

Long Term Database Archiving



Author



This presentation was prepared by:

Jack Olson
CTO

NEON Enterprise Software, Inc.
11044 Research, Suite D300
Austin, TX 78730
Tel: 512-241-7335
E-mail: jack.olson@neonesoft.com

This document is protected under the copyright laws of the United States and other countries as an unpublished work. This document contains information that is proprietary and confidential to NEON Enterprise Software, which shall not be disclosed outside or duplicated, used, or disclosed in whole or in part for any purpose other than to evaluate NEON Enterprise Software products. Any use or disclosure in whole or in part of this information without the express written permission of NEON Enterprise Software is prohibited.

© 2004 NEON Enterprise Software (Unpublished). All rights reserved.

Emergence of Data Management Functions

The Long Term Data Storage Problem

Long Term Data Storage Solutions

Database Archiving Capabilities

Difference between DBA and DM

■ Database Administration

- Backup/Recovery
- Disaster Recovery
- Reorganization
- Performance Monitoring
- Application Call Level Tuning
- Data Structure Tuning
- Capacity Planning

Managing the database environment

■ Data Management

- Database Security
- Data Privacy Protection
- Data Quality Improvement
- Data Quality Monitoring
- Database Archiving
- Data Extraction
- Metadata Management

Managing the content and uses of data

Data Management Functions

■ Database Security

- Authorization Auditing
- Access Auditing
- Intrusion Detection
- Replication Auditing

■ Data Quality

- Data Profiling
- Data Quality Assessment
- Data Cleansing
- Data Quality Filtering
- Data Profile Monitoring

■ Data Archiving

- Short term Reference Database
- Long Term Database Archiving

■ Data Extraction

- Maintain privacy
- Maintain Security

■ Metadata Management

- Complete Encapsulation
- Change History Auditing

Database Administration Functions



- Very well defined tasks
- Very well defined Job Title and Description
- Overwhelming vendor support
- DBMS architectures fully supportive
- Functions fall entirely in IT
- Must be done well to support efficient operational environment

Database Management Functions

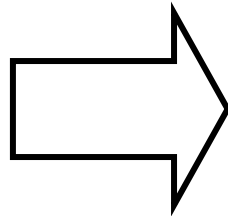
- **Tasks definitions are emerging**
- **No standard Job Titles or Descriptions**
- **More aligned with business units than IT**
- **IT management has not been supportive (NMP)**
- **Executive management has not been supportive**
- **DBMS architectures built without consideration of DM**
- **Little Vendor Support**
- **Companies have accrued many penalties for not paying attention to DM requirements**

Emerging Data Management Drivers

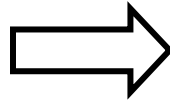


Recent Regulations:

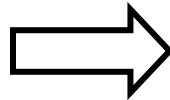
- Corporate Governness
- Data Privacy
- Data Retention
- Data Accuracy



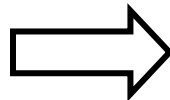
Increasing Data Quality Costs



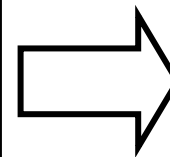
Increasing Data Volumes



Increasing uses/ users of data

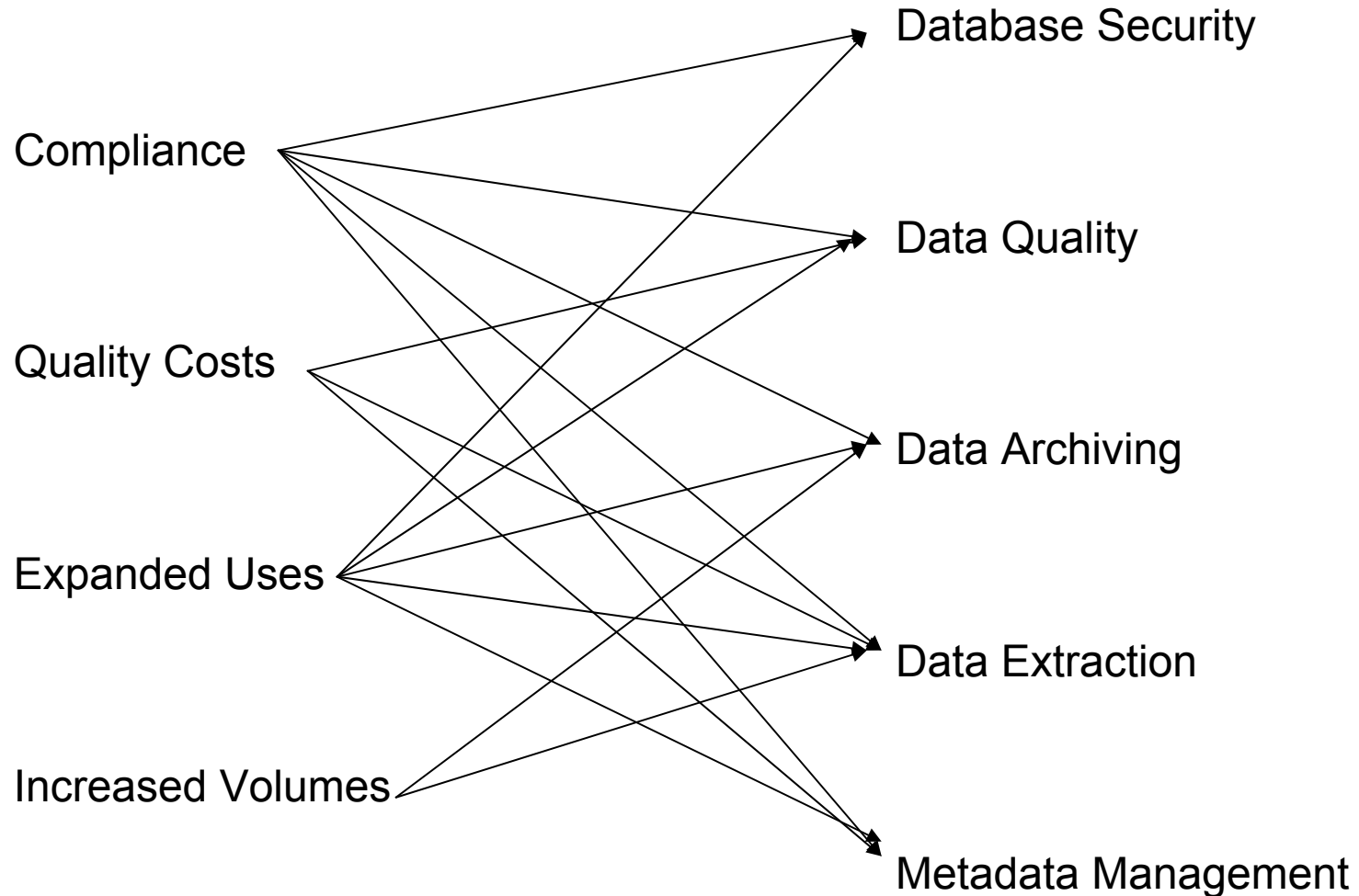


More
Emphasis and
Spending on
Data Management
Functions



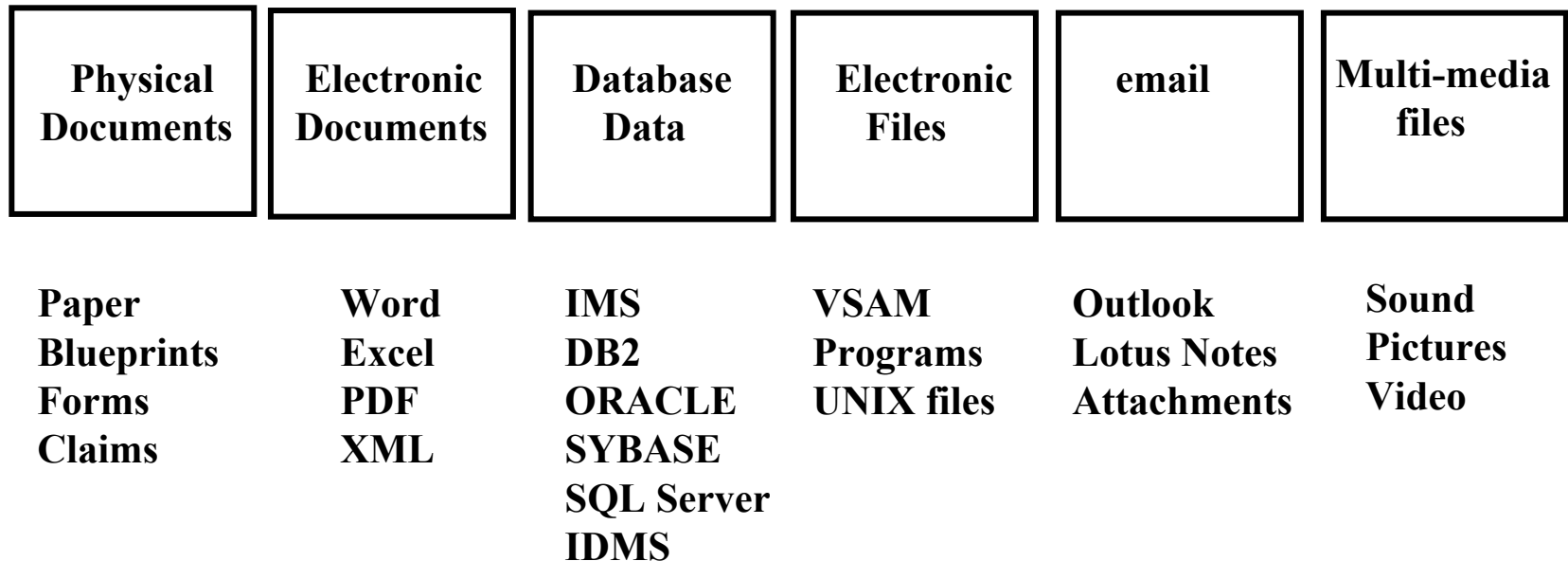
Significant
Tangible
Benefits

Drivers Impacts on Functions



Long Term Database Archiving

Data Archiving

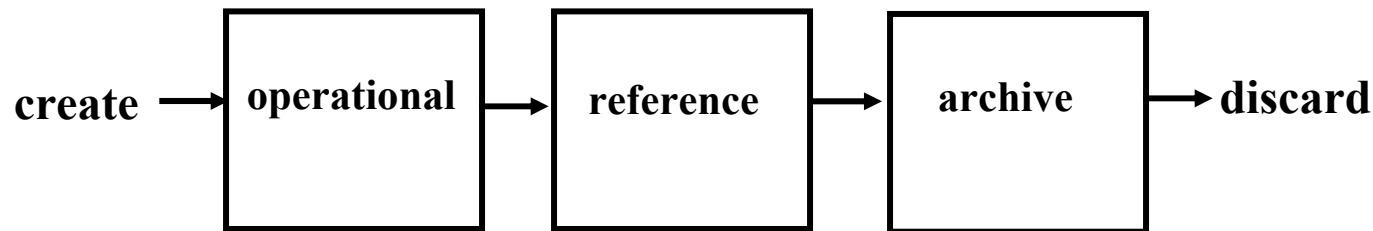


Database Archiving

Database Archiving:

The process of removing selected data records from operational databases that are not expected to be referenced again and storing them in an archive database where they can be retrieved if needed.

Database Data States



needed for
completing
business
transactions

needed for
reporting
or
expected
queries

no expected
needs for
business
transactions
or reference

mandatory retention period



Data Retention Requirements refer to the length of time you need to keep data

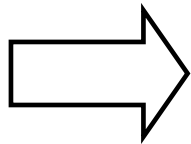
determined by laws
determined by business needs

Data Archiving is a process used to move data from the operational database to another data store to be kept for the duration of the retention period when it is unacceptable to keep the data in the operational database for that long.

large volumes of data interfering with operations
need for better protection from modification
need for isolation of content from changes

Why Is This Important

External Regulations
Internal needs for analytic applications



We need to keep more data: a lot more data
for more years: a lot more years
We need to preserve original content and meaning

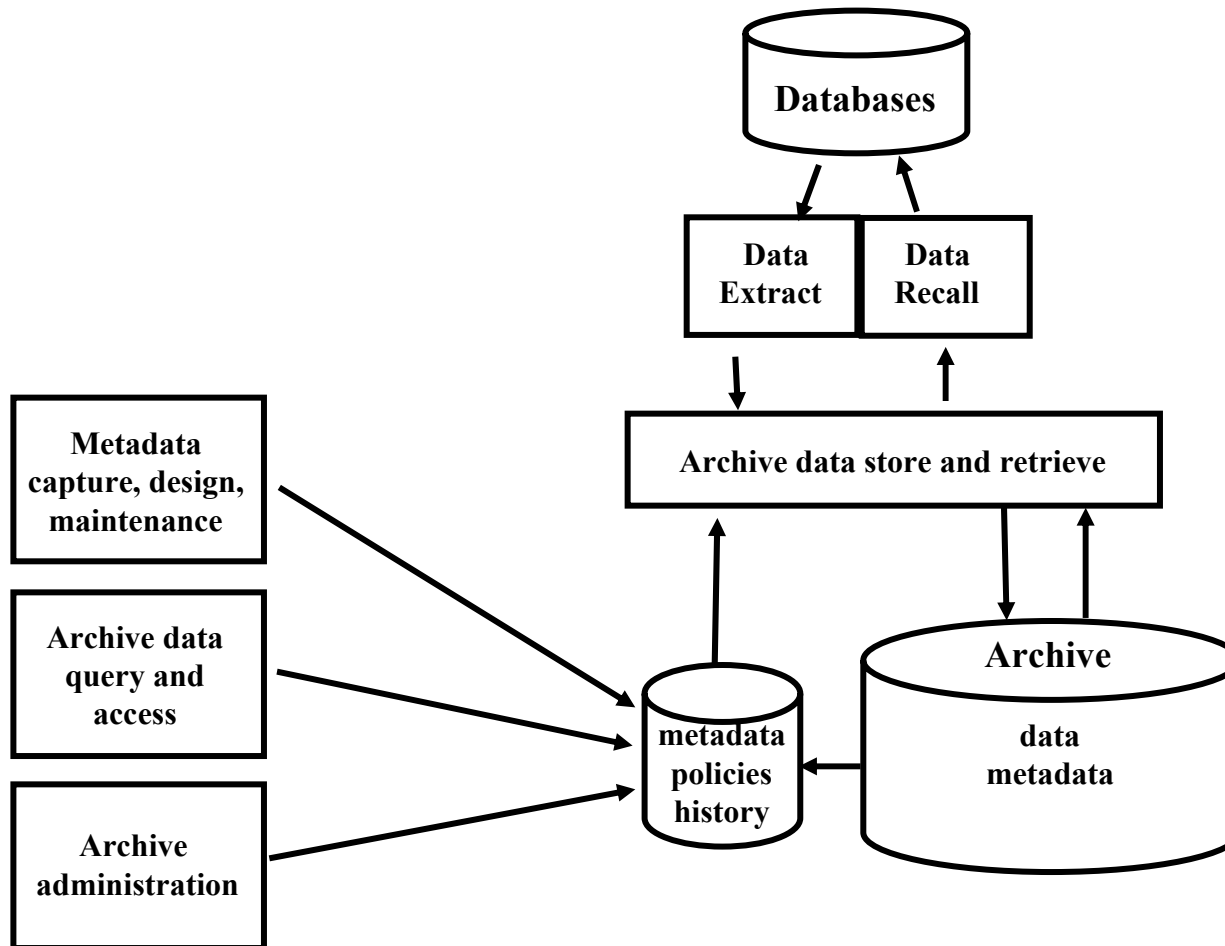
—————→ Old retention period

—————→ New retention period

What Solutions Are Out There?

- **Keep Data in Operational Database**
- **Store Data in UNLOAD files**
- **Move Data to a Parallel Reference Database**
- **Move Data to a Database Archive**

Components of a Database Archiving Solution



What Do You Need to Support Database Archiving?

- Policy based archiving: logical selection
- Keep data for very long periods of time
- Store very large amounts of data in archive
- Maintain Archives for ever changing operational systems
- Become independent from Applications/DBMS/Systems
- Become independent from Operational Metadata
- Protect data from unwanted changes
- Guarantee authenticity of data
- Access data when needed; as needed
- Discard data after retention period

Policy based archiving

■ Why :

- Business objects are archived, not files
- Rules for when something is ready can be complex
- Data ready to be archived is distributed over database

■ Implications:

- User must provide policies for when something is moved

■ How:

- Full metadata description of data
- Flexible specification of policy : “WHERE clause”
- Support accessing data outside archive set

Keep Data for a Long Time

- **Why : retention requirements in decades**
- **Implications:**
 - Archive will outlive applications/DBMS/systems that generated them
 - Archive will outlive people who designed and managed operational systems
 - Archive will outlive media we store it on
- **How:**
 - Unique data store
 - Application/DBMS/system independence
 - Metadata independence
 - Continuous management of storage
 - Continuous management of archive content

Keep Very Large Amounts of Data

■ Why :

- Large volumes of data today
- Increasing rates of data volume growth
- Long retention periods

■ Implications:

- Archive won't fit in DBMS solutions
- Must partition contents
- Cannot read all of archive to satisfy queries
- Must support management functions at partition level

■ How:

- Unique data store
 - Supports partitioning of data
 - Unlimited number of partitions
 - Manages partitions independently
- Indexing and scoping

Maintain Archive for Changing Operational Systems

■ Why :

- Metadata changes frequently
- Applications are re-engineered periodically
 - Change DBMS platform
 - Change System platform
 - Replace with new application
 - Consolidate after M&A

■ Implications:

- Archive must support multiple variations of an application
- Archive must deal with metadata changes

■ How:

- Manage applications as major archive streams having multiple minor metadata differences streams
- Achieve independence from operating environment

Achieve Application Independence



■ Why :

- Operational applications will not be available
- Operational systems will not be available

■ Implications:

- Archive must satisfy all query requirements from within
- Archive data must include metadata needed for interpretation of data
- Archive system will be moved to new systems from time to time

■ How:

- Store metadata and data in archive together
- Implement archive system on multiple systems
- Implement archive system on new systems

Achieve Metadata Independence

■ Why :

- Operational metadata is inadequate
- Operational metadata changes
- Operational systems keep only the “current” metadata
- Data in archive often does not mirror data in operational structures

■ Implications:

- Archive must encapsulate metadata
- Metadata must be improved

■ How:

- Metadata Capture, Validate, Enhance capabilities
- Store structure that encapsulates with data
- Keeps multiple versions of metadata

Protect From Unwanted Changes

■ Why :

- Potential use in lawsuits/ investigations
- Potential use in business analysis

■ Implications:

- Cannot be managed in operational environment

■ How:

- Special storage structure that does not support I/U/D
- Do not modify archive data on metadata changes
- Checksum for detection of sabotage
- Limit access to functions
- Audit use of functions
- Maintain offsite backup copies for restore if sabotaged

Achieve Authenticity of Data

■ Why :

- Need to guarantee authenticity in lawsuits/ investigations

■ Implications:

- Cannot be managed in operational environment

■ How:

- Full data protection as described above
- Ability to return data to original form bit-for-bit

Access Data Directly From Archive



■ Why :

- Cannot depend on application environment

■ Implications:

- Full access capability within archive system

■ How:

- SQL like interface
- LOAD format output for
 - For load into a database
 - May be different from database came from
- Recall format output for
 - Showing original input bit-for-bit
- Requires full and accurate metadata
- Ability to review metadata
- Ability to function across metadata changes

Discard Function

- **Why :**
 - Legal exposure for data kept too long
- **Implications:**
 - Data cannot be kept in archive beyond retention period
 - Must be removed with no exposure to forensic software
- **How:**
 - Policy based discard
 - System level function
 - Tightly controlled and audited
 - True “zero out” capability
 - Discard from backups as well

So Where do we store the archive?

■ Relational Database

- NOT
- Only supports 1 definition of data
- Problem with very large amounts of data
- Cannot protect from unwanted changes
- Requires excessive administration

■ New Database Archive Structure

- Stores data and metadata
- Partitions data by metadata groupings
- Unlimited number of partitions
- Does not support INSERT/UPDATE/DELETE functions
- Manages by partitions
- Indexed and scoped

Summary Points

- Keeping data in operational systems is a bad idea
- Putting data in UNLOAD files is a bad idea
- Putting data in a parallel references database is a bad idea
- Using a DBMS to store the archive does not work
- Database archiving requires a great deal of data design
 - Establishing and maintaining metadata
 - Designing how data looks in the archive
 - Achieving application independence
- Database archives must be continuously managed
 - Copying data for storage problems (e.g. media rot)
 - Copying data for system changes
 - Copying data for data encoding standard changes
 - Logging, auditing, and monitoring
 - Archive events
 - Partition management
 - Accesses
- **MUST HAVE full time Database Archivist on staff**



Intelligent Solutions for Enterprise Data. Depend On It.