1. **Data Catalog:**

**Definition:** A data catalog is an inventory of all data assets within an organization. It helps data professionals find relevant data for analytical or business purposes by providing necessary information about the data.

**Example Use-Case:** Imagine a large retail company with diverse data sources. The data catalog allows their data analysts to quickly locate and assess relevant datasets. For instance, a marketing team can use the data catalog to find customer order data for personalized campaign analytics.

1. **Data Anonymization/Data Masking:**

**Definition:** Data anonymization (also known as data masking) is the process of transforming sensitive or personally identifiable information (PII) into a format that cannot be linked back to an individual. It ensures privacy while allowing data to be used for testing, development, or analysis.

**Example Use-Case:** A healthcare organization needs to share medical records for research purposes without revealing patient identities. Data anonymization masks names, addresses, and other PII, preserving privacy while enabling analysis.

1. **PII Data:**

**Definition:** PII data refers to personally identifiable information. It includes any data that can identify an individual, such as names, social security numbers, email addresses, or biometric data.

**Example Use-Case:** An insurance company handles PII data when processing claims. Protecting this sensitive information is crucial to comply with privacy regulations like GDPR or HIPAA.

1. **Data Democratization:**

**Definition:** Data democratization aims to make data accessible to a broader audience within an organization. It involves providing self-service access to data, empowering business users to explore and analyze it without heavy reliance on IT.

**Example Use-Case:** A retail company implements a user-friendly dashboard tool that allows store managers to analyze sales data independently. This democratization of data enhances decision-making across the organization.

1. **Data Modeling:**

**Definition:** Data modeling involves designing the structure and relationships of data entities in a database. It helps represent real-world concepts, ensuring efficient data storage and retrieval.

**Example Use-Case:** A financial institution creates a data model to define customer accounts, transactions, and relationships. This model guides database design and query optimization.

1. **Dashboards:**

**Definition:** Dashboards are visual representations of data, often displayed as charts, graphs, or tables. They provide a concise overview of key metrics and trends.

**Example Use-Case:** An e-commerce company uses a sales dashboard to monitor daily revenue, conversion rates, and customer demographics. This helps executives make informed decisions.

1. **Data Ecosystem:**

**Definition:** A data ecosystem encompasses all data-related components within an organization, including databases, applications, tools, and processes.

**Example Use-Case**: A tech startup’s data ecosystem includes cloud storage, data pipelines, machine learning models, and reporting tools. Integrating these components efficiently ensures smooth data flow.

1. **Data Enrichment:**

**Definition:** Data enrichment involves enhancing existing data with additional information. This could be through external sources, APIs, or data cleansing.

**Example Use-Case:** A travel agency enriches customer profiles with geolocation data, preferences, and social media activity. This enables personalized travel recommendations.

1. **Data Exchange:**

**Definition:** Data exchange refers to sharing data between different systems, organizations, or platforms. It ensures seamless data flow across boundaries.

**Example Use-Case:** A supply chain network exchanges real-time inventory data between suppliers, manufacturers, and retailers. This optimizes inventory management.

1. **Data Extraction:**

**Definition:** Data extraction involves retrieving specific data from various sources, such as databases, APIs, or web scraping.

**Example Use-Case:** A market research firm extracts pricing data from competitor websites to analyze market trends and adjust pricing strategies.

1. **Data Governance:**

**Definition:** Data governance is a concept within the discipline of data management that takes a holistic approach to an organization’s data and its lifecycle. It encompasses various aspects such as data ingestion, data catalogs, retention, storage management, sharing, archiving, backup, recovery, loss prevention, removal, and deletion.

**Example Use-Case:** Imagine a large financial institution. Data governance ensures that customer data adheres to privacy regulations, maintains data quality, and aligns with business strategy. For instance, defining access controls for sensitive financial records ensures compliance and protects customer information.

1. **1Data Ingestion:**

**Definition:** Data ingestion involves moving and replicating data from various sources (databases, files, streaming, etc.) to a destination such as a cloud data lake or data warehouse. It sets the foundation for downstream analytics.

**Example Use-Case:** An e-commerce company ingests real-time customer transaction data from its online platform into a cloud data lake. This data is then used for personalized recommendations, fraud detection, and sales forecasting.

1. **Data Joins:**

**Definition:** Data joins combine rows from multiple tables based on related columns. They allow us to retrieve data from different tables as if they were a single table.

**Example Use-Case:** Suppose we want to analyze sales data. We can join the “sales” table with the “products” table using a common product ID column. This allows us to see which products are selling well and their associated details.

1. **Data Lineage:**

**Definition:** Data lineage tracks the journey of data from its origin to its current state, including all transformations it undergoes. It provides visibility into data flow, helping identify errors and ensuring data quality.

**Example Use-Case:** In a complex big data environment, data lineage helps trace issues. If a report shows unexpected results, following the lineage reveals where the data originated and which transformations caused discrepancies.

1. **Data Mesh:**

**Definition:** Data mesh is a decentralized data architecture treating data as a product. It shifts data ownership to dedicated data product owners within business units. It operates based on domain-driven design, product thinking, and federated governance.

**Example Use-Case:** A retail company adopts a data mesh approach. Each business unit owns specific data products (e.g., customer profiles, inventory data). This decentralization improves agility, data quality, and domain-specific analytics.

1. **Data Portability:**

**Definition:** Data portability allows individuals to obtain and reuse their personal data across different services. It enables moving, copying, or transferring data from one IT environment to another securely without affecting its usability.

**Example Use-Case:** A user switches from one cloud storage provider to another. Data portability ensures they can seamlessly transfer their files, photos, and documents without loss or disruption.

1. **Data Replication:**

**Definition:** Data replication involves copying data from one location to another (e.g., between databases or systems). It ensures real-time consistency across data resources.

**Example Use-Case:** An e-commerce platform replicates customer order data from its primary database to regional replicas. This maintains up-to-date inventory and order information for efficient order processing.

1. **Data Privacy:**

**Definition:** Data privacy focuses on keeping data safe and private. It involves handling data with consent, notice, and adherence to regulatory obligations.

**Example Use-Case:** A healthcare provider ensures patient medical records are accessible only to authorized personnel, protecting sensitive health information.

1. **Data Consistency:**

**Definition:** Data consistency ensures uniformity, accuracy, and coherence of data across databases or systems. Values in different locations must be identical.

**Example Use-Case:** An e-commerce company maintains consistent product model records between its engineering and sales teams. Inconsistent data could lead to errors in order fulfillment.

1. **Data Quality:**

**Definition:** Data quality assesses how well a dataset meets standards for accuracy, completeness, reliability, and timeliness. High-quality data is trustworthy and suitable for analysis and decision-making.

**Example Use-Case:** A financial institution ensures that customer account balances are accurate and complete for financial reporting and risk assessment.

1. **Data Silo:**

**Definition:** Data silo refers to isolated data repositories within an organization that hinder data sharing and collaboration. Silos prevent seamless access to information across departments.

**Example Use-Case:** Marketing and sales teams struggle to align customer data due to separate CRM and marketing automation systems. Breaking down data silos improves customer insights and campaign effectiveness.

1. **Data Validation:**

**Definition:** Data validation ensures the accuracy and quality of data. It involves checks to confirm data type, code validity, range, format, consistency, and uniqueness.

**Example Use-Case:** An e-commerce platform validates customer order data to prevent incorrect entries and maintain data accuracy.

1. **Data Wrangling:**

**Definition:** Data wrangling transforms raw data into usable formats. It includes merging data sources, filling gaps, deleting irrelevant data, and handling outliers.

**Example Use-Case:** A data scientist cleans and reshapes messy survey responses into a structured dataset for sentiment analysis.

1. **Database Schema:**

**Definition:** A database schema formally defines the logical structure of data within a Database Management System (DBMS). It specifies how data is formatted, stored, processed, and accessed.

**Example Use-Case:** A retail company designs a schema for its product inventory database, defining tables for products, categories, and suppliers.

1. **Data Stewardship:**

**Definition:** Data stewardship manages data throughout its lifecycle. It includes creating metadata, ensuring data quality, applying standards, and maintaining data models.

**Example Use-Case:** A financial institution appoints a data steward responsible for maintaining accurate customer records and enforcing data policies.

1. **EDI Data Standards:**

**Definition:** EDI (Electronic Data Interchange) data standards define formats for business messaging. They enable automated exchange of documents like purchase orders, invoices, and requests for quotations.

**Example Use-Case:** Two trading partners use EDI standards to seamlessly exchange purchase orders and invoices, streamlining their supply chain.

1. **Observability:**

**Definition:** Observability refers to the ability to gain insights into the internal behaviour of a system by collecting, analysing, and visualizing relevant data. It encompasses monitoring, tracing, and logging to understand how components interact and identify issues.

**Use-case Example:** Imagine an e-commerce platform experiencing slow response times during peak shopping hours. By implementing observability practices, the operations team can monitor server metrics, trace API calls, and analyse logs to pinpoint bottlenecks and optimize performance.

1. **Streaming Data:**

**Definition:** Streaming data refers to continuous, real-time data flow from various sources (e.g., sensors, social media, IoT devices) that arrives in small, frequent chunks. It contrasts with batch processing, where data is processed in fixed intervals.

**Use-case Example:** A financial institution processes stock market data as it arrives, updating stock prices, calculating moving averages, and triggering alerts for specific events (e.g., sudden price drops). Streaming data enables timely decision-making and automated actions.

1. **Data Lake:**

**Definition:** A data lake is a centralized repository that stores raw, unstructured, and structured data at scale. Unlike traditional databases, data lakes accommodate diverse data types and allow flexible querying and analysis.

**Use-case Example:** A healthcare organization maintains a data lake containing patient records, medical images, research papers, and sensor data. Researchers can explore this vast dataset to discover correlations, develop predictive models, and improve patient care.

1. **Lakehouse Architecture:**

**Definition:** Lakehouse architecture combines the benefits of data lakes and data warehouses. It integrates data storage (like a data lake) with features like schema enforcement, indexing, and ACID transactions (like a data warehouse).

**Use-case Example:** An e-commerce company adopts a lakehouse architecture. Raw customer clickstream data is ingested into the data lake, but it’s also transformed into structured tables with defined schemas. Analysts can query both raw and curated data efficiently for business intelligence and reporting.

**2. Differentiate between Monolith vs Micro-service Architecture.**

| **Aspect** | **Monolithic Architecture** | **Microservices Architecture** |
| --- | --- | --- |
| **Definition** | A single, tightly coupled application where all components (UI, business logic, database) share the same codebase. | A collection of smaller, independently deployable services, each focusing on specific business capabilities. |
| **Unit of Deployment** | Entire application is deployed as one unit. | Individual services are deployed independently. |
| **Scalability** | Limited scalability due to centralized nature. | Scalability is flexible; services can scale independently based on demand. |
| **Updates and Maintenance** | Updates require modifying the entire stack. | Services can be updated, deployed, and maintained independently. |
| **Flexibility** | Less flexible; changes impact the entire system. | Highly flexible; updates to one service don’t affect others. |
| **Resilience** | Failures can impact the entire application. | Failures in one service don’t affect the entire system. |
| **Use-case Example** | Small-scale projects with limited complexity. | Large-scale applications with varying workloads and complex requirements (e.g., Netflix). |

**3. Write about the AWS services.**

1. **Amazon S3 (Simple Storage Service)**:
   * **Description**: Amazon S3 is an object storage service that provides scalable, durable, and secure storage for various data types, including files, images, videos, and backups.
   * **Use-cases**:
     + Storing and serving static website content.
     + Data backup and archival.
     + Hosting media files for applications.
     + Integrating with other AWS services like EC2, Lambda, and Redshift.
2. **Amazon S3 Glacier**:
   * **Description**: S3 Glacier is designed for long-term data archiving at a low cost. It provides three storage classes with varying retrieval times: Instant Retrieval, Flexible Retrieval, and Deep Archive.
   * **Use-cases**:
     + Archiving historical data.
     + Compliance records.
     + Medical images.
     + Genomics data.
3. **Amazon Redshift**:
   * **Description**: Redshift is a fully managed data warehouse service. It’s optimized for analytics and supports complex queries on large datasets.
   * **Use-cases**:
     + Business intelligence and reporting.
     + Data warehousing.
     + Data analytics.
4. **Amazon RDS (Relational Database Service)**:
   * **Description**: RDS simplifies database management by handling tasks like backups, patching, and scaling. It supports various database engines such as MySQL, PostgreSQL, and SQL Server.
   * **Use-cases**:
     + Running relational databases.
     + Migrating from on-premises databases.
     + Enterprise applications.
5. **Amazon DynamoDB**:
   * **Description**: DynamoDB is a NoSQL database service that provides fast and flexible storage for applications. It’s designed for low-latency, high-throughput workloads.
   * **Use-cases**:
     + Real-time applications.
     + Gaming leaderboards.
     + IoT data storage.
6. **Amazon EC2 (Elastic Compute Cloud)**:
   * **Description**: EC2 provides resizable compute capacity in the cloud. It allows you to launch virtual servers (instances) and manage them as needed.
   * **Use-cases**:
     + Hosting web applications.
     + Running custom software.
     + Scalable workloads.
7. **Amazon Lightsail**:
   * **Description**: Lightsail is a simplified compute service for developers who need an easy-to-use platform. It’s ideal for simpler workloads and quick deployments.
   * **Use-cases**:
     + Simple web applications.
     + Dev/test environments.
     + Small-scale projects.
8. **AWS Lambda**:
   * **Description**: Lambda is a serverless compute service that runs code in response to events. It automatically manages compute resources, allowing you to focus on writing code.
   * **Use-cases**:
     + Event-driven applications.
     + Data preprocessing.
     + Serverless APIs.
9. **Amazon SNS (Simple Notification Service)**:
   * **Description**: SNS is a messaging service that enables pub/sub communication. It allows you to send notifications to multiple subscribers via various protocols.
   * **Use-cases**:
     + Sending SMS alerts.
     + Distributing notifications to mobile devices.
     + Integrating with other AWS services.
10. **Amazon CloudWatch and CloudTrail**:
    * **Description**:
      + **CloudWatch**: Monitors AWS resources and applications, providing metrics, logs, and alarms.
      + **CloudTrail**: Records API calls made on your account, helping with auditing and security analysis.
    * **Use-cases**:
      + **CloudWatch**: Monitoring EC2 instances, RDS databases, and custom metrics.
      + **CloudTrail**: Tracking changes to resources and troubleshooting issues.
11. **Amazon SageMaker**:
    * **Description**: SageMaker is a fully managed machine learning service. It simplifies building, training, and deploying ML models.
    * **Use-cases**:
      + Developing ML models.
      + Natural language processing.
      + Computer vision.
12. **AWS Step Functions**:
    * **Description**: Step Functions coordinate multiple AWS services into serverless workflows. They allow you to build and visualize complex workflows.
    * **Use-cases**:
      + Orchestrating microservices.
      + Workflow automation.
      + State machine management.