Athanasios Velios and Nicholas Pickwoad

Current use and future development of the database of the St. Catherine's Library Conservation Project

Saint Catherine's Monastery in Sinai, Egypt, holds one of the most important collections of ancient manuscripts in the world. Since 2000, conservators from Camberwell College of Arts in London have been visiting the monastery to collect detailed records of the manuscripts' structures and materials, and assess their preservation condition.[1] This condition assessment has been undertaken in order to plan the long-term conservation work in the library. The information has been collected on specially designed paper forms instead of a fully digital recording system for a variety of reasons, mainly in view of the remote location of the monastery which makes software and hardware support extremely difficult and costly and the convenience of drawing on paper rather than on a computer screen.[2]

Each paper form holds a detailed record of the manuscript's structure and condition. Retrieving information about a specific manuscript is therefore easy by examining the form. However, fast retrieval of information about a specific observation made over the entire collection is practically impossible, as it would mean searching through all individual forms (3306 in total). Patterns of observations can be evident by browsing a sample of the forms, but this is time-consuming, difficult to document, and the results may not represent collective information as accurately as results from the whole set of forms. Such collective information is necessary for planning long-term conservation work in the library. For this reason, a digital database has been developed to accompany the paper forms and enhance the existing potential for searching the data. Our main requirements from the database were:

1. Storage of the existing information collected on the paper forms.
2. Fast retrieval of observations about bookbinding structure and condition by members of the project team, so that conservation work can be planned more efficiently.

Although at the moment only project members have access to the database, it is hoped that it will be more generally available in the future.

The database structure has been based on the structure of the paper form. Each page of every form has been digitized and saved as an image. The information captured in these images is being inputted to the database and we hope that this process will be finished by the end of 2006.

In this paper, we will discuss the principles followed for structuring the database using the relational model and how they are relevant to the design of the paper forms. Some of the benefits of the relational model are detailed searching capabilities and the potential for statistical analysis of the data. Several examples of results returned from the database illustrating these benefits will be discussed. In the St. Catherine's library condition-assessment survey, information is collected following a *hierarchical* methodology, starting from the general observations (*parent* of the hierarchy) and continuing with the more detailed ones (*child*). The relational model, however, has certain limitations in the way hierarchical data is stored, which became apparent while we were exploring its potential. We plan in the future to switch from the relational model to a more recent technology, namely the eXtensible Markup Language (XML), which is ideal for encapsulating hierarchical data. XML is also recommended for the long-term archiving of digital data, a major concern of the project, without compromising the research capacity as offered by a relational database. Until the transition is completed, the relational

[1] Pickwoad, N., 'The condition survey of the manuscripts in the monastery of Saint Catherine on Mount Sinai', *The Paper Conservator* 28 (2004): 33-61.

[2] Velios, A. and N. Pickwoad, 'Collecting digital data on paper', University of the Aegean: *Proceedings of the 2nd International Conference on Museology — Technology for Cultural Heritage* (Mitilini, in press).

database will be used for all data-management needs.

The main focus of this article will therefore be a description of the structure of the database as opposed to the interface and user experience which are independent of the database structure and can be tailored according to the users' needs. For the project's users who are mainly book conservators, a web-based interface to the database has been developed which allows querying using criteria from any combination of recorded characteristics. Assistance for the development of particularly complex queries is offered by the authors on demand. The underlying technology of this interface is likely to change when the relational database is transformed to an XML database.

This paper may be of interest to conservators as it discusses the concept of documentation based on structured information. It may be of particular interest to book conservators currently working on binding documentation issues, as the principles of database design followed for the St. Catherine database may apply to any collection. Finally, those who are particularly interested in the St. Catherine's database will find this information useful as a user's reference. While a basic understanding of relational databases and XML is assumed, appendices have been included with short introductions to concepts, along with references for further reading.

Principles of the database structure

1. Analysis of paper form

The St. Catherine's database has been developed in order to accommodate the information collected during the condition assessment at the library of the monastery in Sinai, Egypt. The detailed information recorded on the forms could lead to the development of an extremely complex database structure. However, such a structure could make the development of an interface for data retrieval difficult, and also the database itself could be difficult to maintain and update. Therefore, our efforts focussed on simplifying the database structure as much as possible, without compromising the potential for storing detailed information.

Designing the paper forms was done in such a way as to minimize the time spent on each manuscript during the assessment and to ensure that a complete record was provided for each manuscript. The paper form has been particularly helpful to the assessors as it functions as a logical route through the elements which need to be checked to record the material, structure, and condition of each book. Although such optimization is essential during recording, database structures are developed using different optimization criteria known as *normalisation rules* which help make the database efficient and easy to maintain (Appendix 1).[3] We tried to preserve the structure of the paper form in the database to ensure the association of tables with the information they hold, although in many cases, this was not possible. In order to identify places where the normalisation rules disagreed with the paper form, we first deconstructed the paper form and rearranged the information to fit the relational database model and its normalisation rules.

2. Information deconstruction: Sections and libraries

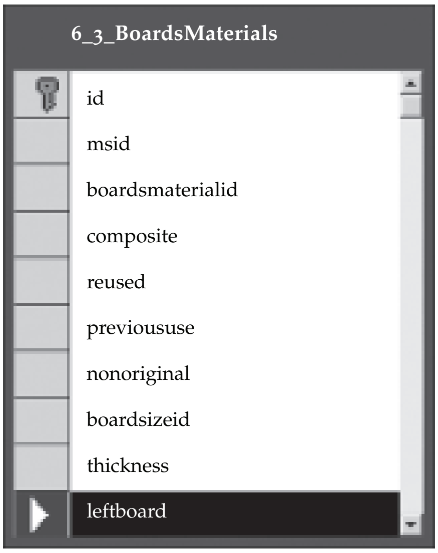
Three characteristics of the form influenced the database structure. First, each section of the form deals with a distinct part of the binding (e.g. Section 7.2 deals with the tooling on the covering). Second, identical information is recorded for different parts of the binding (e.g. left and right board or multiple page markers). Finally, types of observations are repeated in the records of different elements of the binding (e.g. paper can be a material for the text leaves, boards or linings).

Given the logical division of the paper form into sections which examine individual elements of the book, the information stored in each database table corresponds to a specific section of the form. Therefore, each section of the form produces a database table which carries that section's number and name (e.g. the table name for page marker section on Page 1 is *1_2_PageMarkers*). By keeping this analogy between the paper form and the database, anyone familiar with the paper form will find it easy to use the database. As the paper form has already

[3] Codd, E.F., *The relational model for database management: Version 2* (Reading, MA; Wokingham: Addison;Wesley, 1990).

been well documented by Pickwood, this documentation is applicable to the database.[4] These initial section-tables form the basic structure of the database onto which all other tables are linked.

Due to the symmetry of Byzantine bindings, in many cases, identical information needs to be recorded from different parts of the book. Moreover, features recorded individually may have multiple occurrences on a book. Although recording these features on the paper form demands separate sections (e.g. left board material on Section 6.3a and right board material on Section 6.3b and the whole of page 1a for multiple bookmarks, page markers and lifting tabs), for simplicity and in compliance with normalisation rules, this information is stored in a single table of the database. In most cases, individual occurrences of information corresponding to the same manuscript need to be identified separately and for this reason an additional column in the table is needed to hold this information. For example, the table for the left and right board material in Fig. 1, includes a column called *leftboard*, which indicates whether the current record of the board material for a manuscript is the left board or not (if it is not the left board, it can only be the right board, hence the column requires a *boolean datatype* to indicate the either/or state of this information) (Appendix 1).


6_3_BoardsMaterials

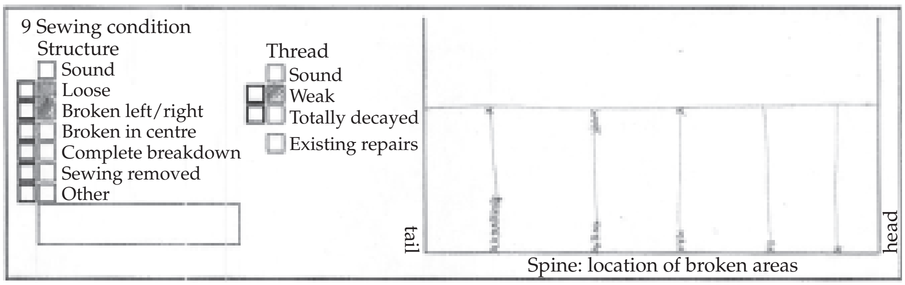
id
msid
boardsmaterialid
composite
reused
previoususe
nonoriginal
boardsizeid
thickness
leftboard

Fig. 1 Column list of the table which stores material information about the boards. Column *leftboard* is highlighted.

The assessment form contains multiple references to identical definitions of materials, conditions, colours, etc. As mentioned in the example above, *paper* can be text-leaf material but also the material for boards or linings. However, the word *paper* describing paper as a material in the database does not change despite the different elements of the book in which it is observed. This indicates that there should be only one occurrence of the word *paper* in the database which can be used as a reference by all tables that need to store information on paper as a material. In the St. Catherine's database, there are such reference tables, called *libraries*, which store the whole list of definitions for the following: materials (table name: *MaterialList*); colours (table name: *ColourList*); parts of the book (table name: *LocationList*); condition types (table name: *DamageList*).

The libraries store definitions of all terms which can be accessed by any table. However, as this is not always needed, i.e. sections of the form need access to only part of the libraries and not the whole set of definitions, additional section-specific tables have been implemented, called *sub-libraries,* whose role is to filter the library terms. For example, different metals are listed in the material library because they need to be referred to by the furniture sections (Page 9 of the form). However, metallic materials can never be used for text leaves and therefore the text leaves materials' section (Page 2 of the form) should not be able to refer to them. The sub-library *TextLeavesMaterialList* filters the material records from the table *MaterialList* so that tables from Page 2 of the form only have access to the materials which apply to text leaves (paper, parchment and papyrus). In this way, when a new record of a book's text leaves is created, the available options for the material are limited, thus reducing the probability of a mistake by making data inputting easier. A different way of implementing that functionality would be to perform filtering directly into the *MaterialList* table by including a new field. This would certainly be possible if each material referred to a single section of the paper form. However, it is often the case that a material appears in different sections of the form and, therefore, separate filtering for the specific section is necessary. Filtering is implemented by using reference values and each row of a library corresponds to an i.d. number which is then linked to the sub-library. The intermediate reference using the i.d. number is necessary in case definitions need to be renamed. This may be needed in the future in the case of unfamiliar characteristics which, when first observed, are given temporary names, which may change. The database needs to offer the capability of quickly updating the names of the definitions but keeping their references intact. With the i.d. reference a definition is renamed once by updating a single table row and the whole database refers to the updated version automatically. This is much simpler than updating every row that this definition has been used for, which would be the case if i.d. numbers were not used.

[4] Pickwood, N., *Assessment Manual*, Camberwell/ St. Catherine Project website, <http://www.arts.ac.uk/research/stcatherines/files/manual20050110.pdf > last accessed 8 Aug 2006.



9 Sewing condition
Structure
□ Sound
▨ Loose
□ Broken left/right
□ Broken in centre
□ Complete breakdown
□ Sewing removed
□ Other

Thread
□ Sound
▨ Weak
□ Totally decayed
□ Existing repairs

tail
head
Spine: location of broken areas

Fig. 2 Scan from Section 9 of Page 5 of the paper form, which records the condition of the spine.

3. Database structure: Tables and relationships

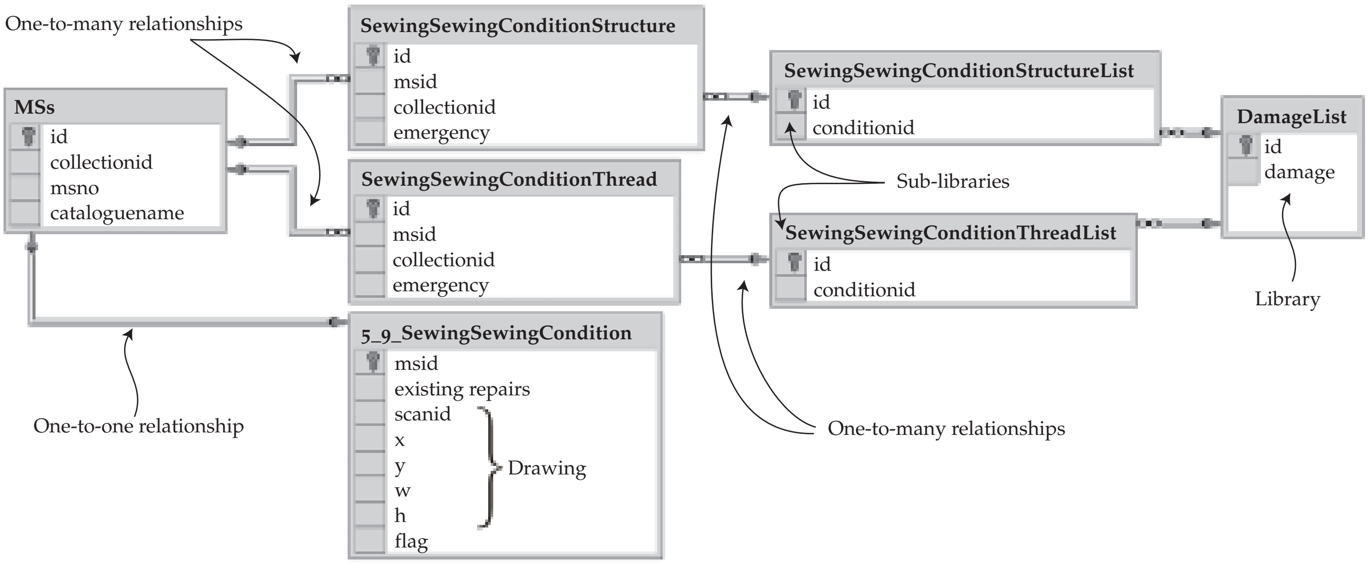
The tables in the database can be divided into four groups according to their functionality (Appendix 1). The first group includes tables with information from a section of the form which is recorded only once per manuscript (*one-to-one relationships*). The second group includes tables with information from a section of the form which is recorded multiple times per manuscript (*one-to-many relationships*). The third group includes the libraries and sub-libraries, and the fourth includes special tables such as the one storing the identity of the manuscripts (table *MSs*).

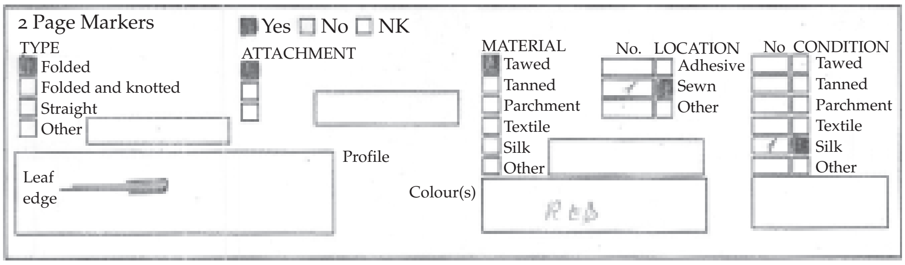
To assist with describing the structure, we consider Section 9 from Page 5 of the paper form which records the sewing condition of the manuscript (Fig. 2). This section holds information about the condition of the structure; the condition of the thread; whether previous repairs exist; and finally the drawing of the spine breaks. The existing repairs checkbox and the drawing are recorded once in each manuscript. However the types of damage observed on the structure and the thread are recorded multiple times per manuscript as a single volume may have more than one type of sewing structure and sewing thread damage. Hence, the drawing and the existing repairs are stored in a database table which has a one-to-one relationship with the manuscript, whereas the damage types are stored in different tables (for the structure and the thread separately) which have one-to-many relationships with the manuscript (Fig. 3).

Having established these links between the manuscript and the individual sections as recorded on the paper form, there is another type of relationship often found in the database structure. Continuing with the above example, the type of damage recorded for the sewing structure can be one from the following list: loose; broken left; broken right; broken in centre; complete breakdown; and sewing removed.

As explained in the previous section, these different types of damage are kept

Fig. 3 Diagrammatic example of the use of relationships in the database.





2 Page Markers ☒ Yes □ No □ NK

TYPE
☒ Folded
□ Folded and knotted
□ Straight
□ Other

ATTACHMENT
☒
□
□

Profile

Leaf edge

MATERIAL
☒ Tawed
□ Tanned
□ Parchment
□ Textile
□ Silk
□ Other

Colour(s) Red

No. LOCATION
□ Adhesive
✓ ☒ Sewn
□ Other

No CONDITION
□ Tawed
□ Tanned
□ Parchment
□ Textile
✓ ☒ Silk
□ Other

Fig. 4 Scan from Section 2 of Page 5 of the paper form which records page marker series.

in the general *DamageList* library and are filtered through a sub-library which is specific to the sewing structure part of Section 5.9. In order to use these terms, a link needs to be established between the table which holds the sewing structure condition for each manuscript, and the table which holds the sewing structure condition sub-library. Again, this is done with a one-to-many relationship where one type of damage from the sub-library can occur multiple times in the structure condition table (Fig. 3).

Several tables in the database play a particularly significant role, as they hold essential information about the identity of the manuscripts and the assessment forms; these are:

1. Manuscripts table (table name: MSs): This is an important table in the database structure as it holds information about the manuscripts' identities. This includes the shelfmark, collection names and a unique sequential identity number which is used to refer to the manuscript throughout the database. In the example of Section 5.9, this identity number is used by the sewing structure conditions table to link each condition with a manuscript.

2. Scanned pages table (table name: Scans). As mentioned above, images of the paper forms are produced by a scanner. The database has a designated table which keeps records of the location of each page's scanned image, the form page number (from 1 to 10) and the manuscript which the page belongs to. This table is used when reference to a scanned page of the form is needed, typically to extract drawings of the page as explained below.

It is often the case that multiple items of an element are recorded on the paper form. For example Section 2 in Page 1 holds information about series of page markers. When many different series of page markers are present on a manuscript, these can be recorded on Page 1a. Although by looking at Page 1, the information about type, attachment and material of page markers appears to have a one-to-one relationship with the manuscript, this is not really the case (Fig. 4). This information is unique for each page markers series and not the manuscript. Therefore, the one-to-one relationship in this case changes to one-to-many, as a manuscript can have more than one series of page markers (Fig. 5). Similar modifications in the structure are needed when symmetric elements of the binding are recorded. For example the same information is recorded for the left and right boards. This information demands a one-to-many relationship with the manuscript, as a manuscript can have many (two) boards.

To summarize, in the previous paragraphs we described how the information contained in specific sections of the paper form can be stored in the database using a combination of one-to-one and one-to-many relationships with special

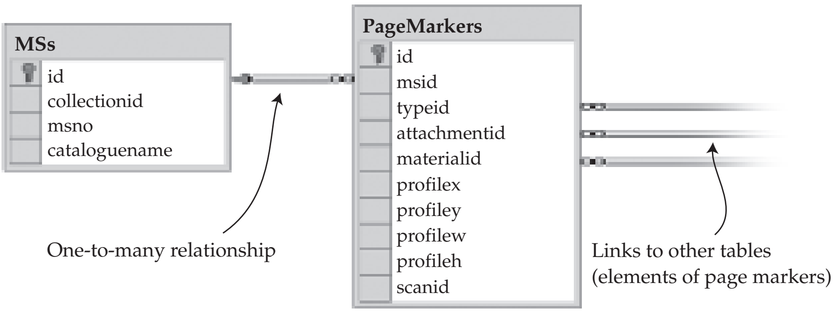


Fig. 5 Diagrammatic structure of database storage of page marker series information.

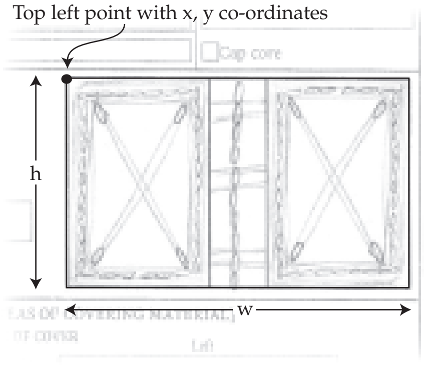


Fig. 6 Coordinates and dimensions recorded to capture a single drawing from a paper form.

tables. The same principles for table linking are followed throughout the database. In general, information which is kept once per manuscript produces a one-to-one relationship with the manuscript. Multiple observations of the same type of information produce one-to-many relationships with the manuscript. Finally, libraries and sub-libraries are needed to produce one-to-many relationships with the rest of the tables.

Drawings

Drawings are often present on the paper forms. Relational databases offer different ways of storing images in tables. A popular one involves the use of *Binary Large Objects* (BLOBs), where images are stored inside the database system as binary data and a reference to them is produced as a record in a table. Another way of storing images involves external storage on the file system and reference to them from within the database. A simple test indicated the difference in speed with which images are retrieved from the database as BLOBs and from the hard disk as files on our system (other systems may have different performance). A request for 50 BLOB images took about 35 seconds to complete on the local machine (no network delay) whereas the same request for images on the disk took about 36 seconds. In our database, queries usually take on average 5–10 seconds to complete, depending on the length of the returned data. Despite the marginally faster retrieval of the drawing data with BLOBs, we decided not to use them because BLOBs do not support direct access to the images as separate files on a disk. Instead, they are encapsulated in the large database files which cannot be read by ordinary software, and this was a feature that we needed in other parts of our work. In addition, the time needed to backup a database increases dramatically when BLOBs are used and a complete backup could take longer than the periods between backups. This would mean that we would have to reduce the number of backups, which is not advisable as our data currently changes very often (on a daily basis). Moreover, our images do not change at all, so they only need to be backed-up initially and subsequently checked on a regular basis and refreshed when necessary. In order to save time, we wanted to avoid including them in each backup job by keeping them externally.

Our images of the scanned paper forms are stored as separate files on the hard disk. The database is aware of the images because references to them are kept in the special table called *Scans*. This holds the location of each image on the disk alongside relevant metadata so that the files can be directly accessed when necessary. However, the drawings on the scanned paper forms only occupy part of the image (the rest being checkboxes and written notes). When querying the database for a specific drawing, the user needs to view only that part. In order to be able to crop the image to the required frame, information about the location of the drawing on the page is needed. Therefore, when a drawing needs to be stored in a table, we identify the scanned page and store information about the coordinates of the top-left corner of the required frame and the width and height of the drawing's frame (Fig. 6). When a user queries the database for a drawing, this information can be used to retrieve the full scanned image, set the cropping boundaries of the requested drawing, and present the cropped image only.

Naming conventions

In order to keep the terminology consistent, the names used for the fields of the paper form are also used for the database tables. There are four types of naming following the different functions of the database tables and these are described below.

Tables holding information with a one-to-one relationship to the manuscript are named after the section name of the form (e.g. *5_9_SewingCondition* for the table of Section 9, Page 5). The table name starts with the page number, followed by the section number and finally the section name. Tables which hold information with a one-to-many relationship to the manuscript are named using the page name, section name and the individual part of this section. For example, *SewingStructureConditions* is the name of the table which stores the damage types

Shelfmark	Main type	Primary type	Secondary type	Primary attachment	Secondary attachment	Colours	Materials
Greek 0128	Compound	1. Span 1	-	2. Frayed	-	Pink	Cord
Greek 0141	Simple	1. Span 1	-	-	-	Red	Natural thread
Greek 0145	Compound	1. Span 1	2. Knotted, single length	1. Knotted	1. Wound	Red	Silk
Greek 0145	Compound	1. Span 1	2. Knotted, single length	1. Knotted	1. Wound	Pink	Silk
Greek 0147	Simple	-	-	1. Knotted	1. Wound	Blue	Natural thread
Greek 0147	Simple	-	-	1. Knotted	1. Wound	Natural	Natural thread
Greek 0153	Loose	-	-	-	-	Blue	Natural thread
Greek 0153	Loose	-	-	-	-	White	Natural thread
Greek 0161	Simple	1. Span 1	-	-	-	Red	Silk
Greek 0161	Loose	-	-	-	-	Red	Silk

Table 1 Sample records of data recorded about bookmarks.

observed in the sewing structure. The ending letter 's' indicates the multiple damage types observed on a given manuscript. Sub-libraries have the same name as the section they apply to including the word 'List' at the end. For example, the different types of damage which can be observed at the spine are listed in a sub-library table called *SewingStructureConditionList*. Finally, libraries follow the same principle but have general names describing the information they hold (e.g. *MaterialList* for the materials' library).

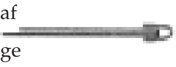
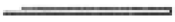
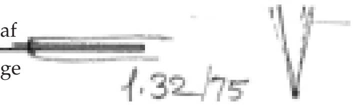
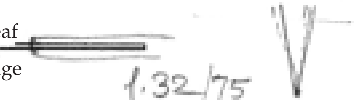
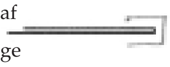
In names, points ('.') have been replaced by underscores ('_'). This was done in order to avoid confusion as relational database systems often use the format *TableName*.*ColumnName* (with the point between the two names) to indicate that a column belongs to a table. Also, because the names of the tables can be comprised of a number of words, to separate the individual components for readability purposes, a capital letter at the beginning of each word has been used (i.e. *SewingStructureConditions* instead of *sewingstructureconditions*). In the case of column names, where the names are shorter, only lowercase letters have been used.

Benefits offered by the relational database

1. Querying and detailed searching

The main benefit of the database is the potential to retrieve information from the whole of the collection as easily as from one manuscript only. The paper form organized information on a *per manuscript* basis as each form corresponds to a manuscript. This makes collective information retrieval practically impossible. In

Table 2 Sample records of data recorded about page markers, including drawings.

Shelfmark	Type	Profile	Attachment	Material	Flag	Colours	Condition	No in Condition	Location	No in location
Arabica 0011	Folded	Leaf edge 	Adhesive	Tanned leather	True	Brown	Sound	2	Foredge	2
Arabica 0013	-	Leaf edge 	-	Textile	True	Green	Sound	1	Foredge	1
Arabica 0032	-	Leaf edge 	Sewn	Natural thread	True	Dark brown	Broken off	5	Head	1
Arabica 0032	-	Leaf edge 	Sewn	Natural thread	True	Dark brown	Broken off	5	Foredge	4
Arabica 0049	Folded	Leaf edge 	Adhesive	Tawed leather	True	Brown	Sound	2	Foredge	2

Date	No. of manuscripts
03/02/2005	23
26/01/2005	20
14/05/2005	20
01/06/2005	20
28/01/2005	19
02/02/2005	19
07/02/2005	19
04/02/2005	18
20/01/2005	16
25/01/2005	16

Table 4 Table showing the number of manuscripts assessed on each date. This list has been sorted in decreasing order.

Shelfmark	Attachment	Colour	Condition	Location	Material	Type
Arabica 0080	Adhesive	Red	Worn	Foredge	Silk	Folded
Arabica 0084	Adhesive	Red	Broken off	Head	Silk	Knotted
Arabica 0381	Adhesive	Green	Sound	Foredge	Silk	Folded
Arabica 0397	Adhesive	Yellow-Green	Sound	Head	Silk	Knotted
Arabica 0397	Adhesive	Yellow-Green	Broken off	Head	Silk	Folded
Arabica 0408	Adhesive	Yellow	Broken off	Foredge	Silk	Folded
Arabica 0438	Adhesive	Pink-Red	Worn	Foredge	Silk	Folded
Georgian 0059	Adhesive	Green	Worn	Foredge	Silk	Knotted
Greek 0153	Adhesive	Patterned	Worn	Head	Silk	Folded
Greek 0639	Adhesive	Deep red	Worn	Foredge	Silk	Folded

Table 3 Sample records of page markers made of silk and attached with adhesive.

the database, the information is organized on a *per element* basis as each element is stored in a table column and the table rows correspond to each manuscript. Information is therefore grouped in tables making collective retrieval much simpler. For example, Table 1 shows the result of a query about the structure of bookmarks (Section 4, Page 1 of the form). Similarly, drawings can be incorporated in the results as shown in Table 2 where descriptions of page markers are shown (Section 2, Page 1). The database can return the shelfmark of the manuscript to which the information of each row corresponds. Although this is not essential for a query to be performed, the information returned is more useful when that reference exists. In general, different elements from any section of the form can be combined in a single query and there is no limit to how these combinations are made. However, meaningful results will be returned only when meaningful questions are asked.

To extend further the searching potential, the user can be given the option to apply conditions to the returned results. This permits detailed querying of the database according to specific criteria based on the records which exist in the sub-libraries. (The records for each sub-library are the available options with which an element can be searched.) For example Table 3 shows the results of a query for all page markers whose material is silk and attachment is adhesive. In many databases such conditions can be applied but there is no guarantee that the requested condition will be fulfilled by any of the records – in other words a user sets conditions which may not be relevant to the specific search. In our case these conditions are totally controlled by the libraries and sub-libraries of the database, ensuring that the applied restrictions on the data will be meaningful as they evolve from the data itself.

2. Statistics and conservation management

By obtaining access to collective data, new ways of using the condition assessment information become possible. Relational databases are particularly useful when it comes to statistically analysing data. Most of the major relational database packages make use of basic mathematical functions to enhance query results (e.g. count records, average values, remove duplicates, etc.). The development of the database has allowed the use of these possibilities on the data of the condition assessment. For example, Table 4 shows the number of manuscripts surveyed for each day of work at the monastery indicating that 3 Feb 2005 was the most productive day. Another example in Table 5 shows the percentage of lifting tabs (Section 3, Page 1 of the form) which were originally attached to a manuscript but are now missing.

Table 5 Percentage of missing lifting tabs as a result of database information.

Total no of lifting tabs	568
No of missing lifting tabs	86
Percentage of missing lifting tabs	15.14%

Similar queries can be constructed for any element of the binding and

Shelfmark	Left board	Right board	Left of centre	Centre	Right of centre	Ranking
Greek 1426	110	110	140	160	160	5
Greek 1428	130	90	100	90	90	5
Greek 1431	100	95	95	90	90	5
Greek 1451	90	90	85	90	90	3
Greek 1475	90	85	100	110	100	3
Greek 1478	95	90	85	120	110	3
Greek 1585	90	80	70	90	60	1
Greek 1588	85	85	70	80	75	1
Greek 1594	95	90	90	75	80	1
Greek 1602	90	60	90	120	60	1

Table 6 Possible methodology for ranking the suitability of the manuscripts for digitization. The numbers correspond to degrees up to which the book can open without risking damaging the spine. Books with high degrees get a high ranking as they are easy to photograph. Books with low degrees are difficult or impossible to photography and therefore get a low ranking.

condition recorded on the forms. If the results of these queries are combined with current conservation expertise, it will be possible to identify trends about the state of preservation of the whole collection quantitatively. Moreover, planning conservation work in the library will be done based on true data and will result in more accurate estimates of the resources needed. Due to the remote location of the library, this is particularly important as careful planning for any visit is essential. An example of combining the database potential with conservation expertise is the proposal of a manuscript ranking system according to their suitability for digitization, done at the request of Dr David Cooper, a consultant to the monastery on digitization issues. The results presented in Table 6 are based on a number of smaller queries about the flexibility of the manuscripts' spines and their capability to open at adequate angles to accommodate the photography of each folio.

Standard model

The results presented above have been exported from the database in a simple text format. They have been retrieved through a webpage which connects to the database. The underlining technology for relational databases has long been standardized (Structured Query Language became an ISO Standard in 1987) and it is possible to export data from a relational database in a wide range of computer programs.[5] This standardization is an additional benefit as it ensures cross-platform compatibility, straightforward data retrieval, and reliable data copying from one database system to another.

The advantages offered by the relational model are important for the work done for the St. Catherine's project. However, with continuing developments in the data structure and information retrieval fields, there are further demands for better records of conservation-related information. In the next sections we explore some of the limitations of the relational model and a proposed method for overcoming them.

Complex data

As mentioned previously, the design of the St. Catherine's database allows for searching binding structures using criteria about every element of the binding and its condition. The efficient implementation of the database permits combinations of such criteria to be made. In order to make this possible within the relational model, the database design demanded the introduction of certain limiting features, which are explained below.

1. Recording structured elements

Recording bindings in detail means keeping a record of the individual elements comprising a binding, for example, a board. These elements consist of other

[5] International Organisation for Standardisation, ISO/IEC 9075-1, 'Information Technology – Database Languages – SQL' (Geneva: ISO, 1987)..

elements, or sub-elements. For example, a board is made of a material and has specific dimensions. The recording of the board itself indicates whether there is a board or not. However, in order for the record to be of any use beyond that, information about the board's properties (e.g. dimensions and material) needs to be recorded as well and the sub-elements will depend on the element. For example, if *paper* is the material element of a board, its sub-elements would be the *process type* (whether it is laminated or pulp paper) and the *paper type* (printed or MS). Arguably, one would also be able to check the ruling on the paper or other material features. This example indicates that there is a structure in this information which falls into a hierarchical parent-child arrangement (Fig. 7).

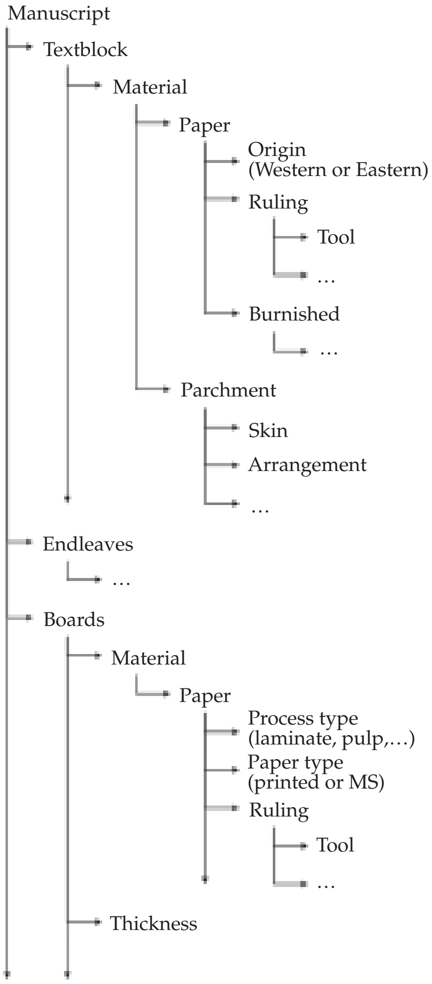


Fig. 7 Example of hierarchical structure of data.

To continue this example, an accurate record of the board should arguably include both the information and the hierarchy of the information, namely the fact that certain information comprises larger, more general information. Although this is certainly possible to implement using the relational model, previous experience has shown that the resulting database structure is rather abstract. Good examples of this problem are the various online Content Management Systems (CMSs) such as Drupal or Mambo.[6, 7] The flexibility of such structures allows the mapping of a hierarchy, however this results in a database structure which is designed to store hierarchies and it is not focussed on the actual data. For example, a single database table would hold information about a variety of data which in our case could range from manuscript boards to endleaves, only because this data happens to be in a similar hierarchical relationship to the manuscript. In the St. Catherine's database we wanted to avoid such complex database structures and keep the table structure focussed on the type of information. The database is not designed to store hierarchical data, but this can be retrieved at the moment by using the assessment manual. Our proposal for a way of embedding the hierarchy information in our database will be introduced later when discussing XML.

2. Recording variable data

Hierarchical information introduces an additional anomaly in the relational database design. Often, the child elements of a parent may demand different element descriptions. To continue with the previous example, the board material can be either *paper* or *wood*. If it is paper, then a set of child elements needs to be recorded (i.e. paper process, paper type, ruling, etc.). If it is wood, a different set of child elements needs to be recorded (i.e. grain size, grain direction, whether it is hardwood, softwood, etc.).

The set of wood elements is different than the set of paper elements and therefore these two sets cannot be accommodated in the same table of a relational database. For example, one field recorded about wood is whether it is hardwood or softwood. A table would need a *boolean* field to indicate that. When paper is recorded, the hardwood/softwood choice is out of context and therefore a record for paper in the same table will result in 'null', which contradicts the principles of the normalisation rules. This problem can be solved by creating separate tables for different materials. However, this introduces the obvious limitation that whenever a new material is observed during data inputting, the database structure needs to change in order to accommodate the child elements of the new material. Again, this is possible in the relational model but is not an elegant approach, as the structure of the database is affected by the data and in order to input new data substantial redevelopment of the database is needed, including building new tables and relationships. This is one of the most important drawbacks of the relational model, which led us to consider alternative data-storage methods. In the next section we will discuss our proposals for implementing the database in a way which allows flexibility of recording variable and hierarchical data.

Future work

1. Hierarchical structures

In the previous sections the concept of the hierarchical information structure was introduced by giving selected examples. However, the hierarchical structure is

[6] http://drupal.org.

[7] http://www.mamboserver.com.

met in every part of the manuscript record. The *bound book* can be the beginning of the hierarchy (otherwise called the *root*). Under the root, the basic elements of the binding follow (e.g. text-block, boards, endleaves, etc.). By exploring one of these elements, new child elements emanate. For example, a text-block has elements describing the material, size, ruling, etc. Or in a more explicit record, a text-block may have each individual folio as a child element which will then have other sub-elements describing the specific folio. The same hierarchical principle of presenting information can be applied to any part of the binding description. The hierarchy provides space for any observation to be stored, but it does not require all information to be there. A diagrammatic example of such a hierarchy is shown in Fig. 7.

A good way of implementing hierarchical structures is by using XML, which has been designed with this principle in mind (Appendix 2). Every XML document is required to have a root element (elements are also called *tags*) within which all other tags are located and organized hierarchically. Parent tags contain nested child tags which may contain further nested tags to describe the full hierarchy of the data. An XML document could, therefore, be used to describe the binding of a manuscript using a hierarchical structure. In XML documents, elements are described inside tags by simple text. The use of simple text makes editing XML documents easy, but it also increases the risk of using multiple hierarchical structures which by mistake do not agree. In order to produce a record of a binding consistent with everyone's records, one would need a description of the hierarchical structure for bookbinding, with which every XML bookbinding document must comply (see the concept of *schemas* in Appendix 2). This compliance must be on both the hierarchical structure and the values that each XML tag can accept. Recent funding for the St. Catherine's project from the Arts and Humanities Research Council (AHRC) has initiated a research project for producing such a hierarchy in the form of an XML glossary. The planned outcome of this research project is a widely accepted hierarchy of XML elements which will be used when describing Byzantine bindings. To the authors' knowledge such a hierarchy is not currently available for Byzantine bookbinding or bookbinding in general. Choosing an optimal hierarchy for bookbinding description is a matter of wider scholarly discussion and our intention is to start an inclusive discussion with other experts working on the history of Byzantine binding.

Having established such a hierarchy in XML, the St. Catherine's database will then be translated from the current relational model to a collection of XML documents, one for each manuscript. The main advantage of XML is that it allows recording of hierarchical structures by using nested tags as explained above. XML also offers the potential for using records semantically while storing information in a database (see the concept of *namespaces* in Appendix 2). The relational model for database design was not developed with such functionality in mind but rather to accelerate the searching capabilities of strictly relational data. However, the important advantages of relational databases, namely speed and connectivity, are not absent in XML. Most of the new versions of commercial and open source databases support XML. This makes storing XML documents in databases possible and allows the use of equally fast search tools for information retrieval. Tools like XQuery and XPath replicate the searching functionality of the relational databases and compete in speed. The increase in computational power and the advance of XML software can only make XML data retrieval faster than it is today. Current software is adequate for serving XML queries and a good example of a database with a web interface currently online is the Text Encoding Initiative (TEI) website.[8] For these reasons, the St. Catherine's database project will not abandon the concept of a database but will switch from the relational model to XML in order to achieve better representation of our hierarchical data. The relational structure will be transferred to XML and data which cannot be represented successfully in the relational model will be restructured in XML. We hope that our transition from relational to XML databases will be a success and that it will set the standard for conservation recording in book conservation and other fields.

[8] http://www.tei-c.org.

id	century	title
1	9	Books of Job, Daniel, Jeremiah and Ezekiel
2	10	Pentateuch
3	14	Pentateuch
4	10	Pentateuch
5	13	Exodus – commentary
6	16	Exodus – commentary
7	10	Chronicles I–II
8	13	Isiah, Hosea, Joel, Amos, Obadiah, Jonas...
9	13	Genesis, Exodus, Leviticus, Numbers, Deuteronomy
10	12	Prophetologion – Lessons from the prophets recited...

Table 7 Sample records of the manuscripts' century and title.

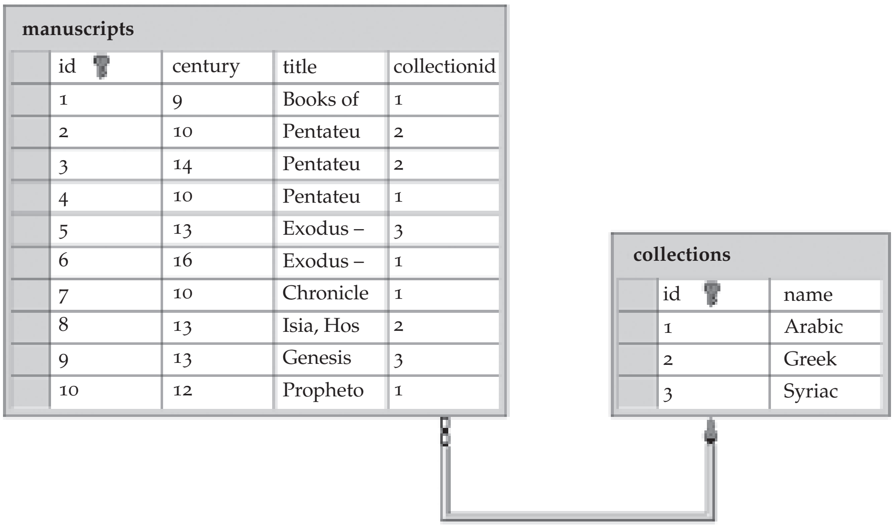
Appendix 1 The relational model.

Relational databases are widely used to store large sets of similar records. Data in relational databases are stored in *tables*. Tables consist of *columns* and *rows*, with columns indicating the kind of information stored and rows corresponding to the individual records. Each column can store one kind of data (e.g. numbers, characters, dates, etc.). The kind of data stored in a column is called *datatype*. A list of the most widely supported datatypes follows:

1. Integer: stores any integer number.
2. Floating numbers: stores any fraction number.
3. String: stores any characters (usually there is a limitation in the length of characters allowed).
4. Date/time: stores any date and/or time.
5. Boolean: stores either a positive (yes) value or a negative (no) value.
6. Binary objects (BLOBs): stores any computer file, including images.

A table can, therefore, store any combination of data by assigning each kind of data to one of its columns. Every record (row) of the table can use the available columns to insert new data. The Relational Model requires that no two identical records should exist in a table. However different database implementations (including the SQL standard) do not necessarily apply that limitation. In general it is preferable that the value of at least one column is different between any two records so that records can be distinguished. Usually a table has an integer column which serves as an *identity* of the record or the *primary key*. Every record gets a unique integer number and hence it can be identified in the table, also ensuring that there are no two identical records (Table 7).

A relational database can contain many tables. Each of the tables holds a set of logically grouped data. Combinations of data from different tables are possible by using



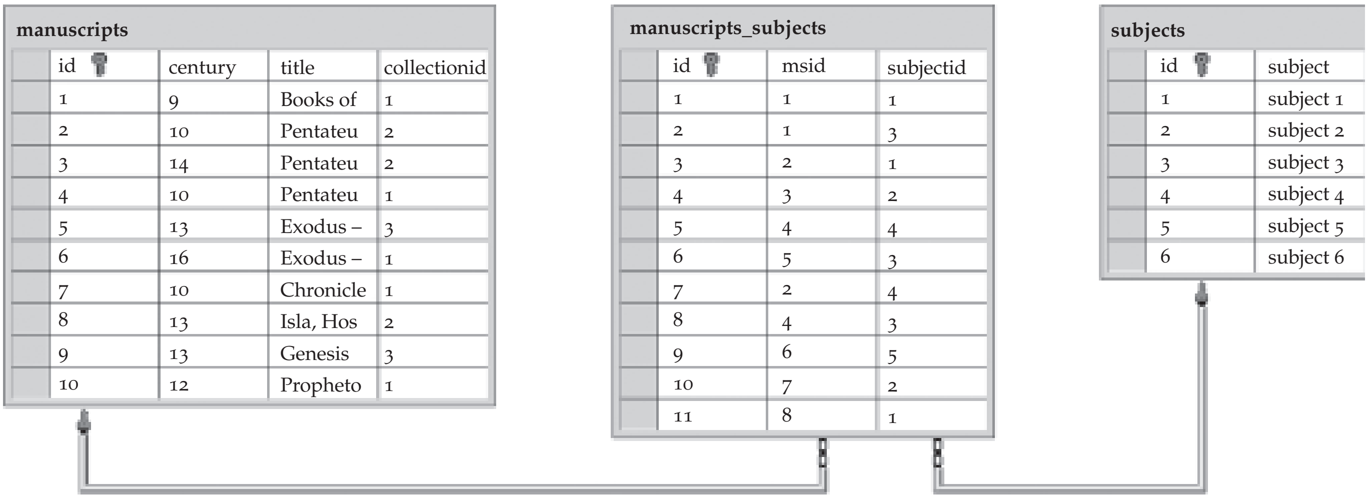
manuscripts

id	century	title	collectionid
1	9	Books of	1
2	10	Pentateu	2
3	14	Pentateu	2
4	10	Pentateu	1
5	13	Exodus –	3
6	16	Exodus –	1
7	10	Chronicle	1
8	13	Isia, Hos	2
9	13	Genesis	3
10	12	Propheto	1

collections

id	name
1	Arabic
2	Greek
3	Syriac

Fig. 8 Example of *one-to-many* relationship.



manuscripts

id	century	title	collectionid
1	9	Books of	1
2	10	Pentateu	2
3	14	Pentateu	2
4	10	Pentateu	1
5	13	Exodus –	3
6	16	Exodus –	1
7	10	Chronicle	1
8	13	Isla, Hos	2
9	13	Genesis	3
10	12	Propheto	1

manuscripts_subjects

id	msid	subjectid
1	1	1
2	1	3
3	2	1
4	3	2
5	4	4
6	5	3
7	2	4
8	4	3
9	6	5
10	7	2
11	8	1

subjects

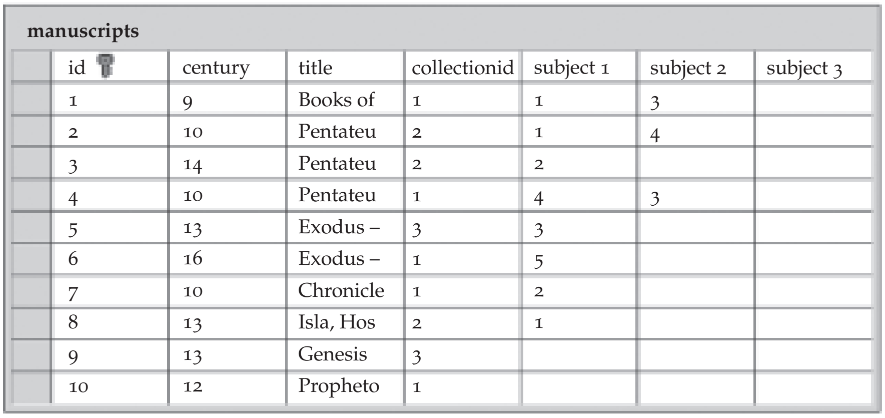
id	subject
1	subject 1
2	subject 2
3	subject 3
4	subject 4
5	subject 5
6	subject 6

Fig. 9 Example of *many-to-many* relationship.

relationships. For example in Fig. 8, the *manuscripts* table is linked to the *collections* table by the use of an additional column (*collectionid*) which indicates the collection that the manuscript belongs to. This column takes values from the primary key column of the *collections* table and is called a *foreign key*. Therefore, manuscript number 1 belongs to the Arabic collection, manuscript number 2 belongs to the Greek collection, manuscript number 3 to the Greek collection and so on. In this particular case the relationship between the two tables is called a *one-to-many* relationship as one collection can have many manuscripts, but a manuscript can only belong to one collection. Other types of relationships exist. The *one-to-one* relationship indicates that each record of a table corresponds to a record of another table. Such relationships are used when a large number of columns exist in one table and for practical reasons it is more convenient to split the table in two. A more complicated relationship is created when many records in a table may correspond to many records in another table. For example, if we were devising thematic collections of manuscripts, a single manuscript may fall into more than one category (e.g. astronomy and mathematics). In this case a *many-to-many* relationship is created by using a new table as shown in Fig. 9.

Instead of a many-to-many relationship, another way of implementing our thematic catalogue would be by producing multiple columns in the manuscript table with each one of them storing a subject (Fig. 10). Although technically this is possible, it is not recommended practice for two important reasons. First, if we allowed space for, say, three subject areas per manuscript, the manuscripts which belonged to less than three subject areas would leave empty cells, whereas the manuscripts affiliated with more than three subject areas would lack the necessary space. Second, if we decided to change the name of a subject area, this change would have to take place as many times as the subject area has been affiliated to manuscripts.

To avoid such problems in the relational model, certain general rules have been described (called *normalisation rules*) which help optimize a database structure. These rules are not obligatory and it is up to the database developer to choose whether to use them or not, but in general they are widely accepted guidelines for database development. The two



manuscripts

id	century	title	collectionid	subject 1	subject 2	subject 3
1	9	Books of	1	1	3	
2	10	Pentateu	2	1	4	
3	14	Pentateu	2	2		
4	10	Pentateu	1	4	3	
5	13	Exodus –	3	3		
6	16	Exodus –	1	5		
7	10	Chronicle	1	2		
8	13	Isla, Hos	2	1		
9	13	Genesis	3			
10	12	Propheto	1			

Fig. 10 Bad example of database design with repeating columns of the same information.

most important of these are that columns in a table must hold as simple information as possible. For example a column should not hold mixed data like *red silk thread* as this includes both colour and material information in the same column. Second, columns of the same type of information (e.g. colour) should not be repeated in the same table. Hence, we should not allow columns like: colour 1, colour 2, colour 3, etc. in the same table. These rules comprise the *1st normalisation form* (INF) of a database. That is, if a database complies with these rules then it can be described as a *1NF* database. There are many other rules for database optimization, but it is not our intention to describe them here.

Data is retrieved from tables which are interlinked by using the Structured Query Language (SQL) which has been specially designed for this purpose. Its purpose is to formalize statements which describe what kind of data is needed. A detailed description of SQL is beyond the scope of this paper. An example of an SQL statement is: *SELECT id, title FROM manuscripts WHERE collectionid=1.* This statement instructs the database to return the identity number of each record (id) and the text in the title column (title) which are stored in table (manuscripts) and whose corresponding collection (collectionid) is equal to 1 (Arabic collection), hence all manuscripts of the *Arabic* collection.

Relational databases are a rather large field to cover in the limited space of an appendix. The information and examples given above illustrate the very basic principles of relational databases, which should be adequate for following the ideas described in the main text. However, the reader may want to retrieve more details about SQL and the relational model by referring to Allen *et al*, or Fleming.[9, 10]

Appendix 2 eXtensible Markup Language (XML).

XML is a computer language recommended by the *World Wide Web Consortium* (W3C) for describing and transferring data.[11] XML documents are extensively used for long-term data storage as they are based on simple text and hence they are easily readable by humans with only minimal software.

Each XML file consists of a series of *tags* which are expressed with the use of the '<' and '>' symbols. In the example *<ThreadColour>red</ThreadColour>*, *ThreadColour* is the tag which characterizes the word *red*. Red in this case refers to the colour of the thread. Notice the '/' symbol before the second *ThreadColour*, which indicates that the tag terminates and therefore the information about the colour of the thread has ended. To continue with this example we now present a more extensive view of the tagging:

```
<Thread>
        <Material>silk</Material>
        <Colour>red</Colour>
        <Thickness>medium</Thickness>
</Thread>
```

We have introduced a general *Thread* tag which includes the colour, material, and thickness tags of the thread, to indicate that these properties are part of the thread and incorporated logically inside it. The *Thread* tag is part of the endband-material tag, which is part of the endband tag and so on. In XML documents, information is described hierarchically starting from the most general (for example a *Manuscript* tag) to the more specific (as in the endband-thread example).

The meaning of the tags and the way they are located within each other is of crucial importance when it comes to transferring data from one system to another. Unless the tags of both systems refer to the same type of data, the transfer is impossible. Therefore, XML files need to follow a design principle which will allow them to indicate the meaning of their data to other systems. This design principle is called a *schema*. Schemas are agreed by professional bodies or interested parties who want to use a common standard for transferring information. An example of an XML schema is the *HTML specification* for creating documents on the World Wide Web. Schemas allow the standardization of information which leads to *semantic recording*.

In order to explain the concept of semantic recording we are going to start from current computer searching techniques. Computer text searching is mainly performed by comparing a sequence of characters (search string) to a resource and returning the locations where it occurs. This searching is done on a character level with the computer being unaware of what it is searching for. This is why if we search for *boards*, as in bookbinding, it is not unusual to get results about *boards* as in *notice boards*, since the search engine cannot understand the difference between a board in bookbinding and a board for notices. In XML however, it is possible to search for a board in bookbinding only with a combination of simple text searching and the use of *namespaces*. As mentioned earlier, XML allows tagged text data. These tags give meaning to the data by using the concept of namespaces. A namespace is a way of identifying the subject field of an XML document.

[9] Allen, C., Creary, C. and S. Chatwin, *Introduction to Relational Databases and SQL Programming* (Burr Ridge, IL: McGraw-Hill Technology Education, 2004).

[10] Fleming, C., *Handbook of Relational Database Design* (Boston, MA: Addison-Wesley, 1989).

[11] http://www.w3.org.

Therefore, a namespace for bookbinding would assign the correct meaning to the term *board* and allow the computer to direct the search request appropriately. This is called a *semantic search* which forms the basic idea for the semantic web and allows search engines to look intelligently for information in appropriate resources.

There is extensive literature on XML but a good starting point are the tutorials published online by the W3C.[12]

Acknowledgements
The authors would like to thank Dr Peter McBrien from the Department of Computing, Imperial College for his advice on certain aspects of the database structure.

[12] http://www.w3schools.com.

Summary
The St. Catherine's library conservation project has focussed so far on the condition assessment of the collection, which took place in the St Catherine's Monastery in Sinai, Egypt. Detailed information of each of the 3306 manuscripts of the library has been collected. Approximately 1000 observations were made on each manuscript. During the assessment, this information was kept on paper. However, a computer filing system was necessary to store the information in digital format which would then allow quick searching of the data. In this paper, the authors describe the design and implementation of a computer database system which helps organize, categorize and access the collected data quickly and efficiently. The article includes an extensive description of the principles behind the database design. It also refers to the complexity of the recorded data and how this fits the relational database model. The paper concludes by exploring the future development of the database and its transition to XML.

Summary
The St. Catherine's library conservation project has focussed so far on the condition assessment of the collection, which took place in the St Catherine's Monastery in Sinai, Egypt. Detailed information of each of the 3306 manuscripts of the library has been collected. Approximately 1000 observations were made on each manuscript. During the assessment, this information was kept on paper. However, a computer filing system was necessary to store the information in digital format which would then allow quick searching of the data. In this paper, the authors describe the design and implementation of a computer database system which helps organize, categorize and access the collected data quickly and efficiently. The article includes an extensive description of the principles behind the database design. It also refers to the complexity of the recorded data and how this fits the relational database model. The paper concludes by exploring the future development of the database and its transition to XML.

Summary
The St. Catherine's library conservation project has focussed so far on the condition assessment of the collection, which took place in the St Catherine's Monastery in Sinai, Egypt. Detailed information of each of the 3306 manuscripts of the library has been collected. Approximately 1000 observations were made on each manuscript. During the assessment, this information was kept on paper. However, a computer filing system was necessary to store the information in digital format which would then allow quick searching of the data. In this paper, the authors describe the design and implementation of a computer database system which helps organize, categorize and access the collected data quickly and efficiently. The article includes an extensive description of the principles behind the database design. It also refers to the complexity of the recorded data and how this fits the relational database model. The paper concludes by exploring the future development of the database and its transition to XML.

Summary
The St. Catherine's library conservation project has focussed so far on the condition assessment of the collection, which took place in the St Catherine's Monastery in Sinai, Egypt. Detailed information of each of the 3306 manuscripts of the library has been collected. Approximately 1000 observations were made on each manuscript. During the assessment, this information was kept on paper. However, a computer filing system was necessary to store the information in digital format which would then allow quick searching of the data. In this paper, the authors describe the design and implementation of a computer database system which helps organize, categorize and access the collected data quickly and efficiently. The article includes an extensive description of the principles behind the database design. It also refers to the complexity of the recorded data and how this fits the relational database model. The paper concludes by exploring the future development of the database and its transition to XML.

Biographies

Dr Athanasios Velios studied archaeological conservation at the Technological Educational Institute (TEI), Athens. He completed his PhD at the Royal College of Arts, London, on Computer Application to Conservation and has been working in digital documentation in conservation for the past eight years. Velios has been a lecturer in digital documentation methods at TEI, and since 2003 has been Research Fellow at the University of the Arts, London/Camberwell College of Arts for the St. Catherine's Library Conservation Project, working mainly on the digital documentation of Byzantine bookbindings.

Professor Nicholas Pickwoad trained with Roger Powell and ran his own workshop from 1977 to 1989. He has been Advisor on Book Conservation to the National Trust of Great Britain from 1978, and was editor of volumes 8–13 of *The Paper Conservator.* He taught book conservation at Columbia University Library School, New York, from 1989 to 1992 and was Chief Conservator in the Harvard University Library from 1992 to 1995. He is now project leader of the St. Catherine's Library Conservation Project based at the University of the Arts, London/Camberwell College of Arts. He also teaches courses in both Europe and America on the history of European bookbinding.

Contact addresses
Dr Athanasios Velios
Camberwell College of Arts
Wilson Building
Wilson Road
London SE5 8LU
UK
Email: a.velios@gmail.com

Prof. Nicholas Pickwoad
Camberwell College of Arts
Wilson Building
Wilson Road
London SE5 8LU
UK
Email: npickwoad@paston.co.uk