

## Parte 1 – Personalizando acessos com views

### 1. Número de empregados por departamento e localidade

```
CREATE VIEW EmpregadosPorDeptLocalidade AS
SELECT departamento_id, localidade, COUNT(empregado_id) AS numero_empregados
FROM empregados
GROUP BY departamento_id, localidade;
```

### 2. Lista de departamentos e seus gerentes

```
CREATE VIEW DepartamentosGerentes AS
SELECT d.departamento_id, d.nome AS departamento, e.nome AS gerente
FROM departamentos d
JOIN empregados e ON d.gerente_id = e.empregado_id;
```

### 3. Projetos com maior número de empregados

```
CREATE VIEW ProjetosMaisEmpregados AS
SELECT p.projeto_id, p.nome AS projeto, COUNT(ep.empregado_id) AS numero_empregados
FROM projetos p
JOIN empregados_projetos ep ON p.projeto_id = ep.projeto_id
GROUP BY p.projeto_id
ORDER BY numero_empregados DESC;
```

### 4. Lista de projetos, departamentos e gerentes

```
CREATE VIEW ProjetosDeptGerentes AS
SELECT p.projeto_id, p.nome AS projeto, d.nome AS departamento, e.nome AS gerente
FROM projetos p
JOIN departamentos d ON p.departamento_id = d.departamento_id
JOIN empregados e ON d.gerente_id = e.empregado_id;
```

### 5. Quais empregados possuem dependentes e se são gerentes

```
CREATE VIEW EmpregadosComDependentes AS
SELECT e.empregado_id, e.nome, e.eh_gerente, COUNT(d.dependente_id) AS
numero_dependentes
FROM empregados e
LEFT JOIN dependentes d ON e.empregado_id = d.empregado_id
GROUP BY e.empregado_id, e.nome, e.eh_gerente;
```

## Definindo permissões de acesso

-- Criando usuário gerente

```
CREATE USER 'gerente'@'localhost' IDENTIFIED BY 'senha_gerente';
```

```

-- Concedendo permissões ao gerente
GRANT SELECT ON database.EmpregadosPorDeptLocalidade TO 'gerente'@'localhost';
GRANT SELECT ON database.DepartamentosGerentes TO 'gerente'@'localhost';
GRANT SELECT ON database.ProjetosMaisEmpregados TO 'gerente'@'localhost';
GRANT SELECT ON database.ProjetosDeptGerentes TO 'gerente'@'localhost';
GRANT SELECT ON database.EmpregadosComDependentes TO 'gerente'@'localhost';

-- Criando usuário empregado
CREATE USER 'empregado'@'localhost' IDENTIFIED BY 'senha_empregado';

-- Concedendo permissões ao empregado
GRANT SELECT ON database.EmpregadosComDependentes TO 'empregado'@'localhost';

```

## Parte 2 – Criando gatilhos para cenário de e-commerce

### 1. Trigger de remoção: before delete

```

CREATE TRIGGER before_delete_usuario
BEFORE DELETE ON usuarios
FOR EACH ROW
BEGIN
    INSERT INTO usuarios_historico (usuario_id, nome, email, data_criacao, data_remocao)
    VALUES (OLD.usuario_id, OLD.nome, OLD.email, OLD.data_criacao, NOW());
END;

```

### 2. Trigger de atualização: before update

```

CREATE TRIGGER before_update_salario
BEFORE UPDATE ON empregados
FOR EACH ROW
BEGIN
    IF NEW.salario_base <> OLD.salario_base THEN
        INSERT INTO historico_salarios (empregado_id, salario_antigo, salario_novo,
        data_atualizacao)
        VALUES (OLD.empregado_id, OLD.salario_base, NEW.salario_base, NOW());
    END IF;
END;

```