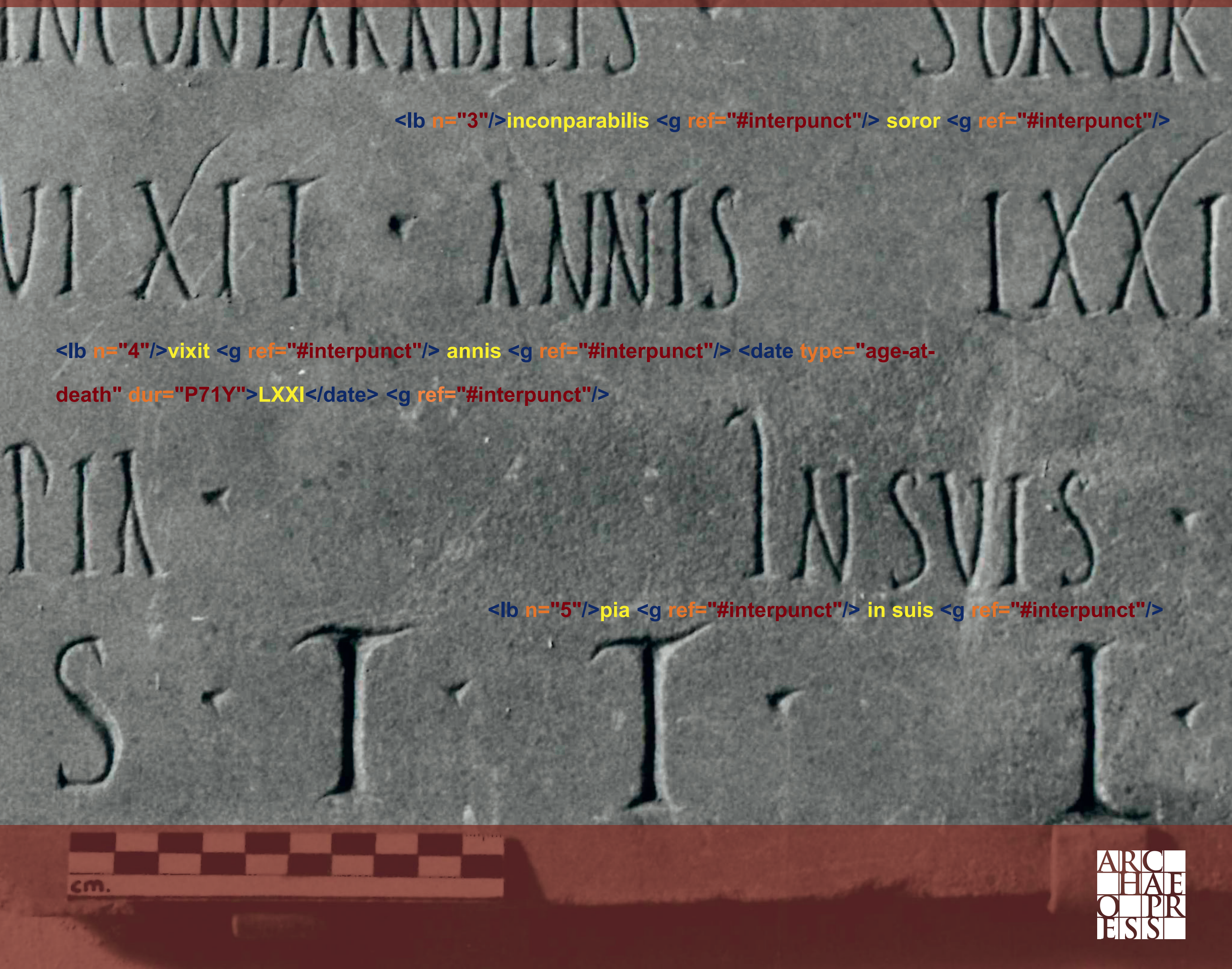
Epigraphy in the Digital Age

Opportunities and Challenges in the Recording, Analysis and Dissemination of Inscriptions

edited by

Isabel Velázquez Soriano and David Espinosa Espinosa



ARCHAEOPRESS

Epigraphy in the Digital Age

Epigraphy in the Digital Age

Opportunities and Challenges in the Recording, Analysis and Dissemination of Inscriptions

edited by

Isabel Velázquez Soriano
and
David Espinosa Espinosa

Archaeopress Archaeology

Oxford 2021

ARCHAEOPRESS PUBLISHING LTD
Summertown Pavilion
18-24 Middle Way
Summertown
Oxford OX2 7LG

www.archaeopress.com

ISBN 978-1-78969-987-6
ISBN 978-1-78969-988-3 (e-Pdf)

Cover image: *CIL* II²/7, 522 (Córdoba, Spain), archive of the Centro *CIL* II (University of Alcalá).

Epigraphy in the Digital Age. Opportunities and Challenges in the Recording, Analysis and Dissemination of Inscriptions has been funded within the research projects "Archivo virtual para las investigaciones sobre patrimonios epigráficos medievales de la Comunidad de Madrid (AVIPES-CM)" (H2019/HUM-5742, Comunidad de Madrid – Fondo Social Europeo) and "*Epigraphica* 3.0: Hacia la creación y diseño de un corpus digital de inscripciones latinas de la provincia de Ourense" (ED481D2017/013, Xunta de Galicia). Also, this book benefited from the funding provided by the research project "Corpus de textos e inscripciones hispano-latinos sobre arquitectura religiosa y civil altomedieval (CITHARA)" (PR87/19-22659, Complutense University of Madrid – Banco Santander) and the research group "Ciudades Romanas (GICIUR)" (Ref. no. 930692)" (Complutense University of Madrid).

Comunidad de Madrid

European Union
European Social Fund

XUNTA DE GALICIA

UNIVERSIDAD COMPLUTENSE MADRID

In Memoriam Prof. Joaquín L. Gómez-Pantoja Fernández-Salguero
(1953–2020)

Contents

Chapter 21

EPIHUM, a Database for Renaissance Epigraphy from Portugal and Spain

Manuel Blázquez-Ochando and Manuel Ramírez-Sánchez

Abstract: This chapter introduces the project 'Escritura expuesta y poder en España (siglos XVI-XVII): catálogo epigráfico online' (EPIHUM Project), the first epigraphic database of inscriptions from Portugal and Spain in the Renaissance (15th and 16th centuries), which gathers together inscriptions in any language of the Iberian Peninsula during that time (Latin, Spanish and Portuguese). To date, the study of inscriptions of this period has been very limited, and has usually been embedded in studies of architecture or sculpture of these centuries. There is no detailed study of the period's inscriptions whether from Spain or from Portugal, even though both countries have a long-standing tradition in the study of epigraphy of ancient and medieval times. The EPIHUM database will be multi-platform, multi-user, extensible and with expanded description capabilities, as well as compatible with the standards of the semantic web in RDF format, allowing file sharing in EpiDoc and TEI XML formats. The new system for cataloguing, recording and managing epigraphic inscriptions will be the only one of its kind, not just because of the information it will contain (unpublished inscriptions of the 15th and 16th centuries in Europe), but also due to its technological advances in information processing and retrieval techniques.

Keywords: Renaissance, digital epigraphy, EpiDoc, linked data, interoperability.

Introduction

The study of humanistic epigraphy in Spain and Portugal lacks a reference work that includes a catalogue of all inscriptions collected to date. In contrast to other historical periods between Classical Antiquity and the Middle Ages, which have been scientifically studied for more than a century and can draw from a long-standing tradition of variously extensive catalogues, the corpus of humanistic inscriptions of the Iberian Peninsula lacks a focused examination of the study of 'exposed writing'.[1] Up till now, the publication of humanistic inscriptions in the Iberian Peninsula has been tightly bound to the study of architectural or sculptural supports, analysed by researchers working in the History of Art such as María José Redondo Cantera.[2] Worthy of note among the scarce epigraphic catalogues published in Spain in this period is Maria José Rubio's work focused on inscriptions from Alcalá de Henares,[3] in which she included those from the modern era. Based on insights from the social history of written culture in the Renaissance, study of these written testimonies has been a fertile area for investigation in recent decades. Exemplary of this area are the works of Antonio Castillo, which have analysed the close-knit relationship between writing and political power in the city of Alcalá de Henares during the 15th and 16th centuries.[4] Also in this area are the works of Francisco Gimeno,[5] in which he examines the gradual process by which Gothic graphic forms were replaced by 'Roman capitals', as they were known in some documents of the period. For the same reasons as in Spain, Portugal's Latin and Romance-language epigraphic texts of the time were not systematically studied either, though there are a few works worthy of mention due to their inclusion of some humanistic inscriptions.[6] In addition to these publications, there are also some recent ones which have addressed the research opportunities afforded by certain Iberian epigraphic programmes,[7] such as those at the Monastery of San Lorenzo de El Escorial or the Portuguese city of Évora.[8]

Given the lack of a catalogue of humanistic inscriptions from the Iberian Peninsula that might interweave historical investigative interests with artistic or philological ones, and that might do so through a published corpus that gives researchers access to the sum total of surviving 16th- and 17th-century inscriptions, an interdisciplinary team of Spanish and Portuguese researchers has worked under the aegis of the EPIHUM project since 2016 to publish just such an epigraphic catalogue.[9] It is clear that a project of this nature could and should situate itself within the ambit of the Digital Humanities, by way of a tailor-made database that draws on the experience of more than two decades of similar projects, covering Classical Antiquity up to the Middle Ages.[10] In effect, epigraphic

1 Petrucci 1986.
2 Redondo Cantera 1987.
3 Rubio 1994.
4 Castillo Gómez 1997; 2000; 2009.
5 Gimeno Blay 2002; 2005; 2007.
6 Cordeiro de Sousa 1983.
7 Ramírez-Sanchez 2012.
8 Ramírez-Sanchez 2017.
9 EPIHUM Project (https://www.epigrafiahumanistica.eu, accessed on 4 June 2020).
10 Ramírez-Sanchez 2018.

databases have up till now mostly covered Ancient Greek and Roman epigraphy, and to a lesser extent that of the medieval period. These were all created due to the Internet's development in the 1990s,[11] and barring exceptions like the *Epigraphische Datenbank Heidelberg*, which has in recent years responded to the most demanding requirements of such online resources, most are invisible to Web 3.0, despite the abundant information and economic resources dedicated to them.[12]

Having paved the way for what is Spain and Portugal's first humanistic epigraphy database as well as Europe's first in the field of Renaissance epigraphy, the EPIHUM project's challenge has become to incorporate the vast potential of the Semantic Web and Linked Data into this type of resource,[13] and thus expand the so-called Digital Humanities.[14] For several years now, projects such as EAGLE have been working precisely along this track,[15] but their results have been limited to Ancient epigraphy and have not evolved into subsequent projects.[16] In addition, the EPIHUM project aims to push forward collaborative work in the recording, documenting and dissemination of the epigraphic heritage, following the lead of projects undertaken outside the field of epigraphic research.[17]

EPIHUM's database gathers together documentation currently published in diffuse manner – mainly in monographs or journals of local and regional history – when not simply ignored by historiography, and it therefore facilitates access to historical sources that might otherwise be hard to find. In this chapter we present the main characteristics of an epigraphic database built according to the principles of information science, thus permitting an organisational model based on epigraphic classification, its technical and physical characterisation, its dating, geolocation, photographic management, epigraphical-archival analysis, definition of authorities and applied information retrieval systems. Given the limited space available, we will focus here on aspects related to the table structure of the EPIHUM project database, leaving comprehensive explanation of the entire system, including full description of all modules and functions, to another work.[18]

Table Structure

Using MySQL, a relational design has been developed to record and relate authorities, sources of information and locations with the epigraphic records of the catalogue, which is also known as the main table and is classified using tags. A details table has also been added to facilitate the recording of observations about parts of the document based on its photographs. Added to all this is the inclusion of a NoSQL table system, based on multidimensional arrays that serve to classify the contents of the tables – for example, the type, occupation and functions of authorities, types of document or material, inscription techniques, typefaces, languages, religious cults, countries, conservation statuses, types of locations, or publication status of the record (Figure 1).

The 'authorities' Table

This is considered an essential table for registering all types of personal authorities, entities, families or organisations, in accordance with ISAAR-CPF standards (Table 1). The recording of authorities directly affects the inscriptions studied by the EPIHUM project, thus serving as an indispensable access point.

Table 1

The 'authorities' Table	
id	Autoincremented primary key field – Unique record identifier, assigned internally by the system.
datereg	Automatic date field – Date of recording of the authority. This is the date the authority is inserted into the system, in order to differentiate it from other authorities.
tg	Variable character field (length 25) – Relational type – Relationship with the 'authorities' table – Records the highest level of authority identifier in a hierarchy, for example, the social class. This always refers to authority identifiers, and in no case identifiers from other tables.
historical	Variable character field (length 25) – Records a Boolean value of 0 or 1 as an indicator of historical authority. In this way, reference can be made to an individual recognized by other sources.
typeauthority	Variable character field (length 250) – Relational type – Relationship with NoSQL table 'arraylist.typeauthority.php' – Records the type of authority among those available in a list of parameters. When a parameter of the NoSQL table is modified, an automatic modification process is triggered in each record of the database whose type of authority matches that case.

[11] Bodel 2012.
[12] Ramírez-Sánchez 2010.
[13] Bodard 2010; Blanke *et al.* 2012.
[14] Blázquez-Ochando 2021.
[15] EAGLE Europeana (http://www.eagle-network.eu, accessed on 4 June 2020).
[16] Orlandi *et al.* 2014.
[17] Durán Salado and Ortiz Lozano 2017; Ramírez-Sánchez 2021.
[18] Blázquez-Ochando 2021.

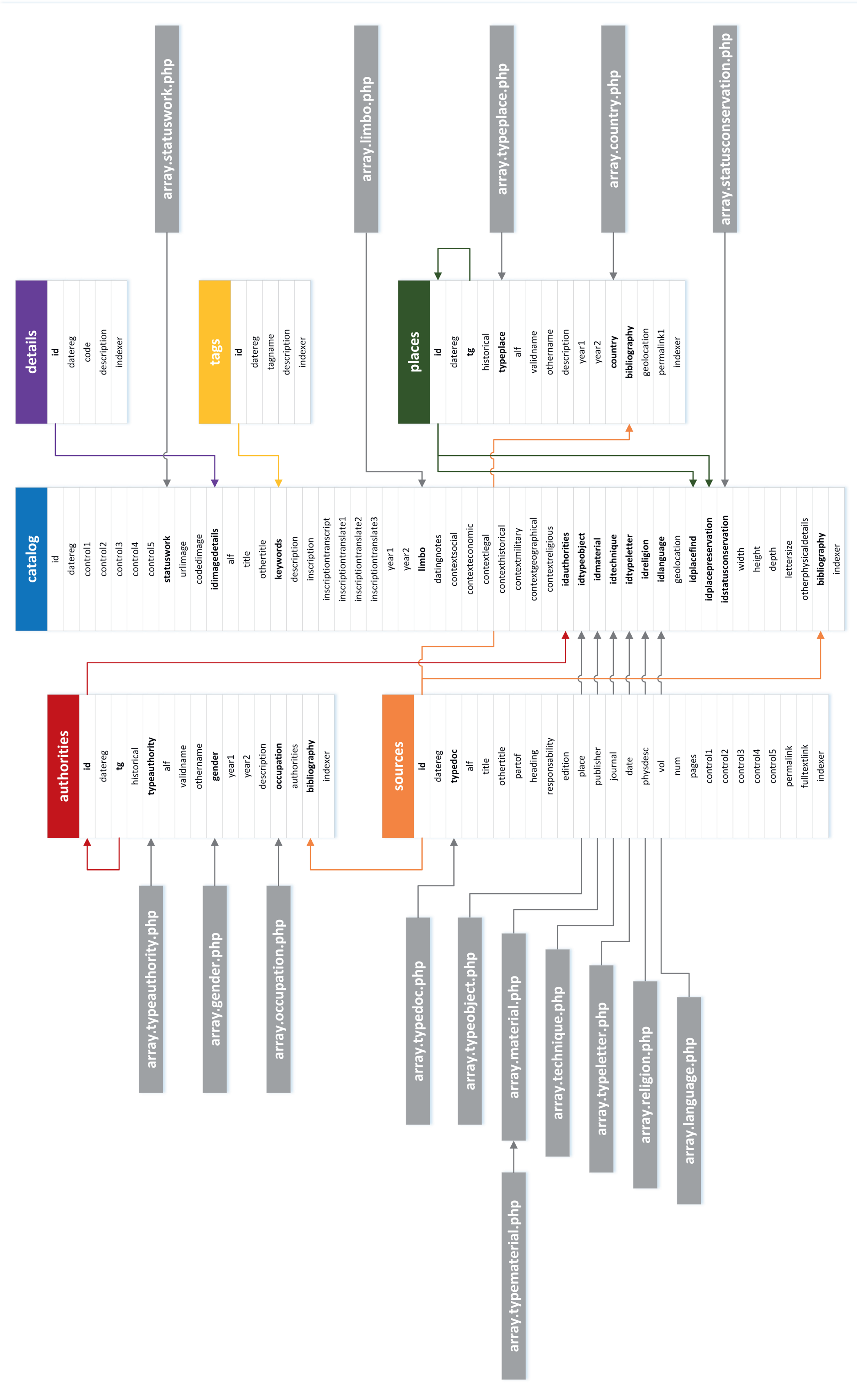


Figure 1. Relational schema of database tables (the schema can be viewed on https://mblazquez.es/wp-content/uploads/esquemaRelacionalTablasEpibase.png).

The 'authorities' Table	
alf	Variable character field (length 25) – The first characters of the authority's title or name are recorded and rules of alphabetization applied, such as the elimination of stop words, in order to facilitate natural alphabetical ordering.
validname	Variable character field (length 500) – Record of the accepted name of the authority. This is by distinction from other names that are not accepted or standardized.
othername	Text field – Records the rest of the names or designations by which an authority is known.
gender	Variable character field (length 25) – Records the gender of the individual authority, or indicates that it is a legal entity or person.
year1	Variable character field (length 25) – Date of birth or commencement.
year2	Variable character field (length 25) – Date of death or expiration.
description	Text field – Historical or biographical information about the authority.
occupation	Text field – Relational type – Records the occupations or predefined functions of the authority. These are defined in the parameter list. When a parameter in the NoSQL file is modified, an automatic modification process is triggered in each record of the database whose type of authority matches that case.
authorities	Text field – Relational type – Records the identifiers of related authorities.
bibliography	Long text field – Relational type – Relationship with the 'sources' table – Records the identifiers of the information sources and bibliography related to the authority in question.
indexer	Long text field – Full-text indexing type – Collection of all text recorded for an authority, including all forms of names whether valid or unauthorized, dates, description, functions and occupations. The text is processed to facilitate the retrieval of information. This involves removing stop words, removing tabs and double spaces, conversion to lowercase, removing accents and punctuation marks, etc.

The 'authorities' Table
SQL Table Code

```
CREATE TABLE `authorities` (
 `id` int(11) NOT NULL,
 `datereg` timestamp NOT NULL DEFAULT CURRENT_
TIMESTAMP,
 `tg` varchar(25) DEFAULT NULL,
 `historical` varchar(25) DEFAULT NULL,
 `typeauthority` varchar(250) DEFAULT NULL,
 `alf` varchar(25) DEFAULT NULL,
 `validname` varchar(500) DEFAULT NULL,
 `othername` text,
 `gender` varchar(25) DEFAULT NULL,
 `year1` varchar(25) DEFAULT NULL,
 `year2` varchar(25) DEFAULT NULL,
 `description` text,
 `occupation` text,
 `authorities` text NOT NULL,
 `bibliography` longtext,
 `indexer` longtext NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 ROW_
FORMAT=COMPACT;

ALTER TABLE `authorities`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `authorities` ADD FULLTEXT KEY `indexer`
(`indexer`);

ALTER TABLE `authorities`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

The 'catalog' Table

This is the main table of the programme. Its object is the recording of the project's inscriptions as well as relating all possible access points and details. For this reason, its design is indebted to the metadata and description fields of the main cataloguing norms and standards, namely ISBD, MARC, TEI, ISAD-G and Dublin Core. This table's development has led to the delineation of common descriptive elements essential to the identification and elucidation of any epigraphic object; these are its characterisation, typification, attributed titles, use of keywords, tags, review of inscriptions and evaluations, translation, interpretation, definition of contexts, date ranges, georeferencing and relationship with authorities, sources and places (Table 2).

Table 2

The 'catalog' Table	
id	Autoincremented primary key field – Unique record identifier, assigned internally by the system.
datereg	Automatic date field – Date of recording of the epigraphic document.

The 'catalog' Table	
control1	Variable character field (length 25) – Field that records the registration code of the epigraphic document for the database. The nomenclature is the prefix 'epihum' followed by the 'id' number assigned by the system. This code is used to create permanent links for the catalogue cards of the epigraphic documents recorded by the system.
control2	Variable character field (length 25) – Auxiliary control fields. These serve to record other control codes or identifiers, from other sources or reference databases in which the epigraphic document in question has been partially or totally described. Although they are fields that will not be used in the early stages of the project, they can be used later to automate the referencing of epigraphic documents in the databases of other similar projects.
control3	
control4	
control5	
statuswork	Variable character field (length 100) – Relational type – Relationship with NoSQL table 'arraylist.statuswork.php' – Records the status code of the catalogue card. The default values are 'In progress', 'Pending review' and 'Published'. Documents listed as 'In progress' are not displayed in the system's public OPAC catalogue online. Those marked 'Pending review' are awaiting review by editors and system administrators. Epigraphic documents listed as 'Published' can be retrieved from the public and internal OPAC.
urlimage	Variable character field (length 500) – Records the URL for a high-resolution image of the epigraphic document, which is linked to the catalogue card.
codedimage	Text field – Stores the image of the epigraphic document, encoded in base64, so that its digital preservation can be ensured.
idimagedetails	Variable character field (length 250) – Relational type – Relationship with the 'details' table – The identifiers of clippings of the epigraphic document image are recorded. These are the details that the cataloguer can generate with the selection tools provided by Epibase.

The 'catalog' Table	
alf	Variable character field (length 25) – The first characters from the title attributed to the epigraphic document or its transcription are recorded according to rules of alphabetization, such as the elimination of stop words, in order to facilitate natural alphabetical ordering.
title	Text field – Title of the epigraphic document, should it have one, or the title attributed by the researcher or cataloguer.
othertitle	Text field – Other forms of the title or other titles known to the researcher, in order to facilitate retrieval.
keywords	Text field – Relational type – Relationship with the 'tags' table – Records the identifiers of standardized tags. These act as descriptors or key words in the retrieval of the epigraphic document.
description	Long text field – The description of the epigraphic object or document is recorded, taking into account all its details and singularities. The text is indexed, along with the rest of the descriptive elements, for full text retrieval.
inscription	Long text field – Represents the reading of the epigraphic document. These are the original inscriptions, as they appear on the object, without acronyms or full transcription.
inscriptiontranscript	Long text field – Records a complete transcription of the inscribed text, expanding all acronyms and abbreviations, as well as restoring parts of the text that may be lost.
inscriptiontranslate1	Long text field – The transcript's translation into Spanish, Portuguese and English is recorded, so that its full text might be retrieved.
inscriptiontranslate2	
inscriptiontranslate3	
year1	Variable character field (length 25) – Start year when dating by ranges.
year2	Variable character field (length 25) – End year when dating by ranges.
limbo	Variable character field (length 25) – Relational type – Relationship with NoSQL table 'arraylist.limbo.php' – Records date range extremes defined by researchers, according to the custom parameter 'limbo'.
datingnotes	Text field – Annotations regarding dating are recorded, in order to clarify or disambiguate the status of the inscription.

The 'catalog' Table	
contextsocial	Text field – The social context of the inscription, if appropriate.
contexteconomic	Text field – If applicable, a description of the economic situation in which the inscription was made, or the context to which the text might refer.
contextlegal	Text field – The legal considerations of the document are described, if it is a law, norm or regulation, or refers to the context in which the document was discovered.
contexthistorical	Text field – Describes the historical context, or the events referred to in the text of the inscription.
contextmilitary	Text field – Description of the military aspects, if applicable, whether regarding the content of the inscription or its chronological or historical context.
contextgeographical	Text field – If the inscription refers to a place name, the field can be used to clarify the toponymy or location alongside any other context.
contextreligious	Text field – Examines, if applicable, any religious aspect of the document, or its historical, geographical or chronological context relating to religion or religious events.
idauthorities	Variable character field (length 250) – Relational type – Relationship with the 'authorities' table – Stores the identifier of the authorities related to the epigraphic document.
idtypeobject	Variable character field (length 25) – Relational type – Relationship with NoSQL table 'arraylist.typeobject.php' – Stores the identifier of the epigraphic object type, according to the parameters defined by the researcher.
idmaterial	Variable character field (length 25) – Relational type – Relationship with NoSQL table 'arraylist.material.php' – Records the identifier of the type of material of the epigraphic object, according to the parameters defined by the researcher.
idtechnique	Variable character field (length 25) – Relational type – Relationship with NoSQL table 'arraylist.technique.php' – Records the identifier of the technique employed to inscribe the epigraphic object, according to the parameters defined by the researcher.

The 'catalog' Table	
idtypeletter	Variable character field (length 25) – Relational type – Relationship with NoSQL table 'arraylist.typeletter.php' – Records the identifier of the type of writing used in the epigraphic document, according to the parameters defined by the researcher.
idreligion	Variable character field (length 25) – Relational type – Relationship with NoSQL table 'arraylist.religion.php' – Records the identifier of the religion or religious context of the epigraphic document, according to the parameters defined by the researcher.
idtypeinscription	Variable character field (length 25) – Relational type – Relationship with NoSQL table 'arraylist.typeinscription.php' – Records the identifier of the type of inscription present in the document, according to the parameters defined by the researcher.
idlanguage	Variable character field (length 25) – Relational type – Relationship with NoSQL table 'arraylist.language.php' – Records the identifier of the language in which the epigraphic document is written, according to the parameters defined by the researcher.
geolocation	Variable character field (length 500) – Records the coordinates of the primary exhibition context of the inscription.
idplacefind	Variable character field (length 25) – Relational type – Relationship with the 'places' table – Stores the identifier of the primary exhibition location of the inscription.
idplacepreservation	Variable character field (length 25) – Relational type – Relationship with the 'places' table – Stores the identifier of the place where the object is currently preserved (if different to the previous field).
idstatusconservation	Variable character field (length 25) – Relational type – Relationship with NoSQL table 'arraylist.statusconservation.php' – Records the conservation status of the inscription, according to the parameters defined by the researcher.
width	Variable character field (length 100) – width of the epigraphic support in centimetres.
height	Variable character field (length 100) – height of the epigraphic support in centimetres.

The 'catalog' Table	
depth	Variable character field (length 100) – depth of the epigraphic support in centimetres.
lettersize	Variable character field (length 100) – size of the letters of the text.
otherphysicaldetails	Medium text field – Other details of physical description.
bibliography	Medium text field – Relational type – Relationship with the 'sources' table – Stores the identifiers of the documents or information sources directly related to the epigraphic object or document.
indexer	Long text field – Full-text indexing type – Collection of all text on the catalogue card, including all defined titles, tags, description, reading notes, inscription, transcription, translations of the transcription, annotations and contexts. The text is processed to facilitate the retrieval of information.

SQL Table Code

```
CREATE TABLE `catalog` (
  `id` int(11) NOT NULL,
  `datereg` timestamp NOT NULL DEFAULT CURRENT_
TIMESTAMP,
  `control1` varchar(25) DEFAULT NULL,
  `control2` varchar(25) DEFAULT NULL,
  `control3` varchar(25) DEFAULT NULL,
  `control4` varchar(25) DEFAULT NULL,
  `control5` varchar(25) DEFAULT NULL,
  `statuswork` varchar(100) DEFAULT NULL,
  `urlimage` varchar(500) DEFAULT NULL,
  `codedimage` varchar(500) DEFAULT NULL,
  `idimagedetails` varchar(250) DEFAULT NULL,
  `alf` varchar(25) DEFAULT NULL,
  `title` text,
  `othertitle` text,
  `keywords` text,
  `description` longtext,
  `inscription` longtext,
  `inscriptiontranscript` longtext,
  `inscriptiontranslate1` longtext,
  `inscriptiontranslate2` longtext,
  `inscriptiontranslate3` longtext,
  `year1` varchar(25) DEFAULT NULL,
  `year2` varchar(25) DEFAULT NULL,
  `limbo` varchar(25) DEFAULT NULL,
  `datingnotes` text,
  `contextsocial` text,
  `contexteconomic` text,
  `contextlegal` text,
  `contexthistorical` text,
  `contextmilitary` text,
  `contextgeographical` text,
  `contextreligious` text,
  `idauthorities` varchar(250) DEFAULT NULL,
  `idtypeobject` varchar(25) DEFAULT NULL,
  `idmaterial` varchar(25) DEFAULT NULL,
  `idtechnique` varchar(25) DEFAULT NULL,
  `idtypeletter` varchar(25) DEFAULT NULL,
  `idreligion` varchar(25) DEFAULT NULL,
  `idtypeinscription` varchar(25) DEFAULT NULL,
  `idlanguage` varchar(25) DEFAULT NULL,
  `geolocation` varchar(500) DEFAULT NULL,
  `idplacefind` varchar(25) DEFAULT NULL,
  `idplacepreservation` varchar(25) DEFAULT NULL,
  `idstatusconservation` varchar(25) DEFAULT NULL,
  `width` varchar(100) DEFAULT NULL,
  `height` varchar(100) DEFAULT NULL,
  `depth` varchar(100) DEFAULT NULL,
  `lettersize` varchar(100) DEFAULT NULL,
  `otherphysicaldetails` mediumtext,
  `cat1` varchar(500) DEFAULT NULL,
  `cat2` varchar(500) DEFAULT NULL,
  `bibliography` mediumtext,
  `indexer` longtext NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 ROW_
FORMAT=COMPACT;

ALTER TABLE `catalog`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `catalog` ADD FULLTEXT KEY `indexer`
(`indexer`);

ALTER TABLE `catalog`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

The 'details' Table

This table is intended to record the details of photographs of epigraphic objects, so that their descriptions might serve as new access points to the main document (Table 3). Therefore, this table is auxiliary to 'catalog', enabling a direct relationship between the main images and their clippings, ensuring the descriptions are as exhaustive and granular as possible. On the other hand, the details described can be used for the purposes of retrieval, to search within the system for the entire documentary set. This offers an opportunity to make an approximate analysis of the parts by their common denominators, based on evidence and facts described within the system.

Table 3

The 'details' Table	
id	Autoincremented primary key field – Unique record identifier, assigned internally by the system.
code	Text field – Records the detail reference code, composed of the prefix "epigraphic" followed by the detail number and a random 10-character alphanumeric code. In addition, it corresponds to the name of the image's clipping, making retrieval simple and effective.

The 'details' Table	
description	Medium text field – Stores a description of the detail – an explanation of the elements observed in the clipping from the main image.
indexer	Long text field – Full-text indexing type – Collection of all the detail descriptions for exhaustive retrieval. The text is processed in the same way as previous cases.

SQL Table Code

```
CREATE TABLE `details` (
 `id` int(11) NOT NULL,
 `code` text,
 `description` mediumtext,
 `indexer` longtext NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

ALTER TABLE `details`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `details` ADD FULLTEXT KEY `indexer`
(`indexer`);

ALTER TABLE `details`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

The 'places' Table

This table collects the places, locations, sites or whereabouts, whether geographical, physical, political, historical, regional, etc., which are directly related to the recorded inscriptions (Table 4). The structure is very similar to the 'authorities' table, since it has much the same fields, with the addition of recording coordinates, country of the location, and permanent link to other georeferencing projects external to the system.

Table 4

The 'places' Table	
id	Autoincremented primary key field – Unique record identifier, assigned internally by the system.
datereg	Automatic date field – Date of recording of the place. This is the date the place is inserted, in order to differentiate it from the others.
tg	Variable character field (length 25) – Relational type – Relationship with the "places" table – Records the highest level of authority identifier in a hierarchy, for example, a city relative to a region, or a region relative to a country. This always refers to place identifiers, and in no case identifiers from other tables.
historical	Variable character field (length 25) – Records a Boolean value of 0 or 1 as an indicator of a place or historical location. In this way, reference can be made to a historically recognized space.
typeplace	Variable character field (length 250) – Relational type – Relationship with NoSQL table "arraylist.typeplace.php" – Records the identifier of the type of place, according to parameters defined by the researcher. When a parameter of the NoSQL table is modified, an automatic modification process is triggered in each record of the database whose type of authority matches that case.
alf	Variable character field (length 25) – The first characters of the place's name are recorded and rules of alphabetization applied, such as the elimination of stop words, in order to facilitate natural alphabetical ordering.
validname	Text field – Accepted name for the place or location.
othername	Text field – Other forms of the name or of mentioning the place in question, in order to facilitate its retrieval.
description	Text field – Outline of the place's history and its specificities.
year1	Variable character field (length 25) – Starting point of dating range, which could be associated with the moment the place begins to be named, or its date of discovery, or any other possibility determined by the researcher.
year2	Variable character field (length 25) – End point of dating range. It may be associated with the naming of the place.
country	Variable character field (length 25) – Country where the place is found.
bibliography	Long text field – Relational type – Relationship with the "sources" table – Records the identifiers of the information sources and bibliography related to the place in question.
geolocation	Variable character field (length 500) – Records the coordinates of the place or location.
permalink1	Variable character field (length 500) – Permanent link to a source of information relating to the place described.
indexer	Long text field – Full-text indexing type – Collection of all text for exhaustive retrieval, including the name of the place, other designations, description and country. The text is processed in the same way as previous cases.

The 'places' Table
SQL Table Code

```
CREATE TABLE `places` (
 `id` int(11) NOT NULL,
 `datereg` timestamp NOT NULL DEFAULT CURRENT_
TIMESTAMP,
 `tg` varchar(25) DEFAULT NULL,
 `historical` varchar(25) DEFAULT NULL,
 `typeplace` varchar(250) DEFAULT NULL,
 `alf` varchar(25) DEFAULT NULL,
 `validname` text,
 `othername` text,
 `description` text,
 `year1` varchar(25) DEFAULT NULL,
 `year2` varchar(25) DEFAULT NULL,
 `country` varchar(25) DEFAULT NULL,
 `address` text,
 `maplocation` text,
 `bibliography` longtext,
 `geolocation` varchar(500) DEFAULT NULL,
 `permalink1` varchar(500) DEFAULT NULL,
 `indexer` longtext NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 ROW_
FORMAT=COMPACT;

ALTER TABLE `places`
 ADD PRIMARY KEY (`id`);
ALTER TABLE `places` ADD FULLTEXT KEY `indexer`
(`indexer`);

ALTER TABLE `places`
 MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

The 'sources' Table

All sources of information are recorded in this table, including bibliographic references, scientific articles, monographs, conference proceedings, web resources, etc. (Table 5). Each of these sources directly or indirectly cites an authority or inscription included in the system. In this way, it functions as an auxiliary to the 'authorities' table and the very catalogue of inscriptions itself.

Table 5

The 'sources' Table	
id	Autoincremented primary key field – Unique record identifier, assigned internally by the system.
datereg	Automatic date field – Date of recording of the document. This is the date the new source or bibliographic reference is inserted into the system, in order to differentiate it from the rest.
typedoc	Variable character field (length 250) – Relational type – Relationship with NoSQL table "arraylist.typedoc.php" – Records the identifier of the type of document, according to parameters defined by the researcher. When a parameter of the NoSQL table is modified, an automatic modification process is triggered in each record of the database whose type of authority matches that case.
alf	Variable character field (length 25) – The first characters of the title, or in its absence, of the author's name are recorded and rules of alphabetization applied, such as the elimination of stop words, in order to facilitate natural alphabetical ordering.
title	Variable character field (length 750) – Title of the document itself.
othertitle	Variable character field (length 750) – Other designations or forms of the title, including attributed or parallel titles.
partof	Variable character field (length 750) – When the document in question is a part of another document, for example, a chapter of a book.
heading	Variable character field (length 500) – Document heading. Depending on the case, the document may be headed by author or title, in accordance with ISBD standards.
responsibility	Variable character field (length 750) – Statement of responsibility, authors, secondary authors and other statements of personal or corporate authority regarding who participated in the editing or preparation of the document.
edition	Variable character field (length 100) – Document edition data, such as the edition number, its editors, possible reprints, corrections, etc.
place	Variable character field (length 100) – Place of publication.
publisher	Variable character field (length 500) – Publisher.
journal	Variable character field (length 500) – Journal.
date	Variable character field (length 100) – Date of publication.
physdesc	Variable character field (length 500) – Physical description.
vol	Variable character field (length 25) – Volume (applicable if a journal article).
num	Variable character field (length 25) – Number (applicable if a journal article).
pages	Variable character field (length 50) – Page range.
control1	Variable character field (length 25) – ISBN – International Standard Book Number.

The ‘sources’ Table	
control2	Variable character field (length 25) – NIPO – Identification Number Official Publications.
control3	Variable character field (length 25) – ISSN – International Standard Serial Number.
control4	Variable character field (length 25) – DL – Legal Deposit (*Depósito Legal*).
control5	Variable character field (length 25) – Non-specific control code left open and available for any other numbering not previously covered that might serve to identify the bibliographic reference.
doi	Variable character field (length 500) – DOI – Digital Object Identifier.
permalink	Variable character field (length 500) – Used to add a permanent link, especially when it comes to journal articles and information sources on the web.
fulltextlink	Variable character field (length 500) – Holds a link to the information source where the full text is provided.
indexer	Long text field – Full-text indexing type – Collection of all text for exhaustive retrieval, including the title, other forms of the title, responsibility statement, edition, publication data, publisher, journal and physical description. The text is processed in the same way as previous cases.

SQL Table Code

```
CREATE TABLE `sources` (
  `id` int(11) NOT NULL,
  `datereg` timestamp NOT NULL DEFAULT CURRENT_
TIMESTAMP,
  `typedoc` varchar(25) NOT NULL,
  `alf` varchar(25) DEFAULT NULL,
  `title` varchar(750) DEFAULT NULL,
  `othertitle` varchar(750) DEFAULT NULL,
  `partof` varchar(750) DEFAULT NULL,
  `heading` varchar(500) DEFAULT NULL,
  `responsability` varchar(750) DEFAULT NULL,
  `edition` varchar(100) DEFAULT NULL,
  `place` varchar(100) DEFAULT NULL,
  `publisher` varchar(500) DEFAULT NULL,
  `journal` varchar(500) DEFAULT NULL,
  `date` varchar(100) DEFAULT NULL,
  `physdesc` varchar(500) DEFAULT NULL,
  `vol` varchar(25) DEFAULT NULL,
  `num` varchar(25) DEFAULT NULL,
  `pages` varchar(50) DEFAULT NULL,
  `control1` varchar(25) DEFAULT NULL,
  `control2` varchar(25) DEFAULT NULL,
  `control3` varchar(25) DEFAULT NULL,
  `control4` varchar(25) DEFAULT NULL,
  `control5` varchar(25) DEFAULT NULL,
  `doi` varchar(500) DEFAULT NULL,
  `permalink` varchar(500) DEFAULT NULL,
  `fulltextlink` varchar(500) DEFAULT NULL,
  `indexer` longtext NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 ROW_
FORMAT=COMPACT;
```

The ‘sources’ Table

```
ALTER TABLE `sources`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `sources` ADD FULLTEXT KEY `indexer`
(`indexer`);

ALTER TABLE `sources`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

The ‘tags’ Table

This table is used to record all the tags, labels or descriptors used to classify documents (Table 6). The reasoning behind the use of tags is to make quickly and efficiently classifying documents as simple and dynamic a process as possible, and maintain a set vocabulary of terms. This is more versatile than other traditional classification systems, such as thesauruses, subject headings or decimal classifications. However, it also has some drawbacks inherent to its methodology, such as the difficulty for users of the application to establish convention, consensus and standardisation of the specialised vocabulary used, in this case that belonging to epigraphy. Nevertheless, once said vocabulary is fixed, the retrieval and classification of documents tends to be swiftly and reliably performed.

Table 6

The ‘tags’ Table	
id	Autoincremented primary key field – Unique record identifier, assigned internally by the system.
datereg	Automatic date field – Date of recording of the authority. This is the date the tag is inserted into the system, in order to differentiate it from the rest.
tagname	Variable character field (length 500) – Name of the tag, or a descriptor or accepted term.
description	Text field – If apt, a description of the term or descriptor, in order to improve retrieval.
indexer	Long text field – Full-text indexing type – Collection of all text for exhaustive retrieval, including the descriptor or tag along with its description. The text is processed in the same way as previous cases.

The 'tags' Table

SQL Table Code

```
CREATE TABLE `tags` (
 `id` int(11) NOT NULL,
 `datereg` timestamp NOT NULL DEFAULT CURRENT_
TIMESTAMP,
 `tagname` varchar(500) DEFAULT NULL,
 `description` text,
 `indexer` longtext NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

ALTER TABLE `tags`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `tags` ADD FULLTEXT KEY `indexer`
(`indexer`);

ALTER TABLE `tags`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

NoSQL Tables

When MySQL-style database tables are not appropriate for consideration, NoSQL-type data structures are an option. This is because they do not make use of Structured Query Language, also commonly known as SQL. In this way, the MySQL database is geared towards administration of epigraphic records, sources of bibliographic information, authorities, details and classification methods, and the NoSQL tables are focused on the treatment of auxiliary data, typologies and characterisation.

The structure of NoSQL tables is based on multidimensional arrays written in the PHP programming language. The following table shows an example of a multidimensional array composed of three fields, namely (Table 7): the 'id' field, which is the corresponding identification number; the 'name' field, which is the name visible in the system; and the 'value' field, which records the canonical name of the lettering type, without accents, punctuation marks or capital letters, presenting a standardised text that can be used for full-text retrieval.

Table 7

Example of a multidimensional array in a NoSQL table dedicated to typeface

```
<?php
$typeletter[] = array(id => "1", name => "Gótica", value =>
"gotica");
$typeletter[] = array(id => "2", name => "Humanística",
value => "humanistica");
$typeletter[] = array(id => "3", name => "Mixta", value =>
"mixta");
?>
```

In order to edit a data file, like the one in the previous table, a specialised editing programme was developed (Figure 2).

Figure 2. Interface of the NoSQL administration programme.

This programme enables the explicit editing of each NoSQL table, providing a graphical interface by way of forms, with options for creating records or editing existing ones. This is achieved by listing all the available records in the multidimensional array. Table 8 demonstrates this method. First, the text-indexing and normalisation function 'func.indexer.php' is loaded, which will be used later in the programme. Second, instructions for the form are included so that the user can conveniently operate upon the data files. To do this, the file 'arraylist.typeletter.php' is loaded, which contains the multidimensional array with records of typefaces. The number of available records is then counted. Next, the form is formatted as HTML so as to add or insert records into the aforementioned fields of the array. Note the autoincremented identifier, which is indicated in the variable '$ntypeletter'. After this, the values of the data in the array are listed using the loop 'foreach($typeletter as $one){}', in order to generate a new form for editing, modifying or if necessary deleting the typefaces already recorded.

Table 8

Fragment of the interface forms used for editing the NoSQL table

```
// Carga de funciones
include("func.indexer.php");

// typeletter Form
// ====================================>>>
include("arraylist.typeletter.php"); $ntypeletter =
count($typeletter); $ntypeletter++;
$string_typeletter .= "
<div class='parambox'>
<div class='pad20'>
<div style='font-size: 17px; margin-bottom: 9px;'>Tipo de
escritura</div>
<div>
<input value='Id' class='paramtitlecell'/>
```

Fragment of the interface forms used for editing the NoSQL table

```
<input value='Denominación' class='paramtitlecell'/>
</div>
<form action='$_SERVER[PHP_SELF]' method='post'>
<div>
<input type='text' name='id' value='$ntypeletter'
class='paramcellnew'/>
<input type='text' name='name'
placeholder='Denominación' class='paramcellnew'/>
<input type='hidden' name='value' placeholder='Código'
class='paramcellnew'/>
<input type='submit' name='newtypeletter'
class='newparam' value='Añadir nuevo'/>
</div>
</form>
<form action='$_SERVER[PHP_SELF]' method='post'>
";
$a=0;
if(count($typeletter) == 0) {} else {
		foreach($typeletter as $one){
				$string_typeletter .= "
<div>
<input type='hidden' id='del$a' name='status[]' value='0'/>
<input type='text' name='id[]' value='$one[id]'
class='paramcell'/>
<input type='text' name='name[]' value='$one[name]'
class='paramcell'/>
<input type='hidden' name='value[]' value='$one[value]'
class='paramcell'/>
<input type='checkbox' onclick=\"document.
getElementById('del$a').value='1';\"/>
</div>
				";
				$a++;
		}
}
$string_typeletter .= "<input type='submit'
name='savetypeletter' class='saveparam' value='Guardar
cambios'/><input type='submit' name='deletetypeletter'
class='deleteparam' value='Borrar marcados'/></form></
div></div>";
echo "$string_typeletter";
```

Once the operations interface is generated, the programme runs the method to save the information, in case any of the records were modified. This can be verified in the code in Table 9. The programme receives the inputted data from the form, corresponding to the 'id', 'name' and 'value' fields of each record. This is inclusive of the field to be modified and regenerates the multidimensional array in the file 'arraylist.typeletter.php', the content of which resembles the data structure previously demonstrated in Table 7.

Table 9

Fragment of the NoSQL programme for saving changes to records

```
// typeletter Save
// ==================================>>>
if($_POST[savetypeletter]){

		$id = $_POST['id'];
		$name = $_POST['name'];
		$value = $_POST['value'];

		$fp = fopen("arraylist.typeletter.php", "w");
		fwrite($fp, "<?php\n\n");
		foreach($id as $key => $n) {

// Valor normalizado
$newindexvalue = indexlight($name[$key]);
$newindexvalue = strtolower($newindexvalue);
$newindexvalue = preg_replace("/ /", "", $newindexvalue);

fwrite($fp, "\$typeletter[] = array(id => \"$id[$key]\", name
=> \"$name[$key]\", value => \"$newindexvalue\");\n");

		}
		fwrite($fp, "\n\n?>");
		fclose($fp);
		echo "<script>window.location.href = window.
location.href;</script>";

}
```

Another essential function of the programme is the insertion of records, which can be seen in Table 10. In this case, the data file 'arraylist.typeletter.php' is regenerated with the values previously available, plus the last one inserted by the user. Once the operation is complete, the information listed onscreen is updated.

Table 10

Fragment of the NoSQL programme responsible for inserting records into the table

```
// typeletter New
// ==================================>>>
if($_POST[newtypeletter]){

$fp = fopen("arraylist.typeletter.php", "w");
fwrite($fp, "<?php\n\n");
foreach($typeletter as $one){
fwrite($fp, "\$typeletter[] = array(id => \"$one[id]\", name
=> \"$one[name]\", value => \"$one[value]\");\n");
}
```

Fragment of the NoSQL programme responsible for inserting records into the table

// Valor normalizado
$newindexvalue = *indexlight*(**$_POST**[*name*]);
$newindexvalue = *strtolower*(**$newindexvalue**);
$newindexvalue = *preg_replace*("/ /", "", **$newindexvalue**);
 	
fwrite(**$fp**, "\$typeletter[] = array(id => \"**$_POST**[id]\", name => \"**$_POST**[name]\", value => \"**$newindexvalue**\");\n");
fwrite(**$fp**, "\n\n?>");
fclose(**$fp**);
 	
echo "<script>window.location.href = window.location.href;</script>";
 	
}

One final function of the programme is to delete records. In this case, the data file is rewritten, by going over the list of values in the array and reinscribing those values that do not correspond to the one inputted by the user. This is to say that so long as the identifier of the record to be deleted does not match the input, the array value is rewritten. In the opposite case, the record is removed, resulting in the effective erasure of the desired information. Once the rewrite is complete, the page is refreshed, displaying the list without that value. This method can be seen in Table 11.

Table 11

Fragment of the function for deleting records from NoSQL tables

// typeletter Delete
// ===================================>>>
if($_POST[*deletetypeletter*]){
 	
$id = **$_POST**['id'];
$name = **$_POST**['name'];
$value = **$_POST**['value'];
 	
$fp = *fopen*("arraylist.typeletter.php", "w");
fwrite(**$fp**, "<?php\n\n");
foreach($id as $key => **$n**) {
 	
if($status**[**$key] == "1"){} **else** {
fwrite(**$fp**, "\$typeletter[] = array(id => \"**$id**[**$key**]\", name => \"**$name**[**$key**]\", value => \"**$value**[**$key**]\");\n");
}
 	
}
fwrite(**$fp**, "\n\n?>");
fclose(**$fp**);
echo "<script>window.location.href = window.location.href;</script>";
 	
}

As has been explained, the key to managing NoSQL data files or tables lies in the rewriting of files using a lower level of programming, which is an efficient method due to its low execution cost, and a proven one, behaving correctly in its logical operations. However, there are additional editing mechanisms to these, related to updating the data in MySQL. This is to say that making modifications to the values in a NoSQL table might impact the main tables with which it is related. Such situations are highly important, especially when it comes to the modification and deletion of records. For example, if the 'Gothic' typeface is deleted, this means that all epigraphic documents characterised by 'Gothic' script would be orphaned of that auxiliary value. On the other hand, said change would not cause an update and the result would be a misrepresentation of information. Therefore, when any change is made the programme is instructed to modify the affected fields, using the MySQL replacement function, as shown in Table 12.

Table 12

SQL query for replacing values in MySQL
UPDATE catalog SET idtypeletter = REPLACE(idtypeletter, '3', '0') WHERE idtypeletter LIKE ('3');

The instruction in Table 12 corresponds to a SQL query that replaces the font in the main 'catalog' table, assuming that record '3' of the 'mixed' font was deleted from the NoSQL table or data file. In that case, the value is removed or substituted with the value '0'. To ensure the precision of the query, the 'WHERE' clause is added, which establishes the fundamental condition that the 'idtypeletter' field of the 'catalog' table has the value targeted for updating of '3'.

Regarding the auxiliary-type NoSQL tables considered suitable for this task, the following were defined: a) Gender, b) Language, c) Limbo, d) Material, e) Type of material, f) Type of writing/script, g) Occupation, h) Cult/Religion, i) Conservation status, j) Status of the record, k) Country, l) Technique, m) Type of authority, n) Type of document, o) Type of inscription, p) Type of object, q) Type of place.

Conclusions

The programme developed within the framework of the EPIHUM project provides a standard of exhaustive description for epigraphic documents, as pertains to their reading and transcription peculiarities, context, characterisation and geolocation. To this end, the following have been taken into account: the ISBD, MARC-21 and RDA bibliographic description standards; the ISAD-G, ISAAR-CPF archival description standards;

the Epidoc-TEI epigraphic description standards; and the Dublin Core Qualified metadata format. The cataloguing schema resulting from this synthesis can help to create definitive standards for the cataloguing of epigraphic documents that include relational schemas of authorities, types and characterisations. The structure of tables and fields defined in this chapter is a proven and functional model that can serve as the basis for future research addressing this problem.

On the other hand, a hybrid relational database model has been selected, one made up of NoSQL data tables for the management of auxiliary tables. This helps to manage the parameters of characterisation and typification of epigraphic documents, minimizing the amount of system resources dedicated to this end and thus improving performance, given the reduction of operational query transactions in MySQL. This has positive repercussions when it comes to creating multi-user programmes in which the number of concurrences can be high, rendering an improvement in the scalability of the application.

The project has prioritised standardisation, as reflected in the modules and parameters of the system. These are personal, institutional and familial authorities, tags for the classification of documents, topographical authorities, bibliographic references, types of object, materials, supports, techniques, types of writing, types of inscription and document states.

Regarding the interface, work has been done to keep all interactive and cataloguing elements within the researcher's field of vision, without popups or elements extraneous to the main editing screen. This feature can improve productivity in the cataloguing of epigraphic documents, reducing distraction and possible errors in the recording and assignment of authorities. To this end, peripheral tables have been employed, located in parallel to the cataloguing schema of the epigraphic document, so that perspective upon the task at hand is never lost. Thus authorities, tags, places, bibliography, photographs and geolocation can all be assigned without leaving the cataloguing page until the task is finished.

Closely linked to the interface, work has also been done on the query and retrieval views, whether in catalogue form or a graphical mosaic made up of photographs and their clippings. Retrieval via multiple points of view is essential, as is allowing the researcher to decide how best to do so in each case. Nevertheless, the query algorithm has been designed to function with all description fields, with the inscription text, its transcription, translation and characterisation fields, or through full-text searches, which can be filtered by date range or previously defined types.

Acknowledgements

This work was conducted within the framework of the project 'Escritura expuesta y poder en España (siglos XVI-XVII): catálogo epigráfico on-line' (HAR2015-63637-P), funded by the Ministerio de Economía y Competitividad, Gobierno de España (Programa Nacional de I+D+i). English translation by Dr Romén Reyes-Peschl.

Bibliography

Blanke, T., Bodard, G., Bryant, M., Dunn, S., Hedges, M., Jackson, M. and Scott, D. 2012. Linked Data for Humanities Research. The SPQR experiment, in *6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*: 1-6. Milano: Università degli Studi di Milano.

Blázquez-Ochando, M. 2021. Desarrollo de un sistema de gestión documental de inscripciones. El proyecto EPIBASE, in M. Ramírez-Sánchez (ed.) *Escritura expuesta y poder en España y Portugal durante el Renacimiento: de la edición digital al estudio de la epigrafía humanística*: 71-148. Madrid: Sílex.

Bodard, G. 2010. EpiDoc: Epigraphic documents in XML for publication and interchange, in F. Feraudi-Gruénais (ed.) *Latin on Stone: epigraphic research and electronic archives*: 101-118. Lanham, MD: Lexington Books.

Bodel, J. 2012. Latin Epigraphy and the IT Revolution, in J. Davies and J. Wilkes (eds) *Epigraphy and the Historical Sciences*: 275-296. Proceedings of the British Academy 177. Oxford: Oxford University Press.

Castillo Gómez, A. 1997. *Escrituras y escribientes. Prácticas de la cultura escrita en una ciudad del Renacimiento*. Las Palmas de Gran Canaria: Gobierno de Canarias.

Castillo Gómez, A. 2000. Artificios epigráficos. Lecturas emblemáticas del escribir monumental en la ciudad del Siglo de Oro, in V. Mínguez (ed.) *Del libro de emblemas a la ciudad simbólica. Actas del III Simposio Internacional de Emblemática Hispana*, vol. I: 151-168. Castelló de la Plana: Universitat Jaume I.

Castillo Gómez, A. 2009. A la vista de todos. Usos gráficos de la escritura expuesta en la España altomoderna. *Scripta* 2: 73-90.

Cordeiro de Sousa, J.M. 1983. *Apontamentos de epigrafía portuguesa.* Coimbra: Universidade de Coimbra.

Durán Salado, M.I. and Ortiz Lozano, L. 2017. La participación social en la documentación del patrimonio cultural, in V. Muñoz Cruz, S. Fernández Cacho and J. Arenillas Torrejón (eds) *Introducción a la documentación del patrimonio cultural*: 298-319. Seville: Junta de Andalucía, Consejería de Cultura.

Gimeno Blay, F.M. 2002. Capitales renacentistas, libros humanísticos. Representaciones de la cultura escrita en la pintura valenciana (XV-XVI), in L. Miglio and P. Supino (eds) *Segni per Armando Petrucci*: 159-175. Rome: Bagatto.

Gimeno Blay, F.M. 2005. *Admiradas mayúsculas. La recuperación de los modelos gráficos romanos.* Salamanca: Fundación Germán Sánchez Ruipérez.

Gimeno Blay, F.M. 2007. De la *luxurians litera* a la *castigata et clara*. Del orden gráfico medieval al humanístico (siglos XV-XVI). *Litterae Caelestes* 2(1): 9-51.

Orlandi, S., Santucci, R., Casarosa, V. and Liuzzo, P.M. (eds) 2014. *Information Technologies for Epigraphy and Cultural Heritage. Proceedings of the First EAGLE International Conference*. Collana Convegni 26. Rome: Sapienza Università Editrice.

Petrucci, A. 1986. *La scrittura. Ideologia e rappresentazione*. Turin: Einaudi.

Ramírez-Sánchez, M. 2010. Epigrafía e Internet en España hoy: una introducción necesaria, in J. M. Iglesias Gil (ed.), *Cursos sobre Patrimonio Histórico, 14. Actas de los XX Cursos monográficos sobre el Patrimonio Histórico*: 15-41. Santander: Universidad de Cantabria.

Ramírez-Sánchez, M. 2012. La tradición de la epigrafía antigua en las inscripciones hispanas de los siglos XV y XVI. *Veleia* 29: 255-277.

Ramírez-Sánchez, M. 2017. La *civitas* romana en la memoria del Renacimiento: una aproximación a través de los programas epigráficos de Évora, in A. Ruiz-Gutiérrez and C. Cortés-Bárcena (eds) Memoriae civitatum. *Arqueología y epigrafía de la ciudad romana. Estudios en homenaje de José Manuel Iglesias Gil*: 493-520. Santander: Universidad de Cantabria.

Ramírez-Sánchez, M. 2018. Las Ciencias de la Antigüedad en España en la encrucijada: retos y oportunidades en la sociedad digital, in E. Ortiz-de-Urbina and J.M. Vallejo (eds) Métodos y técnicas en Ciencias de la Antigüedad: estudios sobre investigación y docencia: 13-36. Vitoria-Gasteiz: Universidad del País Vasco.

Ramírez-Sánchez, M. 2021. La escritura expuesta en España y Portugal durante el Renacimiento: edición de corpus en la era digital, in M. Ramírez-Sánchez (ed.) *Escritura expuesta y poder en España y Portugal durante el Renacimiento: de la edición digital al estudio de la epigrafía humanística*: 21-70. Madrid: Sílex.

Redondo Cantera, M.J. 1987. *El sepulcro en España en el siglo XVI. Tipología e iconografía.* Madrid: Ministerio de Cultura.

Rubio Fuentes, M.J. 1994. *Catálogo epigráfico de Alcalá de Henares*. Alcalá de Henares: Fundación Colegio del Rey.