**1.** **Write the definitions for the data terms slides and provide an example use-case for each term**

**1.Data Catalog**: A data catalog is a centralized metadata repository, often implemented using services like AWS Glue, AWS Lake Formation, and Amazon Athena, allowing users to efficiently discover, manage, and analyze data assets across various data sources within their organization.

**real-world test cases:**

Test the scalability of the data catalog solution to accommodate growing data volumes and diverse data formats without degradation in performance or reliability.

**2. Data Anonymization/Data Masking**: Data anonymization or data masking involves obscuring sensitive information within datasets to protect the privacy of individuals, typically achieved by replacing identifiable data with fictitious or generalized values while preserving the dataset's utility for analysis or testing purposes.

**real-world test cases:**

Execute SQL queries against a sample dataset in Amazon Redshift to verify that masked data fields return consistent and anonymized results without revealing original values.

**3. PII Data**: Personally Identifiable Information (PII) data refers to any information that can be used to identify an individual, such as names, social security numbers, email addresses, or biometric data, which must be handled with care to comply with privacy regulations.

**real-world test cases:**

Conduct regular compliance assessments and audits to ensure that data handling practices for PII data stored in AWS comply with GDPR, CCPA, or other applicable data protection regulations.

**4. Data Democratization**: Data democratization refers to the process of providing access to data and analytics capabilities to a broader range of users within an organization, enabling self-service analytics and empowering users to make data-driven decisions without extensive technical expertise.

**real-world test cases**

Monitor user adoption metrics and usage patterns to track the impact of data democratization initiatives on decision-making processes and organizational performance indicators.

**5. Data Modeling**: Data modeling involves designing the structure and relationships of data entities within a database or data warehouse, typically using techniques such as entity-relationship diagrams or schema design to organize and represent data for efficient storage and retrieval.

**real-world test cases**

Implement data validation checks and constraints within database schemas to prevent data integrity issues, such as duplicate records or invalid data types, from being introduced during data ingestion or updates.

**6. Dashboards**: Dashboards are visual interfaces that display key performance indicators (KPIs), metrics, and other relevant data in a concise and interactive format, allowing users to monitor and analyze data trends, make informed decisions, and track progress towards organizational goals.

**real-world test cases**

Validate the accessibility of dashboards on different devices and screen sizes, including desktops, tablets, and mobile phones, to ensure consistent rendering and usability across platforms**.**

**7. Data Ecosystem**: A data ecosystem encompasses the interconnected set of tools, services, and processes for managing, analyzing, and deriving value from data across an organization, including data storage, processing, integration, analytics, and governance components.

**real-world test cases**

Validate compliance with data governance policies and regulatory requirements across all components of the data ecosystem, ensuring data security, privacy, and regulatory compliance.

**8. Data Enrichment**: Data enrichment involves enhancing existing datasets with additional contextual information or attributes obtained from external sources, such as demographic data, geographic information, or market trends, to improve the quality and relevance of the data for analysis or decision-making.

**real-world test cases**

Evaluate the impact of data enrichment on data storage and processing costs within the data ecosystem, identifying opportunities for optimization and cost reduction.

**9. Data Exchange**: Data exchange involves the secure sharing and transfer of data between different systems, applications, or organizations, typically facilitated by APIs, data pipelines, or integration platforms to enable seamless data interoperability and collaboration.

**real-world test cases**

Evaluate the scalability of data exchange solutions to handle varying data volumes and processing loads, ensuring that performance remains consistent under peak usage conditions.

**10. Data Extraction**: Data extraction refers to the process of retrieving data from various sources, such as databases, files, or APIs, and transferring it to a destination for further processing, analysis, or storage, often performed using ETL (Extract, Transform, Load) tools or services.

**real-world test cases**

Validate compliance with data governance policies and regulatory requirements during data extraction processes, ensuring that sensitive information is handled securely and in accordance with data privacy regulations.

**11. Data Governance**: Data governance encompasses the policies, processes, and controls for ensuring the availability, integrity, security, and compliance of data assets across an organization, including data management, access control, and regulatory compliance measures.

**real-world test cases**

Validate data quality monitoring and remediation processes to detect and address data quality issues such as duplicates, inconsistencies, and missing values, ensuring data reliability and fitness for use.

**12. Data Ingestion**: Data ingestion involves the process of collecting, receiving, and loading data from various sources into a data storage or processing system, such as a data lake, data warehouse, or analytics platform, typically performed using automated pipelines or ingestion tools.

**real-world test cases**

Perform integration testing to verify the interoperability of data ingestion pipelines with external systems, applications, or data formats, ensuring seamless data integration and interoperability.

**13. Data Joins**: Data joins refer to the operation of combining related datasets based on common keys or attributes to create a unified view of the data, enabling analysis and insights generation across multiple sources, often performed in SQL queries or data processing workflows.

**real-world test cases**

Verify the effectiveness of query optimization techniques such as index usage, partitioning, and data denormalization in improving join performance and reducing query execution times.

**14. Data Lineage**: Data lineage involves tracking the origins, transformations, and movements of data throughout its lifecycle, providing visibility into how data is created, used, and modified across different systems, processes, and analytical workflows.

**real-world test cases**

Validate the accessibility and usability of data lineage information by data consumers and analysts, ensuring that lineage information is presented in a clear and intuitive manner to facilitate data discovery and understanding.

**15. Data Mesh**: Data mesh is an architectural approach that advocates for decentralizing data ownership and management by treating data as a product, enabling cross-functional teams to manage their own data domains, and providing self-serve data infrastructure and tools to enable data democratization and agility.

**real-world test cases**

Perform performance testing to evaluate the scalability and throughput of data mesh architecture in handling increasing data volumes and processing loads, ensuring that performance remains consistent under peak usage conditions.

**16. Data Portability**: Data portability refers to the ability to easily move data between different storage systems, services, or regions within the AWS ecosystem, ensuring flexibility and agility in managing data assets.

**real-world test cases**

Validate data synchronization between AWS and an external data warehouse by replicating changes bidirectionally.

**17. Data Replication**: Data replication involves copying data from one storage location or service to another, typically for purposes such as disaster recovery, high availability, or data distribution across multiple regions or environments.

**real-world test cases**

Test the failover process in a multi-AZ setup to ensure seamless transition and minimal data loss during failover events.

**18. Data Privacy**: Data privacy involves protecting sensitive information and ensuring compliance with data protection regulations by implementing appropriate security measures, encryption techniques, access controls, and data governance practices to safeguard data against unauthorized access or disclosure.

**real-world test cases**

Validate compliance with data privacy regulations by conducting audits and assessments of data handling practices, encryption mechanisms, and access controls.

**19. Data Consistency**: Data consistency refers to the reliability and accuracy of data across different systems or replicas, ensuring that all copies of the data are synchronized and up to date to maintain data integrity and reliability for applications and users.

**real-world test cases**

Implement automated data consistency checks and alerts to detect and resolve inconsistencies in real-time

**20. Data Quality**: Data quality refers to the level of accuracy, completeness, consistency, and reliability of data stored and processed within the AWS ecosystem, often addressed through data validation, cleansing, enrichment, and monitoring processes to ensure high-quality data for analytics and decision-making.

**real-world test cases**

Monitor data quality metrics such as completeness, accuracy, and consistency over time to identify trends and patterns in data quality.

**21. Data Silo**: A data silo refers to a situation where data is stored or managed in isolated or fragmented systems or environments, leading to inefficiencies, duplication, and barriers to data sharing and collaboration across an organization.

**real-world test cases**

Monitor key performance indicators (KPIs) related to data accessibility and collaboration to track the effectiveness of data silo reduction initiatives over time.

**22. Data Validation**: Data validation involves the process of checking and verifying the accuracy, integrity, and compliance of data against predefined rules, standards, or requirements, typically performed using automated validation routines or manual review processes to ensure data quality and reliability.

**real-world test cases**

Integrate data validation tests into CI/CD pipelines to automate the validation process and ensure continuous data quality monitoring throughout the software development lifecycle.

**23. Data Wrangling**: Data wrangling refers to the process of preparing and transforming raw or unstructured data into a usable format for analysis or consumption, involving tasks such as cleansing, parsing, aggregating, and structuring data using tools and services like AWS Glue, Data Pipeline, or Amazon EMR.

**real-world test cases**

Evaluate collaboration features and version control capabilities of data wrangling platforms to support teamwork and facilitate knowledge sharing among data wranglers and data scientists.

**24. Database Schema**: A database schema defines the structure, organization, and relationships of data elements within a database, specifying the tables, fields, constraints, and indexes that govern how data is stored, accessed, and manipulated.

**real-world test cases**

Evaluate the impact of database schema changes on downstream applications and analytics processes to minimize disruptions and ensure compatibility with existing data pipelines.

**25. Data Stewardship**: Data stewardship involves assigning responsibility and accountability for managing and protecting data assets within an organization, including defining data governance policies, overseeing data usage and access, and ensuring compliance with data privacy regulations and industry standards.

**real-world test cases:** Monitor key performance indicators (KPIs) related to data stewardship, such as data quality scores, data compliance metrics, and data governance maturity levels, to track progress and identify areas for improvement.

**26. EDI Data Standards**: Electronic Data Interchange (EDI) data standards define formats, protocols, and syntax for exchanging structured data electronically between different business systems or trading partners, facilitating seamless integration and interoperability of data across supply chains and business processes.

**real-world test cases**

Validate the reliability and fault tolerance of EDI data exchange mechanisms to ensure that data transmissions are secure, reliable, and auditable, with mechanisms in place for error handling and message retries.

**27. Observability**: Observability refers to the ability to monitor, measure, and understand the behavior and performance of distributed systems, applications, and services, using metrics, logs, traces, and other telemetry data to identify issues, troubleshoot problems, and optimize resource utilization and user experience.

**real-world test cases**

Validate the integration of observability solutions with other AWS services and third-party tools to ensure seamless data ingestion, analysis, and visualization for monitoring and troubleshooting purposes.

**28. Streaming Data**: Streaming data refers to continuous and real-time data streams generated by devices, sensors, applications, or online transactions, which are processed, analyzed, and acted upon in near-real-time using streaming data services such as Amazon Kinesis or AWS Lambda to enable use cases such as real-time analytics, monitoring, and alerting.

**real-world test cases**

Verify end-to-end data delivery and processing from data ingestion to data storage or analytics endpoints, ensuring that streaming data is processed accurately and delivered in a timely manner.

**29. Data Lake**: A data lake is a centralized repository that stores large volumes of structured, semi-structured, and unstructured data in its native format, providing scalable storage and processing capabilities for data analytics, machine learning, and other data-driven applications.

**real-world test cases**

Test data cataloging and metadata management capabilities to ensure that data assets stored in the data lake are well-documented, organized, and searchable.

**30. Lakehouse Architecture**: Lakehouse architecture combines the features and benefits of data lakes and data warehouses, enabling organizations to store and analyze both raw and structured data in a unified platform, leveraging services like AWS Glue, Amazon Redshift, and Apache Spark to support diverse analytics workloads and use cases with improved performance, cost-effectiveness, and ease of management.

**real-world test cases:**

Ensure that real-time data querying meets business needs and expectations.

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

**Monolithic:** One big block handling everything, like a single large building where all functions are under one roof.

**Microservices**: Breaking down functions into smaller, independent units, like a city with different buildings for different purposes, each working independently but contributing to the whole.

comparison between Monolithic and Microservices Architecture:

1. Structure:

- Monolithic: One big block of code containing all parts of the application.

- Microservices: Application divided into smaller, independent services, each responsible for a specific task.

2. Scalability:

- Monolithic: Scaling means replicating the entire application, which can be inefficient.

- Microservices: Each service can be scaled independently, saving resources and handling varying workload demands more efficiently.

3. Development and Deployment:

- Monolithic: Changes or updates require redeployment of the entire application, leading to longer cycles and increased risk.

- Microservices: Services can be developed, tested, and deployed independently, allowing for faster iteration and continuous delivery.

4. Technology Stack:

- Monolithic: Uses a single technology stack for the entire application.

- Microservices: Allows flexibility in choosing technology stacks for individual services, optimizing for specific requirements.

5. Resilience and Fault Isolation:

- Monolithic: Failure in one part can bring down the entire application.

- Microservices: Failures are contained within individual services, ensuring better fault tolerance and easier recovery.

6. Complexity:

- Monolithic: Becomes complex and hard to manage as it grows.

- Microservices: Offers flexibility and modularity but introduces complexity in managing distributed systems and communication between services.

**4. Write about the following AWS services.**

1. S3 and S3 Glacier:

- S3 (Simple Storage Service): Picture a massive digital storage facility accessible from anywhere online. It's commonly used for storing website data, backups, or large files like images and videos.

- S3 Glacier: Imagine a long-term storage option for your data, like a deep freezer. It's cost-effective for archiving data you don't need to access frequently but want to keep securely stored.

2. Redshift, Amazon RDS, and DynamoDB:

- Redshift: Think of it as a high-speed, robust database made for handling big data analysis. It's perfect for running complex queries and data analysis tasks efficiently.

- Amazon RDS (Relational Database Service): It's like having expert database managers handling your databases for you. RDS simplifies setting up, operating, and scaling relational databases like MySQL, PostgreSQL, or SQL Server in the cloud.

- DynamoDB: Imagine a lightning-fast, endlessly scalable database that's fully managed. DynamoDB is a NoSQL database service ideal for applications needing quick, high-performance data access.

3. EC2 and LightSail:

- EC2 (Elastic Compute Cloud): It's akin to renting virtual computers in the cloud. You can launch and manage virtual servers, known as instances, to run applications or host websites.

- LightSail: Think of it as a simpler version of EC2. LightSail offers an easy, straightforward way to launch and manage virtual private servers without the complexities of EC2. It's great for small projects or beginners looking to start quickly.

4. Lambda:

- Lambda is like having your own team of servers ready to execute your code, but you only pay for the time your code runs. It's a serverless compute service where you upload your code, and AWS handles infrastructure, scaling, and maintenance automatically.

5. Amazon SNS:

- SNS (Simple Notification Service) is like a cloud-based messaging system. It allows you to send notifications or messages to numerous subscribers via email, SMS, or other protocols. It's useful for sending alerts, updates, or messages to users or customers.

6. CloudWatch and CloudTrail:

- CloudWatch: Think of it as a dashboard for monitoring your AWS resources and applications. CloudWatch gathers and tracks metrics, monitors logs, and sets alarms to keep an eye on your AWS infrastructure's health and performance.

CloudTrail: It's like having a detailed log of all your AWS account activity. CloudTrail records every API call made within your AWS account, providing a trail of actions taken by users, services, or resources, crucial for security, auditing, and compliance.

7. Sage maker:

- Sage maker is akin to having a powerful AI assistant to aid in building, training, and deploying machine learning models. It's a fully managed service with built-in algorithms, development tools, and infrastructure, simplifying the machine learning workflow.s

8. Step Functions:

Step Functions act as orchestrators for your application's workflows. Think of them as flowcharts defining the sequence of steps or tasks to execute in response to events or triggers, making it easy to build and coordinate complex, stateful workflows in the cloud’s Step Functions is a fully managed service provided by Amazon Web Services (AWS) that allows you to coordinate and automate the components of your application's workflow. It enables you to build scalable, resilient applications by defining and executing stateful workflows in the cloud.