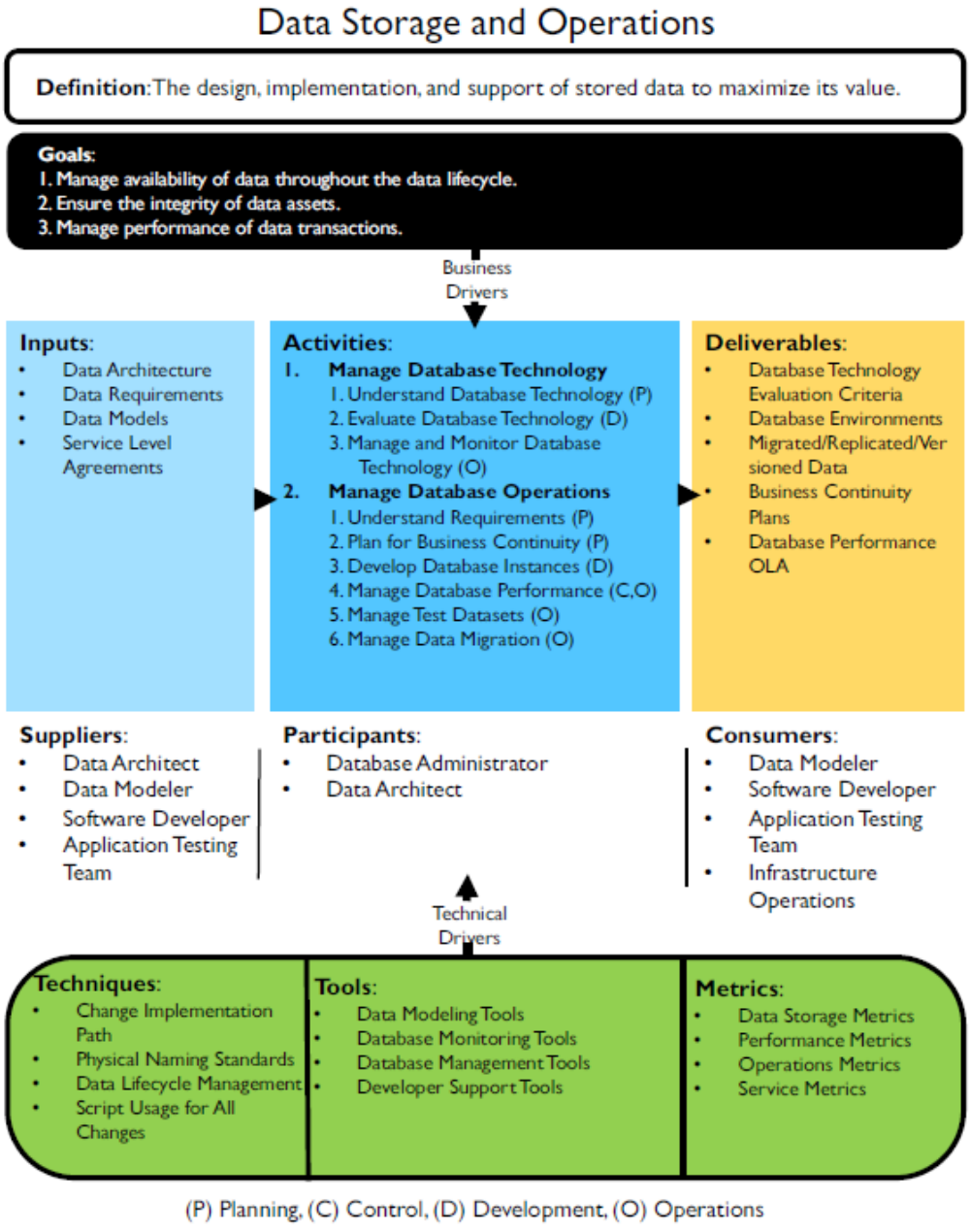
Data Storage and Operations

1 Introduction

True custodians of the data – data at rest.

Data Storage and Operations includes the design, implementation and support of stored data, to maximise its value throughout the lifecycle, from creation to disposal. Two sub-activities:

- **Database support:** activities related to the data lifecycle from implementation of a database environment, through obtaining, backing up and purging data. It includes ensuring the database performs well. Monitoring and tuning.
- **Database technology support:** Defining technical requirements that meet organisational needs, defining technical architecture, installing, maintaining technology, and resolving related issues.



1.1 Business Drivers

Business continuity

1.2 Goals and principles

Goals:
1. Manage availability of data throughout the data lifecycle.
2. Ensure the integrity of data assets.
3. Manage performance of data transactions.

Highly technical side of data management. Guiding principles:

- **Identify and act on automation opportunities:** Automate database processes, develop tools and processes that shorten cycles, reduce errors and development rework. Do in collaboration with data modelling and Data Architecture.
- **Build with reuse in mind:** Develop abstracted and reusable data objects
- **Understand and appropriately apply best practices:**
- **Connect database standards to support requirements:** SLAs reflect DBA-recommended and developer-accepted methods of ensuring integrity and data security.
- **Set expectation for the DBA role in project work:** Include the DBA in all SDLC phases of a project

1.3 Essential Concepts

1.3.1 Database Terms

- **Database:** Any collection of stored data
- **Instance:** An execution of database software controlling access to a certain area of storage.
- **Schema:** A subset of database objects contained within a database or instance, usually with something in common. Used to isolate sensitive data from general user base.
- **Node:** An individual computer part of a distributes database
- **Database abstraction:** An API (common Application Interface) is used to call database functions.

1.3.2 Data Lifecycle Management

The DBA is the custodian of all database changes. The DBA defines the precise changes, implements and controls the changes. The DBA must have a plan to back out the changes.

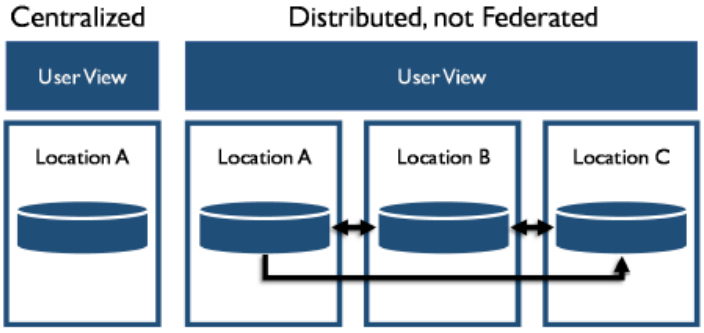
1.3.3 Administrators

- **Production DBA:** Responsible for data operations management
    - Ensures performance and reliability of the database – performance tuning
    - Backup and recovery mechanisms
    - Implement clustering and failover of the database if continual data availability is required
    - Other database maintenance activities such as archiving.
    - Deliverables of Production DBAs:
        - A production database environment
        - Mechanisms and processes for controlled implementation of changes to databases in the production environment
        - Mechanisms for ensuring availability, integrity and recoverability of data in response to all circumstances that could result in loss or corruption of data.
        - Mechanisms for detecting and reporting database errors

- Data availability, recovery and performance in accordance with SLAs
- Mechanisms and processes for monitoring database performance

- **Application DBA:** Responsible for databases in all environments (Development/test, QA and production). Part of an application support team
- **Procedural and Development DBAs:**
  - Administration of procedural database objects (stored procedures, triggers and user-defined functions)
  - Development DBAs focus on creating and managing special use databases such as "sandbox".
- **NSA:** Network Storage Administrators support data storage arrays.

1.3.4 Database Architecture Types



1.3.4.1 Centralised Databases

All the data in one place. All users come to the one system, no alternatives if it is unavailable.

1.3.4.2 Distributed Databases

Quick access to data over many of nodes, each offering local computation and storage. DBMS replicates across nodes, and can detect and handle failures. MapReduce divides a data request into small fragments over the nodes.

1.3.4.2.1 Federated Databases

A federated database system maps multiple autonomous database systems into a single federated database, connected via a network. Remain autonomous but allow sharing of data.

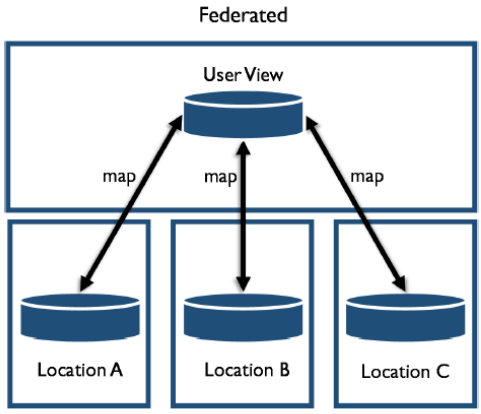
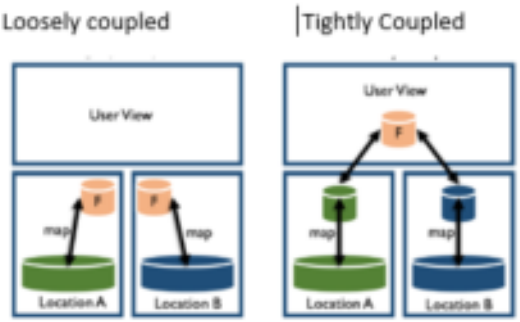


Figure 56 Federated Databases

A FDBMS can be loosely or tightly coupled:

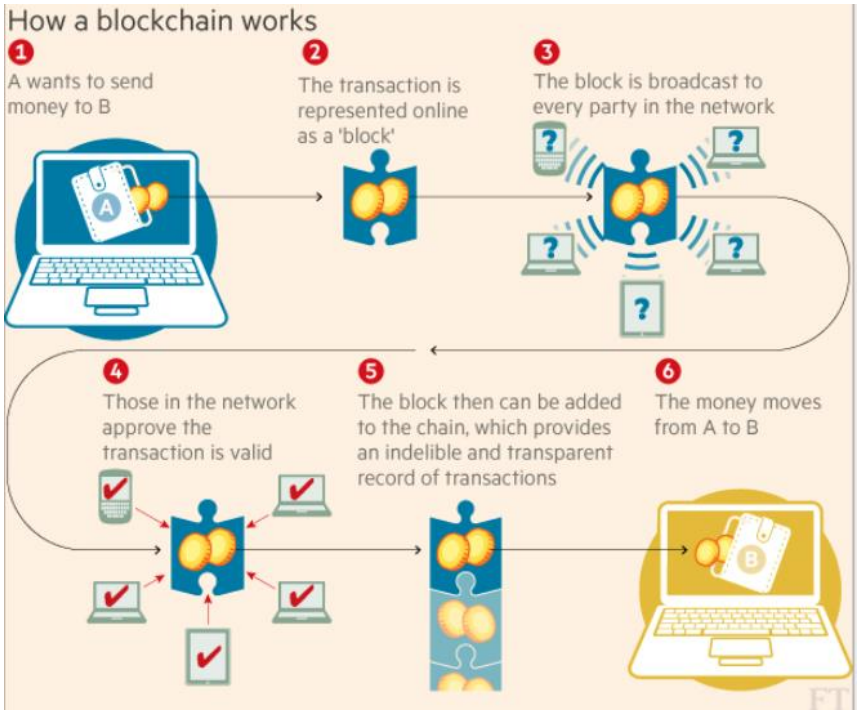
- **Loosely coupled**: Component databases construct their own federated schema.

- **Tightly coupled**: Component systems use independent processes to construct and publish an integrated federated schema



1.3.4.2.2 Blockchain database

A type of federated database used to securely manage financial transactions. Each transaction has a record. The database creates chains of time bound groups (blocks) that also contain information from the previous block in the chain. Hash algorithms create information on the transactions which can never change. Tampering is evident if the hash values no longer match.



1.3.4.3 Virtualisation / Cloud platforms

Virtualisation (Cloud computing) provides computation, software, data access and storage without end-user knowledge of the physical location or configuration of the system.

- **Virtual machine image:** Users purchase virtual machine instances for a limited time. Either load own database or use an optimised installation of the database.
- **Database-as-a-service (DaaS):** Database service provider installs and maintains the database and application owners pay according to their usage.
- **Managed database hosting in the cloud:** The cloud provider hosts the database and manages it on the application owner's behalf.

DBAs and Network and System Administrators need to establish a systematic project approach to the following functions (as well as the security aspects):

- **Standardisation/consolidation:** Based on Data Governance policy, consolidation reduces the number of data storage locations and standard procedures are developed by DBAs and Data Architects.
- **Server virtualisation:** Enables reduction of cost in infrastructure management.
- **Automation:** of data tasks
- **Security:** Integrate security of virtual systems with security of physical infrastructures

1.3.5 Database Processing Types

ACID: 1980s – reliability within database transactions. Relational – SQL Server.

- **Atomicity:** All operations are performed, or none are. If one fails, the entire transaction fails
- **Consistency:** the transaction must meet all rules defined by the system at all times and must void half-completed transaction
- **Isolation:** Each transaction is independent
- **Durability:** Once complete, the transaction cannot be undone

BASE: Increase in volumes and variability of data, the need to document and store **l**ess structured data, read-optimised data workloads, greater flexibility of scaling and design

- **Basically available:** System guarantees some level of data availability even if there are node failures
- **Soft state:** System in constant state of flux. Data may be available but not current
- **Eventual consistency:** Data eventually becomes consistent through nodes.

Item	ACID	BASE
Casting (data structure)	Schema must exist	Dynamic
	Table structure exists	Adjust on the fly
	Columns data typed	Store dissimilar data
Consistency	Strong Consistency Available	Strong, Eventual, or None
Processing Focus	Transactional	Key-value stores
Processing Focus	Row/Column	Wide-column stores
History	1970s application storage	2000s unstructured storage
Scaling	Product Dependent	Automatically spreads data across commodity servers
Origin	Mixture	Open-source
Transaction	Yes	Possible

- **CAP:** Brewers theorem – the larger the distributed system, the lower the compliance to ACID. At most two of the following properties can exist in a shared data system:
  - **Consistency:** The system must operate as designed and expected at all times
  - **Availability:** The system must be available and respond to each request
  - **Partition Tolerance:** The system must be able to continue operations during occasions of data loss or partial system failure.

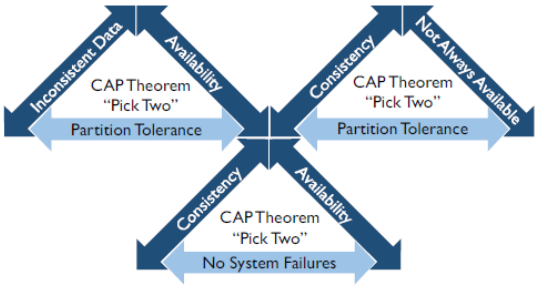


Figure 58 CAP Theorem

Used to drive Lambda Architecture for big data, which uses two paths for data: (refers to ch14 where it appears as 14.1.3.7 Services-Based Architecture (SBA))

- Speed path: Availability and Partition Tolerance
- Batch path: Consistency and Availability

1.3.6 Data Storage Media

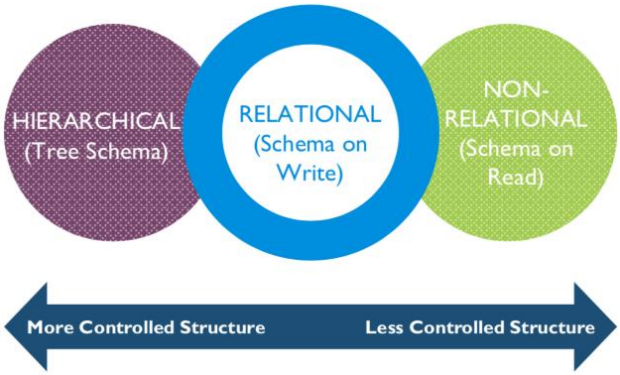
- **Disk and Storage Area Networks (SAN):** Disk arrays and SAN. Persistent storage.
- **In-Memory:** In-Memory Databases (IMDB) are loaded into volatile memory where all processing takes place. Faster response than disk.
- **Columnar Compression Solution:** For data sets where data values are repeated to a large extent and may be compressed. Reduces I/O time
- **Flash Memory:** Solid state

1.3.7 Database Environments

There are various environments used during the system development lifecycle.

- **Production Environments:** The technical environment where all business processes occur. The DBA team should be the only team implementing changes, adhering strictly to standards and procedures.
- **Pre-Production Environments:**
    - **Development:** Slimmer version of production for developers to test new code or patches
    - **Test:** Used for:
        - **Quality Assurance testing (QA)**
        - **Integration testing**
        - **User Acceptance Testing (UAT)**
        - **Performance Testing**
    - **Sandboxes or Experimental Environments:** Allows read only access to production data for experimentation.

1.3.8 Database Organisation



1.3.8.1 Hierarchical

Oldest database model. Tree structure with one to many parent-child relationships.

1.3.8.2 Relational

Based on set theory and relational algebra. Set operations (union, intersection, minus) in the form of SQL (Structured Query Language) are used to retrieve data. To write data the schema must be known (schema on write). Relational databases are row-oriented. The database management system is called RDBMS

- **Multidimensional:** Allows searching using several data element filters simultaneously. Uses a type of SQL called MDX (Multidimensional eXpression)
- **Temporal:** Built in support for handling time.

1.3.8.3 Non-relational (NoSQL – Not only SQL)

Schema-on-read allows data to be read in different ways. NoSQL means the storage structure is not bound by tabular design. NoSQL databases are used in Big Data and real-time web applications as they are optimised for retrieval and appending operations.

- **Column-oriented:** Used in BI applications as redundant data can be compressed. Difference between column-oriented and row-oriented (usually relational):
    - **Column-oriented is more efficient when:**
        - aggregating over many rows
        - new values for that column are applied at once as there is no need to touch columns for the other rows
        - Online Analytical Processing (OLAP)
    - **Row-oriented is more efficient when:**
        - Many columns of a single row required at once
        - Writing a whole row of new data
        - Online Transaction Processing (OLTP)
- **Spatial:** Store and query data that represents objects defined in geometrical space. Use special indexes to perform database operations. Open Geospatial Consortium standard.
- **Object/Multi-media:** Hierarchical storage Management System manages the media objects
- **Flat File Database:** Data in rows and columns as a single file. Used by Hadoop databases.
- **Key-Value Pair:** A key identifier and a value:
    - **Document Databases:** Each document has a key. Use XML or JSON (Java Script Object Notation) structures.
    - **Graph Databases:** Key-value pairs where the focus is on the relationship between nodes rather than the nodes themselves.

- **Triplestore:** A data entity composed of subject-predicate-object. Best for taxonomy and thesaurus management. Resource Description Framework (RDF):
    - **subject:** a resource
    - **predicate:** relationship between the subject and object
    - **object:** the object

1.3.9 Specialised Databases

- Computer Assisted Design and Manufacturing (CAD/CAM): Object database
- Geographical Information Systems (GIS): Geospatial databases
- Shopping-cart applications: XML databases to store customer order data

1.3.10 Common Database Practices

1.3.10.1 Archiving

The process of moving data off immediately accessible media onto less expensive media with lower retrieval performance. Schedule regular restoration tests.

Archival processes must be aligned with the partitioning strategy to ensure optimal availability and retention:

- Create a secondary area on a secondary database server
- Partition database tables into archival blocks
- Replicate to the separate database
- Create backups (tape or disk)
- Create jobs that periodically purge unneeded data

1.3.10.2 Capacity and growth projections

Decide how much storage is needed, and work out expansion needs

1.3.10.3 Change Data Capture (CDC)

Detecting that data has changed and ensure information relevant to the change is stored appropriately. Log-based replication.

1.3.10.4 Purging

Purging is the process of completely removing data from media so that is cannot be retrieved.

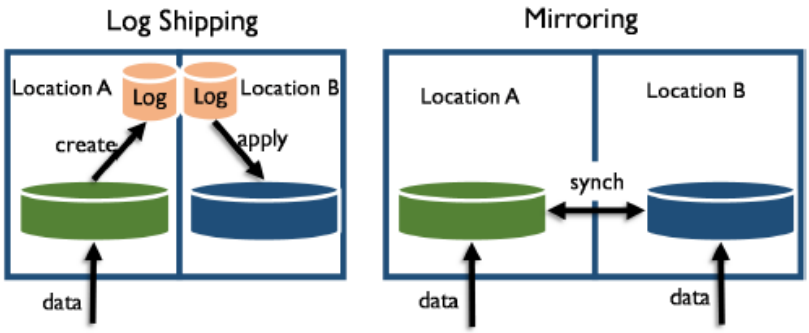
1.3.10.5 Replication

Replication transparency: The same data is stored on multiple storage devices and is consistent throughout the database system, so that users cannot tell which database copy they are using.

- **Active replication:** Create and store the same data at every replica from every other replica
- **Passive replication:** recreating and storing data on the primary, then transferring it to secondary replicas

Two primary replication patterns:

- **Mirroring:** Updates to the primary are replicated immediately to the secondary as part of a two-phase commit**.**
- **Log Shipping:** The secondary receives and applies copies of the primary database's transaction log at regular intervals



1.3.10.6 Resiliency and Recovery

Resiliency is a measure of how tolerant a system is to error conditions, and how well it recovers. Three recovery types:

- **Immediate recovery:** Predicting and automatically resolving issues
- **Critical recovery:** Plan to restore the system quickly to minimise delays to business processes
- **Non-critical recovery**: Restoration can be delayed until critical systems have been restored

1.3.10.7 Retention

How long data is kept available. Affects capacity planning. Data retention plans are also affected by Data Security, as some data has legal requirements to be retained a specific time.

1.3.10.8 Sharding

Small chunks of the database are isolated so replication is merely a file copy.

2 Activities

Data Technology Support (selecting and maintaining the software that stores and manages the data) and Data Operations Support (the data and processes that the software manages).

2.1 Manage Database Technology

The Information Technology Infrastructure Library (ITIL) is the leading reference model.

2.1.1 Understand Database Technology Characteristics

DBAs and Data Architects combine their knowledge of available tools with business requirements to suggest the best technology.

2.1.2 Evaluate Database Technology

Some of the factors to consider when selecting a DBMS:

- Product architecture and complexity
- Volume and velocity limits
- application profile such as transaction processing and BI
- Hardware platform and operating system support
- Availability of supporting software tools
- Performance benchmarks
- Scalability
- Software, memory and storage requirements
- Resiliency, including error handling and reporting

Price and the possible necessity to employ extra staff.

2.1.3 Manage and Monitor Database Technology

DBAs need to be trained to function as Level 2 technical support. They need to have a working knowledge of application development skills. DBAs work with business users and application developers to ensure the most effective use of technology.

2.2 Manage Databases

Database support is provided by DBAs and NSAs (Network Storage Administrators).

2.2.1 Understand Requirements

- **Define storage requirements:** Initial capacity estimate for first year and growth projection for next few years. Take data storage, indexes, logs and mirrors into account.
- **Identify usage patterns:** Databases have predictable usage patterns:
  - Transaction based
  - Large data set write or retrieval
  - Time based – month-end, lighter on weekends
  - Location based – more populated areas have more transactions
  - Priority based – some departments or batch IDs have higher priority
- **Define access requirements:** The authorisation to access different data files, and the standard languages and methods to do it.

2.2.2 Plan for Business Continuity

Recovery plan for all databases and database servers in the event of a disaster which could result in loss or corruption of data. Identify critical databases which need to be restored first.

This plan should be reviewed by the business continuity group.

- **Make backups:**
  - Back up databases and transaction logs
  - Backup frequency determined by SLA.
  - Incremental and complete backups
  - Keep backups on a separate file system
- **Recover data:** DBA executes restoration of data. Test recovery periodically

2.2.3 Develop Database Instances

- Installing and updating DBMS software
- Maintaining multiple environment installations, including different DBMS versions
- Installing and administering related data technology

2.2.3.1 Manage the Physical Storage Environment

Storage environment management needs to follow Software Configuration Management (SCM) processes or ITIL methods to record modification to the database configuration. Four processes:

- **Configuration Identification:** DBAs with Data Stewards, Data Architects and Data Modellers to identify attributes that define end-user configuration. These must be baselined, recorded and only changed with formal change control.
- **Configuration change control:** Processes and approval stages to change the above attributes
- **Configuration status accounting:** Report on the configuration at any point in time
- **Configuration audits:**
  - Physical configuration audit: an item is installed in accordance with design documentation

- Functional configuration audit: Performance attributes of an item are achieved

DBAs must communicate any changes to the physical attributes to modellers, developers and Metadata managers.

DBAs also maintain metrics on data volume, capacity projections, query performance and statistics on the physical objects.

2.2.3.2 Manage Database Access Controls

DBAs oversee the following functions to protect data assets:

- **Controlled Environment:** DBAs and NSAs. Network roles and permissions, 24/7 monitoring, firewall management, patch management
- **Physical security:** Simple Network Management Protocol (SNMP)-based monitoring, data audit logging, disaster management and database backup plans.
- **Monitoring:** Continuous monitoring of servers
- **Controls:** Access controls, database auditing, intrusion detection and vulnerability assessment tools

2.2.3.3 Create Storage Containers

All data must be stored on the physical drive and organised for ease of load, search and retrieval.

2.2.3.4 Implement Physical Data Models

DBAs implement the physical layout of the data model in storage. The physical data model includes storage objects, indexing objects, and any encapsulated code objects required to enforce quality rules, connect database objects and achieve database performance.

2.2.3.5 Load Data

DBMS should have a bulk load facility to load data into a new database. Data must be in the right format for the target object.

Third party data can be updated regularly by the service. DBAs must be aware of any legal restrictions before loading third party data

DBAs may load data manually, or automate and schedule loading.

Responsibility for data acquisition services in a managed environment is centralised with data analysts who document it in the logical data model. Developers create scripts if necessary. DBA implements the processes to load the data into the database.

2.2.3.6 Manage Data Replication

DBAs advise data replication decisions on:

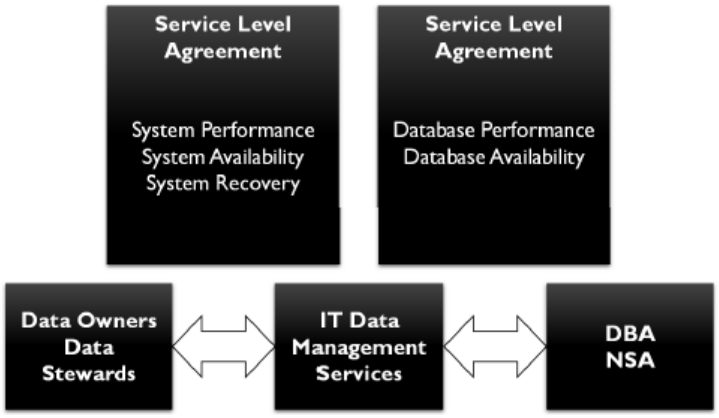
- Active or passive replicating
- Distributed concurrency control from distributed data systems
- The appropriate methods to identify updates to data under the Change Data Control process:
  - Timestamp
  - Version numbers

2.2.4 Manage Database Performance

Database performance depends on availability and speed. Availability of space and query optimisation are part of performance.

2.2.4.1 Set Database performance service levels

System performance, data availability and recovery expectations, and expectations for teams to respond are governed by Service Level Agreements (SLAs) between IT data management services and data owners.



2.2.4.2 Manage Database Availability

Availability is the percentage of time that a system or database is available for productive work. Four factors affect availability:

- **Manageability:** The ability to create and maintain an environment
- **Recoverability:** Establish service after interruption, and correct the errors caused
- **Reliability:** Ability to deliver service at specified levels for a stated period
- **Serviceability:** Identify and diagnose problems and solve them

Some things that prevent databases from being available:

- Planned outages
- Unplanned outages
- Application problems
- Data problems
- Human error

DBAs are responsible for doing everything possible to ensure databases stay online and operational:

- Run database backup utilities
- Run database reorganisation utilities
- Run statistics gathering utilities
- Run integrity gathering utilities
- Automating the above utilities
- Exploiting table clustering and partitioning
- Replicating data across mirror databases to ensure high availability

2.2.4.3 Manage Database Execution

DBAs manage database execution, logging and log sizes and synchronisation.

2.2.4.4 Maintain database performance service levels

DBAs generate performance analysis reports regularly and compare them with previous reports to identify negative trends and analyse problems over time.

- **Transaction performance vs Batch performance**: Batch jobs must complete within a batch window
- **Issue remediation:** Common reasons for poor database performance are:
    - Memory allocation or contention:
    - Locking and blocking
    - Inaccurate database statistics
    - Poor coding
    - Inefficient complex table joins
    - Insufficient indexing
    - Application activity
    - Overloaded servers
    - Database volatility
    - Runaway queries

2.2.4.5 Maintain alternate environments

Types of alternate environments:

- **Development:** Test changes that will be implemented in production
- **Test:** QA, Integration testing, UAT and performance testing
- **Sandboxes:** Experimental environments
- **Alternate production environments:** support failover, offline backups and resiliency support systems

2.2.5 Manage Test Data Sets

Efficient testing requires high quality test data to be generated and managed

2.2.6 Manage data Migration

Data Migration is the process of transferring data between storage types, formats or computer systems with as little change as possible. Usually performed programmatically, automated based on rules.

3 Tools

- Data modelling tools
- Database monitoring tools
- Database Management tools
- Developer support tools

4 Techniques

4.1 Test in lower environments

Test upgrades and patches on the lowest level first, development. Then install and test on higher levels, production last.

4.2 Physical naming standards

ISO/IEC 11179 – Metadata Registries (MDR)

4.3 Script usage for all changes

Test any change scripts in non-production before applying.

5 Implementation Guidelines

5.1 Readiness assessment/Risk assessment

Two central ideas:

- **Data Loss**: SLAs specify general requirements for protection. Ongoing assessment to ensure robust technical responses to cyber threat are in place.
- **Technology readiness**: Does the organisation have the skill set to implement newer technology.

5.2 Organisation and Cultural Change

- **Proactively communicate**: DBAs should be in close communication with project teams at all stages of the project, to detect and resolve issues as early as possible
- **Communicate with people on their level and in their terms**
- **Stay business focussed:** objective is to meet business requirements and derive maximum value
- **Be helpful:** Not helping may force people to ignore standards and find another way
- **Learn continually:** Setbacks and problems are lessons which can be applied later.

Understand stakeholders and their needs. Develop clear, concise, practical, business focussed standards for doing the best possible work in the best possible way. Teach and implement those standards in a way that provides maximum value to stakeholders and earns their respect.

6 Data Storage and Operations Governance

6.1 Metrics

Storage metrics may include:

- Count of databases by type
- aggregated transaction stats
- Capacity metrics
- Storage service usage
- requests made against storage services
- Performance improvements of applications using service

Performance metrics:

- Transaction frequency and quantity
- Query performance
- API service performance

Operational metrics:

- Aggregated statistics about data's retrieval times
- Backup size
- Data quality measurement
- Availability

Service metrics:

- Issue submission, resolution and escalation count by type
- Issue resolution time

6.2 Information asset Tracking

Ensure organisation complies with software licensing and annual support agreements.

Determine TCO (total cost to ownership) for each technology product.

6.3 Data Audits and Data validation

A data audit is the evaluation of data based on defined criteria, typically performed to investigate specific concerns about the data. A data audit includes:

- Project specific checklist
- comprehensive checklist
- Required deliverables
- Quality control criteria

Data validation is the process of evaluating stored data against established acceptance criteria (from the Data Quality team or customer specifications) to determine its quality and usability.

DBAs support data audits and validation by:

- Help develop and review the approach
- Perform preliminary data screening and review
- Develop data monitoring methods
- Applying statistical, geo-statistical and bio-statistical techniques to optimise the data
- Support sampling and analysis
- Review data
- Provide support for data discovery
- Act as the SME for questions related to database administration