3p3 BOM – IMPLEMENTATION GUIDE FOR 3 UNIVERSAL TABLES

From 150 Tables to 3 Ontological Tables

For: Cyril Amegah
From: Luca Meggiolaro
Date: August 22, 2025
Objective: Implement EXACTLY the 3P3 system with 3 universal tables

---

⚠ FUNDAMENTAL PARADIGM SHIFT

STOP ❌

- **DO NOT create** 150 separate tables (TSK_PHONE, TSK_EMAIL, etc.)
- **DO NOT use** traditional FileMaker relationships
- **DO NOT duplicate** data structures

START ✅

- **ONLY 3 physical tables** in the database
- **Differentiation via TYPE field**
- **SQL for dynamic queries**

STOP ❌ START ✅

Create TSK_PHONE table → Record in ETY with SubType='PHONE'
Create TSK_EMAIL table → Record in ETY with SubType='EMAIL'
150 FileMaker relationships → SQL query on 3 tables
Fixed fields per process → Dynamic JSONData

📊 THE THREE UNIVERSAL TABLES – EXACT SPECIFICATION

1. CMP TABLE (Components/Catalog)

Purpose: Contains ALL templates and reusable components

```
CREATE TABLE CMP (
  -- IDENTITY
  CmpID         VARCHAR(30) PRIMARY KEY,  -- DNA: MAT_2025_0001
```

```
    CmpType        VARCHAR(20) NOT NULL,     -- MAT|CDL|PRX|TSK|PRD
    CmpSubType     VARCHAR(30),              -- PHONE|EMAIL|PLANCIA|etc
    CmpCode        VARCHAR(50) UNIQUE,       -- Human-readable code
    CmpName        VARCHAR(100),             -- Descriptive name

    -- STRUCTURE
    ParentCmpID    VARCHAR(30),              -- For hierarchies
    IsTemplate     BOOLEAN DEFAULT TRUE,     -- Always template in CMP
    JSONStructure  TEXT,                     -- Specific attributes in JSON

    -- PHYSICAL ATTRIBUTES
    Weight         DECIMAL(10,3),
    Dimension      VARCHAR(50),
    Color          VARCHAR(30),
    Cost           DECIMAL(10,2),

    -- CONTROL
    Active         BOOLEAN DEFAULT TRUE,
    CreatedDate    TIMESTAMP,
    CreatedBy      VARCHAR(30)
);
```

RECORD EXAMPLE:

```
INSERT INTO CMP VALUES (
    'TSK_TEMPLATE_PHONE',     -- CmpID
    'TSK',                    -- Type = Task
    'PHONE_CALL',             -- Process-specific SubType
    'TSK_PHONE_001',          -- Code
    'Phone Call Template',    -- Name
    NULL,                     -- No parent
    TRUE,                     -- Is template
    '{"duration_min": 15, "fields": ["number", "outcome", "notes"]}',  -- Specific JSON
    NULL, NULL, NULL, 25.00,   -- Hourly cost
    TRUE, NOW(), 'Admin'
);
```

2. ETY TABLE (Entity/Orchestrator)

Purpose: ALL instances created from CMP templates

```
CREATE TABLE ETY (
    -- PROCESS IDENTITY
    EtyID          VARCHAR(30) PRIMARY KEY,  -- DNA: TSK_2025_0089
    ProcessType    VARCHAR(20) NOT NULL,     -- TSK|BOM|PRJ|JOB|ORDER
    ProcessSubType  VARCHAR(30),             -- PHONE|EMAIL|etc (from CMP)
    FromTemplate   VARCHAR(30),              -- Link to CMP template

    -- RELATIONSHIPS
    ProjectID      VARCHAR(30),              -- Related project
    ParentEtyID    VARCHAR(30),              -- Parent process
    ClientID       VARCHAR(30),              -- Client

    -- STATUS
    Status         VARCHAR(20) DEFAULT 'TO_DO',
    Priority       INTEGER DEFAULT 3,
    Progress       INTEGER DEFAULT 0,

    -- SPECIFIC DATA (replaces 150 tables!)
    JSONData       TEXT,                   -- ALL specific attributes

    -- TEMPORAL
    DateCreated     TIMESTAMP,
    DateStarted     TIMESTAMP,
    DateCompleted   TIMESTAMP,
    DueDate         DATE,

    -- RESPONSIBILITY
    ResponsibleUser VARCHAR(30),
    AssignedTo      VARCHAR(30),
    Department      VARCHAR(30)
);
```

EXAMPLE: Phone Call Instance (instead of separate TSK_PHONE table)

```
INSERT INTO ETY VALUES (
    'TSK_2025_0089',          -- Unique EtyID
    'TSK',                    -- Type
    'PHONE_CALL',             -- SubType (was TSK_PHONE_CALL)
    'TSK_TEMPLATE_PHONE',     -- From which template
    'PRJ_2025_0001',          -- Project
    NULL,                     -- No parent
    'LILA_COSMETIC',          -- Client
    'IN_PROGRESS',            -- Status
    2, 50,                    -- Priority, Progress
    '{
        "phone_number": "+39 333 1234567",
        "contact_person": "Mario Rossi",
        "call_duration": 12,
        "call_outcome": "Interested",
        "notes": "Requires urgent quote",
        "recall_date": "2025-08-25"
    }',                       -- SPECIFIC DATA in JSON!
    '2025-08-22 10:00:00',    -- Created
    '2025-08-22 10:05:00',    -- Started
    NULL,                     -- Not completed
    '2025-08-23',             -- Due date
    'Marco', 'Marco', 'Sales'
);
```

3. LOG TABLE (Actions/Universal Register)

Purpose: Tracks EVERY action on CMP and ETY

```
CREATE TABLE LOG (
    -- IDENTITY
    LogID          VARCHAR(40) PRIMARY KEY,  -- ACT_20250822_100500_001
    ActionType     VARCHAR(20) NOT NULL,     -- CREATE|UPDATE|DELETE|STATUS

    -- TARGET
    TargetTable    VARCHAR(3),               -- CMP|ETY
    TargetID       VARCHAR(30),              -- Target entity ID
    TargetType     VARCHAR(20),              -- TSK|BOM|PRJ etc

    -- ACTION DETAILS
    FieldChanged   VARCHAR(50),              -- Field changed
    OldValue       TEXT,                     -- Previous value
    NewValue       TEXT,                     -- New value
```

```
  ActionJSON     TEXT,                  -- Complex details

  -- TEMPORAL
  Timestamp      TIMESTAMP DEFAULT NOW(),
  Duration       INTEGER,               -- Minutes

  -- ACTOR
  UserID         VARCHAR(30),
  UserRole       VARCHAR(30),
  IPAddress      VARCHAR(45),

  -- HIERARCHY
  SessionID      VARCHAR(50),           -- Groups actions
  ParentLogID    VARCHAR(40)            -- Parent log
);
```

🔄 HOW TO CONVERT THE 150 TABLES

INSTEAD OF:

```
TSK_PHONE_CALL (separate table)
TSK_EMAIL (separate table)
TSK_MEETING (separate table)
... 150 tables
```

WE DO:

```
-- EVERYTHING in ETY with ProcessSubType as differentiator!

-- Query to find all phone calls:
SELECT * FROM ETY
WHERE ProcessType = 'TSK'
AND ProcessSubType = 'PHONE_CALL'
AND Status = 'TO_DO';

-- Query to find all tasks of Marco:
SELECT * FROM ETY
WHERE ProcessType = 'TSK'
AND ResponsibleUser = 'Marco';
```

📝 ESSENTIAL FILEMAKER SCRIPT

Script: Create_Universal_Entity

```
# Input: $ProcessType, $ProcessSubType, $TemplateID

# STEP 1: Generate unique DNA
Set Variable [$NewID; Value: $ProcessType & "_" & Year(Get(CurrentDate)) & "_" &
        GetAsNumber(ExecuteSQL("SELECT MAX(SUBSTR(EtyID, -4)) FROM ETY
        WHERE EtyID LIKE '" & $ProcessType & "_" & Year(Get(CurrentDate)) & "_%'"; ""; "")) + 1]

# STEP 2: Retrieve template from CMP
```

```
Go to Layout ["CMP"]
Enter Find Mode []
Set Field [CMP::CmpID; $TemplateID]
Perform Find []
Set Variable [$TemplateJSON; Value: CMP::JSONStructure]

# STEP 3: Create record in ETY
Go to Layout ["ETY"]
New Record/Request
Set Field [ETY::EtyID; $NewID]
Set Field [ETY::ProcessType; $ProcessType]
Set Field [ETY::ProcessSubType; $ProcessSubType]
Set Field [ETY::FromTemplate; $TemplateID]
Set Field [ETY::Status; "TO_DO"]
Set Field [ETY::JSONData; $TemplateJSON]  # Copy template structure
Set Field [ETY::DateCreated; Get(CurrentTimeStamp)]
Set Field [ETY::ResponsibleUser; Get(UserName)]

# STEP 4: Log creation
Go to Layout ["LOG"]
New Record/Request
Set Field [LOG::LogID; "ACT_" & Get(CurrentTimeStamp)]
Set Field [LOG::ActionType; "CREATE"]
Set Field [LOG::TargetTable; "ETY"]
Set Field [LOG::TargetID; $NewID]
Set Field [LOG::TargetType; $ProcessType]
Set Field [LOG::NewValue; "Created from template " & $TemplateID]
Set Field [LOG::UserID; Get(UserName)]

# STEP 5: Return to ETY and show proper form
Go to Layout ["ETY"]
# USE ProcessSubType to determine which layout to show!
If [$ProcessSubType = "PHONE_CALL"]
   Go to Layout ["ETY_Form_Phone"]  # Specific layout with parsed JSON fields
Else If [$ProcessSubType = "EMAIL"]
   Go to Layout ["ETY_Form_Email"]
# ... etc for each type
End If
```

⚡ IMMEDIATE ADVANTAGES

1. **Database 95% smaller** (3 tables vs 150)
2. **New process = new CMP template** (not a new table!)
3. **Unified reporting** across ALL processes
4. **Very simple backup/restore**
5. **Better performance** (fewer JOINs)

🚫 ERRORS TO AVOID

1. **DO NOT create separate tables** for each process

2. **DO NOT duplicate fields** already common (use ETY base fields)
3. **DO NOT forget JSONData** for specificity
4. **DO NOT skip LOG** for every operation

✅ VALIDATION CHECKLIST

- Only 3 physical tables in the database?
- ProcessSubType differentiates types?
- JSONData contains specific fields?
- Templates in CMP, instances in ETY?
- LOG records everything?
- Script generates unique DNA?
- Layouts adapt to SubType?

📞 SUPPORT

Technical doubts? JSONData can contain ANY specific structure:

- Phone call fields
- Email details
- Meeting data
- Production info
- ... anything that would have required a separate table!

Example FileMaker query to extract from JSON:

```
JSONGetElement(ETY::JSONData; "phone_number")
JSONGetElement(ETY::JSONData; "call_duration")
```

🎯 NEXT IMMEDIATE STEP

1. **STOP developing separate tables**
2. **CREATE the 3 tables** as specified
3. **IMPLEMENT first test** with TSK_PHONE_CALL
4. **VALIDATE** that everything works
5. **SCALE** to other processes

THIS IS THE REAL 3P3 SYSTEM!