DOMAIN 2 DATA STORAGE & DATA MANAGEMENT

AWS-Certified-Data-Engineer-Associate_Exam-Study-Guide Content outline

- This exam guide includes weightings, content domains, and task statements for the exam. This guide does not provide a comprehensive list of the content on the exam. However, additional context for each task statement is available to help you prepare for the exam.
- The exam has the following content domains and weightings:
  - Domain 1: Data Ingestion and Transformation (34% of scored content)
  - Domain 2: Data Store Management (26% of scored content)
  - Domain 3: Data Operations and Support (22% of scored content)
  - Domain 4: Data Security and Governance (18% of scored content)

DOMAIN 2: DATA STORAGE & DATA MANAGEMENT

- 📊 Weight: 26% 📘 Core Competency: Choosing and managing storage solutions, partitioning and formatting datasets, managing metadata catalogs, and applying lifecycle, access, and encryption policies.

🔹 2.1 Storage Solutions on AWS

- 🟦 Amazon S3 (Simple Storage Service)
  - Durability: 99.999999999% (11 9s)
  - Storage Classes:
    - Standard (frequent access)
    - Intelligent-Tiering (automated cost optimization)
    - Glacier & Deep Archive (cold storage)
  - Encryption:
    - SSE-S3 (managed keys)
    - SSE-KMS (customer-managed keys)
    - CSE (client-side)
  - Performance Tips:
    - Use multipart uploads for files >100MB
    - Leverage transfer acceleration for cross-region ingestion
- 🟩 Amazon Redshift
  - Columnar storage optimized for analytics
  - Redshift Spectrum:
    - Query S3 data without loading into Redshift
    - Uses Glue Data Catalog for schema
  - Distribution Styles: AUTO, EVEN, KEY, ALL
  - Sort Keys: Optimize query performance
  - Compression: Columnar compression + column pruning
- 🟧 Amazon DynamoDB
  - For storing metadata or small key-value datasets
  - Used alongside data lakes for lookup tables

🔹 2.2 Data Partitioning & Layout Optimization

- 📁 Partitioning in S3
  - Split data into folders using key=value format: Example: s3://bucket/events/year=2024/month=04/day=01/
  - Tools: Glue Crawler auto-detects partitions
  - Best Practices:
    - Don't over-partition (can cause too many small files)
    - Balance partitioning depth vs. query filtering power
- 📐 File Formats

Format	Type	Strengths	Use Cases
Parquet	Columnar	Efficient analytics, compression	Athena, Redshift Spectrum
ORC	Columnar	Hive optimized, high compression	EMR, Presto
Avro	Row	Schema evolution support	Kafka, Glue Streaming
JSON	Row	Human-readable, flexible	Logging
CSV	Row	Simple, but inefficient	Legacy data loads

- Use columnar formats for analytical workloads
- Compress files (Snappy, GZIP) for performance and cost

🔹 2.3 AWS Glue Data Catalog & Metadata

- AWS Glue Data Catalog
  - Central metadata repository for datasets
    - Integrated with:
      - Athena
      - Redshift Spectrum
      - EMR, Glue, DataBrew
  - Glue Crawlers: Automatically detect schema and partitions
  - Schema Versions: Track schema evolution for formats like Avro
- 🗂️ Schema Management
  - Store schema with:
    - Avro files
    - Glue Catalog tables
  - Schema Compatibility:
    - Forward-compatible: new optional fields added
    - Backward-compatible: fields removed or made optional
  - Best Practice: Maintain schema definitions in Glue and document changes

🔹 2.4 Lake Formation Security and Access Control

- 🔐 AWS Lake Formation
  - Data lake security and fine-grained permissions on S3 data
  - Integrates with:
    - Glue Catalog
    - Redshift, Athena, EMR
  - Access Controls: Table-level, column-level, and row-level permissions
  - Tag-based Access Control (LF-TBAC): Dynamic access using data classification tags
- 🛡️ Security & Encryption
  - KMS: AWS Key Management Service for encrypting S3, Redshift, and Glue
  - S3 Bucket Policies: Control access with IAM or resource policies
  - CloudTrail: Auditing data access events

🔹 2.5 Lifecycle Management and Storage Optimization

- ⏳ S3 Lifecycle Policies
  - Transition rules (e.g., Standard → Glacier after 30 days)
  - Expiration rules to delete old versions or temp data
  - Storage Class Analysis: Recommend transitions
- 💸 Cost Optimization
  - Tools:
    - S3 Storage Class Analysis
    - Redshift Advisor (table size, compression, skew)
    - Cost Explorer
  - Tactics:
    - Use compressed formats
    - Convert CSV → Parquet before analytics
    - Delete temp/intermediate files after ETL

📝 Sample Use Case Scenarios

- Store petabyte-scale clickstream data:Use S3 (Parquet format, partitioned by date), Glue Catalog, Athena for querying.
- Secure sensitive financial data in a data lake:S3 + Lake Formation + KMS encryption + fine-grained permissions.
- Query semi-structured data from S3 without loading into Redshift:Use Redshift Spectrum with Glue Catalog.
- Manage evolving schema for streaming IoT data:Use Avro + schema stored in Glue + crawler updates.

✅ Best Practices Summary

- Use Parquet/ORC + partitioning + compression for S3 data
- Store metadata centrally in Glue Data Catalog
- Use Lake Formation for secure, fine-grained access
- Apply S3 Lifecycle Policies to optimize storage cost
- For queries across S3, use Athena or Redshift Spectrum
- Separate raw, processed, and curated data layers in your lake

📚 Official Resources

- What is Amazon S3?
- AWS Glue components
- Security in AWS Lake Formation
- Introduction to Amazon Redshift
- Top 10 Performance Tuning Tips for Amazon Athena