MayaCal

Function of Tzolk'in Calendar structures in the Postclassic Maya Codices

Stage Report 2

Jakub Špoták

December 2024

Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I03-03-V04-00243.

Funded by the European Union
NextGenerationEU

RECOVERY AND RESILIENCE PLAN

MINISTRY OF EDUCATION, RESEARCH, DEVELOPMENT AND YOUTH OF THE SLOVAK REPUBLIC

Report 2: Structural Encoding of Mesoamerican Codices and Tzolk'in Calendar in a Relational Database

I. Introduction

This report presents a comprehensive overview of the structural logic underpinning the design and implementation of a relational database dedicated to the study of Mesoamerican codices. These ancient manuscripts, primarily of Maya and Mixtec origin, exhibit complex layers of visual, textual, and calendrical information. The goal of this database is not only to serve as a digital repository, but also as a functional analytical tool for researchers examining epigraphy, iconography, and timekeeping systems. The database was developed using FileMaker Pro, offering a modular and visually intuitive interface that mirrors the hierarchical organization of the codices themselves.

II. Hierarchical Structure of Codex Records

Codices were structured as folded manuscripts composed of multiple pages. Each page could feature several registers—subdivided segments used to isolate visual scenes or calendrical structures. In our database, this structure is modeled with distinct but interrelated tables: `COD_Codices` (codices), `CPG_Codices_Pages` (pages), `CPG_Codices_Registers` (registers), and `CDA_Codices_Almanac` (almanacs). Each register is linked to a specific page, and each almanac to a specific register, forming a multilevel hierarchy that reflects the embedded logic of the original manuscripts. This system allows researchers to access and cross-reference visual and textual content precisely at the level of its original context.

III. Almanacs as Vessels of Calendrical Knowledge

Calendrical almanacs are one of the most intellectually rich features of Mesoamerican codices. In the Classic and Postclassic periods, almanacs encoded ritual schedules, omens, prognostications, and agricultural cycles. They typically rely on the 260-day Tzolk'in calendar, composed of 20 named days and 13 day numbers, which combine into a cycle that repeats every 260 days. Within the codices, almanacs are arranged in rows or blocks, often repeating sequences with structured variation. In the database, each almanac record includes metadata fields such as the number of initial days, step values for calendar progression, and distance numbers for computation. This structure allows both static representation and dynamic calculation of full almanac sequences.

Codex	Frames Found	Almanacs Found
Dresden	620	98
Grolier	20	1
Madrid	926	255
Paris	126	11

Figure 1 Distribution of almanacs in major codices.

IV. Modeling Tzolk'in Sequences in the Database

The database includes a dedicated reference table for all 260 unique Tzolk'in day combinations. Almanacs are initiated with one or more base days (`Tzolkin_1` to `Tzolkin_10`) and paired distance numbers (`DN_1` to `DN_10`) representing the number of days to advance in the cycle. A calculation script processes these inputs, determines the resultant day position using modulo 260 logic, and populates a secondary table `Almanac_Row`, which stores fully resolved sequences. This process emulates the cyclic logic of the original manuscripts and enables users to analyze or compare calendar patterns across registers and codices.

Result_1	Calculation	from T13_CODICES_ALMANAC_ROW, = Let ([base= Tzolkin_1; d= DN_1]; Mod (base + d - 1; 260)+1)
Result_2	Calculation	from T13_CODICES_ALMANAC_ROW, = Let ([base= Result_1; d= DN_2]; Mod (base + d - 1; 260)+1)
Result_3	Calculation	from T13_CODICES_ALMANAC_ROW, = Let ([base= Result_2; d= DN_3]; Mod (base + d - 1; 260)+1)
Result_4	Calculation	from T13_CODICES_ALMANAC_ROW, = Let ([base= Result_3; d= DN_4]; Mod (base + d - 1; 260)+1)
Result_5	Calculation	Unstored, from T13_CODICES_ALMANAC_ROW, = Let ([base= Result_4; d= DN_5]; Mod (base + d - 1; 260)+1)
Result_6	Calculation	Unstored, from T13_CODICES_ALMANAC_ROW, = Let ([base= Result_5; d= DN_6]; Mod (base + d - 1; 260)+1)
Result_7	Calculation	Unstored, from T13_CODICES_ALMANAC_ROW, = Let ([base= Result_6; d= DN_7]; Mod (base + d - 1; 260)+1)
Result_8	Calculation	Unstored, from T13_CODICES_ALMANAC_ROW, = Let ([base= Result_7; d= DN_8]; Mod (base + d - 1; 260)+1)
Result_9	Calculation	Unstored, from T13_CODICES_ALMANAC_ROW, = Let ([base= Result_8; d= DN_9]; Mod (base + d - 1; 260)+1)
Result_10	Calculation	Unstored, from T13_CODICES_ALMANAC_ROW, = Let ([base= Result_9; d= DN_10]; Mod (base + d - 1; 260)+1)
Label_Result_1	Calculation	from T13_CODICES_ALMANAC_ROW, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";Result_1)
Label_Result_2	Calculation	from T13_CODICES_ALMANAC_ROW, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";Result_2)
Label_Result_3	Calculation	from T13_CODICES_ALMANAC_ROW, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";Result_3)
Label_Result_4	Calculation	from T13_CODICES_ALMANAC_ROW, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";Result_4)
Label_Result_5	Calculation	Unstored, from <unknown>, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";<Field Missing>)
Label_Result_6	Calculation	Unstored, from <unknown>, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";<Field Missing>)
Label_Result_7	Calculation	Unstored, from <unknown>, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";<Field Missing>)
Label_Result_8	Calculation	Unstored, from <unknown>, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";<Field Missing>)
Label_Result_9	Calculation	Unstored, from <unknown>, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";<Field Missing>)
Label_Result_10	Calculation	Unstored, from <unknown>, = ExecuteSQL ("SELECT Day_text FROM CDA_Tzolkin_reference WHERE ID_Tzolkin = ?" ; "" ; "";<Field Missing>)
id_almanac	Text	Indexed

Figure 2 Calculated fields in Almanac_Row table for Tzolk'in computations.

V. Computational Architecture and Calculated Fields

Scripts play a critical role in automating the generation of almanac sequences. When an almanac is created, the user specifies how many rows should be generated and which Tzolk'in days and distance numbers serve as inputs. The script loops through these inputs, retrieves values from the reference table, and inserts computed rows into the child table. Calculated fields such as `Result_1` through `Result_10` apply mathematical formulas to determine subsequent Tzolk'in positions, while SQL-based fields such as `Label_Result_1` retrieve the corresponding day

names. This framework ensures consistency, accuracy, and reusability across different codices and almanac structures.

```
Set Variable [ $almanacID ; Value: T13_CODICES_ALMANACS::ID_Almanac ]
Set Variable [ $registerID ; Value: T13_CODICES_ALMANACS::id_codices_registers ]
Set Variable [ $rowCount ; Value: T13_CODICES_ALMANACS::Calendar_rows ]
If [ IsEmpty ( $rowCount ) or $rowCount  < 1 ]
    Show Custom Dialog [ "Error" ; "Add number of Calendar rows" ]
    Exit Script [ Text Result:  ]
End If

Set Variable [ $DN_1 ; Value: T13_CODICES_ALMANACS::DN_1 ]
Set Variable [ $DN_2 ; Value: T13_CODICES_ALMANACS::DN_2 ]
Set Variable [ $DN_3 ; Value: T13_CODICES_ALMANACS::DN_3 ]
Set Variable [ $DN_4 ; Value: T13_CODICES_ALMANACS::DN_4 ]
Set Variable [ $DN_5 ; Value: T13_CODICES_ALMANACS::DN_5 ]
Set Variable [ $DN_6 ; Value: T13_CODICES_ALMANACS::DN_6 ]
Set Variable [ $DN_7 ; Value: T13_CODICES_ALMANACS::DN_7 ]
Set Variable [ $DN_8 ; Value: T13_CODICES_ALMANACS::DN_8 ]
Set Variable [ $DN_9 ; Value: T13_CODICES_ALMANACS::DN_9 ]
Set Variable [ $DN_10 ; Value: T13_CODICES_ALMANACS::DN_10 ]

Set Variable [ $i ; Value: 1 ]
Loop [ Flush: Always ]
    Exit Loop If [ $i > $rowCount ]
    Set Variable [ $fieldName ; Value: "Tzolkin_" & $i ]
    Set Variable [ $tzolkinVal ; Value: GetField ( $fieldName ) ]
    If [ IsEmpty ( $tzolkinVAL ) ]
        Show Custom Dialog [ "Missing values" ; "Some Tzolkin fields are missing" ]
        Exit Script [ Text Result:  ]
    End If
    Go to Layout [ “T13a_codices_almanac_row” (T13_CODICES_ALMANAC_ROW) ; Animation: None ]
    New Record/Request
    Set Field [ T13_CODICES_ALMANAC_ROW::id_almanac ; $almanacID ]
    Set Field [ T13_CODICES_ALMANAC_ROW::id_codices_registers ; $registerID ]
    Set Field [ T13_CODICES_ALMANAC_ROW::Tzolkin_1 ; $tzolkinVal ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_1 ; $DN_1 ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_2 ; $DN_2 ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_3 ; $DN_3 ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_4 ; $DN_4 ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_5 ; $DN_5 ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_6 ; $DN_6 ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_7 ; $DN_7 ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_8 ; $DN_8 ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_9 ; $DN_9 ]
    Set Field [ T13_CODICES_ALMANAC_ROW::DN_10 ; $DN_10 ]

```

Figure 3 Script for generating almanac rows based on Tzolk'in values.

VI. Data Coverage and Codical Distribution

To date, the database includes multiple codices (Dresden, Madrid, Paris, Grolier), each with dozens to hundreds of pages and registers. The almanac count differs across manuscripts due to their original structure and preservation state: for example, the Madrid Codex contains over 250 identifiable almanacs, while the Grolier Codex includes only one. This distribution has been documented and visualized in summary tables, enabling quantitative assessments of calendrical density, usage, and distribution patterns.

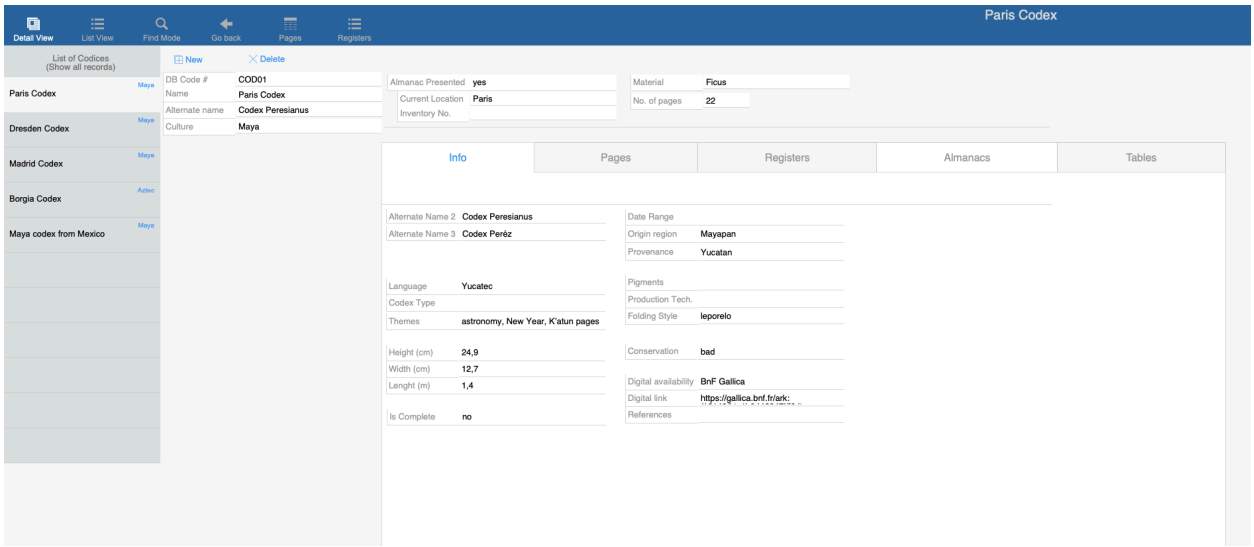

Figure 4 Layout view: metadata for codices.

VII. User Interface and Visualization

The FileMaker Pro layouts are designed to visually reflect the codex structure. Users begin at the level of a codex record and drill down into associated pages, registers, and almanacs through relational portals. Images of original manuscripts, including scans and reconstructions, are embedded alongside data entry fields. Each layout supports metadata management, calendrical computation, and visual navigation. This hybrid design approach—combining structured relational logic with image-driven context—mirrors the interpretive process of working with the original manuscripts.

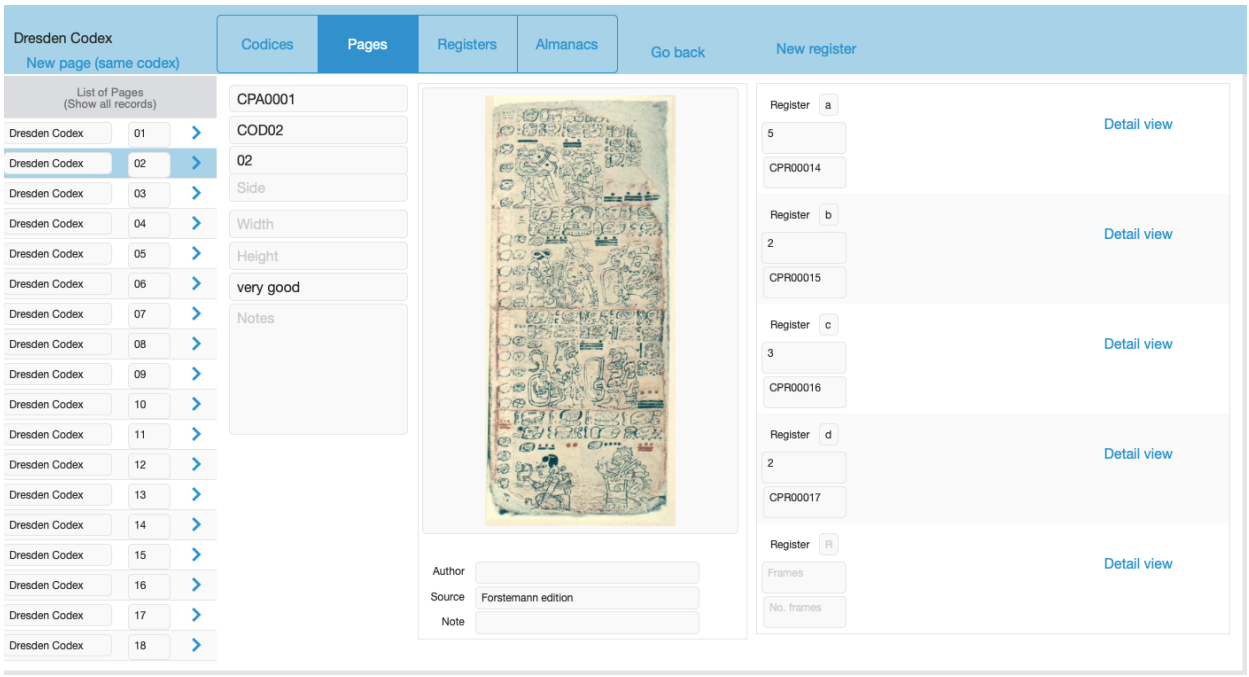

Figure 5 Layout view: Individual codex pages.

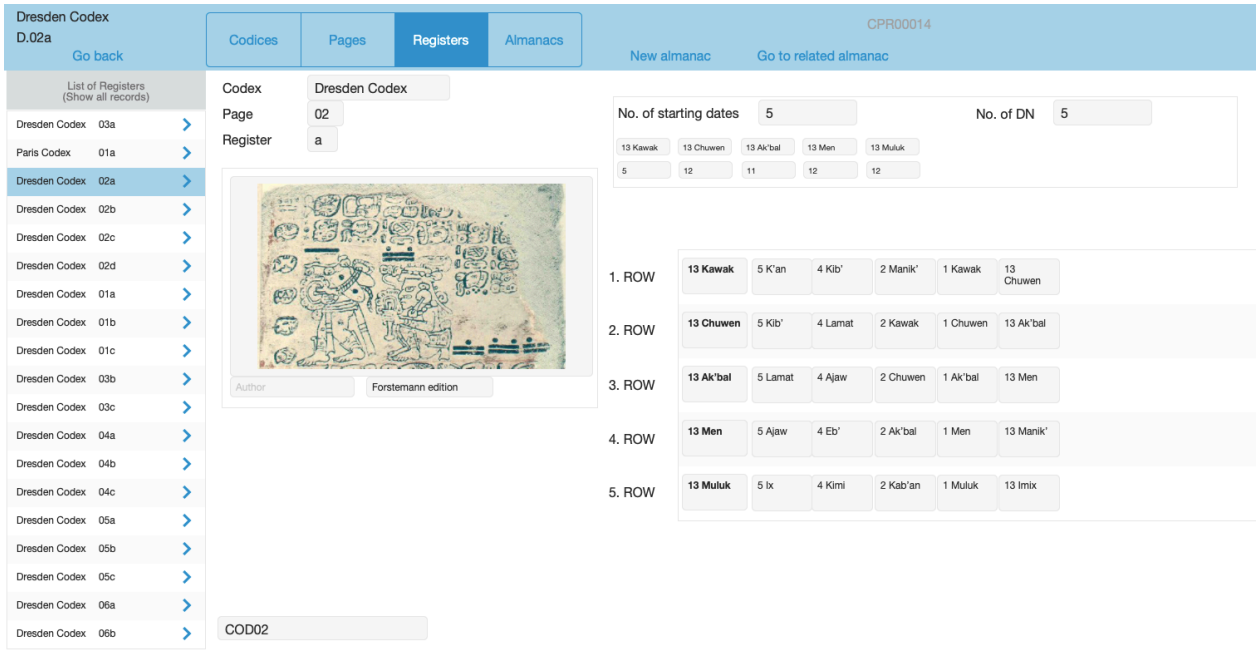

Figure 6 Layout view: Registers and associated almanacs.

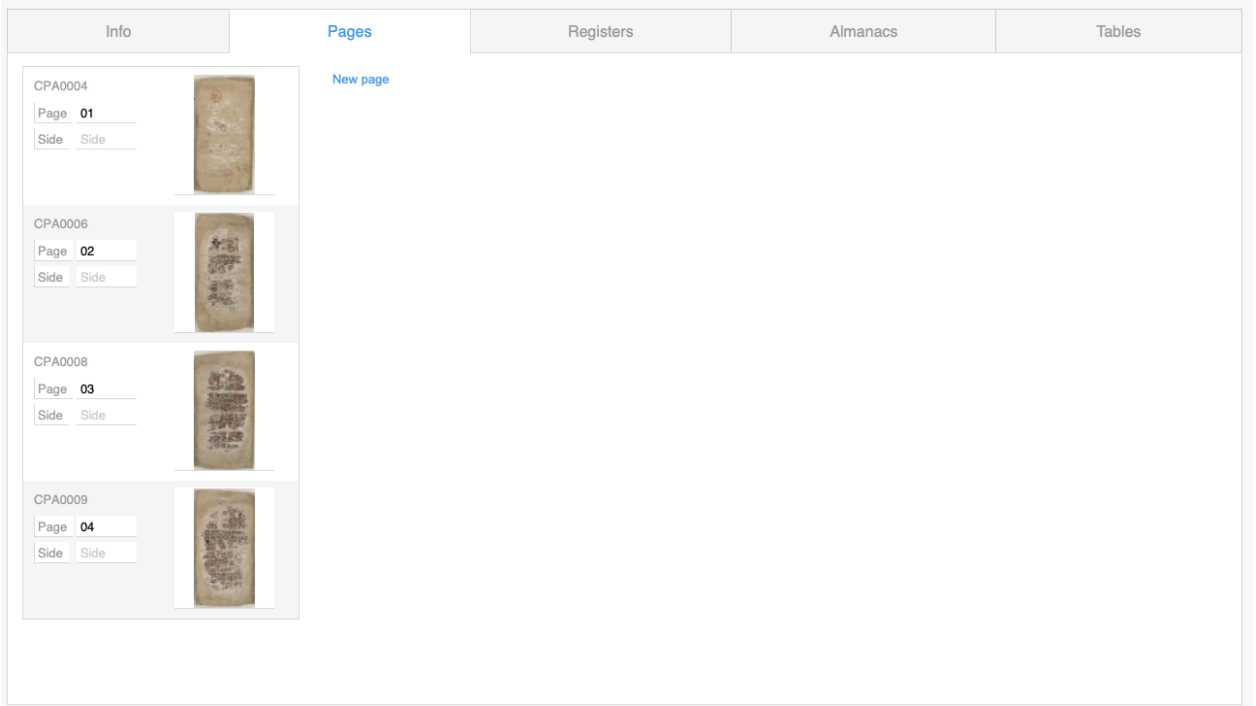

Figure 7 Page thumbnails within the codex layout.

VIII. Conclusion and Scholarly Impact

By encoding the layered, cyclic logic of Mesoamerican calendrical thought into a digital relational model, this database provides a novel tool for scholarly analysis. It allows for both granular documentation and high-level pattern recognition. Researchers can now trace calendrical themes, evaluate structural repetition, and compare manuscript content across corpora. The approach offers a replicable framework for other manuscript traditions with embedded calendar systems, and supports both academic research and digital preservation.