade":"Curitiba",
"estado":"PR"},{"id_moto":7,"placa":"STU9G01","modelo":"ADV 160","ano":2023,
"status":"Disponivel","cidade":"Brasília","estado":"DF"}]
Procedimento executado com sucesso - exceções tratadas internamente


Procedimento PL/SQL concluído com sucesso.
```

5. TESTES DE PROCEDIMENTO

1. Teste Procedimento 1


```
== Teste Procedimento 1 ==
JSON de motos e localizacao:
[{"id_moto":1,"placa":"ABC1A23","modelo":"CG 160","ano":2022,
"status":"Disponivel","cidade":"São Paulo","estado":"SP"},{"id_moto":2,
"placa":"DEF4B56","modelo":"NMAX 160","ano":2023,"status":"Disponivel",
"cidade":"São Paulo","estado":"SP"},{"id_moto":3,"placa":"GHI7C89",
"modelo":"PCX 160","ano":2021,"status":"Manutencao","cidade":"Rio de Janeiro",
"estado":"RJ"},{"id_moto":4,"placa":"JKL0D12","modelo":"Fazer 250","ano":2020,
"status":"Disponivel","cidade":"Rio de Janeiro","estado":"RJ"},{"id_moto":5,
"placa":"MNO3E45","modelo":"XTZ 250","ano":2022,"status":"Disponivel",
"cidade":"Belo Horizonte","estado":"MG"},{"id_moto":6,"placa":"PQR6F78",
"modelo":"CB 300","ano":2024,"status":"Disponivel","cidade":"Curitiba",
"estado":"PR"},{"id_moto":7,"placa":"STU9G01","modelo":"ADV 160","ano":2023,
"status":"Disponivel","cidade":"Brasília","estado":"DF"}]
```


2. Teste Procedimento 2

```
== Teste Procedimento 2 ==
Agencia   Conta   Saldo
-----------------------------
1         1            370.00
1         2            830.00
Sub Total             1200.00
2         3            470.00
2         4            600.00
Sub Total             1070.00
3         5            350.00
Sub Total              350.00
4         6            220.00
Sub Total              220.00
5         7            420.00
Sub Total              420.00
Total Gera            3260.00
```

6. RESUMO DOS TESTES REALIZADOS

```
================================================================================
5. RESUMO DOS TESTES REALIZADOS
================================================================================

*** FUNÇÕES TESTADAS: ***
- fn_motos_em_json: Execução normal e com erro (tabela inexistente)
- fn_valida_senha: Vários cenários (senha forte, fraca, nula, extremos)

*** PROCEDIMENTOS TESTADOS: ***
- pr_listar_motos_json: Execução normal e com erro
- pr_saldos_manutencao_relatorio: Execução normal e com erro

*** TRIGGER TESTADO: ***
- TR_AUDITORIA_RESERVA: INSERT, UPDATE, DELETE e tratamento de exceções

*** EXCEÇÕES DEMONSTRADAS: ***
- Tabelas inexistentes (ORA-00942)
- Dados nulos e inválidos
- Cenários extremos
- Tratamento silencioso de erros no trigger
```

7. CÓDIGO 2 SPRINT

```sql
-- Tabela: usuarios
CREATE TABLE usuarios (
    id_usuario NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    nome VARCHAR2(100) NOT NULL,
    email VARCHAR2(100) UNIQUE NOT NULL,
    senha VARCHAR2(100) NOT NULL,
    tipo_usuario VARCHAR2(20) NOT NULL
        CONSTRAINT chk_tipo_usuario_user CHECK (tipo_usuario IN ('cidadao',
'voluntario', 'orgao_publico')),
    data_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Tabela: tipos_ocorrencia
CREATE TABLE tipos_ocorrencia (
    id_tipo NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    nome VARCHAR2(50) NOT NULL
```

```
);

-- Tabela: ocorrencias
CREATE TABLE ocorrencias (
    id_ocorrencia NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    id_usuario NUMBER REFERENCES usuarios(id_usuario),
    id_tipo NUMBER REFERENCES tipos_ocorrencia(id_tipo),
    descricao VARCHAR2(1000),
    latitude NUMBER(10,7),
    longitude NUMBER(10,7),
    data_ocorrencia TIMESTAMP NOT NULL,
    nivel_risco VARCHAR2(20) NOT NULL
        CONSTRAINT chk_nivel_risco_ocor CHECK (nivel_risco IN ('baixo',
'moderado', 'alto')),
    status VARCHAR2(20) NOT NULL
        CONSTRAINT chk_status_ocor CHECK (status IN ('em andamento',
'resolvido'))
);

-- Tabela: alertas
CREATE TABLE alertas (
    id_alerta NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    titulo VARCHAR2(100) NOT NULL,
    mensagem VARCHAR2(1000) NOT NULL,
    nivel_urgencia VARCHAR2(20) NOT NULL
        CONSTRAINT chk_nivel_urgencia_alert CHECK (nivel_urgencia IN ('baixa',
'media', 'alta')),
    data_emissao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    id_ocorrencia NUMBER REFERENCES ocorrencias(id_ocorrencia),
    fonte VARCHAR2(50) NOT NULL
        CONSTRAINT chk_fonte_alert CHECK (fonte IN ('sistema', 'iot'))
);

-- Tabela: abrigos
CREATE TABLE abrigos (
    id_abrigo NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    nome VARCHAR2(100) NOT NULL,
    endereco VARCHAR2(150),
    capacidade_total NUMBER NOT NULL,
    vagas_disponiveis NUMBER NOT NULL,
    telefone VARCHAR2(20),
    status VARCHAR2(20) NOT NULL
        CONSTRAINT chk_status_abrigo CHECK (status IN ('disponivel', 'lotado',
'inativo'))
);

-- Tabela: checkins_abrigos
```

```sql
CREATE TABLE checkins_abrigos (
    id_checkin NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    id_usuario NUMBER REFERENCES usuarios(id_usuario),
    id_abrigo NUMBER REFERENCES abrigos(id_abrigo),
    data_entrada TIMESTAMP NOT NULL,
    data_saida TIMESTAMP
);

--INSERT
INSERT INTO usuarios (nome, email, senha, tipo_usuario) VALUES
('Jo?o Silva', 'joao.silva@email.com', 'senha123', 'cidadao');
INSERT INTO usuarios (nome, email, senha, tipo_usuario) VALUES
('Maria Santos', 'maria.santos@email.com', 'senha456', 'voluntario');
INSERT INTO usuarios (nome, email, senha, tipo_usuario) VALUES
('Carlos Oliveira', 'carlos.oliveira@gov.br', 'senha789', 'orgao_publico');
INSERT INTO usuarios (nome, email, senha, tipo_usuario) VALUES
('Ana Costa', 'ana.costa@email.com', 'senha321', 'cidadao');
INSERT INTO usuarios (nome, email, senha, tipo_usuario) VALUES
('Pedro Almeida', 'pedro.almeida@email.com', 'senha654', 'voluntario');

INSERT INTO tipos_ocorrencia (nome) VALUES ('Enchente');
INSERT INTO tipos_ocorrencia (nome) VALUES ('Deslizamento');
INSERT INTO tipos_ocorrencia (nome) VALUES ('Inc?ndio');
INSERT INTO tipos_ocorrencia (nome) VALUES ('Acidente de Tr?nsito');
INSERT INTO tipos_ocorrencia (nome) VALUES ('Queda de ?rvore');


INSERT INTO ocorrencias (id_usuario, id_tipo, descricao, latitude, longitude,
data_ocorrencia, nivel_risco, status) VALUES
(1, 1, 'Alagamento na Rua das Flores', -23.5505, -46.6333, SYSDATE-5, 'alto',
'em andamento');
INSERT INTO ocorrencias (id_usuario, id_tipo, descricao, latitude, longitude,
data_ocorrencia, nivel_risco, status) VALUES
(2, 2, 'Deslizamento de terra no Morro da Vista', -23.5489, -46.6388, SYSDATE-
3, 'alto', 'resolvido');
INSERT INTO ocorrencias (id_usuario, id_tipo, descricao, latitude, longitude,
data_ocorrencia, nivel_risco, status) VALUES
(1, 3, 'Princ?pio de inc?ndio em vegeta??o', -23.5567, -46.6298, SYSDATE-2,
'moderado', 'resolvido');
INSERT INTO ocorrencias (id_usuario, id_tipo, descricao, latitude, longitude,
data_ocorrencia, nivel_risco, status) VALUES
(4, 4, 'Colis?o entre dois ve?culos', -23.5445, -46.6356, SYSDATE-1, 'baixo',
'resolvido');
INSERT INTO ocorrencias (id_usuario, id_tipo, descricao, latitude, longitude,
data_ocorrencia, nivel_risco, status) VALUES
(5, 5, '?rvore ca?da bloqueando via', -23.5523, -46.6311, SYSDATE, 'moderado',
'em andamento');
```

```
INSERT INTO alertas (titulo, mensagem, nivel_urgencia, id_ocorrencia, fonte)
VALUES
('Alerta de Enchente', '□rea de risco na Rua das Flores - Evite circular',
'alta', 1, 'sistema');
INSERT INTO alertas (titulo, mensagem, nivel_urgencia, id_ocorrencia, fonte)
VALUES
('Deslizamento Controlado', 'Situa□□o normalizada no Morro da Vista', 'baixa',
2, 'iot');
INSERT INTO alertas (titulo, mensagem, nivel_urgencia, id_ocorrencia, fonte)
VALUES
('Inc□ndio Extinto', 'Fogo controlado - □rea liberada', 'baixa', 3,
'sistema');
INSERT INTO alertas (titulo, mensagem, nivel_urgencia, id_ocorrencia, fonte)
VALUES
('Via Liberada', 'Acidente removido - tr□nsito normalizado', 'baixa', 4,
'iot');
INSERT INTO alertas (titulo, mensagem, nivel_urgencia, id_ocorrencia, fonte)
VALUES
('Bloqueio de Via', '□rvore sendo removida - desvie pela Rua Alternativa',
'media', 5, 'iot');

INSERT INTO abrigos (nome, endereco, capacidade_total, vagas_disponiveis,
telefone, status) VALUES
('Abrigo Municipal Centro', 'Rua Central, 100', 50, 30, '11-3333-1111',
'disponivel');
INSERT INTO abrigos (nome, endereco, capacidade_total, vagas_disponiveis,
telefone, status) VALUES
('Centro Comunit□rio Norte', 'Avenida Norte, 200', 80, 0, '11-3333-2222',
'lotado');
INSERT INTO abrigos (nome, endereco, capacidade_total, vagas_disponiveis,
telefone, status) VALUES
('Escola Municipal Sul', 'Rua do Sul, 300', 60, 45, '11-3333-3333',
'disponivel');
INSERT INTO abrigos (nome, endereco, capacidade_total, vagas_disponiveis,
telefone, status) VALUES
('Gin□sio Poliesportivo', 'Rua dos Esportes, 400', 100, 80, '11-3333-4444',
'disponivel');
INSERT INTO abrigos (nome, endereco, capacidade_total, vagas_disponiveis,
telefone, status) VALUES
('Centro de Conven□□es', 'Avenida das Conven□□es, 500', 200, 150, '11-3333-
5555', 'disponivel');

INSERT INTO checkins_abrigos (id_usuario, id_abrigo, data_entrada) VALUES
(1, 1, SYSDATE-2);
INSERT INTO checkins_abrigos (id_usuario, id_abrigo, data_entrada, data_saida)
VALUES
```

```
(2, 2, SYSDATE-5, SYSDATE-3);
INSERT INTO checkins_abrigos (id_usuario, id_abrigo, data_entrada) VALUES
(4, 3, SYSDATE-1);
INSERT INTO checkins_abrigos (id_usuario, id_abrigo, data_entrada, data_saida)
VALUES
(5, 1, SYSDATE-4, SYSDATE-2);
INSERT INTO checkins_abrigos (id_usuario, id_abrigo, data_entrada) VALUES
(1, 5, SYSDATE);



--UPDATE
UPDATE usuarios SET email = 'joao.silva.novo@email.com' WHERE id_usuario = 1;
UPDATE usuarios SET nome = 'Maria Santos Silva' WHERE id_usuario = 2;
UPDATE usuarios SET tipo_usuario = 'voluntario' WHERE id_usuario = 4;
UPDATE usuarios SET senha = 'novaSenha123' WHERE id_usuario = 3;
UPDATE usuarios SET nome = 'Pedro Almeida Junior' WHERE id_usuario = 5;

UPDATE tipos_ocorrencia SET nome = 'Enchente Urbana' WHERE id_tipo = 1;
UPDATE tipos_ocorrencia SET nome = 'Deslizamento de Terra' WHERE id_tipo = 2;
UPDATE tipos_ocorrencia SET nome = 'Inc□ndio Florestal' WHERE id_tipo = 3;
UPDATE tipos_ocorrencia SET nome = 'Acidente Veicular' WHERE id_tipo = 4;
UPDATE tipos_ocorrencia SET nome = 'Queda de Vegeta□□o' WHERE id_tipo = 5;

UPDATE ocorrencias SET status = 'resolvido' WHERE id_ocorrencia = 1;
UPDATE ocorrencias SET nivel_risco = 'baixo' WHERE id_ocorrencia = 5;
UPDATE ocorrencias SET descricao = 'Alagamento controlado na Rua das Flores'
WHERE id_ocorrencia = 1;
UPDATE ocorrencias SET latitude = -23.5510, longitude = -46.6340 WHERE
id_ocorrencia = 2;
UPDATE ocorrencias SET data_ocorrencia = SYSDATE-1 WHERE id_ocorrencia = 3;

UPDATE alertas SET nivel_urgencia = 'baixa' WHERE id_alerta = 1;
UPDATE alertas SET titulo = 'Situa□□o Controlada' WHERE id_alerta = 2;
UPDATE alertas SET mensagem = '□rea totalmente liberada para circula□□o' WHERE
id_alerta = 3;
UPDATE alertas SET fonte = 'sistema' WHERE id_alerta = 4;
UPDATE alertas SET nivel_urgencia = 'baixa' WHERE id_alerta = 5;

UPDATE abrigos SET vagas_disponiveis = 25 WHERE id_abrigo = 1;
UPDATE abrigos SET status = 'disponivel', vagas_disponiveis = 10 WHERE
id_abrigo = 2;
UPDATE abrigos SET telefone = '11-3333-3334' WHERE id_abrigo = 3;
UPDATE abrigos SET capacidade_total = 120, vagas_disponiveis = 100 WHERE
id_abrigo = 4;
UPDATE abrigos SET endereco = 'Avenida das Conven□□es, 555' WHERE id_abrigo =
5;
```

```sql
--DELETE
DELETE FROM alertas WHERE id_alerta = 4;
DELETE FROM checkins_abrigos WHERE id_checkin = 2;
DELETE FROM ocorrencias WHERE id_ocorrencia = 4;
DELETE FROM ocorrencias WHERE id_tipo = 5;
DELETE FROM ocorrencias WHERE id_usuario = 5;
DELETE FROM tipos_ocorrencia WHERE id_tipo = 5;
DELETE FROM usuarios WHERE id_usuario = 5;


-- FUNCOES PARA RETORNO DE DADOS PROCESSADOS

--Calcular risco medio
CREATE OR REPLACE FUNCTION calcular_risco_medio(p_id_tipo NUMBER)
RETURN NUMBER
IS
    v_risco_medio NUMBER;
BEGIN
    SELECT AVG(CASE
                WHEN nivel_risco = 'baixo' THEN 1
                WHEN nivel_risco = 'moderado' THEN 2
                WHEN nivel_risco = 'alto' THEN 3
               END)
    INTO v_risco_medio
    FROM ocorrencias
    WHERE id_tipo = p_id_tipo;

    RETURN v_risco_medio;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 0;
END;
/

--Total de ocorrencias em andamento
CREATE OR REPLACE FUNCTION total_ocorrencias_em_andamento
RETURN NUMBER
IS
    v_total NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_total
    FROM ocorrencias
    WHERE status = 'em andamento';
```

```
    RETURN v_total;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 0;
END;
/

--Total de ocorrencias resolvidas
CREATE OR REPLACE FUNCTION total_ocorrencias_resolvidas
RETURN NUMBER
IS
    v_total NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_total
    FROM ocorrencias
    WHERE status = 'resolvido';

    RETURN v_total;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 0;
END;
/



-- BLOCOS ANONIMOS COM CONSULTAS COMPLEXAS

SET SERVEROUTPUT ON;

-- Bloco Anonimo 1: Relatario de Ocorrencias por Usuario
DECLARE
    v_total_ocorrencias NUMBER;
    v_nome_usuario VARCHAR2(100);
    v_tipo_usuario VARCHAR2(20);

    CURSOR c_usuarios IS
        SELECT u.id_usuario, u.nome, u.tipo_usuario,
               COUNT(o.id_ocorrencia) as total_ocorrencias
        FROM usuarios u
        LEFT JOIN ocorrencias o ON u.id_usuario = o.id_usuario
        GROUP BY u.id_usuario, u.nome, u.tipo_usuario
        HAVING COUNT(o.id_ocorrencia) > 0
        ORDER BY COUNT(o.id_ocorrencia) DESC;
BEGIN
    DBMS_OUTPUT.PUT_LINE('=== RELAT□RIO DE OCORR□NCIAS POR USU□RIO ===');
```

```
    FOR rec IN c_usuarios LOOP
        IF rec.total_ocorrencias > 1 THEN
            DBMS_OUTPUT.PUT_LINE('Usu□rio: ' || rec.nome ||
                                 ' (' || rec.tipo_usuario || ')' ||
                                 ' - Total: ' || rec.total_ocorrencias || '
ocorr□ncias');
        ELSE
            DBMS_OUTPUT.PUT_LINE('Usu□rio: ' || rec.nome ||
                                 ' (' || rec.tipo_usuario || ')' ||
                                 ' - Total: ' || rec.total_ocorrencias || '
ocorr□ncia');
        END IF;
    END LOOP;
END;
/

-- Bloco An□nimo 2: Analise de Ocupacao de Abrigos
DECLARE
    v_taxa_ocupacao NUMBER;
    v_status_abrigo VARCHAR2(20);
    v_contador NUMBER := 0;

    CURSOR c_abrigos IS
        SELECT a.nome, a.capacidade_total, a.vagas_disponiveis, a.status,
               ROUND(((a.capacidade_total - a.vagas_disponiveis) /
a.capacidade_total) * 100, 2) as taxa_ocupacao
        FROM abrigos a
        WHERE a.status != 'inativo'
        ORDER BY taxa_ocupacao DESC;
BEGIN
    DBMS_OUTPUT.PUT_LINE('=== AN□LISE DE OCUPA□□O DE ABRIGOS ===');

    FOR rec IN c_abrigos LOOP
        v_contador := v_contador + 1;

        IF rec.taxa_ocupacao >= 80 THEN
            v_status_abrigo := 'CR□TICO';
        ELSIF rec.taxa_ocupacao >= 50 THEN
            v_status_abrigo := 'MODERADO';
        ELSE
            v_status_abrigo := 'NORMAL';
        END IF;

        DBMS_OUTPUT.PUT_LINE(v_contador || '. ' || rec.nome ||
```

```
                          ' - Ocupa□□o: ' || rec.taxa_ocupacao || '% (' ||
v_status_abrigo || ')');
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Total de abrigos analisados: ' || v_contador);
END;
/

-- CURSORES EXPLICITOS

-- Cursor para Listar Alertas com Informacoes Detalhadas
DECLARE
    CURSOR c_alertas_detalhados IS
        SELECT a.titulo, a.mensagem, a.nivel_urgencia, a.data_emissao,
               t.nome as tipo_ocorrencia, o.descricao as desc_ocorrencia
        FROM alertas a
        JOIN ocorrencias o ON a.id_ocorrencia = o.id_ocorrencia
        JOIN tipos_ocorrencia t ON o.id_tipo = t.id_tipo
        ORDER BY a.data_emissao DESC;

    v_alerta c_alertas_detalhados%ROWTYPE;
    v_contador NUMBER := 0;  -- Variavel para contar registros
BEGIN
    DBMS_OUTPUT.PUT_LINE('=== RELAT□RIO DETALHADO DE ALERTAS ===');

    OPEN c_alertas_detalhados;

    LOOP
        FETCH c_alertas_detalhados INTO v_alerta;
        EXIT WHEN c_alertas_detalhados%NOTFOUND;

        v_contador := v_contador + 1;  -- Incrementa contador

        DBMS_OUTPUT.PUT_LINE('T□tulo: ' || v_alerta.titulo);
        DBMS_OUTPUT.PUT_LINE('Tipo: ' || v_alerta.tipo_ocorrencia);
        DBMS_OUTPUT.PUT_LINE('Urg□ncia: ' || v_alerta.nivel_urgencia);
        DBMS_OUTPUT.PUT_LINE('Data: ' || TO_CHAR(v_alerta.data_emissao,
'DD/MM/YYYY HH24:MI'));
        DBMS_OUTPUT.PUT_LINE('Mensagem: ' || v_alerta.mensagem);
        DBMS_OUTPUT.PUT_LINE('---');
    END LOOP;

    CLOSE c_alertas_detalhados;

    DBMS_OUTPUT.PUT_LINE('Total de registros processados: ' || v_contador);
END;
```

```
/


--CONSULTAS SQL COMPLEXAS (RELATARIOS)

-- Relat□rio 1: Estatasticas Gerais de Ocorrencias por Tipo
SELECT
    t.nome as tipo_ocorrencia,
    COUNT(o.id_ocorrencia) as total_ocorrencias,
    SUM(CASE WHEN o.status = 'resolvido' THEN 1 ELSE 0 END) as resolvidas,
    SUM(CASE WHEN o.status = 'em andamento' THEN 1 ELSE 0 END) as
em_andamento,
    ROUND(AVG(CASE
              WHEN o.nivel_risco = 'baixo' THEN 1
              WHEN o.nivel_risco = 'moderado' THEN 2
              WHEN o.nivel_risco = 'alto' THEN 3
             END), 2) as risco_medio
FROM tipos_ocorrencia t
LEFT JOIN ocorrencias o ON t.id_tipo = o.id_tipo
GROUP BY t.id_tipo, t.nome
HAVING COUNT(o.id_ocorrencia) > 0
ORDER BY total_ocorrencias DESC;

-- Relat□rio 2: Usuarios Mais Ativos com Detalhes de Participacao
SELECT
    u.nome,
    u.tipo_usuario,
    COUNT(o.id_ocorrencia) as total_ocorrencias_reportadas,
    COUNT(c.id_checkin) as total_checkins_abrigos,
    (COUNT(o.id_ocorrencia) + COUNT(c.id_checkin)) as pontuacao_atividade
FROM usuarios u
LEFT JOIN ocorrencias o ON u.id_usuario = o.id_usuario
LEFT JOIN checkins_abrigos c ON u.id_usuario = c.id_usuario
GROUP BY u.id_usuario, u.nome, u.tipo_usuario
HAVING (COUNT(o.id_ocorrencia) + COUNT(c.id_checkin)) > 0
ORDER BY pontuacao_atividade DESC, total_ocorrencias_reportadas DESC;

-- Relatario 3: Analise de Alertas por Periodo e Urgencia
SELECT
    a.nivel_urgencia,
    COUNT(*) as total_alertas,
    COUNT(CASE WHEN a.fonte = 'manual' THEN 1 END) as alertas_manuais,
    COUNT(CASE WHEN a.fonte = 'iot' THEN 1 END) as alertas_iot,
    MIN(a.data_emissao) as primeiro_alerta,
    MAX(a.data_emissao) as ultimo_alerta
FROM alertas a
WHERE a.data_emissao >= SYSDATE - 30
```

```sql
GROUP BY a.nivel_urgencia
ORDER BY
    CASE a.nivel_urgencia
        WHEN 'alta' THEN 1
        WHEN 'media' THEN 2
        WHEN 'baixa' THEN 3
    END;

-- Relatario 4: Eficiencia de Resolucao de Ocorrencias por Tipo
SELECT
    t.nome as tipo_ocorrencia,
    COUNT(*) as total_ocorrencias,
    SUM(CASE WHEN o.status = 'resolvido' THEN 1 ELSE 0 END) as resolvidas,
    ROUND(
        (SUM(CASE WHEN o.status = 'resolvido' THEN 1 ELSE 0 END) / COUNT(*)) *
100,
        2
    ) as taxa_resolucao_pct,
    AVG(
        CASE WHEN o.status = 'resolvido'
        THEN EXTRACT(DAY FROM (SYSDATE - o.data_ocorrencia))
        END
    ) as tempo_medio_resolucao_dias
FROM ocorrencias o
JOIN tipos_ocorrencia t ON o.id_tipo = t.id_tipo
GROUP BY t.id_tipo, t.nome
HAVING COUNT(*) >= 1
ORDER BY taxa_resolucao_pct DESC, tempo_medio_resolucao_dias ASC;

-- Relatario 5: Ranking de Abrigos por Utilizacao e Capacidade
SELECT
    a.nome as nome_abrigo,
    a.capacidade_total,
    a.vagas_disponiveis,
    (a.capacidade_total - a.vagas_disponiveis) as ocupacao_atual,
    ROUND(((a.capacidade_total - a.vagas_disponiveis) / a.capacidade_total) *
100, 2) as taxa_ocupacao_pct,
    COUNT(c.id_checkin) as total_checkins_historico,
    COUNT(CASE WHEN c.data_saida IS NULL THEN 1 END) as hospedes_atuais
FROM abrigos a
LEFT JOIN checkins_abrigos c ON a.id_abrigo = c.id_abrigo
WHERE a.status != 'inativo'
GROUP BY a.id_abrigo, a.nome, a.capacidade_total, a.vagas_disponiveis,
a.status
ORDER BY taxa_ocupacao_pct DESC, total_checkins_historico DESC;
```

```sql
-- EXECUCAO DAS FUNCOES CRIADAS
-- Teste da funcao de risco medio
SELECT calcular_risco_medio(1) as risco_medio_enchente FROM DUAL;

-- Teste das funcoes de total por status
SELECT total_ocorrencias_em_andamento() as total_em_andamento FROM DUAL;
SELECT total_ocorrencias_resolvidas() as total_resolvidas FROM DUAL;

-- Comparativo dos status
SELECT
    total_ocorrencias_em_andamento() as em_andamento,
    total_ocorrencias_resolvidas() as resolvidas,
    (total_ocorrencias_em_andamento() + total_ocorrencias_resolvidas()) as
total_geral
FROM DUAL;
```