

# DOCUMENT 3: 3P3 DATABASE ARCHITECTURE

Complete Implementation Schema for THE BRIDGE

**For:** Cyril Amegah - Technical Assessment THE BRIDGE

**Date:** August 10, 2025

**Purpose:** Complete blueprint for 3P3 FileMaker database development

## PART A: CORE FOCUS - PRODUCT AS ELEMENTARY ENTITY

### MASTER USE CASE: KOOL TOOL Color Charts

**Why this example is perfect:**

- **Real industrial complexity** (120 strands with different recipes)
- **Complete workflow** (from client phone call to finished product)
- **Critical traceability** (every thread must be traceable)
- **Infinite variants** (infinite color combinations)
- **Web integration** (online client approval)

### FINAL PRODUCT ENTITY:

```
CARTELLA_COLORI_4ANTE = {  
    CLIENT VALUE: 76€  
    COMPONENTS: 120 strands + 4 pages + accessories + packaging  
    PROCESSES: TSK→PRJ→RCH→TEH→APR with 15+ sub-processes  
    RESOURCES: 5 operators + 3 machines + 20+ materials  
    DNA: PRJ25001\CARTELLA_LILA_4ANTE_120CIOCCHI  
}
```

### COMPLETE OPERATIONAL WORKFLOW

#### 1. CLIENT PHONE CALL (TSK25001)

```
TRIGGER: LILA COSMETIC client request  
ACTION: Simona phones client, collects specifications  
INPUT: Client briefing + reference samples  
OUTPUT: Technical specifications + commercial possibility  
DNA: TSK25001  
STATUS: TO_DO → IN_PROGRESS → DONE
```

#### 2. PROJECT OPENING (PRJ25001)

TRIGGER: Positive phone call, interested client  
ACTION: Luca opens official project  
INPUT: Phone call report + feasibility assessment  
OUTPUT: Active project + sample request authorization  
DNA: PRJ25001  
PARENT: TSK25001

### **3. SAMPLE REQUEST (RCH25001)**

TRIGGER: Approved project  
ACTION: Giuseppe registers request for 120 strand samples  
INPUT: Client specifications + requested color list  
OUTPUT: List of samples to produce  
DNA: RCH25001  
PARENT: PRJ25001

### **4. TECHNICAL SHEET (TEH25001)**

TRIGGER: Defined samples  
ACTION: Marco creates product technical sheet  
INPUT: Sample list + technical specifications  
OUTPUT: Definitive product sheet + initial BOM  
DNA: TEH25001  
PARENT: RCH25001

### **5. FINAL APPROVAL (APR25001)**

TRIGGER: Completed technical sheet  
ACTION: Client approves via web interface  
INPUT: Technical sheet + physical samples  
OUTPUT: Production approval or modification requests  
DNA: APR25001  
PARENT: TEH25001

## **SUB-PROCESSES RCH (Detailed 120 Strand Management):**

### **RCH25002: SWATCHES REGISTRATION**

DNA: RCH25002  
PARENT: RCH25001  
ACTION: Giuseppe registers list of 120 strands requested by client  
INPUT: Client email with color list + physical samples  
OUTPUT: 120 rows in system, each with specific DNA

## RCH25003: SWATCHES SAMPLING

DNA: RCH25003  
PARENT: RCH25002  
ACTION: Marco prepares workspace and necessary spools  
INPUT: List of 120 strands + warehouse spool availability  
OUTPUT: Work plan + allocated spools

## RCH25004-RCH25123: SWATCHES CREATION (120 Specific Processes)

### STRAND EXAMPLE 1:

DNA: RCH25004\_BLONDE\_L8\_R001  
PARENT: RCH25003  
ACTION: Marco assembles BLONDE\_L8 strand  
RECIPE: BOB001(3 parts) + BOB002(2 parts)  
MACHINE: ASSEMBLAGGIO\_001  
TIME: 45 seconds  
OUTPUT: BLONDE\_L8 sample strand

### STRAND EXAMPLE 2:

DNA: RCH25005\_BROWN\_M7\_R045  
RECIPE: BOB003(4 parts) + BOB001(1 part)  
... (118 remaining strands with unique DNA and recipes)

## RCH25124: SWATCHES LABELING

DNA: RCH25124  
PARENT: RCH25004-RCH25123 (strand batch)  
ACTION: Anna prints labels with QR code for each strand  
INPUT: 120 finished strands + label templates  
OUTPUT: 120 applied labels + generated QR codes

## APR25002: SWATCHES MASTER APPROVAL

DNA: APR25002  
PARENT: RCH25124  
ACTION: Client accesses via browser and approves/rejects strands  
WEB\_URL: <https://kooltool.eu/approval/APR25002>  
INPUT: Strands with labels + web interface  
OUTPUT: Approval list for each strand + confirmed recipes

## PART B: 12-15 CORE TABLES WITH DETAILED FIELDS

### 1. LOG TABLE (Master Process Controller)

sql

LOG TABLE - Universal Existential Attributes:

— ProcessID (Text, 50, Primary Key, Indexed)

Format: PRXYYNNNN

Example: "RCH25001"

— ProcessType (Text, 10, Indexed, Dropdown)

Values: TSK|PRJ|RCH|TEH|APR

Example: "RCH"

— ProcessFamily (Text, 20, Calculation)

Formula: Case(ProcessType = "RCH"; "SAMPLES";

ProcessType = "TEH"; "TECHNICAL"; ProcessType)

Example: "SAMPLES"

— ParentProcessID (Text, 50, Indexed)

Parent process DNA for hierarchy

Example: "PRJ25001"

— ClientID (Text, 20, Foreign Key, Indexed)

Connection to CLIENTS table

Example: "CLI\_LILA\_COSMETIC"

— ResponsibleID (Text, 10, Foreign Key, Indexed)

Connection to USERS table

Example: "GIU" (Giuseppe)

— CreatorID (Text, 10, Foreign Key, Indexed)

Who created the process

Example: "MAR" (Marco)

— Status (Text, 20, Indexed, Dropdown)

Values: TO\_DO|IN\_PROGRESS|DONE|BLOCKED|CANCELLED

Example: "IN\_PROGRESS"

— Priority (Text, 10, Dropdown)

Values: HIGH|MEDIUM|LOW

Example: "HIGH"

— DateCreated (Timestamp, Auto-enter Creation)

Example: "19/08/2025 09:30:15"

— DateStarted (Timestamp)

Set when Status = IN\_PROGRESS

Example: "19/08/2025 10:15:30"

— DateCompleted (Timestamp)

- | Set when Status = DONE
- | Example: "19/08/2025 16:45:20"
- |
- |   |— EstimatedDuration (Number, Minutes)
- |   | Estimated time for completion
- |   | Example: 240 (4 hours)
- |
- |   |— ActualDuration (Number, Calculation)
- |   | Formula: If(DateCompleted; DateCompleted - DateStarted; "")
- |   | Example: 375 (6.25 actual hours)
- |
- |   |— Description (Text, 500)
- |   | Free description of process
- |   | Example: "120 strand samples for LILA chart"
- |
- |   |— Notes (Text, 2000)
- |   | Operational notes and observations
- |   | Example: "Client very demanding on blonde tones"
- |
- |   |— QRCode (Container, Image)
- |   | QR code automatically generated with DNA
- |
- |   |— LastModified (Timestamp, Auto-enter Modification)
- |   | Example: "22/08/2025 14:22:10"

## 2. RCH TABLE (Research/Samples Management - Specific)

sql

RCH TABLE - Specific Sampling Attributes:

  |— ProcessID (Text, 50, Primary Key = FK to LOG)

    |— Example: "RCH25001"

  |— SwatchesTotal (Number, Integer)

    |— Total number of requested strands

    |— Example: 120

  |— SwatchesCompleted (Number, Calculation)

    |— Formula: Count(RCH\_DETAIL::ProcessID where Parent = This ProcessID)

    |— Example: 120

  |— SwatchesApproved (Number, Calculation)

    |— Formula: Count(RCH\_DETAIL::ProcessID where ApprovalStatus = "APPROVED")

    |— Example: 115

  |— SwatchesRejected (Number, Calculation)

    |— Formula: Count(RCH\_DETAIL::ProcessID where ApprovalStatus = "REJECTED")

    |— Example: 5

  |— ClientSampleReceived (Date)

    |— Date of sample receipt from client

    |— Example: "18/08/2025"

  |— ClientSampleFiles (Container, Multiple)

    |— Digital client sample files

    |— Example: "LILA\_samples\_001.jpg, LILA\_samples\_002.jpg"

  |— SwatchType (Text, 20, Dropdown)

    |— Values: COLOR\_MATCH|TEXTURE\_MATCH|CUSTOM\_BLEND

    |— Example: "COLOR\_MATCH"

  |— DeliveryMethod (Text, 20, Dropdown)

    |— Values: PHYSICAL\_MAIL|COURIER|PICKUP|DIGITAL\_ONLY

    |— Example: "COURIER"

  |— PackagingType (Text, 30)

    |— Packaging type for sample shipping

    |— Example: "Presentation folder 15x20cm"

  |— ApprovalDeadline (Date)

    |— Deadline for client approval

    |— Example: "25/08/2025"

  |— ShippingAddress (Text, 200)

    |— Sample shipping address

Example: "LILA COSMETIC, Via Roma 123, Milano 20100"

QualityStandard (Text, 50, Dropdown)

Values: BASIC|PREMIUM|LUXURY|CUSTOM

Example: "PREMIUM"

SpecialInstructions (Text, 1000)

Specific instructions from client

Example: "Avoid too cold tones on blondes"

### 3. RCH\_DETAIL TABLE (Individual Strands - Specific)

sql

RCH\_DETAIL TABLE - Specific Individual Strand Attributes:

  |— ProcessID (Text, 60, Primary Key = FK to LOG)

  |— Format: PRXYYNNN\_RECIPES\_CODE

  |— Example: "RCH25004\_BLONDE\_L8\_R001"

  |— SwatchCode (Text, 30, Indexed)

    |— Unique strand code for client

    |— Example: "BLONDE\_L8"

  |— ColorName (Text, 50)

    |— Commercial color name

    |— Example: "Blonde Level 8 Ash"

  |— ColorFamily (Text, 20, Dropdown)

    |— Values: BLONDE|BROWN|BLACK|RED|GREY|FANTASY

    |— Example: "BLONDE"

  |— ColorLevel (Number, 1-10)

    |— Lightening level (1=black, 10=extra light blonde)

    |— Example: 8

  |— ColorTone (Text, 10, Dropdown)

    |— Values: NATURAL|ASH|GOLD|RED|VIOLET|PEARL

    |— Example: "ASH"

  |— RecipeCode (Text, 30, Indexed)

    |— Unique recipe code

    |— Example: "BLONDE\_L8\_R001"

  |— RecipeFormula (Text, 200)

    |— Complete recipe formula

    |— Example: "BOB001(3parts) + BOB002(2parts)"

  |— BobineUsed (Text, 100)

    |— List of spools used

    |— Example: "BOB001,BOB002"

  |— BobineQuantities (Text, 50)

    |— Quantity for each spool

    |— Example: "3,2"

  |— TotalWeight (Number, Decimal)

    |— Total strand weight in grams

    |— Example: 5.2

  |— ProductionTime (Number, Seconds)

| Production time in seconds  
| Example: 45

|  
| MachineUsed (Text, 20)  
| Machine used for assembly  
| Example: "ASSEMBLAGGIO\_001"

|  
| OperatorID (Text, 10, FK to USERS)  
| Operator who assembled  
| Example: "MAR"

|  
| QualityCheck (Text, 20, Dropdown)  
| Values: OK|KO|PENDING|REVISION  
| Example: "OK"

|  
| ClientApproval (Text, 20, Dropdown)  
| Values: APPROVED|REJECTED|PENDING|REVISION  
| Example: "APPROVED"

|  
| ProductionNotes (Text, 1000)  
| Production notes  
| Example: "BOB001 spool tension slightly high"

|  
| ClientNotes (Text, 1000)  
| Notes from client  
| Example: "Perfect, use for series production"

|  
| QRCodeImage (Container, Image)  
| QR code for physical traceability

## 4. TSK TABLE (Task Management - Specific)

sql

TSK TABLE - Specific Task/Communication Attributes:

  |— ProcessID (Text, 50, Primary Key = FK to LOG)

  |— Example: "TSK25001"

  |— TaskType (Text, 20, Dropdown)

  |— Values: PHONE\_CALL|EMAIL|MEETING|VISIT|FOLLOW\_UP

  |— Example: "PHONE\_CALL"

  |— ContactMethod (Text, 50)

  |— Contact method used

  |— Example: "+40-21-555-0123"

  |— ContactPerson (Text, 100)

  |— Person contacted

  |— Example: "Maria Rossi - Procurement Manager"

  |— Duration (Number, Minutes)

  |— Actual communication duration

  |— Example: 15

  |— TaskOutcome (Text, 20, Dropdown)

  |— Values: SUCCESS|PARTIAL|FAILED|RESCHEDULED

  |— Example: "SUCCESS"

  |— TaskSummary (Text, 500)

  |— Brief communication summary

  |— Example: "Interested client, requests 120 strand samples"

  |— TaskDetails (Text, 2000)

  |— Complete communication details

  |— Example: "Discussed 4-panel folder specifications..."

  |— FollowUpRequired (Boolean)

  |— Requires follow-up?

  |— Example: TRUE

  |— FollowUpDate (Date)

  |— Expected follow-up date

  |— Example: "22/08/2025"

  |— NextAction (Text, 500)

  |— Next action to take

  |— Example: "Open project and send sample list"

└─ Attachments (Container, Multiple)

Files attached to communication

## 🎯 5. PRJ TABLE (Project Management - Specific)

sql

PRJ TABLE - Specific Project Attributes:

  |— ProcessID (Text, 50, Primary Key = FK to LOG)

  |— Example: "PRJ25001"

  |— ProjectName (Text, 100)

  |— Commercial project name

  |— Example: "LILA COSMETIC - Premium Color Charts"

  |— ProjectCode (Text, 30, Unique)

  |— Internal project code

  |— Example: "LILA\_CART\_2025\_001"

  |— ProjectType (Text, 20, Dropdown)

  |— Values: STANDARD|CUSTOM|PROTOTYPE|RUSH

  |— Example: "CUSTOM"

  |— ProjectCategory (Text, 30, Dropdown)

  |— Values: CARTELLE|CIOCCHI|STAMPATI|ACCESSORI|MIXED

  |— Example: "CARTELLE"

  |— EstimatedValue (Number, Currency)

  |— Estimated project value

  |— Example: 1500.00

  |— ActualValue (Number, Currency)

  |— Final confirmed value

  |— Example: 1650.00

  |— ProjectDeadline (Date)

  |— Project delivery deadline

  |— Example: "30/08/2025"

  |— ProjectStatus (Text, 20, Dropdown)

  |— Values: PLANNING|ACTIVE|ON\_HOLD|COMPLETED|CANCELLED

  |— Example: "ACTIVE"

  |— ProjectManager (Text, 10, FK to USERS)

  |— Project manager

  |— Example: "LUC"

  |— EstimatedHours (Number, Decimal)

  |— Total estimated hours

  |— Example: 24.5

  |— ActualHours (Number, Calculation)

  |— Actual hours (sum from LOG children)

| Example: 28.2

|   | ProjectScope (Text, 1000)

|   | Detailed scope description

|   | Example: "Production of 120 strands + 4-panel folder..."

|   | ProjectNotes (Text, 2000)

|   | Project notes and observations

|   | Example: "Client very demanding on color quality"

|   | ProjectFolder (Text, 200)

|   | Link to Google Drive folder

|   | Example: "<https://drive.google.com/folders/xyz123>"

## 6. TEH TABLE (Technical Management - Specific)

sql

TEH TABLE - Specific Technical Aspect Attributes:

  |— ProcessID (Text, 50, Primary Key = FK to LOG)

  |— Example: "TEH25001"

  |— TechnicalType (Text, 30, Dropdown)

  |— Values: DATA\_SHEET|BOM|SPECIFICATIONS|DRAWING|PROTOTYPE

  |— Example: "DATA\_SHEET"

  |— ProductCode (Text, 30)

  |— Technical product code

  |— Example: "CART\_LILA\_4ANTE\_120"

  |— TechnicalTitle (Text, 100)

  |— Technical document title

  |— Example: "LILA Premium Chart Technical Sheet"

  |— ProductDimensions (Text, 50)

  |— Finished product dimensions

  |— Example: "15x20x2 cm"

  |— ProductWeight (Number, Decimal)

  |— Finished product weight in grams

  |— Example: 145.5

  |— MaterialsList (Text, 1000)

  |— List of necessary materials

  |— Example: "120 strands, 4 pages 200gr cardboard, glue..."

  |— ProductionSteps (Text, 2000)

  |— Production phases

  |— Example: "1. Strand preparation, 2. Page printing..."

  |— QualityStandards (Text, 1000)

  |— Required quality standards

  |— Example: "Colors conform to sample, strand tension..."

  |— EstimatedProductionTime (Number, Minutes)

  |— Estimated production time

  |— Example: 180

  |— TechnicalDrawings (Container, Multiple)

  |— Attached technical drawings

  |— TechnicalSpecs (Text, 3000)

  |— Complete technical specifications

  |— Example: "4-panel folder with 120 strands arranged..."

- |—— BOMReference (Text, 50)
  - |—— Bill of materials reference
  - |—— Example: "BOM\_CART\_LILA\_001"
- |—— RevisionNumber (Number, Integer)
  - |—— Document revision number
  - |—— Example: 2
- |—— ApprovalRequired (Boolean)
  - |—— Requires client approval?
  - |—— Example: TRUE
- |—— TechnicalNotes (Text, 2000)
  - |—— Additional technical notes
  - |—— Example: "Be careful not to over-tighten strands"

## 7. APR TABLE (Approval Management - Specific)

sql

APR TABLE - Specific Approval Attributes:

  |— ProcessID (Text, 50, Primary Key = FK to LOG)

  |— Example: "APR25001"

  |— ApprovalType (Text, 20, Dropdown)

  |— Values: CLIENT|INTERNAL|TECHNICAL|QUALITY|FINAL

  |— Example: "CLIENT"

  |— ApprovalSubject (Text, 100)

  |— Approval subject

  |— Example: "LILA strand sample approval"

  |— ApprovalItem (Text, 50)

  |— What needs to be approved

  |— Example: "120 strand samples"

  |— ApproverName (Text, 100)

  |— Name of person who must approve

  |— Example: "Maria Rossi - LILA COSMETIC"

  |— ApproverEmail (Text, 100)

  |— Approver email

  |— Example: "m.rossi@lila-cosmetic.com"

  |— ApprovalDeadline (Date)

  |— Approval deadline

  |— Example: "25/08/2025"

  |— ApprovalMethod (Text, 20, Dropdown)

  |— Values: WEB\_INTERFACE|EMAIL|PHYSICAL|PHONE|MEETING

  |— Example: "WEB\_INTERFACE"

  |— ApprovalURL (Text, 200)

  |— URL for web approval

  |— Example: "<https://kooltool.eu/approval/APR25001>"

  |— ApprovalStatus (Text, 20, Dropdown)

  |— Values: PENDING|APPROVED|REJECTED|PARTIAL|REVISION

  |— Example: "APPROVED"

  |— ApprovalDate (Timestamp)

  |— Approval date/time

  |— Example: "24/08/2025 16:30:22"

  |— ApprovalFile (Container, PDF/Image)

  |— Signed approval file

```
  └── ApprovalNotes (Text, 2000)
      ├── Approver's notes
      │   Example: "Perfect, proceed with production"
      └── RejectionReason (Text, 1000)
          ├── Reason for eventual rejection
          │   Example: "BLONDE_L8 color too dark"
          └── RevisionRequests (Text, 2000)
              ├── Modification requests
              │   Example: "Lighten the blonde and darken the brown"
              └── InternalNotes (Text, 1000)
                  ├── Internal team notes
                  │   Example: "Very precise client, prepare variants"
```

## PART C: TABLE OPERATIONAL PRINCIPLES

### 🔗 OPTIMIZED RELATIONSHIPS

#### RELATIONSHIP GRAPH STRUCTURE:

```
LOG (Master Hub)
  └── 1:1 → TSK (ProcessID match)
  └── 1:1 → PRJ (ProcessID match)
  └── 1:1 → RCH (ProcessID match)
  └── 1:1 → RCH_DETAIL (ProcessID match)
  └── 1:1 → TEH (ProcessID match)
  └── 1:1 → APR (ProcessID match)
  └── 1:∞ → LOG (Self-join via ParentProcessID)
  └── ∞:1 → CLIENTS (ClientID match)
  └── ∞:1 → USERS (ResponsibleID match)
  └── ∞:1 → USERS (CreatorID match)
```

#### PERFORMANCE KEYS:

- INDEX: ProcessID, ProcessType, Status, ClientID, ResponsibleID
- CALCULATION: ActualDuration, ProcessFamily
- SUMMARY: Count children, Sum hours per project

### 📊 OPERATIONAL QUERY EXAMPLES

#### Query 1: All processes of a project

```
sql
```

```
FIND: LOG::ProcessID = "PRJ25001" OR  
LOG::ParentProcessID = "PRJ25001" OR  
LOG::ParentProcessID LIKE "*PRJ25001*"
```

RESULT:

- └─ PRJ25001 (Main project)
- └─ RCH25001 (Sample request)
- └─ RCH25002-25003 (Sample sub-processes)
- └─ RCH25004-25123 (120 individual strands)
- └─ TEH25001 (Technical sheet)
- └─ APR25001-25002 (Various approvals)

## Query 2: Daily operational dashboard

sql

```
FIND: LOG::Status = "IN_PROGRESS" AND  
LOG::DateStarted >= Today
```

PORTRAITS:

- └─ Active processes per operator
- └─ Imminent deadlines (< 3 days)
- └─ Pending approvals
- └─ Quality problems to solve

## Query 3: Automatic project cost report

sql

CALCULATION FIELD in PRJ:

```
TotalProjectCost =  
Sum(Related_LOG::ActualDuration * USERS::HourlyRate) +  
Sum(Related_MATERIALS::Quantity * MATERIALS::UnitCost) +  
Sum(Related_MACHINES::Time * MACHINES::HourlyRate)
```

AUTOMATIC RESULT:

```
Total cost PRJ25001 = 587€  
Margin = 1650€ quote - 587€ costs = 1063€ (64% margin)
```

## PART D: CONCRETE OPERATIONAL BENEFITS

### ⌚ SCENARIO 1: Client Quality Problem

**PROBLEM:** LILA client complains about BLONDE\_L8 strand color

## AUTOMATIC INVESTIGATION:

1. Find strand: RCH25004\_BLONDE\_L8\_R001
2. Find recipe: BOB001(3) + BOB002(2)
3. Find spool batches: BOB001\_LOT\_2025\_045, BOB002\_LOT\_2025\_033
4. Find supplier: TINTURE\_ROSSI for BOB001, TINTURE\_VERDI for BOB002
5. Find operator: Marco (MAR) on 20/08/2025 at 14:30
6. Find machine: ASSEMBLAGGIO\_001
7. Find problems: "BOB001 spool tension slightly high" (from notes)

RESULT: Problem identified in 30 seconds

ACTION: Contact supplier TINTURE\_ROSSI for batch BOB001\_LOT\_2025\_045

## SCENARIO 2: Operator Performance Analysis

**QUERY:** Marco's performance last 30 days

### AUTOMATIC RESULT:

sql

FIND: LOG::ResponsibleID = "MAR" AND  
LOG::DateCreated >= (Today - 30)

#### STATISTICS:

- Completed processes: 45
- Average time per process: 38 minutes
- Delayed processes: 3 (6.7%)
- Average quality: 9.2/10
- Specialization: 85% RCH processes (samples)
- Efficiency: +12% vs company target

## SCENARIO 3: Workload Prediction

**QUERY:** Workload next 15 days

### AUTOMATIC RESULT:

sql

FIND: LOG::Status = "TO\_DO" AND

LOG::EstimatedDuration > 0

#### FORECASTS:

- Marco: 47 scheduled hours (118% capacity) ⚠
- Giuseppe: 32 scheduled hours (80% capacity) ✓
- Anna: 28 scheduled hours (70% capacity) ✓
- Simona: 25 scheduled hours (62% capacity) ✓

#### SUGGESTED ACTIONS:

- Reassign 7 hours **from** Marco **to** Giuseppe
- Anticipate non-urgent Marco processes
- Consider overtime **for** Marco **if** urgent

## PART E: BOM-LOG-ETY TRIAD ARCHITECTURE

"The Operational Trinity that Solves Every Industrial Complexity"

### E.1 The Ontological Revolution

The true 3P3 innovation is not having "many tables" but having **3 elementary tables** that manage any complexity:

MATTER (BOM) + ACTION (LOG) + ENTITY (ETY) = 3P3 UNIVERSE

#### Ontological Correspondence:

- **BOM = ASPECT** (what is needed to make the product)
- **LOG = NATURE** (real actions we are doing)
- **ETY = ENTITY** (orchestrator that coordinates everything)

### E.2 Genial Dual Pattern

**REVOLUTIONARY INSIGHT:** BOM and LOG use the same pattern on different planes!

#### DESIGN PLANE (BOM):

BOM = "What is needed to make the product"

- Defines RECIPE: FIR→BOB→CDL→Strands→Page→Folder
- Defines SEQUENCE: Phase1→Phase2→Phase3→Phase4
- Defines RESOURCES: Who, machines, estimated times
- RESULT: Ideal work plan

#### REALIZATION PLANE (LOG):

LOG = "What we are actually doing"

- └─ Records ACTIONS: Mario made strand at 14:30
- └─ Records TIMES: 45 real seconds vs 60 estimated
- └─ Records PROBLEMS: BOB001 spool with high tension
- └─ RESULT: True story of what happened

## E.3 ETY Manager - The Supreme Orchestrator

ETY is the "director" that:

- **Coordinates BOM and LOG:** Verifies that reality follows plan
- **Manages TreeView:** Maintains infinite entity hierarchy
- **Resolves recursivity:** Strand is semi-finished AND finished product
- **Tracks evolutionary state:** From idea to project to product

## E.4 Infinite Recursive TreeView Management

PRACTICAL EXAMPLE:

CARTELLA\_COLORI (ETY\_001)

- └─ BOM: Strands(120) + Pages(4) + Cover(1) + Glue
- └─ LOG: Complete production process
- └─ ETY Children:
  - └─ CIOCCHI\_120 (ETY\_002)
    - └─ BOM: BOB001(3) + BOB002(2) + assembly
    - └─ LOG: Marco assembles strands 14:30-16:45
    - └─ ETY Children:
      - └─ BOB001 (ETY\_003)
        - └─ BOM: FIR\_BIANCO + TINTURA\_L8
        - └─ LOG: External dyeing, 3 days
        - └─ ETY Children: FIR\_BIANCO (elementary)
      - └─ BOB002 (ETY\_005) - same structure
    - └─ PAGINE\_4 (ETY\_006) - recursive structure
    - └─ COPERTINA (ETY\_007) - recursive structure

## E.5 Implementation Tables

BOM TABLE:

sql

#### BOM TABLE:

- └─ BOM\_ID (**Primary Key**)
- └─ EntityID (FK to ETY)
- └─ ComponentType (FIR|BOB|CDL|LUC|MAT|RES)
- └─ ComponentCode (BOB001, FIR\_BIANCO, etc.)
- └─ Quantity (3, 2, 1, etc.)
- └─ Sequence (assembly **order**)
- └─ Notes (technical details)

#### ETY TABLE:

sql

#### ETY TABLE:

- └─ EntityID (**Primary Key**)
- └─ EntityCode (CARTELLA\_001, CIOCCHI\_120, etc.)
- └─ EntityType (FINISHED|SEMITRANSHED|RAW\_MATERIAL)
- └─ ParentEntityID (for TreeView hierarchy)
- └─ ProcessID (FK to LOG for current process)
- └─ Status (IDEA|DESIGN|PRODUCTION|FINISHED)
- └─ Description (commercial name)

## E.6 Revolutionary Advantage

### With 3 tables we manage:

- Infinite nested recipes
- Complete traceability from thread to product
- Consistency between design and production
- Infinite scalability controlled ontologically

This is the 3P3 core that makes it superior to any existing system!

## CONCLUSION

The 3P3 architecture transforms business management from "isolated data administration" to "orchestration of self-evolving processes".

### IMMEDIATE IMPLEMENTATION for Cyril:

1. **START** with LOG + 5 specific tables (TSK, PRJ, RCH, TEH, APR)
2. **TEST** with complete workflow TSK→PRJ→RCH→TEH→APR
3. **EXPAND** to other macroprocesses using same pattern
4. **IMPLEMENT** BOM-LOG-ETY triad for complex product management

**The system is clear, crystalline and implementable - just like 3P3 theory!**

---

**KOOL TOOL SRL - Craiova, România**

"*3 tables to govern infinite complexity*"