Data Storage Operations Handbook

> **Source File:** generated-documents\dmbok\data-storage-operations-handbook.md
> **Generated:** 30/07/2025 at 06:57:50
> **Generated by:** Requirements Gathering Agent - PDF Converter

Data Storage & Operations Handbook

Generated by adpa-enterprise-framework-automation v3.2.0
Category: dmbok
Generated: 2025-07-17T09:08:13.032Z
Description: Comprehensive guide for database administration, storage management, and data operations following DMBOK standards.

Data Storage & Operations Handbook

For ADPA - Advanced Document Processing & Automation Framework
Aligned with DAMA DMBOK 2.0 Guidelines

---

1. Introduction

1.1 Purpose

This handbook establishes standards, procedures, and best practices for data storage, database administration, and operational management in the ADPA enterprise automation framework. It ensures secure, efficient, and reliable data handling, aligning with DMBOK 2.0 and enterprise requirements for compliance, scalability, and maintainability.

1.2 Scope

This document applies to all systems and teams managing data for ADPA, including:

- Database administrators (DBAs)
- DevOps and operations engineers
- Developers interfacing with storage or data migration
- Compliance and security teams

Covered topics include:

- Database design, implementation, and lifecycle management
- Storage infrastructure (on-premises, cloud, hybrid)
- Operational procedures (monitoring, backup, recovery, maintenance)
- Security, compliance, and access control
- Documentation and continuous improvement

---

2. Database Architecture

2.1 Database Management Systems

ADPA Default Configuration:

- **Primary Data Store:**

    - **Type:** JSON-based configuration files (default for metadata, templates, and settings)
    - **Location:** `/src/documentTemplates/`, `/src/modules/`, `/docs/`
- **Extensibility:**
    - Supports integration with relational (SQL) or NoSQL databases for enterprise deployments
    - **Recommended DBMS for Enterprise:**
        - **Relational:** PostgreSQL, MySQL, Microsoft SQL Server
        - **NoSQL:** MongoDB (for unstructured data, logs, analytics)
- **Data Warehouse (Optional):**
    - For analytics and reporting, integration with cloud warehouses (Azure Synapse, Snowflake, BigQuery)

Best Practice:
Use a modular approach—begin with the default file-based setup for development, and transition to enterprise DBMS for production.

2.2 Database Design Standards

- **Naming Conventions:**
    - Use clear, descriptive names for tables/collections, columns/fields, and indexes (e.g., `user_accounts`, `document_templates`)
    - Follow snake_case or camelCase consistently
- **Schema Design Principles:**
    - Normalize to 3NF where appropriate (relational)
    - For NoSQL, design collections around access patterns and denormalize as needed for performance
- **Indexing Strategy:**
    - Index primary keys and frequently queried fields
    - Monitor index usage and adjust based on query patterns
- **Partitioning Approach:**

    - For large datasets, use table partitioning (date, tenant, or logical domain)
    - In NoSQL, use sharding as appropriate

- **Data Modeling:**
    - Maintain ERD (Entity Relationship Diagrams) for relational schemas
    - Document data models and mappings for NoSQL

---

3. Storage Management

3.1 Storage Infrastructure

- **Local Development:**
    - Use project directory structure (`/docs/` , `/generated-documents/`)
- **Enterprise/Production:**
    - **SAN/NAS:** For on-premises, use Storage Area Networks (SAN) or Network-Attached Storage (NAS) for high availability and redundancy
    - **Cloud Storage:**
        - Use Azure Blob Storage, AWS S3, or Google Cloud Storage for scalable document archives and backups
        - Separate storage accounts/buckets for production, dev, test
    - **Storage Tiering:**
        - Define tiers (hot, warm, cold) for performance/cost optimization. E.g., recent documents on SSD, archives on object storage.

3.2 Capacity Planning

- **Growth Projections:**

    - Estimate data growth (GB/month) based on document volumes, logs, and user base
    - Regularly review storage utilization dashboards

- **Performance Requirements:**
    - Set IOPS/throughput targets per workload (e.g., API response < 200ms for metadata queries)
    - Use SSD for high-transaction workloads

- **Scalability:**
    - Plan for horizontal scaling (sharding, partitioning)
    - Use cloud auto-scaling where available

- **Cost Optimization:**
    - Archive infrequently accessed data to cold storage
    - Purge obsolete data per retention policy

---

4. Database Administration

4.1 Installation and Configuration

- **Standard Procedures:**
    - Follow vendor best practices for DBMS installation (PostgreSQL, MongoDB, etc.)
    - Use Infrastructure-as-Code (IaC) for repeatable deployments (Terraform, ARM templates, CloudFormation)

- **Configuration Parameters:**
    - Set memory, connection pool, and cache settings according to workload
    - Enable slow query logging

- **Security Settings:**

    - Disable default/public accounts
    - Enforce encrypted connections (SSL/TLS)
    - Regularly update DBMS to latest stable version

- **High Availability (HA) Setup:**
    - Use clustering (e.g., PostgreSQL replication, MongoDB replica sets)
    - Set up automated failover and health checks

4.2 Performance Monitoring

- **Key Performance Indicators (KPIs):**
    - Query latency
    - Transaction throughput
    - Cache hit ratio
    - Replication lag
    - Disk usage

- **Monitoring Tools:**
    - Open-source: Prometheus, Grafana, pgAdmin, MongoDB Compass
    - Cloud-native: Azure Monitor, AWS CloudWatch, Google Stackdriver

- **Alert Thresholds:**
    - Set alerts for CPU > 80%, disk > 80%, replication lag, failed backups, etc.

- **Performance Tuning:**
    - Regularly analyze slow queries and add/adjust indexes
    - Vacuum/analyze tables (relational)
    - Rebalance shards/replica sets as needed

---

5. Data Operations

5.1 Backup and Recovery

- **Backup Schedules:**
  - Daily full backups, hourly incremental/differential (for mission-critical data)
  - Store backups on separate physical/cloud infrastructure
- **Retention Policies:**
  - Retain daily/weekly/monthly snapshots per compliance needs (e.g., 7 days, 4 weeks, 12 months)
- **RPO/RTO:**
  - Define Recovery Point Objective (e.g., max 1h data loss)
  - Define Recovery Time Objective (e.g., restore in <2h)
- **Disaster Recovery Procedures:**
  - Document step-by-step restoration; regularly test restores (quarterly DR drills)
  - Store backup/restore scripts in version control

5.2 Data Migration

- **Strategy:**
  - Use ETL (Extract, Transform, Load) pipelines for bulk migrations
  - Validate data types, relationships, and integrity during migration
- **ETL Tools:**
  - Node.js scripts, Azure Data Factory, AWS Glue, custom TypeScript utilities
- **Validation:**

    - Run checksums, record counts, and spot data validation post-migration

- **Rollback:**
    - Always snapshot source data before migration
    - Prepare rollback scripts/procedures for major migrations

---

6. Security and Compliance

6.1 Access Control

- **Authentication:**
    - Use enterprise identity providers (Active Directory, SAML, OAuth2)
    - Enforce strong password and MFA policies
- **RBAC:**
    - Define granular roles (DBA, Developer, ReadOnly, AppUser, etc.)
    - Grant least privilege required; review permissions quarterly
- **Privilege Management:**
    - Use role-based user groups for access assignment
    - Revoke access promptly on role change/termination
- **Audit Logging:**
    - Enable query/user activity logging
    - Store logs securely, monitor for anomalies

6.2 Data Protection

- **Encryption:**

    - Encrypt data at rest (AES-256 for files/db files, cloud-native encryption)
    - Enforce TLS for data in transit

- **Data Masking & Tokenization:**
    - Mask sensitive fields (PII, financial data) in non-prod/test environments
    - Use tokenization for API responses where applicable

- **Key Management:**
    - Use managed key vaults (Azure Key Vault, AWS KMS, GCP KMS)
    - Rotate keys per policy (minimum annually)

- **Regulatory Compliance:**
    - Implement GDPR/SOX/PCI/other required controls
    - Support data subject access requests (DSAR), audit logs, and right-to-be-forgotten

---

7. Maintenance and Support

7.1 Routine Maintenance

- **Health Checks:**
    - Automate daily health checks for all DB systems and storage endpoints

- **Statistics Updates:**
    - Regularly update query planner statistics (PostgreSQL: `ANALYZE`, SQL Server: `UPDATE STATISTICS`)

- **Index Maintenance:**
    - Rebuild/reorganize indexes on schedule (off-peak hours)

- **Storage Optimization:**
    - Monitor disk usage and growth, expand volumes/tune storage as needed

7.2 Incident Management

- **Problem Resolution:**
    - Document and use a standard incident response runbook
    - Collect logs, symptoms, and steps taken for each incident
- **Escalation:**
    - Define on-call rotations and escalation paths for critical incidents
- **Root Cause Analysis:**
    - Conduct RCAs for all major production incidents; document and share findings
- **Preventive Measures:**
    - Implement fixes, automate checks, and update documentation after each incident

---

8. Documentation and Training

8.1 Operational Documentation

- **Runbooks:**
    - Step-by-step guides for common DBA/operations tasks (backups, restores, user management, migrations)
- **Knowledge Base:**

    - Centralized repository for troubleshooting guides, error messages, and solutions

- **Change Logs:**
    - Maintain detailed logs for schema/data changes
- **Best Practices:**
    - Document coding, query, and operational standards

8.2 Training Requirements

- **DBA Training:**
    - Require DBAs to complete training for supported DBMS/cloud platforms
- **User Training:**
    - Provide onboarding for developers and power users on query best practices and data access
- **Certification:**
    - Encourage certification in cloud or database technologies relevant to the deployment (e.g., Azure Database Admin, AWS Solutions Architect)
- **Knowledge Transfer:**
    - Formalize KT sessions for handoffs, onboarding, or after major upgrades

---

9. Continuous Improvement

9.1 Performance Metrics

- **System Trends:**

    - Track response times, error rates, throughput over time
    - Review incidents and outages for patterns

- **User Feedback:**
    - Solicit and review feedback from end-users and developers

- **Benchmarking:**
    - Regularly benchmark storage and DB performance (quarterly or after major upgrades)

9.2 Process Optimization

- **Automation:**
    - Automate repetitive tasks (backups, monitoring, health checks) with scripts or orchestration tools

- **Process Improvement:**
    - Conduct post-mortems and retrospectives after incidents or releases

- **Technology Upgrades:**
    - Stay current with DBMS, storage, and security patches; schedule upgrades proactively

- **Industry Best Practices:**
    - Review and adopt emerging best practices from DMBOK, cloud providers, and database vendors

---

Appendix: ADPA-Specific Considerations

- **JSON-Based Configuration:**

    - For development and small-scale use, maintain configuration and template data in version-controlled JSON files.
    - For enterprise, migrate to SQL/NoSQL for scalability, auditability, and performance.

- **API Audit Trails:**
    - All API actions (generation, upload, publish) should be logged for compliance and traceability.
- **Integration with External Systems:**
    - Use secure connectors and managed identities for Confluence, SharePoint, Adobe, and cloud storage integration.
- **Testing & Validation:**
    - Leverage automated test suites (`npm test` , integration scripts) to validate data integrity and operational readiness of the storage layer.

References

- DAMA DMBOK 2.0
- PMBOK 7th Edition
- BABOK v3
- ADPA Documentation

This handbook is a living document. Review and update at least every 6 months, or after major platform releases or incidents.
