Utah Division of Archives and Records Service

GUIDELINE FOR PRESERVING RECORDS IN DATABASES

February 2021

PURPOSE: Records and Information Management (RIM) guidelines often focus on managing unstructured data (such as Word documents, spreadsheets, and PDF files), or even semi-structured data, such as email. Databases are considered fully structured data, and require a different approach. This guideline is intended to help records officers consider the data (which are government records) that may be contained within their databases, and to provide descriptions of each part of the process of managing the information in databases.

Contents

What Is a Database?

A database is a set of raw data divided into tables that have rows and columns.

The raw data is meaningless without knowing what the fields are named and how they relate to other fields. Here is an example of raw data:

Printer120170202000000002

Toner2201805030000000014

Paper3201809160000000010

A data dictionary or schema describes the fields, their location within the table, and their properties, so that the raw data makes more sense.

A data dictionary or schema is an important piece of information for every database and will be required if you intend to transfer a database to the Division of Archives and Records Service (State Archives). Make sure that your information technology staff understands this requirement and creates this type of documentation.

Each column in the table represents a field, and each column heading contains the field name. For example:

- In the table below, the Item field is a text field that holds the name of the item being logged.
- The Inventory Number holds an integer or number which uniquely identifies the item.
- The Date Acquired field is a date field and is stored differently than either a text field or a number.
- The Quantity field is another number that further describes the item. There are three rows in this table: one for printer, toner, and paper.

Item	Inventory Number	Date Acquired	Quantity
Printer	1	2-Feb-17	2
Toner	2	3-May-18	14
Paper	3	16-Sep-18	10

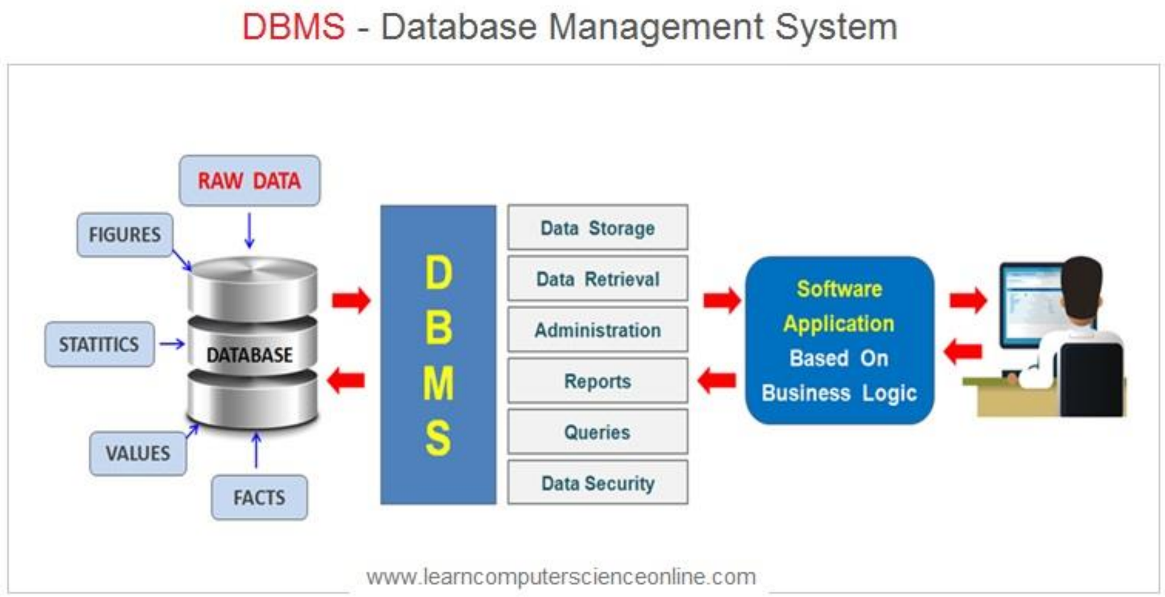
A database management system (DBMS) is software that holds the data and helps keep the integrity of the whole database working properly. Examples include Oracle, SQL Server, and MySQL. Such a system sets rules about the types of fields, and their size and properties that it will accept, as well as security rules about whether an application can view or change a record and how that is done.

To send data to the State Archives from a proprietary database, it must first be exported to a format that is not proprietary, such as .csv spreadsheets or XML.

What Is an Application?

People don’t generally interact directly with a database. Instead, they interact with applications that talk to the database. Do you want to add data, edit something, run a calculation based upon entered values, or print something? The application layer will take care of all of those tasks, and read from or write to the database as needed.

The layers that make up a system include 1) the raw data, 2) the software to manage the data (e.g. Oracle), 3) the application that talks to the database (e.g. FINET), and 4) the operating system (e.g. Windows, Linux). These layers are separate, and can be configured in different ways. Some people may use an application that they purchased, and use one type of database with it, while another customer may purchase the very same application and choose to use a different type of database with it (provided the application knows how to talk to two different types of database management systems).



What Is a Database Record?

From a database manager's point of view, a database record is a row in a table, and nothing more.

From a records manager's point of view, a record is a business transaction that proves that some event transpired, for which documentation is needed and sometimes audited. This transaction may involve more than just an entry into a database row. It often involves multiple tables in a database at once, and it could include email, or even paper in a case file as well. Sometimes those external elements are scanned and attached to a database record, just to keep all the information together.

When you export database records, all of the elements that make up that business record need to be referenced as part of a unit, and not just contain all rows in a given database table.

Why Is This Important to Know?

The Archives will ask you questions about what type of database you are using, and how all the pieces fit together. Preserving a database may mean preserving all of the layers that make up the system, or it may mean creating a migration plan to newer versions of each software layer.

Preserving just the data lessens the ability to use the data the same way it was used in the office. Preserving software is also complicated for the following reasons:

- If the database management system is updated, but the application is not, then the application may no longer be able to connect to the newer database and the system will be unusable. Since database management system software is more widely used than any one application, it is more likely that an application could be abandoned by developers before the database software would be, which creates an imbalance between the two layers. Application software is more susceptible to changing market conditions, so if there is no market for the software, there is no motivation for developers to continue supporting it. Records created with popular software today could become unreadable in the not-too-distant future if developers abandon their support of the application. The application will be stuck in neutral while the database management system moves forward. It will break; the only question is how soon, and how will you move the abandoned records to a new system which is being supported.
- If the application is proprietary and requires a current license to run, but there is no way to pay for a license because the application's creators are no longer in

business, then you likely won't be able to even open the application. If an application is open-source (non-proprietary), it reduces this risk.

- Operating systems represent another layer of the software stack, which impacts both database software and application software. Keeping all of these in sync is extremely challenging, and individual pieces may no longer communicate together once enough time has passed.

One possibile solution is emulation, which simulates the original environment and all of its capabilities. The archival community has made inroads in providing a framework for this, but it is not fully developed and will need ongoing work and support.

If data is exported from the database, then it is no longer connected to the application(s) that supported it, and future users may not understand how the data was used. The context will be lost. If your agency has a continuing need to use this data, then it is likely that your information technology staff will take measures from time to time to update each layer so that the system continues to work. If, on the other hand, the system is due to be decommissioned, and if the records it contains have permanent historical value, the data will need to be sent to State Archives in a way that ensures that the data will remain comprehensible and trustworthy to future users.

We recommend saving screenshots of the user interface, as well as any user manuals, as they can help preserve context.

How Do You Set a Retention on a Database?

Managing retention schedules within a database environment is a difficult prospect because the same field may be used in more than one business workflow. Each transaction may be supported by different work groups which have different functions, and therefore different needs as far as documenting the transaction.

Data should be kept in the live database for as long as all the business groups need that information. If the data tends to be changed over time, and one business function needs to record what the data looked like at a moment in time, then data should be exported for that purpose, and kept for the length of time needed, or else the system should support audit trails directly in the database, where changes to field values are recorded and can be viewed. Audit trails generally include who made the change and when, and may include the original value that was changed.

If your agency is creating a new system, make sure that audit trails are recorded, and that you are provided a way to export entire business records, regardless of the data complexity that makes up that record.

If records should be deleted, make sure system developers understand the requirement to export entire business records, and facilitate a way to do it, where retention length is stored in a field that can be changed if needed. The system should alert you to the records that have met retention, and should provide a way for you to review records that have met retention and then document disposition decisions. An audit trail should record which records were deleted, when, and approved by whom. Ideally, these requirements should be built into any off-the-shelf system solution as well, though this will vary system to system (which, in turn, may impact the ability to easily manage and preserve database records).

To appraise records inside of a database system, identify each workflow supported by the system, and the types of functions that generate records. Describe each business function and determine how long the data is needed to support that function. You may create separate retention schedules and management plans for sets of functions that have similar retention requirements, or create one retention schedule for the whole.

Typically, there are two different use types for databases, which can help you determine the best approach to take for retention:

- The first is the “case file” model where a case file is created, several users add information, the case file stays in the system for a set amount of time, and then is no longer needed and can be exported or deleted.
- The second is the “continuous use” model where information is added and continually changed or updated, never really expiring. In this case, the database may be considered ongoing, and the only way to save information is to create a snapshot at certain intervals, export it, and save it for a desired length of time. This is common for databases where the information is considered “permanent” notwithstanding its constant updating. But you may also use this methodology for temporary records whose fields are likely to be changed by competing workflows. This should be determined on a case-by-case basis, and with the support of any technology solution partners who can help determine the frequency that supported data is backed up on their end.

Determine whether your database follows the “case file” or “continuous use” model. For case files, decide how long a case is active, and what should happen to it when it becomes inactive. For continuous use data, decide how often (if ever) snapshots of data should be taken, based upon frequency of changes to the data, and the interest in having access to the captured data. Snapshots might consist of a copy of the whole database, or a derivative version containing discrete business records if the data is only needed temporarily. If the database is considered permanent with historical value, then copies of the whole database, with full documentation, are needed as described below.

- For case files, do you want to keep the data in the database for the entire time the data is needed, or do you want to export it and store it outside of the database until retention has been met?
    - If exported and deleted from the main system, you are responsible for making sure the data continues to be readable for the entire retention length until final disposition. If that length exceeds 10 years, be aware that storage technology has been known to fail if data is stored for long periods of time, and often even for short periods of time. If data is needed for several decades, the likelihood of needing to migrate the content to newer and more accessible formats increases, as the software used to create the data becomes obsolete. This data will need to be actively managed by you.
        - If you intend to store this data offline yourself, contact the Archives for the latest recommendations of storage media and preservation policies, such as keeping more than one copy, stored in separate locations, and capturing minimal metadata to help ensure file integrity over time. See the “What Is a Checksum” section below.
        - If, on the other hand, you keep the data in the original database for the full length of retention, even during its inactive stages, then regular updates to that system should keep the data in good condition until such time that it can be deleted or sent to the Archives.
- For continuous use data, you may export a copy of the data to use as a snapshot of data values in a moment of time, while the data stays live in the database and is perhaps overwritten.
    - If the retention is temporary, you may choose to condense the output and restructure it to only include data that supports documentation of a transaction or record-creation event, rather than trying to export an entire database. Include metadata such as date captured. The data may be in the form of a report.
    - If the retention is permanent, then it is important to capture a copy of the whole database, and follow the instructions about sending the records to the State Archives (see below).

Sending Records to the State Archives

- Make a plan up front about how transferring records will take place. Record this plan and share it with your RIM Specialist at the State Archives. Ensure the following points are addressed:
  - What kind of data will be sent
  - What format(s) the data will be in
  - How much data will be sent (volume)
  - How frequently the data will be sent
  - Via what method the data will be sent (hard drive, FTP, other)
- Make sure you have a way to export your data to flat file (i.e. a file with a single table of data) so it can be saved in .csv or .xml format. If this is not possible, you will need to partner with the Archives to determine an alternative. If the data is in an unusual form, the Archives may not have the capacity to adequately receive and preserve the information.
- Include all contextual information—screenshots, data dictionaries, user manuals, etc.—which gives the data meaning.
- Talk to your vendor and information technology staff about these requirements. This is especially important for custom-built systems, where it is better to include these requirements prior to a system being launched, so that these tasks can be automated. If you purchased a system that does not have these capabilities, determine if the underlying database is accessible directly to technical staff, and have them come up with a procedure to export the data. This may involve script writing or custom programming. If the underlying database is not accessible, the entire system may need to be migrated to a product that can support these requirements.

Plan for new systems going forward that incorporate retention and export capabilities; ***make these features a priority.***

- The State Archives recommends that agencies send the Archives a copy of permanent electronic records whenever data is exported, so that necessary preservation actions can take place, even if the agency still retains legal custody of the data until retention has been met.
- Data should not sit in a box at the Records Center without preservation action.

Preservation Standards

Electronic records preservation is a complicated process for all types of records, not just databases. If data is maintained in its native format indefinitely, and technology professionals are actively keeping the database in good condition, upgrading when necessary, then the system as a whole is essentially being preserved. The data, however, will only be preserved if it is captured in audit trails or exports before it is changed. For records to be preserved archivally, the following conditions must be met:

- **The data must continue to be accessible over time.** This is much harder to do if data is in a proprietary format. Open source options—or at least open specification options—are preferable, so that the data is recorded in an understandable way, following rules that are known, so that future software can read it and interpret the data correctly.

- **The data must be authentic.** The information should accurately reflect the source from which it came. All of the parts that contribute to this authenticity must be maintained, including documentation of the records transfer process, what software and version were used, how the records were exported, where they were stored, what format they were in, what actions were done to them, what security measures were in place, etc. The internal structure of the information must be maintained as well, including relationships to other records. Preservation metadata includes descriptive and technical metadata of what the files are all about, how they came to be, and how they relate to other records around them.

- **File integrity must be maintained.** This means that the data must not change unless those changes are intentional and well documented. Changes may include file format migration. Each time a file is recorded to media, or a format is transformed to a newer version, a checksum must be created and recorded for that file and periodically re-checked for the entire length of the retention. If the checksum fails against the stored value, then integrity is lost and may only be regained if another copy is available that has the original checksum.

Checksum verification is possible to do with offline media, but can be more efficiently done if the records are stored in a preservation system that performs this function automatically.

Agencies should investigate digital preservation systems if they intend to preserve records themselves instead of sending the data to Archives.

What Is a Checksum?

A checksum is a methodology for determining fixity, or whether or not a record has changed. That's important if you're being audited, or if you need to prove data integrity in court. To capture fixity, an algorithm is run against a file, and a unique string of characters is produced. This acts like taking a fingerprint of the file. Here are some sample checksums, along with their directory name and file name:

↓Checksum	**↓Directory & file name**
5a5b03797cb897b9a4e8e2c2d2d67289	data/11.29.07 Utah Drug Court Conference.pdf
a64e47c82bc8b94b649360649362a034	data/11.15.07 KSLKUED Media Briefing.pdf
084b17ade7c1e9ded9313939a4320a61	data/11.10.07 Veterans Day Program.pdf
aec10b305c1abd651ce1f7c462877379	data/11.28.07 Rural Coalition Meeting.pdf

Depending on the type of algorithm, the length of the string could be short (such as 25 characters, as seen above) or long (such as 256 characters). These different algorithms have names, such as MD5, SHA1, SHA256, SHA512, etc. Software exists which will automatically create a list of files on a specific drive or in a specific folder, along with their associated checksums.

The above checksums were produced by a tool called Bagger (see https://github.com/LibraryOfCongress/bagger), which uses the BagIt specification written by the Library of Congress to facilitate data validation during the records transfer process. This software can also rerun checksum validation on demand to determine if the current value is the same as the original value. If the checksum is the same, congratulations! Your file has not become corrupted. If it has changed, that means that the bits in the file have been altered (a 0 turned into a 1, or a 1 into a 0). Depending on the extent of corruption, the files may or may not be able to be read and interpreted by software. Either way, any data corruption means that the file has lost its integrity and can't be entirely trusted to be authentic. Therefore, it is a best practice to keep multiple copies of files (ideally backed up on a server or in a digital preservation system), so that if one copy gets corrupted, it can hopefully be replaced by an uncorrupted version.

Note that a backup copy is not one of the copies. These multiple copies should be stored separately and independently, and each copy preferably backed up. If a file becomes corrupted, its backup will be, too. The file will need to be restored from one of the separate copies, provided that the separate copy validates with the same checksum of the original. This is also why checksums should be stored independently of storage systems, so you can always prove that file integrity was lost, and not just trust the storage system vendor that "everything's fine."

What Do You Need to Do?

1. Identify each system managed by your agency. Work with the Archives to establish appropriate retention schedules for that information.
2. If your system will remain in-house indefinitely, and the records contained within it are not of permanent value, make sure that system developers understand the business rules required for retention and have a way of following them from within the application (either export and retain outside the system, or purge whole records directly within the system).
3. If your system contains records of permanent value, work with the State Archives and your system developers to facilitate a way to export and fully document the data. Specifically:
    a. Find out what kind of database your system is using.
    b. Find out if the data can be exported to a flat file, in .csv format or .xml.
    c. Ask your technology staff for a copy of the data dictionary or schema and any user manuals which exist.
    d. Export your data.
4. For electronic records sent to the Archives or Records Center:
    a. Use a tool such as Bagger to calculate a checksum of these files.
    b. Write the files to stable media, and revalidate the checksums while on the media. If revalidation fails, rewrite the files again to different media and recheck until you have clean, validated copies on that media.
    c. Make two or more copies of the data. Revalidate each copy. Store them in separate locations.

Please don't hesitate to contact the Archives at recordsmanagement@utah.gov to discuss plans for, or the need to, transfer database records, and we will work with you to find an appropriate solution!

Resources

ARMA International publications:

- Metadata: A Basic Tutorial for Records Managers: An ARMA Standards Report (03-2019)
- Understanding Electronic Records Storage Technologies ARMA International (TR 26-2014)
- Using DoD 5015.02-STD Outside the Federal Government Sector ARMA (TR 04-2009)

Department of Defense publication:

- Electronic Records Management Software Applications Design Criteria Standard (DoD 5015.02-STD)

International Organization for Standardization (ISO) publications:

- ISO 13008:2012: Information and documentation — Digital records conversion and migration process
- ISO 16175-1:2010: Information and documentation – Principles and functional requirements for records in electronic office environments – Part 1: Overview and statement of principles
- ISO 16175-2:2011: Information and documentation – Principles and functional requirements for records in electronic office environments – Part 2: Guidelines and functional requirements for digital records management systems

DAS | Utah Division of Archives and Records Service

346 S Rio Grande St • Salt Lake City, Utah 84101

Telephone (801) 531-3848 • Facsimile (801) 531-3854 • archives.utah.gov