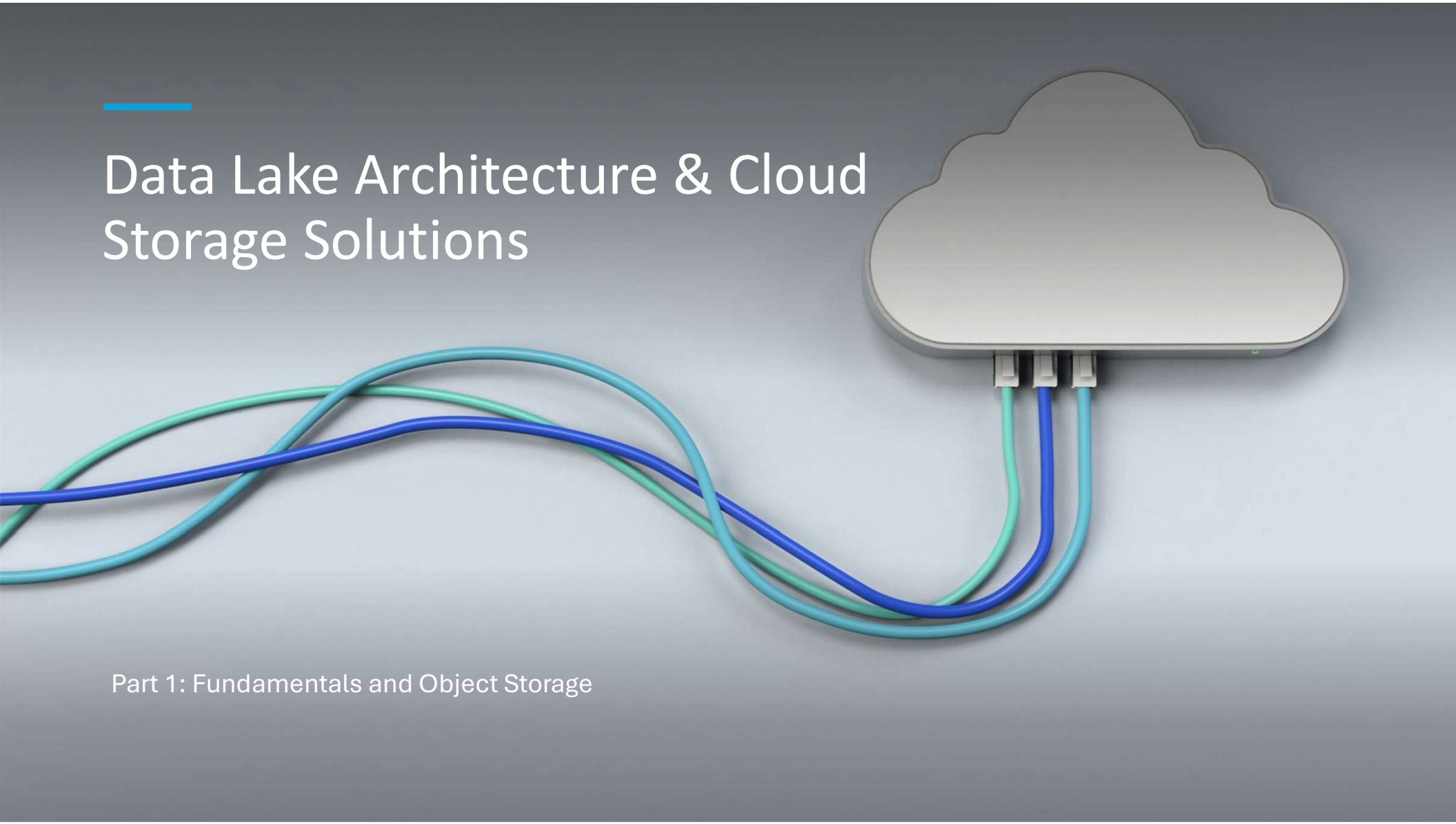
Data Lake Architecture & Cloud Storage Solutions



Part 1: Fundamentals and Object Storage

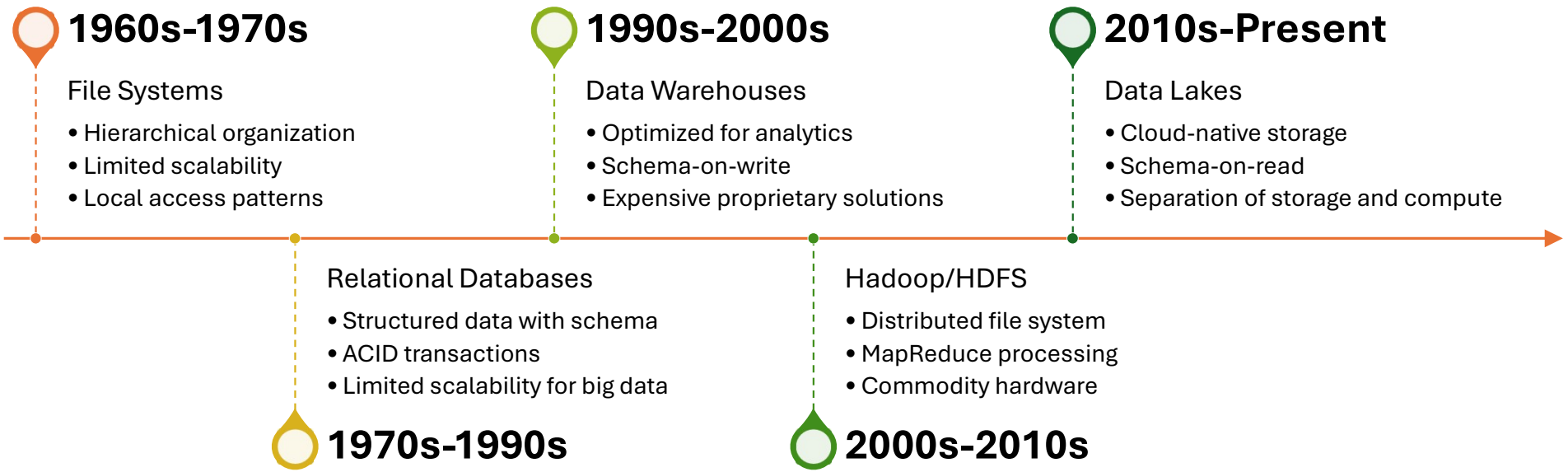
Introduction to Data Lakes

What is a Data Lake?

- A central repository that stores structured, semi-structured, and unstructured data at scale
- Data stored in its raw, native format
- Uses flat architecture and object storage
- Schema-on-read approach (vs. schema-on-write in traditional databases)
- Enables diverse analytics workloads (BI, ML, data science)

Formal Definition: A data lake is a centralized repository designed to store, process, and secure large amounts of structured, semi-structured, and unstructured data.

Evolution of Data Storage Systems



Why Data Lakes

Key Drivers

- **Volume**: Exponential growth in data generation
- **Variety**: Increasing diversity of data types
- **Velocity**: Real-time data processing needs
- **Cost Efficiency**: Lower storage costs compared to data warehouses
- **Flexibility**: Support for diverse workloads
- **Democratization**: Broader access to data across organizations

Business Value

- Single source of truth for enterprise data
- Support for advanced analytics and machine learning
- Reduced data silos and improved data governance
- Cost-effective storage for historical data

Data Lake vs. Data Warehouse

Feature	Data Lake	Data Warehouse
Data type	Raw, unstructured, semi-structured, structured	Processed, structured
Schema	Schema-on-read	Schema-on-write
Users	Data scientists, analysts, engineers	Business analysts, executives
Use cases	Machine learning, exploratory analysis	BI reporting, dashboards
Storage cost	Lower	Higher
Query performance	Variable (may require processing)	Optimized for fast queries
Data quality	Variable, may contain "noise"	Cleansed and validated
Agility	High flexibility for new use cases	Less flexible, predefined models

Core Components of Data Lake Architecture

Logical Architecture

Ingestion Layer

Batch and streaming data intake

Connectors to various data sources

Data validation and initial processing

Storage Layer

Object storage (primary focus today)

Metadata management

Data organization (zones, partitioning)

Processing Layer

Batch processing engines

Stream processing engines

Query engines

Consumption Layer

Analytics tools

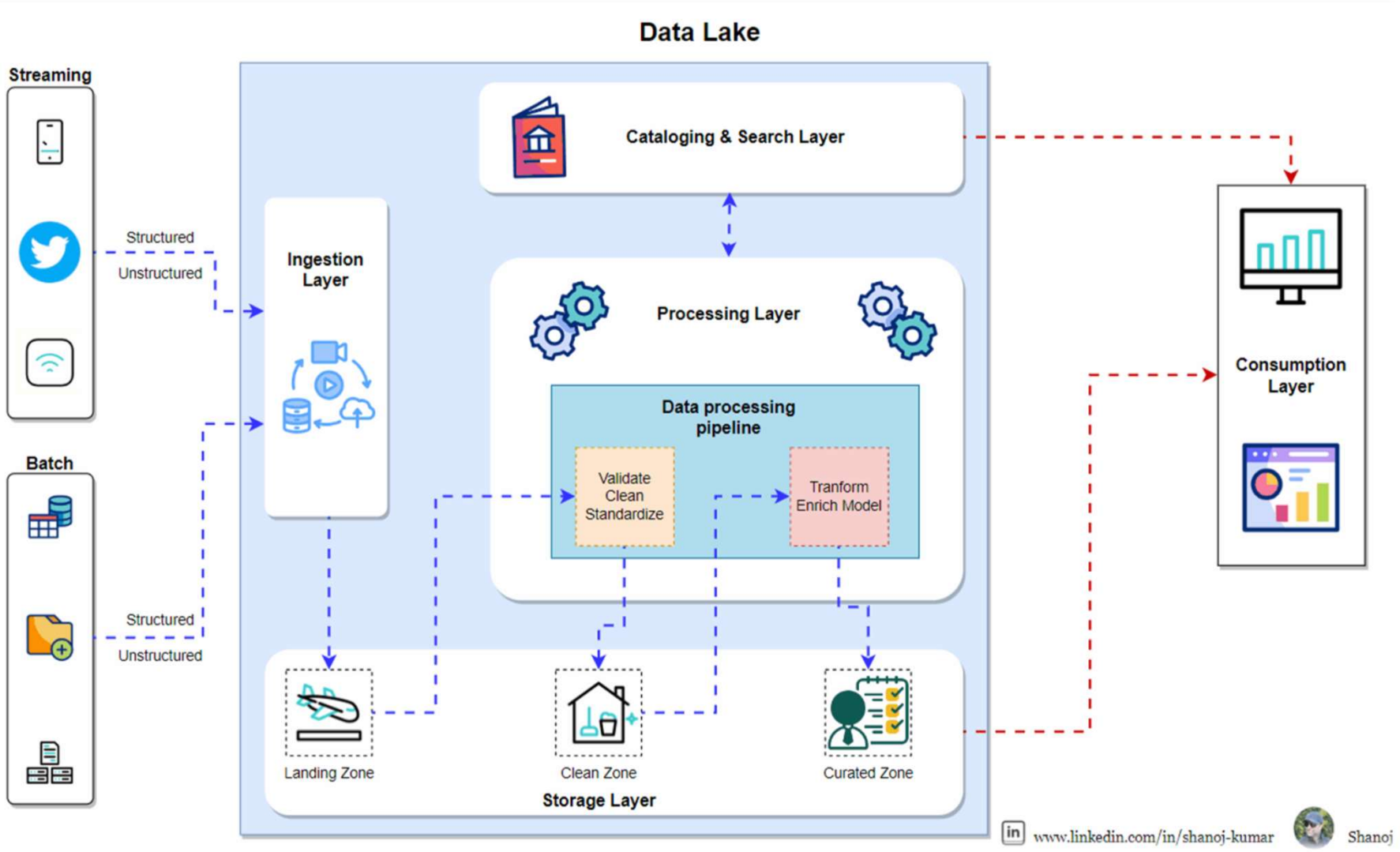
Visualization platforms

Machine learning frameworks

Data Organization in Data Lakes

Zone-Based Architecture

- **Landing/Raw Zone**
  - Original, unaltered data
  - Immutable storage
  - Full historical record
- **Cleansed/Standardized Zone**
  1. Validated data
  2. Standardized formats
  3. Enriched with metadata
- **Curated/Refined Zone**
  1. Transformed for specific use cases
  2. Aggregated and joined data
  3. Optimized for performance
- **Consumption Zone**
  1. Purpose-built datasets
  2. Optimized for specific tools
  3. Often includes data marts



Source: https://aws.plainenglish.io/data-lake-101-architecture-5da905c2256c

Object Storage Fundamentals

What is Object Storage?

- Storage architecture that manages data as objects (vs. files or blocks)
- Each object includes:
  - Data content (the actual data)
  - Metadata (information about the data)
  - Unique identifier (for retrieval)
- Flat address space (no hierarchical structure)
- Highly scalable and durable
- Accessed via HTTP/REST APIs

Key Characteristics

- Immutable objects (create/delete, not update)
- Eventual consistency model
- Unlimited scalability
- Built-in redundancy
- Cost-effective for large datasets

Object Storage vs. Traditional File Systems

Characteristic	Object Storage	Traditional File Systems
Structure	Flat namespace	Hierarchical directories
Scalability	Virtually unlimited	Limited by architecture
Metadata	Rich, customizable	Limited, predefined
Access	HTTP/REST APIs	File system protocols
Consistency	Often eventual	Strong consistency
Mutability	Typically immutable	Mutable
Use cases	Big data, backups, content	Local applications, OS
Cost model	Pay-as-you-go	Capital expenditure

Tiered Storage Approaches

What is Tiered Storage?

- Methodology for categorizing data based on access patterns
- Automatically moves data between storage tiers
- Optimizes for both performance and cost
- Implemented through lifecycle policies

Common Tiers

- **Hot Tier**: Frequently accessed data, optimized for performance
- **Warm Tier**: Occasionally accessed data, balanced performance/cost
- **Cold Tier**: Rarely accessed data, optimized for cost
- **Archive Tier**: Long-term retention, highest latency, lowest cost

Participation Question

Instructions (10 minutes):

For each of the four data artifacts listed below, fill in the table with:

1. **Data Lake Zone** (Raw / Cleansed / Curated / Consumption)
2. **Object Storage Class** (e.g. Hot / Standard, Warm / Intelligent-Tiering, Cold / Glacier, Archive / Deep Archive)
3. **One-Sentence Justification** for your choices

Be ready to share your answers when time is up.

Participation Slides

Artifact	Zone	Storage Class	Justification
A. Website clickstream logs (as JSON files)			
B. End-of-day sales CSVs			
C. Daily aggregated customer data			
D. 2019–2021 historical archive of old files			

Cloud Storage Solutions: Amazon S3

Amazon Simple Storage Service (S3)

- Industry-leading object storage service
- Launched in 2006, pioneered cloud object storage
- 99.99999999% (11 9's) durability
- Virtually unlimited storage capacity

Key Features

- **Storage Classes**: Standard, Intelligent-Tiering, Standard-IA, One Zone-IA, Felxible Glacier, Deep Archive
- **Lifecycle Management**: Automatic transition between storage classes
- **Versioning**: Preserve, retrieve, and restore every version
- **Access Control**: IAM policies, bucket policies, ACLs
- **Encryption**: Server-side and client-side options
- **Event Notifications**: Trigger workflows based on object changes
- **S3 Select**: SQL-like queries on objects

Cloud Storage Solutions

Google Cloud Storage (GCS)

- Google's object storage service
- Global edge network for low-latency access
- 99.999999999% (11 9's) durability
- Integrated with Google's analytics services

Key Features

- **Storage Classes**: Standard, Nearline, Coldline, Archive
- **Object Lifecycle Management**: Automatic class transitions
- **Strong Consistency**: All operations are strongly consistent
- **Uniform Access Control**: IAM permissions model
- **Customer-Managed Encryption Keys**: Control your own keys
- **Object Versioning**: Maintain history of objects
- **Object Holds and Retention Policies**: Compliance features

Cloud Storage Solutions

Azure Blob Storage

- Microsoft's object storage solution
- Integrated with Azure ecosystem
- 99.999999999% (11 9's) durability
- Hierarchical namespace option (Azure Data Lake Storage Gen2)

Key Features

- **Access Tiers**: Hot, Cool, Archive
- **Lifecycle Management**: Automatic tier transitions
- **Blob Types**: Block blobs, page blobs, append blobs
- **Data Lake Storage Gen2**: Hierarchical namespace
- **Immutable Storage**: WORM (Write Once, Read Many) policies
- **Soft Delete**: Recover accidentally deleted data
- **Static Website Hosting**: Directly serve web content

Data Lake Challenges

Common Issues

Data Swamps: Lack of governance and metadata management

Performance: Direct querying of object storage can be slow

Security and Access Control: Complex in multi-tenant environments

Data Quality: No enforced schema leads to quality issues

Cost Management: Unexpected costs from data access patterns

Integration Complexity: Connecting various tools and services

Emerging Solutions

Delta Lake: ACID transactions on cloud storage

Data Catalogs: Automated metadata management

Query Acceleration: Optimized formats and indexing

Data Governance Tools: Automated policy enforcement

Summary

- Data lakes provide flexible, scalable storage for all data types
- Object storage forms the foundation of modern data lakes
- Cloud providers offer robust object storage solutions with tiered storage options
- Proper organization and governance are critical for success
- Understanding the differences between data lakes and data warehouses helps in designing appropriate solutions