

# DDL - Data Warehouse Orçamentário

## Scripts de Criação de Tabelas

Sistema de Classificação Temática com Índices Padronizados

21 de outubro de 2025

### Sumário

<b>1</b>	<b>Tabelas de Domínio</b>	<b>2</b>
1.1	D_INDICE_ALOCACAO . . . . .	2
1.2	D_ORCAMENTO_TEMATICO . . . . .	2
<b>2</b>	<b>Tabelas Dimensão - Estrutura Orçamentária</b>	<b>3</b>
2.1	D_TEMPO . . . . .	3
2.2	D_FUNCAO . . . . .	3
2.3	D_SUBFUNCAO . . . . .	4
2.4	D_EIXO . . . . .	4
2.5	D_SUBEIXO . . . . .	4
2.6	D_ACAO . . . . .	5
2.7	D_SUBACAO . . . . .	5
2.8	D_UO (Unidade Orçamentária) . . . . .	6
2.9	D_PROGRAMAS_PPAG_ANALITICO . . . . .	6
<b>3</b>	<b>Tabela Fato Central</b>	<b>7</b>
3.1	F_DOTACAO_ORCAMENTARIA . . . . .	7
<b>4</b>	<b>Tabela Associativa - Classificação Temática</b>	<b>9</b>
4.1	F_DOTACAO_ORCAMENTO_TEMATICO . . . . .	9
<b>5</b>	<b>Tabelas de Log e Auditoria</b>	<b>11</b>
5.1	LOG_ERROS_ETL . . . . .	11
5.2	LOG_EXECUCAO_ETL . . . . .	12
<b>6</b>	<b>Views de Apoio</b>	<b>12</b>
6.1	VW_VALIDACAO_INDICES . . . . .	12
6.2	VW_RESUMO_ORCAMENTOS_TEMATICOS . . . . .	13
<b>7</b>	<b>Triggers</b>	<b>14</b>
7.1	Trigger: Calcular Valor Alocado Automaticamente . . . . .	14
7.2	Trigger: Atualizar Data de Alteração . . . . .	14
<b>8</b>	<b>Stored Procedures</b>	<b>15</b>
8.1	Procedure: Validar Soma de Índices . . . . .	15

# 1 Tabelas de Domínio

## 1.1 D\_INDICE\_ALOCACAO

```
-- =====
-- Tabela: D_INDICE_ALOCACAO
-- Indices padronizados: 20, 50, 85, 100
-- =====
CREATE TABLE D_INDICE_ALOCACAO (
    idD_INDICE_ALOCACAO INT PRIMARY KEY AUTO_INCREMENT,
    VALOR_INDICE INT NOT NULL UNIQUE,
    NOME_CLASSIFICACAO VARCHAR(50) NOT NULL,
    DESCRICAO TEXT,
    ORDEM_PRIORIDADE INT,
    STATUS VARCHAR(20) DEFAULT 'ATIVO',
    DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT chk_valor_indice
        CHECK (VALOR_INDICE IN (20, 50, 85, 100)),
    CONSTRAINT chk_status_indice
        CHECK (STATUS IN ('ATIVO', 'INATIVO'))
);

CREATE INDEX idx_indice_valor
    ON D_INDICE_ALOCACAO(VALOR_INDICE);
CREATE INDEX idx_indice_status
    ON D_INDICE_ALOCACAO(STATUS);
```

## 1.2 D\_ORCAMENTO\_TEMATICO

```
-- =====
-- Tabela: D_ORCAMENTO_TEMATICO
-- Cadastro de orcamentos tematicos (OCA, PCD, IDOSO, etc)
-- =====
CREATE TABLE D_ORCAMENTO_TEMATICO (
    idD_ORCAMENTO_TEMATICO INT PRIMARY KEY AUTO_INCREMENT,
    CODIGO_ORCAMENTO VARCHAR(10) NOT NULL UNIQUE,
    NOME_ORCAMENTO VARCHAR(100) NOT NULL,
    SIGLA VARCHAR(10) NOT NULL UNIQUE,
    DESCRICAO TEXT,
    BASE_LEGAL VARCHAR(200),
    PUBLICO_ALVO VARCHAR(100),
    ANO_IMPLANTACAO INT,
    STATUS_ORCAMENTO VARCHAR(20) DEFAULT 'ATIVO',
    EXERCICIO VARCHAR(4),
    OBSERVACOES TEXT,
    DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP,
    DATA_ALTERACAO DATETIME,

    CONSTRAINT chk_status_orcamento
        CHECK (STATUS_ORCAMENTO IN ('ATIVO', 'INATIVO', 'SUSPENSO')),
    CONSTRAINT chk_exercicio_orcamento
```

```

        CHECK (LENGTH(EXERCICIO) = 4)
    );

CREATE INDEX idx_orcamento_codigo
    ON D_ORCAMENTO_TEMATICO(CODIGO_ORCAMENTO);
CREATE INDEX idx_orcamento_sigla
    ON D_ORCAMENTO_TEMATICO(SIGLA);
CREATE INDEX idx_orcamento_status
    ON D_ORCAMENTO_TEMATICO(STATUS_ORCAMENTO);
CREATE INDEX idx_orcamento_exercicio
    ON D_ORCAMENTO_TEMATICO(EXERCICIO);

```

## 2 Tabelas Dimensão - Estrutura Orçamentária

### 2.1 D\_TEMPO

```

-- =====
-- Tabela: D_TEMPO
-- Dimensao temporal (exercicios fiscais)
-- =====
CREATE TABLE D_TEMPO (
    idT_TEMPO INT PRIMARY KEY AUTO_INCREMENT,
    EXERCICIO VARCHAR(4) NOT NULL UNIQUE,
    ANO INT,
    STATUS VARCHAR(20) DEFAULT 'ATIVO',
    DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT chk_exercicio_tempo
        CHECK (LENGTH(EXERCICIO) = 4),
    CONSTRAINT chk_status_tempo
        CHECK (STATUS IN ('ATIVO', 'ENCERRADO', 'PLANEJAMENTO'))
);

CREATE INDEX idx_tempo_exercicio
    ON D_TEMPO(EXERCICIO);
CREATE INDEX idx_tempo_ano
    ON D_TEMPO(ANO);

```

### 2.2 D\_FUNCAO

```

-- =====
-- Tabela: D_FUNCAO
-- Classificacao funcional (Portaria MOG 42/1999)
-- =====
CREATE TABLE D_FUNCAO (
    idD_FUNCAO INT PRIMARY KEY AUTO_INCREMENT,
    CODIGO_FUNCAO VARCHAR(4) NOT NULL UNIQUE,
    NOME_FUNCAO VARCHAR(100) NOT NULL,
    DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT chk_codigo_funcao

```

```

        CHECK (LENGTH(CODIGO_FUNCAO) <= 4)
    );

CREATE INDEX idx_funcao_codigo
    ON D_FUNCAO(CODIGO_FUNCAO);

```

## 2.3 D\_SUBFUNCAO

```

-- =====
-- Tabela: D_SUBFUNCAO
-- Detalhamento da classificacao funcional
-- =====
CREATE TABLE D_SUBFUNCAO (
    idD_SUBFUNCAO INT PRIMARY KEY AUTO_INCREMENT,
    CODIGO_SUBFUNCAO VARCHAR(4) NOT NULL UNIQUE,
    NOME_SUBFUNCAO VARCHAR(100) NOT NULL,
    DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT chk_codigo_subfuncao
        CHECK (LENGTH(CODIGO_SUBFUNCAO) <= 4)
);

CREATE INDEX idx_subfuncao_codigo
    ON D_SUBFUNCAO(CODIGO_SUBFUNCAO);

```

## 2.4 D\_EIXO

```

-- =====
-- Tabela: D_EIXO
-- Eixos estrategicos do planejamento
-- =====
CREATE TABLE D_EIXO (
    idD_EIXO INT PRIMARY KEY AUTO_INCREMENT,
    CODIGO_EIXO VARCHAR(10) NOT NULL UNIQUE,
    NOME_EIXO VARCHAR(100) NOT NULL,
    DESCRICAO TEXT,
    DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP
);

CREATE INDEX idx_eixo_codigo
    ON D_EIXO(CODIGO_EIXO);

```

## 2.5 D\_SUBEIXO

```

-- =====
-- Tabela: D_SUBEIXO
-- Detalhamento dos eixos estrategicos
-- =====
CREATE TABLE D_SUBEIXO (
    idD_SUBEIXO INT PRIMARY KEY AUTO_INCREMENT,
    CODIGO_SUBEIXO VARCHAR(10) NOT NULL UNIQUE,

```

```

        NOME_SUBEIXO VARCHAR(100) NOT NULL,
        CODIGO_EIXO VARCHAR(10),
        DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP
    );

CREATE INDEX idx_subeixo_codigo
    ON D_SUBEIXO(CODIGO_SUBEIXO);
CREATE INDEX idx_subeixo_eixo
    ON D_SUBEIXO(CODIGO_EIXO);

```

## 2.6 D\_ACAO

```

-- =====
-- Tabela: D_ACAO
-- Acoes orcamentarias
-- =====
CREATE TABLE D_ACAO (
    idD_ACAO INT PRIMARY KEY AUTO_INCREMENT,
    CODIGO_ACAO VARCHAR(10) NOT NULL UNIQUE,
    NOME_ACAO VARCHAR(200) NOT NULL,
    TIPO_ACAO VARCHAR(50),
    DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP
);

CREATE INDEX idx_acao_codigo
    ON D_ACAO(CODIGO_ACAO);

```

## 2.7 D\_SUBACAO

```

-- =====
-- Tabela: D_SUBACAO
-- Detalhamento das acoes orcamentarias
-- =====
CREATE TABLE D_SUBACAO (
    idD_SUBACAO INT PRIMARY KEY AUTO_INCREMENT,
    CODIGO_SUBACAO VARCHAR(45) NOT NULL UNIQUE,
    NOME_SUBACAO VARCHAR(200) NOT NULL,
    CODIGO_ACAO VARCHAR(10),
    METAFISICA_2022 INT,
    METAFISICA_2023 INT,
    METAFISICA_2024 INT,
    METAFISICA_2025 INT,
    PRODUTO VARCHAR(100),
    UNIDADEMEDIDA VARCHAR(50),
    OBJETIVO_ORCAMENTARIO_ID INT,
    VALOR_EMPENHADO DECIMAL(15,2),
    VALOR_LIQUIDADO DECIMAL(15,2),
    VALOR_PAGO DECIMAL(15,2),
    DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP
);

CREATE INDEX idx_subacao_codigo

```

```

        ON D_SUBACAO(CODIGO_SUBACAO);
CREATE INDEX idx_subacao_acao
        ON D_SUBACAO(CODIGO_ACAO);

```

## 2.8 D\_UO (Unidade Orçamentária)

```

-- =====
-- Tabela: D_UO
-- Unidades orcamentarias/organizacionais
-- =====
CREATE TABLE D_UO (
    idD_UO INT PRIMARY KEY AUTO_INCREMENT,
    ORGAO_ENTIDADE VARCHAR(100),
    UNIDADE_ORGANIZACIONAL VARCHAR(100),
    SIGLA_UNIDADE_ORGANIZACIONAL VARCHAR(45) NOT NULL UNIQUE,
    UNIDADE_ORGANIZACIONAL_SUPERIOR VARCHAR(100),
    NOME_DO_TITULAR VARCHAR(100),
    ENDERECO VARCHAR(200),
    EMAIL_UNIDADE VARCHAR(100),
    TELEFONE VARCHAR(20),
    DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT chk_email_uo
        CHECK (EMAIL_UNIDADE LIKE '%@%')
);

CREATE INDEX idx_uo_sigla
    ON D_UO(SIGLA_UNIDADE_ORGANIZACIONAL);
CREATE INDEX idx_uo_orgao
    ON D_UO(ORGAO_ENTIDADE);

```

## 2.9 D\_PROGRAMAS\_PPAG\_ANALITICO

```

-- =====
-- Tabela: D_PROGRAMAS_PPAG_ANALITICO
-- Programas do Plano Plurianual detalhados
-- =====
CREATE TABLE D_PROGRAMAS_PPAG_ANALITICO (
    idD_PROGRAMAS_PPAG_ANALITICO INT PRIMARY KEY AUTO_INCREMENT,
    EXERCICIO VARCHAR(4) NOT NULL,
    CODIGO_AREA_DE_RESULTADO VARCHAR(45) UNIQUE,
    NOME_AREA_DE_RESULTADO VARCHAR(200),
    CODIGO_DO_PROGRAMA VARCHAR(45) NOT NULL UNIQUE,
    NOME_DO_PROGRAMA VARCHAR(200) NOT NULL,
    OBJETIVO_DO_PROGRAMA TEXT,
    PUBLICO_ALVO_DO_PROGRAMA VARCHAR(200),
    JUSTIFICATIVA_DO_PROGRAMA TEXT,
    NATUREZA_DO_PROGRAMA VARCHAR(100),
    TIPOLOGIA_DO_PROGRAMA VARCHAR(100),
    CLASSIFICACAO_DE_GOVERNO VARCHAR(100),
    DATAINICIO_PROGRAMA DATE,
    DATAFIM_PROGRAMA DATE,

```

```

UNIDADE_GESTORA VARCHAR(100),
CODIGO_DO_INDICADOR VARCHAR(45),
NOME_DO_INDICADOR VARCHAR(200),
FORMA_DE_CALCULO VARCHAR(200),
UNIDADE_DE_MEDIDA VARCHAR(50),
FONTE_DE_DADOS VARCHAR(200),
VALOR_DE_REFERENCIA VARCHAR(45),
DATA_APURACAO_VALOR_DE_REFERENCIA DATE,
INDICE_ESPERADO_ANO_1 VARCHAR(45),
INDICE_ESPERADO_ANO_2 VARCHAR(45),
INDICE_ESPERADO_ANO_3 VARCHAR(45),
INDICE_ESPERADO_ANO_4 VARCHAR(45),
OBSERVACAO TEXT,
DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP,

CONSTRAINT chk_exercicio_ppag
    CHECK (LENGTH(EXERCICIO) = 4)
);

CREATE UNIQUE INDEX idx_ppag_codigo_programa
    ON D_PROGRAMAS_PPAG_ANALITICO(CODIGO_DO_PROGRAMA);
CREATE UNIQUE INDEX idx_ppag_codigo_area
    ON D_PROGRAMAS_PPAG_ANALITICO(CODIGO_AREA_DE_RESULTADO);
CREATE INDEX idx_ppag_exercicio
    ON D_PROGRAMAS_PPAG_ANALITICO(EXERCICIO);
CREATE INDEX idx_ppag_unidade_gestora
    ON D_PROGRAMAS_PPAG_ANALITICO(UNIDADE_GESTORA);

```

### 3 Tabela Fato Central

#### 3.1 F\_DOTACAO\_ORCAMENTARIA

```

-- =====
-- Tabela: F_DOTACAO_ORCAMENTARIA
-- Tabela fato central - dotacoes orcamentarias
-- =====
CREATE TABLE F_DOTACAO_ORCAMENTARIA (
    idF_DOTACAO_ORCAMENTARIA INT PRIMARY KEY AUTO_INCREMENT,

    -- Chaves Estrangeiras (Dimensoes)
    idD_PROGRAMAS_PPAG INT,
    idD_UO INT,
    idD_FUNCAO INT,
    idD_SUBFUNCAO INT,
    idD_ACAO INT,
    idD_SUBACAO INT,
    idD_EIXO INT,
    idD_SUBEIXO INT,
    idT_TEMPO INT,

    -- Identificadores
    CODIGO_DOTACAO VARCHAR(50) NOT NULL,

```

```

NOME_DOTACAO VARCHAR(200),
EXERCICIO VARCHAR(4) NOT NULL,

-- Metricas
VALOR_DOTACAO DECIMAL(15,2) NOT NULL,
VALOR_EMPENHADO DECIMAL(15,2),
VALOR_LIQUIDADO DECIMAL(15,2),
VALOR_PAGO DECIMAL(15,2),

-- Classificacoes adicionais
NATUREZA_DESPESA VARCHAR(20),
FONTE_RECURSO VARCHAR(20),
MODALIDADE_APLICACAO VARCHAR(20),
ELEMENTO_DESPESA VARCHAR(20),

-- Controle
STATUS VARCHAR(20) DEFAULT 'ATIVA',
DATA_CADASTRO DATETIME DEFAULT CURRENT_TIMESTAMP,
DATA_ALTERACAO DATETIME,

-- Constraints
CONSTRAINT fk_fdot_programas
    FOREIGN KEY (idD_PROGRAMAS_PPAG)
    REFERENCES D_PROGRAMAS_PPAG_ANALITICO(
        idD_PROGRAMAS_PPAG_ANALITICO),
CONSTRAINT fk_fdot_uo
    FOREIGN KEY (idD_UO)
    REFERENCES D_UO(idD_UO),
CONSTRAINT fk_fdot_funcao
    FOREIGN KEY (idD_FUNCAO)
    REFERENCES D_FUNCAO(idD_FUNCAO),
CONSTRAINT fk_fdot_subfuncao
    FOREIGN KEY (idD_SUBFUNCAO)
    REFERENCES D_SUBFUNCAO(idD_SUBFUNCAO),
CONSTRAINT fk_fdot_acao
    FOREIGN KEY (idD_ACAO)
    REFERENCES D_ACAO(idD_ACAO),
CONSTRAINT fk_fdot_subacao
    FOREIGN KEY (idD_SUBACAO)
    REFERENCES D_SUBACAO(idD_SUBACAO),
CONSTRAINT fk_fdot_eixo
    FOREIGN KEY (idD_EIXO)
    REFERENCES D_EIXO(idD_EIXO),
CONSTRAINT fk_fdot_subeixo
    FOREIGN KEY (idD_SUBEIXO)
    REFERENCES D_SUBEIXO(idD_SUBEIXO),
CONSTRAINT fk_fdot_tempo
    FOREIGN KEY (idT_TEMPO)
    REFERENCES D_TEMPO(idT_TEMPO),

CONSTRAINT chk_valor_dotacao
    CHECK (VALOR_DOTACAO >= 0),
CONSTRAINT chk_status_dotacao

```



```

        CHECK (STATUS IN ('ATIVA', 'BLOQUEADA', 'CANCELADA')),
        CONSTRAINT uq_codigo_dotacao_exercicio
        UNIQUE (CODIGO_DOTACAO, EXERCICIO)
    );

-- Indices para performance
CREATE INDEX idx_fdot_codigo
    ON F_DOTACAO_ORCAMENTARIA(CODIGO_DOTACAO);
CREATE INDEX idx_fdot_exercicio
    ON F_DOTACAO_ORCAMENTARIA(EXERCICIO);
CREATE INDEX idx_fdot_programa
    ON F_DOTACAO_ORCAMENTARIA(idD_PROGRAMAS_PPAG);
CREATE INDEX idx_fdot_uo
    ON F_DOTACAO_ORCAMENTARIA(idD_UO);
CREATE INDEX idx_fdot_funcao
    ON F_DOTACAO_ORCAMENTARIA(idD_FUNCAO);
CREATE INDEX idx_fdot_acao
    ON F_DOTACAO_ORCAMENTARIA(idD_ACAO);
CREATE INDEX idx_fdot_tempo
    ON F_DOTACAO_ORCAMENTARIA(idT_TEMPO);
CREATE INDEX idx_fdot_status
    ON F_DOTACAO_ORCAMENTARIA(STATUS);
CREATE INDEX idx_fdot_composto
    ON F_DOTACAO_ORCAMENTARIA(EXERCICIO, STATUS, idD_PROGRAMAS_PPAG);

```

## 4 Tabela Associativa - Classificação Temática

### 4.1 F\_DOTACAO\_ORCAMENTO\_TEMATICO

```

-- =====
-- Tabela: F_DOTACAO_ORCAMENTO_TEMATICO
-- Relacionamento N:N entre dotacoes e orcamentos tematicos
-- Implementa classificacao com indices padronizados (20,50,85,100)
-- =====
CREATE TABLE F_DOTACAO_ORCAMENTO_TEMATICO (
    idF_DOTACAO_ORCAMENTO_TEMATICO INT PRIMARY KEY AUTO_INCREMENT,

    -- Chaves Estrangeiras
    idF_DOTACAO_ORCAMENTARIA INT NOT NULL,
    idD_ORCAMENTO_TEMATICO INT NOT NULL,
    idD_INDICE_ALOCACAO INT NOT NULL,

    -- Metricas
    VALOR_ALOCADO DECIMAL(15,2) NOT NULL,

    -- Metadados
    EXERCICIO VARCHAR(4) NOT NULL,
    JUSTIFICATIVA_ALOCACAO TEXT NOT NULL,
    CRITERIO_ALOCACAO VARCHAR(200),
    META_FISICA VARCHAR(200),
    PUBLICO_ALVO_ESPECIFICO VARCHAR(200),

```

```

-- Controle de Auditoria
DATA_INCLUSAO DATETIME DEFAULT CURRENT_TIMESTAMP,
USUARIO_INCLUSAO VARCHAR(50) NOT NULL,
DATA_ALTERACAO DATETIME,
USUARIO_ALTERACAO VARCHAR(50),
STATUS VARCHAR(20) DEFAULT 'ATIVO',

-- Foreign Keys
CONSTRAINT fk_fdot_tematico_dotacao
    FOREIGN KEY (idF_DOTACAO_ORCAMENTARIA)
    REFERENCES F_DOTACAO_ORCAMENTARIA(idF_DOTACAO_ORCAMENTARIA)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,

CONSTRAINT fk_fdot_tematico_orcamento
    FOREIGN KEY (idD_ORCAMENTO_TEMATICO)
    REFERENCES D_ORCAMENTO_TEMATICO(idD_ORCAMENTO_TEMATICO)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,

CONSTRAINT fk_fdot_tematico_indice
    FOREIGN KEY (idD_INDICE_ALOCACAO)
    REFERENCES D_INDICE_ALOCACAO(idD_INDICE_ALOCACAO)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,

-- Unique Constraint - Uma dotacao nao pode ter mais de uma
-- classificacao no mesmo orcamento tematico no mesmo exercicio
CONSTRAINT uq_dotacao_orcamento_exercicio
    UNIQUE (idF_DOTACAO_ORCAMENTARIA,
            idD_ORCAMENTO_TEMATICO,
            EXERCICIO),

-- Check Constraints
CONSTRAINT chk_valor_alocado_tematico
    CHECK (VALOR_ALOCADO >= 0),
CONSTRAINT chk_status_tematico
    CHECK (STATUS IN ('ATIVO', 'INATIVO', 'CANCELADO')),
CONSTRAINT chk_justificativa_min
    CHECK (LENGTH(JUSTIFICATIVA_ALOCACAO) >= 20),
CONSTRAINT chk_exercicio_tematico
    CHECK (LENGTH(EXERCICIO) = 4)
);

-- Indices para Performance
CREATE INDEX idx_fdot_tema_dotacao
    ON F_DOTACAO_ORCAMENTO_TEMATICO(idF_DOTACAO_ORCAMENTARIA);

CREATE INDEX idx_fdot_tema_orcamento
    ON F_DOTACAO_ORCAMENTO_TEMATICO(idD_ORCAMENTO_TEMATICO);

CREATE INDEX idx_fdot_tema_indice
    ON F_DOTACAO_ORCAMENTO_TEMATICO(idD_INDICE_ALOCACAO);

```

```

CREATE INDEX idx_fdot_tema_exercicio
  ON F_DOTACAO_ORCAMENTO_TEMATICO(EXERCICIO);

CREATE INDEX idx_fdot_tema_status
  ON F_DOTACAO_ORCAMENTO_TEMATICO(STATUS);

CREATE INDEX idx_fdot_tema_composto
  ON F_DOTACAO_ORCAMENTO_TEMATICO(
    idF_DOTACAO_ORCAMENTARIA,
    EXERCICIO,
    STATUS
  );

CREATE INDEX idx_fdot_tema_orcamento_exercicio
  ON F_DOTACAO_ORCAMENTO_TEMATICO(
    idD_ORCAMENTO_TEMATICO,
    EXERCICIO,
    STATUS
  );

```

## 5 Tabelas de Log e Auditoria

### 5.1 LOG\_ERROS\_ETL

```

-- =====
-- Tabela: LOG_ERROS_ETL
-- Registro de erros durante processo ETL
-- =====
CREATE TABLE LOG_ERROS_ETL (
  idLOG_ERRO INT PRIMARY KEY AUTO_INCREMENT,
  TABELA_DESTINO VARCHAR(100),
  PIPELINE VARCHAR(100),
  TIPO_ERRO VARCHAR(50),
  DESCRICAO_ERRO TEXT,
  REGISTRO_ERRO TEXT,
  DATA_ERRO DATETIME DEFAULT CURRENT_TIMESTAMP,
  USUARIO VARCHAR(50),
  EXERCICIO VARCHAR(4),
  STATUS_TRATAMENTO VARCHAR(20) DEFAULT 'PENDENTE',

  CONSTRAINT chk_status_tratamento
    CHECK (STATUS_TRATAMENTO IN ('PENDENTE', 'TRATADO', 'IGNORADO'
    ))
);

CREATE INDEX idx_log_tabela
  ON LOG_ERROS_ETL(TABELA_DESTINO);
CREATE INDEX idx_log_data
  ON LOG_ERROS_ETL(DATA_ERRO);
CREATE INDEX idx_log_status
  ON LOG_ERROS_ETL(STATUS_TRATAMENTO);

```

## 5.2 LOG\_EXECUCAO\_ETL

```
-- =====
-- Tabela: LOG_EXECUCAO_ETL
-- Registro de execucoes dos pipelines ETL
-- =====
CREATE TABLE LOG_EXECUCAO_ETL (
    idLOG_EXECUCAO INT PRIMARY KEY AUTO_INCREMENT,
    PIPELINE VARCHAR(100) NOT NULL,
    TABELA_DESTINO VARCHAR(100),
    DATA_INICIO DATETIME NOT NULL,
    DATA_FIM DATETIME,
    STATUS_EXECUCAO VARCHAR(20),
    REGISTROS_PROCESSADOS INT,
    REGISTROS_INSERIDOS INT,
    REGISTROS_ATUALIZADOS INT,
    REGISTROS_ERRO INT,
    TEMPO_EXECUCAO_SEGUNDOS INT,
    MENSAGEM TEXT,
    USUARIO VARCHAR(50),
    EXERCICIO VARCHAR(4),

    CONSTRAINT chk_status_execucao
        CHECK (STATUS_EXECUCAO IN ('EXECUTANDO', 'SUCESSO', 'ERRO', '
            CANCELADO'))
);

CREATE INDEX idx_log_exec_pipeline
    ON LOG_EXECUCAO_ETL(PIPELINE);
CREATE INDEX idx_log_exec_data
    ON LOG_EXECUCAO_ETL(DATA_INICIO);
CREATE INDEX idx_log_exec_status
    ON LOG_EXECUCAO_ETL(STATUS_EXECUCAO);
```

## 6 Views de Apoio

### 6.1 VW\_VALIDACAO\_INDICES

```
-- =====
-- View: VW_VALIDACAO_INDICES
-- Validacao da soma de indices por dotacao
-- =====
CREATE VIEW VW_VALIDACAO_INDICES AS
SELECT
    fdo.idF_DOTACAO_ORCAMENTARIA,
    fdo.CODIGO_DOTACAO,
    fdo.NOME_DOTACAO,
    fdo.VALOR_DOTACAO,
    fdot.EXERCICIO,
    COUNT(DISTINCT fdot.idD_ORCAMENTO_TEMATICO) as QTD_ORCAMENTOS,
    SUM(ia.VALOR_INDICE) as SOMA_INDICES,
    SUM(fdot.VALOR_ALOCADO) as SOMA_VALORES_ALOCADOS,
```

```

STRING_AGG(
    CONCAT(dot.SIGLA, '(' , ia.VALOR_INDICE, ')'),
    ' + ' ORDER BY ia.VALOR_INDICE DESC
) as DETALHE_CLASSIFICACOES,
CASE
    WHEN SUM(ia.VALOR_INDICE) = 100 THEN 'OK_COMPLETO',
    WHEN SUM(ia.VALOR_INDICE) < 100 THEN 'OK_PARCIAL',
    WHEN SUM(ia.VALOR_INDICE) <= 200 THEN 'ATENCAO',
    ELSE 'CRITICO'
END as STATUS_VALIDACAO
FROM F_DOTACAO_ORCAMENTARIA fdo
JOIN F_DOTACAO_ORCAMENTO_TEMATICO fdot
    ON fdo.idF_DOTACAO_ORCAMENTARIA = fdot.idF_DOTACAO_ORCAMENTARIA
JOIN D_INDICE_ALOCACAO ia
    ON fdot.idD_INDICE_ALOCACAO = ia.idD_INDICE_ALOCACAO
JOIN D_ORCAMENTO_TEMATICO dot
    ON fdot.idD_ORCAMENTO_TEMATICO = dot.idD_ORCAMENTO_TEMATICO
WHERE fdot.STATUS = 'ATIVO'
AND fdo.STATUS = 'ATIVA'
GROUP BY
    fdo.idF_DOTACAO_ORCAMENTARIA,
    fdo.CODIGO_DOTACAO,
    fdo.NOME_DOTACAO,
    fdo.VALOR_DOTACAO,
    fdot.EXERCICIO;

```

## 6.2 VW\_RESUMO\_ORCAMENTOS\_TEMATICOS

```

-- =====
-- View: VW_RESUMO_ORCAMENTOS_TEMATICOS
-- Consolidacao por orcamento tematico
-- =====
CREATE VIEW VW_RESUMO_ORCAMENTOS_TEMATICOS AS
SELECT
    fdot.EXERCICIO,
    dot.CODIGO_ORCAMENTO,
    dot.NOME_ORCAMENTO,
    dot.SIGLA,
    COUNT(DISTINCT fdot.idF_DOTACAO_ORCAMENTARIA) as QTD_DOTACOES,
    SUM(fdot.VALOR_ALOCADO) as VALOR_TOTAL,
    AVG(fdot.VALOR_ALOCADO) as VALOR_MEDIO,
    MIN(fdot.VALOR_ALOCADO) as VALOR_MINIMO,
    MAX(fdot.VALOR_ALOCADO) as VALOR_MAXIMO,
    COUNT(CASE WHEN ia.VALOR_INDICE = 100 THEN 1 END) as
        QTD_EXCLUSIVAS,
    COUNT(CASE WHEN ia.VALOR_INDICE = 85 THEN 1 END) as
        QTD_PREDOMINANTES,
    COUNT(CASE WHEN ia.VALOR_INDICE = 50 THEN 1 END) as QTD_MODERADAS,
    COUNT(CASE WHEN ia.VALOR_INDICE = 20 THEN 1 END) as
        QTD_TANGENCIAIS
FROM F_DOTACAO_ORCAMENTO_TEMATICO fdot
JOIN D_ORCAMENTO_TEMATICO dot
    ON fdot.idD_ORCAMENTO_TEMATICO = dot.idD_ORCAMENTO_TEMATICO

```

```

JOIN D_INDICE_ALOCACAO ia
  ON fdot.idD_INDICE_ALOCACAO = ia.idD_INDICE_ALOCACAO
WHERE fdot.STATUS = 'ATIVO'
GROUP BY
  fdot.EXERCICIO,
  dot.CODIGO_ORCAMENTO,
  dot.NOME_ORCAMENTO,
  dot.SIGLA;

```

## 7 Triggers

### 7.1 Trigger: Calcular Valor Alocado Automaticamente

```

-- =====
-- Trigger: trg_calcula_valor_alocado
-- Calcula VALOR_ALOCADO baseado no indice e valor da dotacao
-- =====
DELIMITER $$

CREATE TRIGGER trg_calcula_valor_alocado
BEFORE INSERT ON F_DOTACAO_ORCAMENTO_TEMATICO
FOR EACH ROW
BEGIN
  DECLARE v_valor_dotacao DECIMAL(15,2);
  DECLARE v_indice INT;

  -- Buscar valor da dotacao
  SELECT VALOR_DOTACAO INTO v_valor_dotacao
  FROM F_DOTACAO_ORCAMENTARIA
  WHERE idF_DOTACAO_ORCAMENTARIA = NEW.idF_DOTACAO_ORCAMENTARIA;

  -- Buscar valor do indice
  SELECT VALOR_INDICE INTO v_indice
  FROM D_INDICE_ALOCACAO
  WHERE idD_INDICE_ALOCACAO = NEW.idD_INDICE_ALOCACAO;

  -- Calcular se nao foi informado ou esta zerado
  IF NEW.VALOR_ALOCADO IS NULL OR NEW.VALOR_ALOCADO = 0 THEN
    SET NEW.VALOR_ALOCADO = v_valor_dotacao * (v_indice / 100.0);
  END IF;
END$$

DELIMITER ;

```

### 7.2 Trigger: Atualizar Data de Alteração

```

-- =====
-- Trigger: trg_atualiza_data_alteracao
-- Atualiza DATA_ALTERACAO automaticamente
-- =====
DELIMITER $$

```

```

CREATE TRIGGER trg_atualiza_data_alteracao
BEFORE UPDATE ON F_DOTACAO_ORCAMENTO_TEMATICO
FOR EACH ROW
BEGIN
    SET NEW.DATA_ALTERACAO = NOW();
END$$

DELIMITER ;

```

## 8 Stored Procedures

### 8.1 Procedure: Validar Soma de Índices

```

-- =====
-- Procedure: sp_validar_soma_indices
-- Valida se soma de indices esta consistente
-- =====
DELIMITER $$

CREATE PROCEDURE sp_validar_soma_indices(
    IN p_exercicio VARCHAR(4),
    IN p_limite_critico INT
)
BEGIN
    SELECT
        fdo.CODIGO_DOTACAO,
        fdo.NOME_DOTACAO,
        fdo.VALOR_DOTACAO,
        SUM(ia.VALOR_INDICE) as SOMA_INDICES,
        COUNT(*) as QTD_CLASSIFICACOES,
        STRING_AGG(dot.SIGLA, ', ') as ORCAMENTOS,
        CASE
            WHEN SUM(ia.VALOR_INDICE) <= 100 THEN 'OK'
            WHEN SUM(ia.VALOR_INDICE) <= 200 THEN 'ATENCAO'
            ELSE 'CRITICO'
        END as STATUS
    FROM F_DOTACAO_ORCAMENTARIA fdo
    JOIN F_DOTACAO_ORCAMENTO_TEMATICO fdot
        ON fdo.idF_DOTACAO_ORCAMENTARIA = fdot.
            idF_DOTACAO_ORCAMENTARIA
    JOIN D_INDICE_ALOCACAO ia
        ON fdot.idD_INDICE_ALOCACAO = ia.idD_INDICE_ALOCACAO
    JOIN D_ORCAMENTO_TEMATICO dot
        ON fdot.idD_ORCAMENTO_TEMATICO = dot.idD_ORCAMENTO_TEMATICO
    WHERE fdot.EXERCICIO = p_exercicio
        AND fdot.STATUS = 'ATIVO'
        AND fdo.STATUS = 'ATIVA'
    GROUP BY
        fdo.CODIGO_DOTACAO,
        fdo.NOME_DOTACAO,
        fdo.VALOR_DOTACAO

```

```
        HAVING SUM(ia.VALOR_INDICE) > p_limite_critico
        ORDER BY SUM(ia.VALOR_INDICE) DESC;
END$$

DELIMITER ;
```