# Collectio: a software especially designed for creating dynamic libraries for fluid and multilingual text traditions

Britt Dahlman[1]

[1] *Lund University, Centre for Theology and Religious Studies, LUX, Box 192, 221 00 Lund, Sweden*

**Abstract**
This contribution presents a new software, Collectio, which can be used for creating highly complex relational MySQL databases, or more accurately, dynamic libraries. These libraries prove particularly well-suited for texts where the material has been organized in different ways and thus represents a 'fluid' textual tradition, or in traditions transmitted in many languages. So far, two libraries have been created using Collectio: APDB (the *Apophthegmata Patrum Database*) and HIPPO, which contains pre-modern hippiatric material. The sources included in the libraries are mainly in the form of manuscripts, editions and modern translations. Collectio employs a unique input model, built upon .txt and .csv files stored in an archive in the folder of the library. The contents of the database tables in the master database are generated from these documents. Since not only texts are registered but also the detailed structure and parallel text segments in other sources, both texts and structures can be systematically compared and analysed within and across language boundaries. In addition to the advanced research tools for comparing texts and structures, the application contains search options, indexes of names, places and concepts, metadata on the sources, pre-written SQL commands and more. A new way of encoding text, which can be converted into TEI/XML, is also introduced.

**Keywords**
relational database, dynamic library, TEI/XML, fluid text, multilingual text tradition, fixed-content miscellanies

## 1. Background

The history of the software called 'Collectio' begins with the research program called 'Early Monasticism and Classical Paideia' (MOPAI). This research program started in 2009 and was headed by Professor Samuel Rubenson at the Centre for Theology and Religious Studies at Lund University.[1] An important part of the research consisted of editing and studying monastic wisdom literature, especially the collections of *Apophthegmata Patrum* (AP), i.e. the sayings of the desert fathers and mothers. They are preserved in multiple languages and the material is both large and complex. The sayings are compiled in numerous collections that have a complicated textual transmission. Since different text redactions and stages in many languages (in particular Greek, Latin, Syriac, Armenian, Arabic and Ethiopic) were to be studied and compared, the idea of a digital tool was soon hatched. Scholars studying the AP tradition have made concordance tables to facilitate a comparative study of the sayings in the various collections.[2] Therefore, a relational MySQL database was considered the best solution, as it seemed especially suitable for comparing 'fluid' texts preserved in different types of organisations and in multiple languages. For the creation of this database, IT architect Kenneth Berg was consulted. Another IT technician was Leif Trulsson, who particularly helped in the creation of the graphical user interface. To secure its longevity, it was decided that the database should be based on .txt and .csv files, since it could be assumed that applications supporting those formats would exist for a long time to come. The tool was called APDB, short for the *Apophthegmata Patrum Database*, and was written in ooRexx. One of the reasons for this choice of programming language was that ooRexx includes some very powerful functions to handle character strings. The aim of the tool was not in the first place to store an archive of digitized editions or to produce digital critical editions (although this

*Huminfra Conference 2024, Gothenburg, 10-11 January 2024.*
britt.dahlman@ctr.lu.se (Britt Dahlman)

was also possible), but to provide data and advanced research tools for comparing and visualizing texts and structures.[3] The graphical interface of APDB was released for internal use in the research team in 2012.

Since APDB proved to be very useful for comparing texts consisting of clearly defined smaller narratives manifested in different types of organisations, it was decided to use the same software technology for another database called HIPPO as part of the project ‘Knowledge, Magic and Horse Medicine in Late Antiquity’ (2021–2025). The project, funded by the Swedish Research Council, is a collaboration between Lund University and the Swedish Institute in Rome. One of the main goals of this project is to make previously relatively unknown source material on horse medicine from Antiquity and the early Middle Ages available on an open access digital platform. Thus, HIPPO will be complemented by a web application called ‘Hippiatrica – a dynamic library and research tool’. This platform will be hosted by the Swedish Institute in Rome and accessible via the Institute’s website and the URBIS Library Network.[4]

In 2022 it was decided to make this useful tool available for a wider public. Thus, the software needed a common name, so that it could be used for creating other dynamic libraries. The name agreed upon was ‘Collectio’. The application is installed on the user’s own computer together with some other applications needed as prerequisites, such as ooRexx and the relational database management systems MySQL or MariaDB. The plan is to publish the source code on GitHub under the MIT license in 2024. Another expectation is to make a version of APDB (the Public DB) with a selection of the most important output functions accessible online at https://apdb.collectio.se in the beginning of 2024. For input of data, you will need to have the program installed on your computer, although there are plans for making it possible in the future for a small group of registered scholars to access the input files of APDB using the web interface.[5]

## 2. Two dynamic libraries created from Collectio: APDB and HIPPO

As for now, two dynamic libraries have been created using the Collectio software: APDB and HIPPO. Regarding the first one, APDB, work has been ongoing since 2012 to constantly improve its functions and the material. The aim was and still is to include as many sources as possible that contain dossiers of AP from editions, manuscripts and modern translations. Descriptions of the many collections of AP and their different organisations have been published in several books and articles and will not be repeated here.[6] However, soon the ambition was widened as it turns out that apophthegmata often appear in larger monastic compilations mixed with other monastic texts, and that hagiographic works often either quote from the apophthegmata or are used in collections of apophthegmata. Therefore, also works like the *Historia Lausiaca* by Palladius, the *Historia Monachorum in Aegypto*, and the *Pratum Spirituale* by John Moschos are registered, and new sources are constantly added.

HIPPO was created in 2021 and is under construction. The material to be included is much more limited compared to the large amount of monastic material that is and could still be included in APDB. Ancient hippiatric literature consists of predominantly Greek and Latin texts on the care and medical treatment of horses written from about the 4th century BC onwards. One of the most prominent veterinarians, who wrote down his advice in Latin, was Pelagonius. Greek veterinary works are generally not preserved in their entirety, but only as excerpts collected in the extensive Greek anthology *Corpus Hippiatricorum Graecorum* (or *Hippiatrica*), probably in the 5th and 6th centuries AD. This collection is transmitted in several different redactions where the material is organized in different ways, mainly alphabetically and thematically. Several works as well as redactions of the *Hippiatrica* were so popular that they were translated into many languages, including Arabic, Syriac, Armenian, and vernacular languages such as Medieval Italian and Spanish.

As in the case with the collections of AP, the texts of the *Hippiatrica* thus represent a ‘fluid’ transmission of structure and text with constant adaptation to new contexts, new audiences or settings, such as geographical, cultural, social or didactic ones. In a way, each manuscript is a unique ‘edition’, where the order and appearance of the texts may differ more or less. The scribes, or rather the compilators, may delete or include new material, reorder it, sometimes according to a recognizable principle, modify the text and sometimes reattribute the pieces. This is characteristic of ‘encyclopedical’ collections and other compilations of e.g. hagiographic, liturgical, monastic or ‘scientific’ material.

Many of them are so-called ‘fixed-content miscellanies’, i.e. manuscripts or compilations containing texts that belong to a more or less fixed genre, but where the occurrence and order of the texts vary.[7] Texts belonging to such genres are problematic to edit according to a stemmatic-genealogic method and present in traditional text-critical editions, because of the risk of contamination and of making new compilations of the material not respecting the variable tradition to which such collections of texts belong.

## 3. Why use Collectio and which functions should it have?

When relational databases are used in text editing projects they are often used as tools for collecting metadata, i.e. data that provides information about the sources, such as bibliographical data for publications or codicological data for manuscripts.[8] One example is *The Digital Victorian Periodical Poetry* project.[9] In this project a relational database was created for collecting metadata on more than 15,000 poems from 19th-century periodicals. Transcriptions of the poems were encoded in TEI/XML files. The data from the relational database was then integrated into an already existing TEI file or, if no transcription of the poem existed, a new TEI file consisting of only metadata from the relational database was created. According to Martin Holmes, the plan was to eliminate the relational database by the end of 2022.[10] However, some projects use relational databases for recording both metadata and text. One example is the *Database of Byzantine Book Epigrams* (DBBE) project.[11] It records both text transcriptions and metadata of book epigrams (‘metrical paratexts’) found in medieval Greek manuscripts dating up to the fifteenth century in a relational (PostgreSQL) database.[12]

A dynamic library created from Collectio is primarily intended to contain material in the form of texts and structural lists (with parallels) of manuscripts, editions and modern translations that are hard to map and analyse with traditional methods due to its ‘fluid’ nature. The scholarly debate about digital editions often concerns problems with how to handle the information found in the apparatuses of critical editions and how to present it.[13] The focus is often on textual variation and not on structural variation.

One of the advantages of relational databases is that not only the texts can be registered but also the detailed structure and parallel text segments in other sources (*loci paralleli*). Collectio allows texts and structures to be displayed in various kinds of output. Visualizations (statistics, diagrams etc.) demonstrating relations and relational distance can be created in other applications (e.g. for spreadsheets) using output from Collectio. Consequently, both texts and structures can be compared and studied systematically within a language as well as across language boundaries.

The prerequisites of Collectio would be the following:

- It should constitute a digital archive of editions, modern translations and manuscript transcriptions, but above all a research tool and laboratory for searches, analyses, comparisons and visualizations.
- It should have output functions for comparisons of both the occurrence and order of text units **structurally** within and across language boundaries and thus be an aid in the reconstruction of stages in the development of compilations/redactions and their relationship to each other.
- It should have output functions for comparisons of **texts** within and across language boundaries and thus be an aid in the reconstruction of stages in the development of compilations/redactions and their relationship to each other, as well as for text-critical investigations.
- It should have export functions that enable exchange with other databases, e.g. through marked-up modified diplomatic transcriptions.
- It should be open access; the source code should be published under the MIT license. For the existing libraries, all data in the Public DBs should be licensed under Creative Commons BY-SA.

Collectio provides an archive of structural tables and texts, indices of names, places, concepts and other term types, metadata on the sources, as well as scholarly annotations on the material, as is standard in text databases. The research tools enable searches for words, names, places and terms, as well as analysis and comparison of relations between separate text entities and collections (text and structure). Visualization in the form of graphs and diagrams can be created in other applications (e.g. for spreadsheets) using output from Collectio. SQL queries can easily be performed, both through a list of pre-written SQL commands and through free text queries.

The files in Collectio are encoded with a new markup language, the Collectio code, consisting of a word (or letters) beginning with a colon (e.g., :codex). The creation of this tag design was inspired by GML (Generalized Markup Language). In addition, texts can be encoded with specific embedded tags consisting of curly brackets, an asterisk, and a number (such as {*9 f.10r} for indicating the beginning of folio 10 recto). They are easy to use and can be converted into TEI/XML and exported to text documents. However, the exported text documents do not contain the full TEI data – there are no teiHeader elements for example – and usually only a limited amount of tags are used for the text entities.

## 4. Database model

The model used for registering input in Collectio is unique and its analytic tools constitute a further innovative step compared to other kinds of digitization projects for text corpora using XML-based platforms. The basis of the database, i.e. the library, is .txt and .csv files using Unicode UTF-8, which currently are maintained using common applications for text documents and spreadsheets in .odt and .ods formats. From these documents, which must be correctly encoded, the contents of the database tables in the master database are created. The files are stored in an archive, which can reside in a local folder, if used in a project involving one single person, or on a file share or an FTP server, if used in a project involving several persons. For changes in the files or new files to be included in the master database, it must be reloaded in a process where it is deleted and then recreated. The reason for this is that changes in one source can affect other sources, which then must be reloaded. These sources may in turn have an impact on even more sources. The most rational way to deal with this is to reload the entire database. For a collaborative project, the process should be initiated regularly by a person in charge. Through an advanced control system, all input is checked, and if there are any technical errors, the loading is interrupted, and a list of errors is exported. The errors often depend on incorrect markup (such as if the same ID has been registered for more than one entity). Then, after the errors have been identified and corrected, the master database can be reloaded. Thus, new data or changes in the data are not immediately included in the database tables, unlike the common procedure in relational databases. This ensures that the database is consistent and technically correct, which is crucial for complex relational databases.

Why not use web applications for input? They are often used in administrative IT systems and in crowdsourced platforms when a tool will be used by many people. Web applications can be of great help in guiding the contributor to make technically correct input if the applications are well designed. However, the more complex a relational database is, the more difficult it is to create web applications for input in every situation which can occur and that you cannot anticipate. This should especially be considered if the input goes directly into the database tables without any control system, as is usually the case. In a relational database it is especially important that the connections are consistent, and no breaks occur in a chain of relations. If severe technical errors, such as inconsistency of relations, occur without the users' attention, it may take a long time before they are detected, and in the meantime, other material could have been inserted. Then it could be difficult to restore the database without information being lost. Even if it is theoretically possible to create web applications for such complex connections as are found in Collectio, and especially for making changes in already existing connections between segments, to make them easier to use for the contributor than what can be done in common applications for spreadsheets is a challenge that few programmers would be willing to take on.

A selection of the contents of the master database, where copyrighted material and work for other reasons are excluded, constitutes the public database, which thus can be published online. A copy of all contents of the master database constitutes the personal and the common databases. To get access to the personal and common databases registration might be required. The difference between the latter two databases is that the personal database allows the user to contribute to the database by uploading files through the interface and updating his/her own personal database.

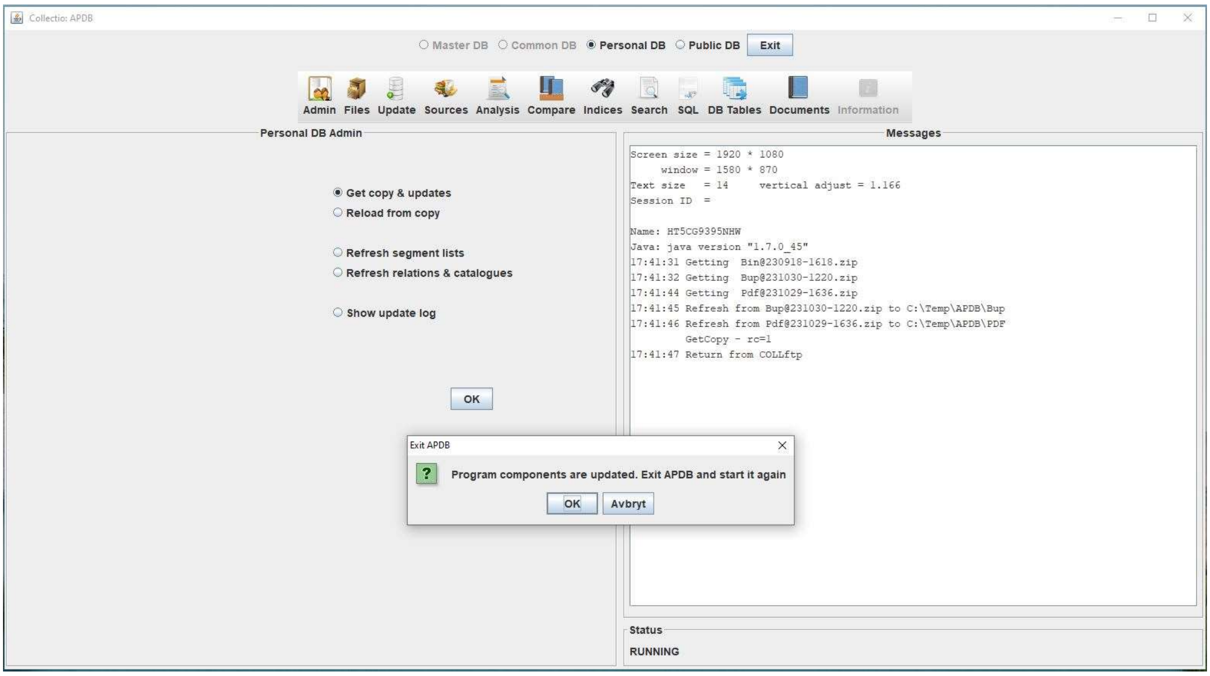


**Figure 1**: The internal interface of APDB. At the top, the buttons for the output functions are placed. In this case, the 'Admin' button and the 'Get copy & updates' option have been selected.

In the internal interface, the individual user of a personal database can download a new copy of the database to replace the old one when this is available (i.e. when a new master database has been created) by clicking on the 'Admin' button and choosing 'Get copy and updates'. The user is also alerted if new program components exist (see Fig. 1).

In addition, the user can also update his/her own personal database. A contributor working on a file can test it by clicking on the 'Update' button and selecting the file in question. This can be done by both new files not yet transferred to the local folder or server and by files outchecked from it. If the file contains any technical errors due to e.g. incorrect tagging, the updating is interrupted, and a list of errors is exported. After a revision of the file, the process can be repeated and, if successful, the data is included in the personal database. In this way the contributor has a 'personalized' database with new data which is available before it is included in the master database. Thus, this new data can be used for various kinds of research (searches, comparisons, and visualizations) together with the other material in the analytic tools. Output from all database versions can be an HTML page that is opened in a web browser, or a file that is opened in either a spreadsheet application or a word processor. Fig. 2 gives a simplified overview of the database model.

The concept of using text files and spreadsheets for input allows the contributor to use the many functions found in common applications, e.g. search and replace letters. Private notes that the contributor does not want to share with the public user of the database can easily be inserted and left untagged, for view and revision later by the contributor. The downside is that it requires much training to acquire the knowledge needed for correctly marking up the documents, in particular the spreadsheets.

As soon as a file is saved with new information and transferred to the local folder or the server, it is marked with a timestamp and stored in an archive. All current active files as well as old files are accessible through the 'Files' button in the interface. This makes it easy to go back and follow the evolution of a source, to track changes, and to see who has done them and when.

## 5. How to use Collectio

Collectio is designed especially for comparing structures and texts consisting of small entities manifested in different types of organisations. Therefore, to facilitate comparisons, all base entities are divided into smaller entities called 'segments', which are given unique IDs. Base entities are 'Unit', 'Title', 'Explicit', and 'Item'. They consist of one or more segments, which are labelled 'a', 'b', 'c', and so on. These are central terms for defining the structure of a source. Through the segments, it is possible to connect these structural entities between sources (and within a source). A unit normally corresponds to a paragraph in a text (e.g. a saying in the AP collections or a text excerpt in the *Hippiatrica* redactions). In printed editions they are often numbered in sequence. An example of an ID for a manuscript source would be 'Athos_Prot_86 V.2a', meaning chapter 'V', unit (i.e. saying) '2', segment 'a' in the source (i.e. the manuscript) labelled 'Athos_Prot_86'.

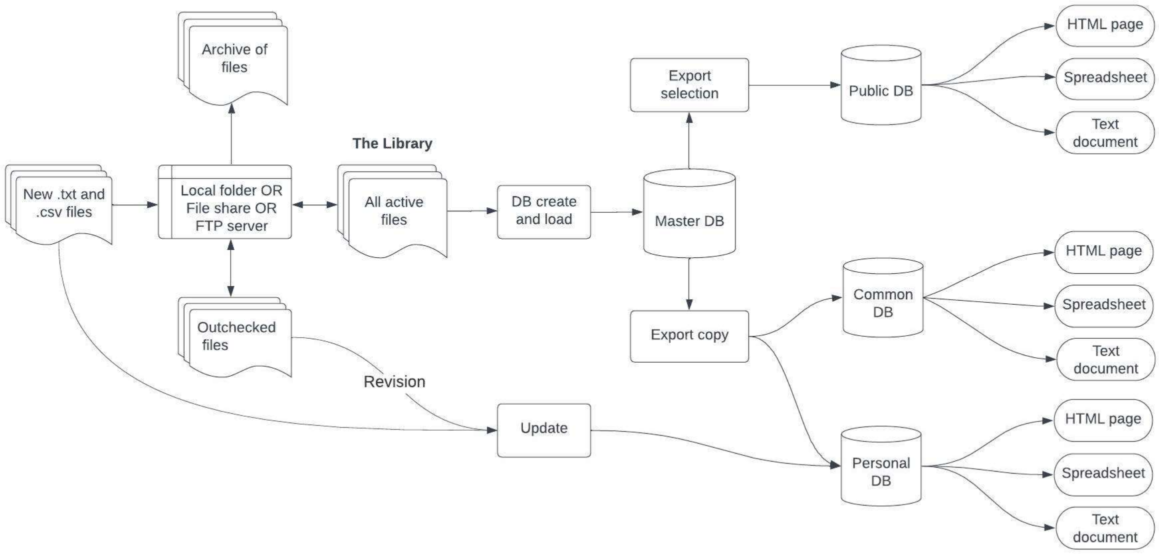


**Figure 2**: Overview of the database model for Collectio.

The connections between the segments are made through so-called reference series, which are lists of unique IDs based on the base entities in one or several manuscripts, editions or translations. Segments having the same registered reference within a source or between sources are thus automatically linked to each other. The connection between two or more segments can be registered in two ways: as being equivalent (marked as '='), or as being similar (marked as '~'). To decide which text segments should be registered as (only) similar to other text segments is not always easy, but usually, the content should be approximately the same, while the wordings of the segments should have substantial differences, such as being longer or shorter. However, there is also the possibility to register only a relation (marked as '-'), which could be described as a more general connection. A relation cannot be registered on the segment level – it is always a relation to a base entity or a higher-level entity.[14] The connections between text segments being equivalent (=) are managed by a number of reference series. If we, for example, look at the connections for Athos_Prot_86 V.2, we find these registered connections (together with many more):

Athos_Prot_86 V.2a = Vat_lat_600 V.6 ~ SyrEn-Bedjan I.587a

Athos_Prot_86 V.2b = Vat_lat_600 V.7 = SyrEn-Bedjan I.587b

It means that the 'a' segment in the Athos manuscript corresponds to unit 'V.6' in the Latin manuscript 'Vat_lat_600', but that they are only similar to unit 'I.587a' in the Syriac edition 'SyrEn-Bedjan'. The 'b' segment corresponds to unit 'V.7' in Vat_lat_600' and to unit 'I.587b' in 'SyrEn-Bedjan'.

Two common ways of organizing material in compilations are alphabetically and thematically. Among the collections of AP, alphabetic collections organize the material according to the names of the desert fathers and mothers, and systematic ones according to themes, usually of virtues and vices. The connections through unique segment numbers make it possible to compare different sources and to use visualization tools to demonstrate relations and relational distance between them. Fig. 3 shows how the apophthegmata in the Delta ('D') section in the Greek alphabetic collection (G) are distributed among the thematic chapters in the Greek systematic collection (GS). In APDB the correspondences between the reference series G and GS can be displayed through a table, as in Fig. 4. The connections can also be visualized in different ways, for example as a diagram demonstrating relational distance created in a spreadsheet application from a table of sequence numbers as output from APDB.

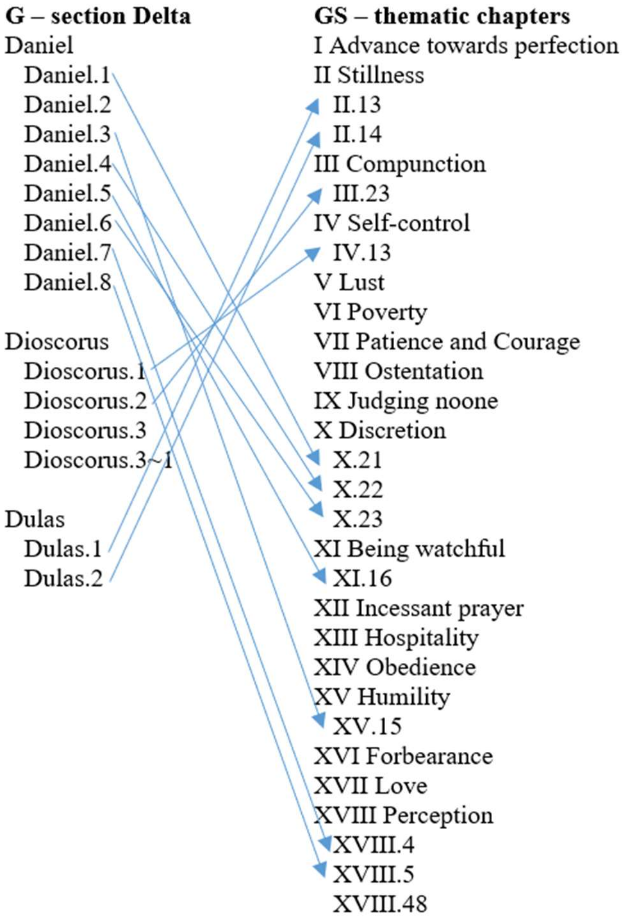


**Figure 3**: Connenctions between parts of G (Greek alphabetic AP) and GS (Greek systematic AP).

| G | GS |
|---|---|
| Daniel.1 | =X.21 |
| Daniel.2 | |
| Daniel.3 | =XV.15 |
| Daniel.4 | =X.22 |
| Daniel.5a | =XI.16a |
| Daniel.5b | =XI.16b |
| Daniel.6a | =X.23a |
| Daniel.6b | =X.23b |
| Daniel.6c (A1) | =X.23c (A1) |
| Daniel.7 (A1) | =XVIII.4 (A1)<br>~XVIII.48 (B1) |
| Daniel.8 | =XVIII.5 |
| Dioscorus.1 | =IV.13 |
| Dioscorus.2 | =III.23a |
| Dioscorus.3 | |
| Dioscorus.3~1a | |
| Dioscorus.3~1b | |
| Dulas.1a | =II.13a |
| Dulas.1b | =II.13b |
| Dulas.2 (A1) | =II.14 (A1) |

**Figure 4**: Table demonstrating the connections between parts of G and GS as output from APDB.

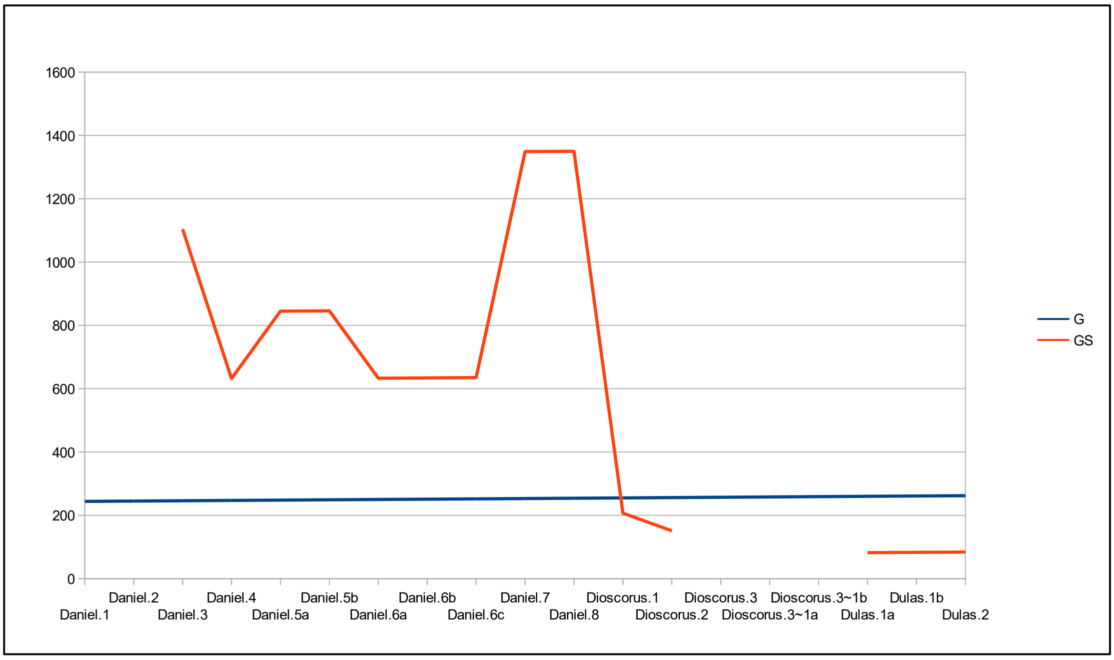


**Figure 5**: Comparison between G and GS visualized in a diagram created from output from APDB.

In Fig. 5, the Delta section in G has been selected as starting point and GS as a source for comparison. In the diagram, the horizontal axis displays the unit IDs in G and the vertical axis the sequence numbers in G and GS. The line of the source selected as starting point, G, is accordingly a straight line, and the other line representing GS illustrates how the segments in GS deviate from the order or are missing compared to G.

## 6. Concluding remarks

'There are no shortcuts in creating a database. The code used in the design of the database sets the limits of the functions. Likewise, the data you put into a database is the data you can process. There is no magic involved in making the data better than it was when it was inserted.' These thoughtful words come from Kenneth Berg, the creator of Collectio, a software written in ooRexx for creating highly complex relational MySQL databases or, to put it more accurately, dynamic libraries. So far, two libraries have been created using Collectio: APDB (the *Apophthegmata Patrum Database*) and HIPPO, which contains pre-modern hippiatric material.

The model used for input is distinguished from other models used in relational databases. The basis of the library is .txt and .csv files, which are stored in an archive in the folder of the library. This folder may reside on a local drive, a shared file system, or an FTP server. From these documents, the contents of the database tables in the master database are created. The master database is regularly reloaded and recreated, and through an advanced control system, all input is checked ensuring that the database is consistent and technically correct. Through a number of advanced research tools Collectio enables texts and structures to be compared and studied systematically on different levels both within a language and across languages. Files in Collectio are encoded with special markup codes that can be converted into TEI/XML and exported to text documents.

Another strength of Collectio is its independence from public (and other) funding. From 2009 to 2015 parts of the development costs were funded by Stiftelsen Riksbankens Jubileumsfond, and for the years 2021–2025, a small portion is funded by the Swedish Research Council (Vetenskapsrådet). However, for the main part, the development of the software is carried out through the voluntary work of dedicated IT technicians – in particular Kenneth Berg – and scholars and students who have given rise to many improvements through their contributions to APDB and HIPPO.

For any public digital library, the question of sustainability is crucial, both in terms of the format of the data sets and regarding a long-lived infrastructure guaranteeing the maintenance. However, to make it possible to publish a library created from such an advanced application as Collectio online, a secured long-term basic maintenance is not enough, if we would like to be able to add new material or correct errors, in particular concerning connections and relations. For what is gained if the content cannot be trusted? Reliable functions for inserting and correcting the connections of segments (such as connection tables or equivalent systems) and control systems are needed to ensure that the relations are consistent, and no technical errors occur. Collectio offers this and creates long-lived scholarly data sets. Working with original documents, spreadsheets and text documents, is a win-win situation – it combines the benefits of a database with those of a printed edition.

## Acknowledgements

This article was supported by funding from the Swedish Research Council (Vetenskapsrådet) for the 'Knowledge, Magic and Horse Medicine in Late Antiquity' project (2020-01788), headed by Elisabet Göransson at Lund University. I am most grateful to Kenneth Berg for his tireless efforts in improving Collectio and his constant readiness to share his competence and support.

## Notes

[1] It was supported by funding from Stiftelsen Riksbankens Jubileumsfond from 2009 to 2015. For a survey of the individual projects that were parts of it, see S. Rubenson, *Det tidiga klosterväsendet och den antika bildningen. Slutrapport från ett forskningsprogram*, vol. 9 of RJ:s skriftserie, Makadam, Göteborg, 2016.

[2] The most known and used general tables are probably those by W. Bousset, *Apophthegmata. Studien zur Geschichte des ältesten Mönchtums*, Tübingen, 1923, and L. Regnault, *Les sentences des pères du désert, troisième recueil & tables*, Solesmes, Sablé-sur-Sarthe, 1976. However, many editors and scholars have made tables for collections and manuscripts within specific languages, such as those by J.-C. Guy, *Recherches sur la tradition grecque des* Apophthegmata Patrum, vol. 36 of Subsidia hagiographica, 2e édition avec des compléments, Société des Bollandistes, Bruxelles, 1984, for the manuscripts of the Greek systematic collection, and C. M. Batlle, *Die "Adhortationes sanctorum Patrum" ("Verba seniorum") im lateinischen Mittelalter: Überlieferung, Fortleben und Wirkung*, Aschendorff, Münster, 1972, for the manuscripts of the Latin systematic collection attributed to Pelagius and John (PJ).
[3] S. Rubenson, "A Database of the Apophthegmata Patrum", in: T. Andrews, C. Macé (Eds.), *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, vol. 1 of Lectio: Studies in the Transmission of Texts & Ideas, Brepols, Turnhout, 2014, pp. 207.
[4] For more on this project, see https://projekt.ht.lu.se/hippo/.
[5] Alongside the development of APDB, from 2013 another tool and digital platform has been created as well: *Monastica* – a dynamic library and research tool, https://monastica.ht.lu.se/. Up to August 2022, *Monastica* was dependent on APDB. *Monastica* imported all data (with some exceptions) from the master database dump of APDB on the FTP server and, thus, had basically the same contents as APDB. Since then, both APDB and Monastica have developed further. As for now (December 2023), *Monastica* has limited possibilities for input and corrections of connections and relations.
[6] For a recent presentation of the collections of AP, see S. Rubenson, "Apophthegmata Patrum", in: D. G. Hunter, P. J. J. van Geest, B. J. Lietaert Peerbolte (Eds.), *Brill Encyclopedia of Early Christianity Online*, 2018. URL: http://dx.doi.org/10.1163/2589-7993_EECO_SIM_00000239. Further descriptions and references are given in S. Rubenson, "A Database of the Apophthegmata Patrum", in: T. Andrews, C. Macé (Eds.), *Analysis of Ancient and Medieval Texts and Manuscripts: Digital Approaches*, vol. 1 of Lectio: Studies in the Transmission of Texts & Ideas, Brepols, Turnhout, 2014, pp. 203–205, and B. Dahlman, E. Göransson, K. Åkerman Sarkisian, "A crosslinguistic approach to the study of the Apophthegmata Patrum. A case study of textual variation in the Greek, Latin and Slavonic traditions". To appear.
[7] For the concept of 'fixed-content miscellanies', see D. Birnbaum, "Computer-Assisted Analysis and Study of the Structure of Mixed Content Miscellanies", *Scripta & e-Scripta* 1 (2003): 15–64, who refers to several articles by A. Miltenova.
[8] For an introduction to how relational databases can be designed within digital humanities projects, see S. Ramsey, "Databases", in: S. Schreibman, R. Siemens, J. Unsworth (Eds.), *A Companion to Digital Humanities*, Blackwell, Oxford, 2004, pp. 177–197.
[9] For more on this project, see https://dvpp.uvic.ca/.
[10] M. Holmes, "Getting Along with Relational Databases", *Journal of the Text Encoding Initiative* 14 (2021). URL: https://journals.openedition.org/jtei/3874. doi:10.4000/jtei.3874.
[11] For more on this project, see https://www.dbbe.ugent.be/.
[12] R. Ricceri, K. Bentein, F. Bernard, A. Bronselaer, E. De Paermentier, et al., "The Database of Byzantine Book Epigrams Project: Principles, Challenges, Opportunities", *Journal of Data Mining and Digital Humanities*, Volume: *On the Way to the Future of Digital Manuscript Studies* (2023). URL: https://hal.science/hal-03833929v3. doi:10.46298/jdmdh.10244.
[13] T. Keeline, "The Apparatus Criticus in the Digital Age", *The Classical Journal* 112:3 (2017): 342–363; S. Douglas Olson, "Further Notes on the Apparatus Criticus in the Digital Age", *The Classical Journal* 114:3 (2019): 330–344; F. Fischer, "Representing the critical text", in: Ph. Roelli (Ed.), *Handbook of Stemmatology: History, Methodology, Digital Approaches*, Berlin: De Gruyter 2020, pp. 405–427 (416–427). See also T. Andrews, "Publication of digitally prepared editions", in: Ph. Roelli (Ed.), *Handbook of Stemmatology: History, Methodology, Digital Approaches*, Berlin: De Gruyter 2020, pp. 427–436.
[14] A relation is typically registered to a Bible verse or a catalogue number, but it can also be registered to base entities in a related source, where the scholar cannot (or does not have the time to) make the correct analysis required for dividing the entities into segments.