Databasics: An Overview of Database Use for Archivists

By Daniel Sokolow, North Shore-LIJ Health System

Increasingly, archivists are turning to database programs as a way to manage their collections. Catalogs, reference requests, accession logs, and other aspects of the trade are often managed via database. It may be helpful to the newly minted (or technologically nervous) business archivist to have a brief overview of the considerations, abilities, and pitfalls of database implementation for business archives.

Which Type of Database to Buy?

Unlike our academic colleagues, most business archives are not subscribers to the major consortia like RLIN that provide database templates for use as a cataloging tool. Nor are many connected to larger library systems with broad electronic catalogs that the archives can piggyback onto. By and large, therefore, business archivists are left to decide for themselves which product best suits their needs. These fall into two basic categories:

- Archives-specific databases
- General database software

The products in the first category, archives-specific databases, are designed with archival management in mind. They have built in modules for collections management, finding aids, accession tracking, etc. For the business archivist with little time or interest in developing their own databases, this can simplify the process. There will inevitably be a learning curve, but on the whole these database packages will have most of what the average archivist will need. The downside to these products is you can only use the tool's built-in capabilities. If you find yourself needing a new database that tracks something completely different, you will not have the ability to customize.

The second category of software, general databases, does provide the kind of flexibility many archivists need. With these products, the scope and functions of the database are in the mind of the creator. Within reason, any database need that develops can be executed using the same techniques for a database of newsletter abstracts as a collections management DB. The down side to this, of course, is that all databases must be created from scratch. Some products have a steeper learning curve than others - they require a certain amount of programming to develop the functionality that is built in to some of the others; on the other hand, if they are already included in other basic desktop software packages, there can be major cost savings.

Some other considerations that should affect the purchasing decision:

- Will these databases only be used internally, or will they be used on an intranet/internet?
- Is the web deployment part of the package, or will other hardware/software be needed?
- How easy is the database to set up and use?
- What is the learning curve for the archivist?
- How much will the IT department need to do to get the system running?
- Will interns or temps be able to learn the system quickly and easily?
- Can images and/or documents be linked directly to the DB record?

Database Design

Whatever package an archives decides to use, the key step before deploying a database is designing the database to meet the needs of the archives. Every archives database should be thoroughly planned out on paper before committing to design within the program. Some questions a database designer should ask:

- What pieces of information do I need to capture?
- Do I need relational databases that will capture information from multiple □ feeder□ databases?
- Do I want fields with controlled vocabularies to keep entries consistent?
- Can I control those fields, or limit who can change them?
- How difficult is it to add or delete fields later on?
- Can I track when a record is modified, and who modified it?
- Should I implement an archival standard of data management?

The critical question is, of course, what information is needed in the database. It is tempting to create a database with every field imaginable in it - "just in case." That approach is usually a poor one. Overlarge databases can slow performance, and much of the information in that scattershot approach can be reasonably separated into more than one DB, and linked together as needed. These multiple databases are known as relational databases, and are linked by common elements - collection ID, provenance, etc. By creating an accession DB, for example, one can create multiple DB records in a collections DB and link them to the same accession DB record. This requires a bit more effort and a good grasp of how to use relational databases, but the end result can save a great deal of time and duplication.

The area of controlled vocabularies is critical for good database design. Anyone who has used a library catalog has seen the results of simple typing errors - four different records for the same subject heading, all because of misplaced periods and missing or added spaces. To avoid these mistakes, many database products will allow a master user to control what entries can be used in certain fields. Thus an entry for "New York Times" must go in exactly that way, not "NY Times" or "Times" or any other variation. This is particularly helpful when dealing with volunteers or interns who may be unfamiliar with the repository's standards. A defined list will allow them to choose the correct entry without causing data problems for later. It does require some thought beforehand about format of entries, who gets to modify the list, and populating the list to begin with. But the long-term integrity of the database makes this an essential part of the planning process.

One major question that should be answered in the design stage is whether or not to encode database records in accordance with any of the various archival data schemes being used in the community. EAD is the most well known, although there are others. In the EAD schema, XML markup language is used to identify content by type, finding aids are encoded using the standard, and then made available on the internet. Databases can be designed with those standards in mind, so that decision should be made at the outset to avoid major headaches later.

Implementation
The most time-consuming, but least difficult aspect of getting archival databases running is the actual population of the database fields. Data entry can be mind numbing, but it is the only way to get a robust database filled in and ready for use. Some database products include shortcuts that can speed data entry. Substitution lists may replace several words with a single keystroke. If a database includes a "duplicate record" function, repetitive entry can be reduced to minor changes when the new record is largely similar to the previous record.

Most of the errors in databases, of course, develop during the data entry phase. Hundreds or thousands of records may get added over a relatively short period of time, and mistakes are inevitable. Establishing a quality control procedure is a good idea, especially if one is using summer students or interns to handle the data entry function. It may be impossible to review every record in a database, but it is a good idea to periodically review some of the records for accuracy.

Access
Presumably, the repository decided back in the discovery phase whether or not to make the catalogs available to the user community. Even if that community is strictly internal, the archives may want to consider restricting access to all of the information within its databases. It seems best not to provide access to the internal management databases of the archives. Accession records, donor records, reference tracking - these internal functions of the archives should be kept internal to the department.

Even in the case of catalog information intended for the user community, there may be reasons to restrict the information that is made available. In the author's health care archive, for example, there are database records that contain patient names. Under the federal government's HIPAA privacy statute, those names are confidential information, and should not be made generally available. As a workaround, in the version of the photographic database available via the intranet, all patient names were redacted from the database record. The original database retains the patient information for continued search needs, but the public is not permitted to see that information. Each archive should be aware of the privacy and security issues surrounding their database information before making the databases available to their users.

In an ideal situation, database catalogs can be made available over the web, via a public internet or a company intranet site. This will hopefully bring additional users to the collections, and make reference easier. Users will be able to search the catalogs from their desks, and explain immediately what it is they need. The author's repository uses an additional software package (an adjunct to its basic database software) to publish catalogs to the intranet. The upside is, of course, greater access. It does raise the issues of additional costs, server management, other layers of complexity, and security issues, so a decision to go to the web should not be made lightly.

Conclusion

In a world of increasing electronic complexity, getting a handle on one's records is essential. There are many tools out there for this purpose, but a database package provides the most flexible, most dynamic tool to organize archival information. There are numerous considerations to moving to electronic data management, including costs, needs, and effort involved. Thorough preparation is essential to proper implementation, and benchmarking against others is usually a good start. Business archivists are expected to perform quickly and efficiently, and a database can make that efficiency easier to achieve.