Day 6 Study Notes: Data Lake Concepts & Modern Storage

Introduction: Modern Data Storage Concepts

Modern data storage has evolved to support the growing volume, variety, and velocity of data generated by organizations. Solutions like data lakes, warehouses, and data marts are designed to address different analytical and operational needs. Understanding these concepts is essential for building scalable, flexible, and governed data platforms.

Data Lake vs Data Warehouse vs Data Mart

Definitions

- Data Lake: A centralized repository that stores raw, structured, semi-structured, and unstructured data at any scale, typically in its native format.
- Data Warehouse: A system optimized for structured, processed data, supporting complex queries and analytics, often with predefined schemas.
- Data Mart: A subset of a data warehouse, focused on a specific business line or team, containing relevant, curated data for targeted analysis.

Key Differences

Aspect	Data Lake	Data Warehouse	Data Mart
Data Type	All (raw, structured, semi-structured, unstructured)	Structured (processed, cleaned)	Structured (subject-specific)
Schema	Schema-on-read	Schema-on-write	Schema-on-write
Purpose	Store everything for future analysis	Business intelligence, reporting	Departmental analytics
Cost	Low (commodity storage)	Higher (optimized storage/computing)	Medium (smaller scale)
Users	Data engineers, scientists	Business analysts, decision makers	Line-of-business analysts

Use Cases

- Data Lake: Machine learning, big data analytics, data archiving.

- Data Warehouse: Financial reporting, operational dashboards, regulatory reporting.
- Data Mart: Marketing campaign analysis, sales performance tracking.

Data Lake Architecture Patterns

- Raw Zone (Landing): Ingests data as-is from multiple sources. Minimal processing; used for retention and traceability.
- Processed Zone (Validated): Data is cleansed, structured, and validated. Errors and duplicates are handled here.
- Curated Zone: Data is enriched, aggregated, and optimized for downstream analytics and consumption.

AWS Lake Formation Basics

- Purpose: Simplifies building secure data lakes on AWS, automating ingestion, cleaning, cataloging, and securing data.
- Features: Centralized access control, automated data import, schema and metadata management, data transformation tools.
- Benefits: Reduced setup time, improved security, integration with AWS analytics and machine learning services.

Metadata Management and Cataloging

Metadata describes the structure, origin, and usage of data. Effective management ensures discoverability, governance, and efficient data usage.

- Importance: Enables search, lineage tracking, access control, and compliance.
- Common Tools: AWS Glue Data Catalog, Apache Atlas, Amundsen.

Master Data Layer in Data Lakes

- Role: Stores cleansed, deduplicated, and authoritative data entities (e.g., customers, products).
- Separation: Kept distinct from raw/curated zones to ensure data consistency and integrity across the organization.

File Format Comparison

Feature	Avro	Parquet	ORC
Storage Layout	**Row-based**	**Columnar**	**Columnar**
Best Use Case	Real-time Streaming (Kafka)	Analytical Queries (OLAP)	Big Data (Hive/Presto)
Write Speed	Very Fast	Slower	Slower
Read Speed	Slower (reads whole row)	Very Fast (selective columns)	Very Fast (best for ACID)
Schema Evolution	Excellent (Schema is in file)	Good	Moderate
Compression	Good	Excellent	Excellent (Best)

Delta Lake: Key Features

- ACID Transactions: Ensures reliability and consistency for concurrent data operations.
- Time Travel: Query previous versions of data for auditing or rollback purposes.
- Schema Evolution: Supports changes to data structure without breaking pipelines.
- Data Reliability: Handles data corruption, duplicates, and updates efficiently.

Apache Iceberg & Hudi: Overview and Use Cases

- Apache Iceberg: Open table format for huge analytic datasets, supports schema evolution, partitioning, hidden partitioning, and high-performance reads/writes. Used with Spark, Trino, Flink, and more.
- Apache Hudi: Manages large analytical datasets with support for upserts, incremental processing, and efficient data versioning. Good for streaming and batch data pipelines.

Lakehouse Architecture

Lakehouse architecture combines the scalability and flexibility of data lakes with the reliability and performance of data warehouses. It allows for unified data storage, processing, and analytics on a single platform, supporting both structured and unstructured data.

Governance Zones in Data Lakes

- Landing/Raw Zone: Ingested data; minimal governance, broad access for ingestion tools.
- Validated/Processed Zone: Cleaned and validated; moderate governance, access for data engineers.
- Curated Zone: Enriched and refined; strict governance, access for analysts and business users.
- Master Zone: Authoritative, high-quality data; strongest governance, limited access.
- Archive Zone: Historical data; minimal access, retention policies enforced.

Access Patterns by Zone

- Raw Zone: Bulk ingestion, limited querying, access for data engineers only.
- Processed Zone: Data transformation, validation, access for ETL processes.
- Curated Zone: BI tools, dashboards, wide access for analytics.
- Master Zone: Reference data for critical processes, tightly controlled access.
- Archive Zone: Occasional retrieval for audits or compliance.

Data Lineage: Importance

- Governance: Tracks how data flows and transforms across zones, supporting compliance and audits.
- Trust: Enables users to understand and trust data origins and transformations.
- Troubleshooting: Helps identify issues or errors in data pipelines by tracing data history.

Conclusion: Key Takeaways and Best Practices

- Choose the right storage paradigm (lake, warehouse, mart) based on data type, use case, and user needs.
- Implement clear governance zones with appropriate access patterns and controls.
- Leverage modern table formats (Delta Lake, Iceberg, Hudi) for reliability and scalability.
- Maintain robust metadata management and data lineage tracking for transparency and compliance.
- Adopt lakehouse architectures to unify analytics and storage capabilities.