BYTEWISE LIMITED

Data Engineering Track

Task: Week – 1 (First Month)

Task No: 2 & 3

Task Date: 15-03-2023

Internee Name: Umer Farooq

Mentor Name: Ahtisham

Task Details:

This task includes the following:

1. Data Marts
2. Data Lakehouse
3. data Mesh
4. DWH vs Data Lake
5. OLTP vs OLAP

Additionally, after completing the tasks, after you complete these topics, answer the following questions in the document:

1. Can a database be used as DWH?
2. Major differences between structured and Un-structured data.
3. What are the duties of a data engineer? (high-level)

1.DATA MARTS:

- A Data Mart is a **subset of a data warehouse** that is designed to **serve the needs of a specific business unit or department** within an organization.
- It contains a subset of the data warehouse's data that is relevant to the business unit, and it is organized in a way that makes it easy to access and analyse.

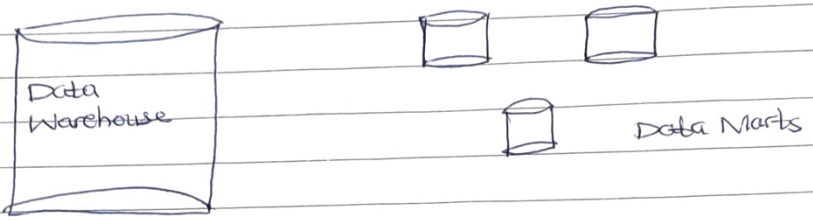
Difference b/w Data Marts & Data Warehouse:

Here are some key differences between Data Marts and Data Warehouses:

- **Scope:** Data Marts are focused on a specific business unit or department, while Data Warehouses are designed to serve the entire organization.
- **Data Volume**: Data Marts contain a smaller amount of data than Data Warehouses.
- **Data Structure**: Data Marts are often structured differently than the Data Warehouse, and may be optimized for specific types of queries and analysis.

②: Data Marts :-

A data mart is a sub-section of the data-warehouse, built specifically for a particular business function, purpose, or community of users.



②: Example :- Sales or finance groups in an organization accessing data for their quarterly reporting & projections.

NADEEM

analytics & use access only.

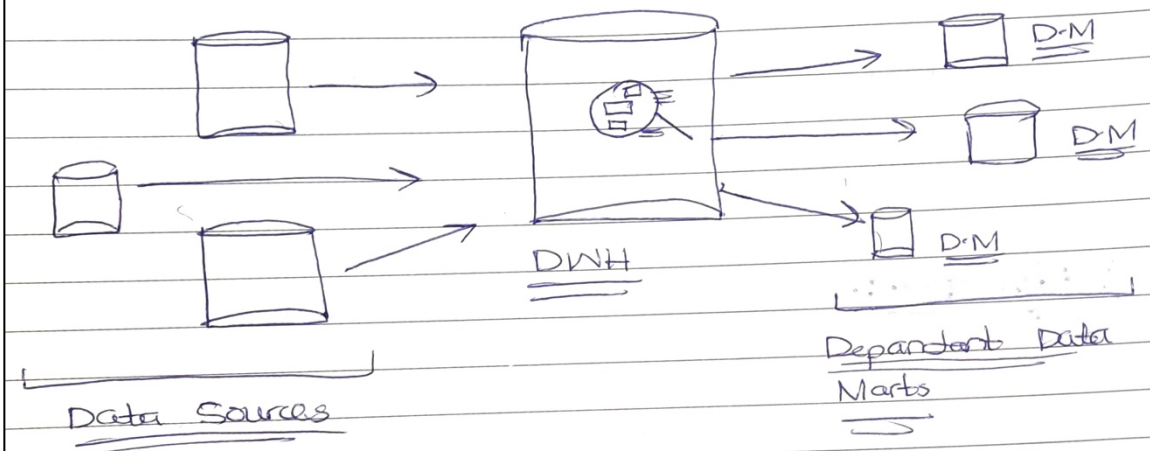
Types of Data Marts :-

There are 3 basic types of data marts :

1. Dependent Data Marts (D-M)
2. Independent D-M
3. Hybrid D-M

1 : Dependant Data Marts :-

- Dependant data marts are a sub-section of an entripse data warehouse.



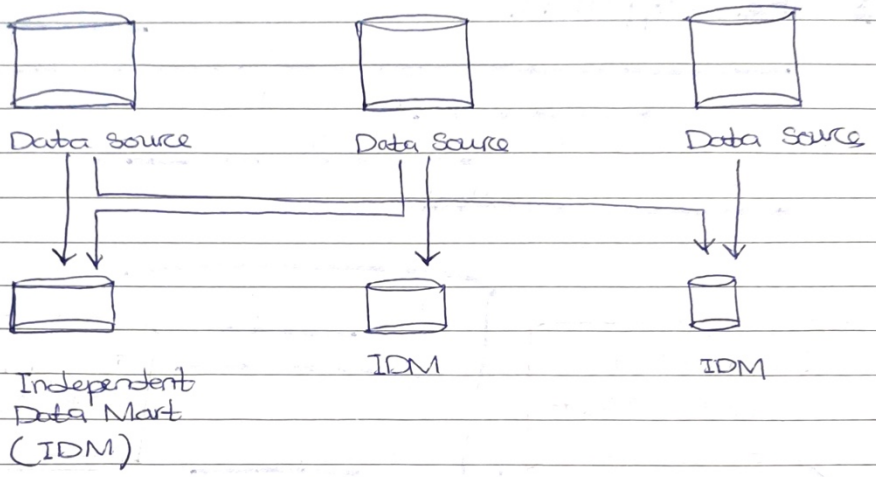
(Since) Dependant Data marts offer analytical capabilities for a restricted area of the data warehouse, it also provides

Date: _____ M T W T F S S

isolated security & isolated performance.

② Independent Data Marts:

- Independent data marts are created from sources other than an Enterprise Data Warehouse, such as internal operational systems (e.g: sales data, website data) or external data (online search queries related to certain products).

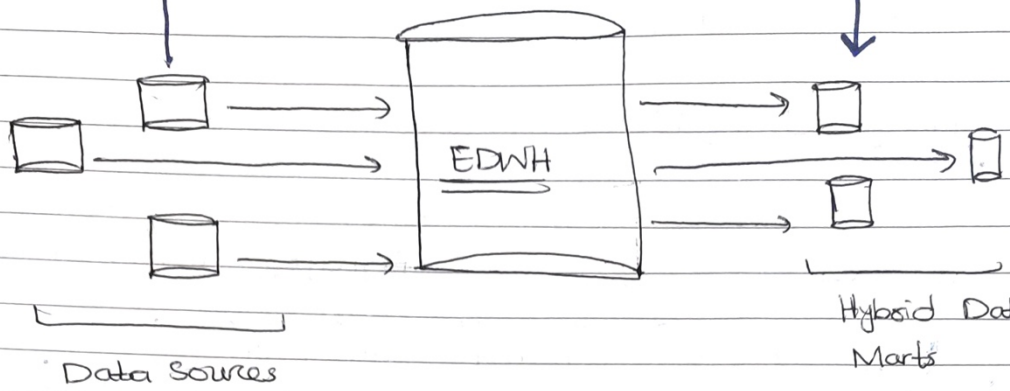


③: Hybrid Data Marts:

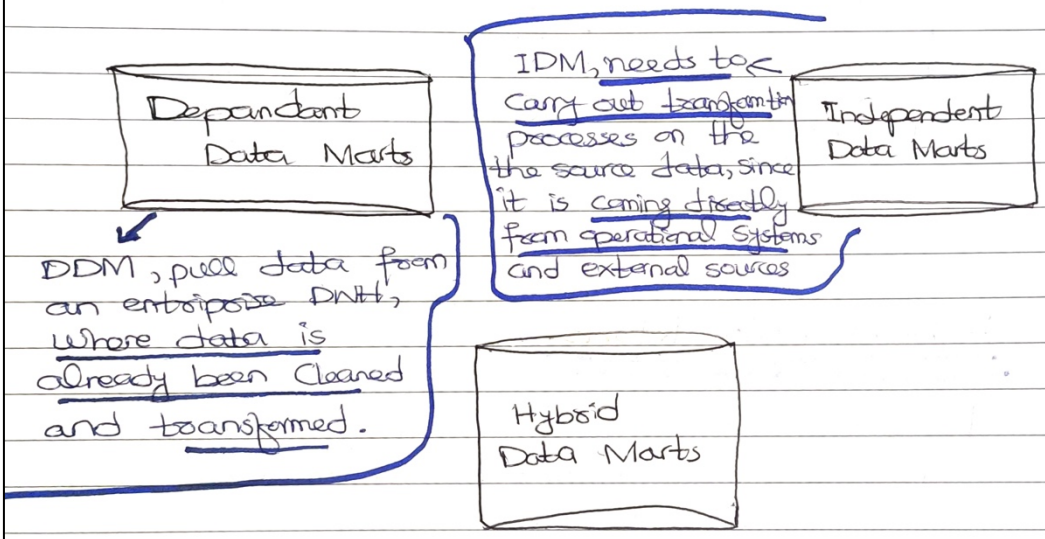
- Hybrid data marts combine inputs from data warehouses, operational systems, & external systems.

NADEEM

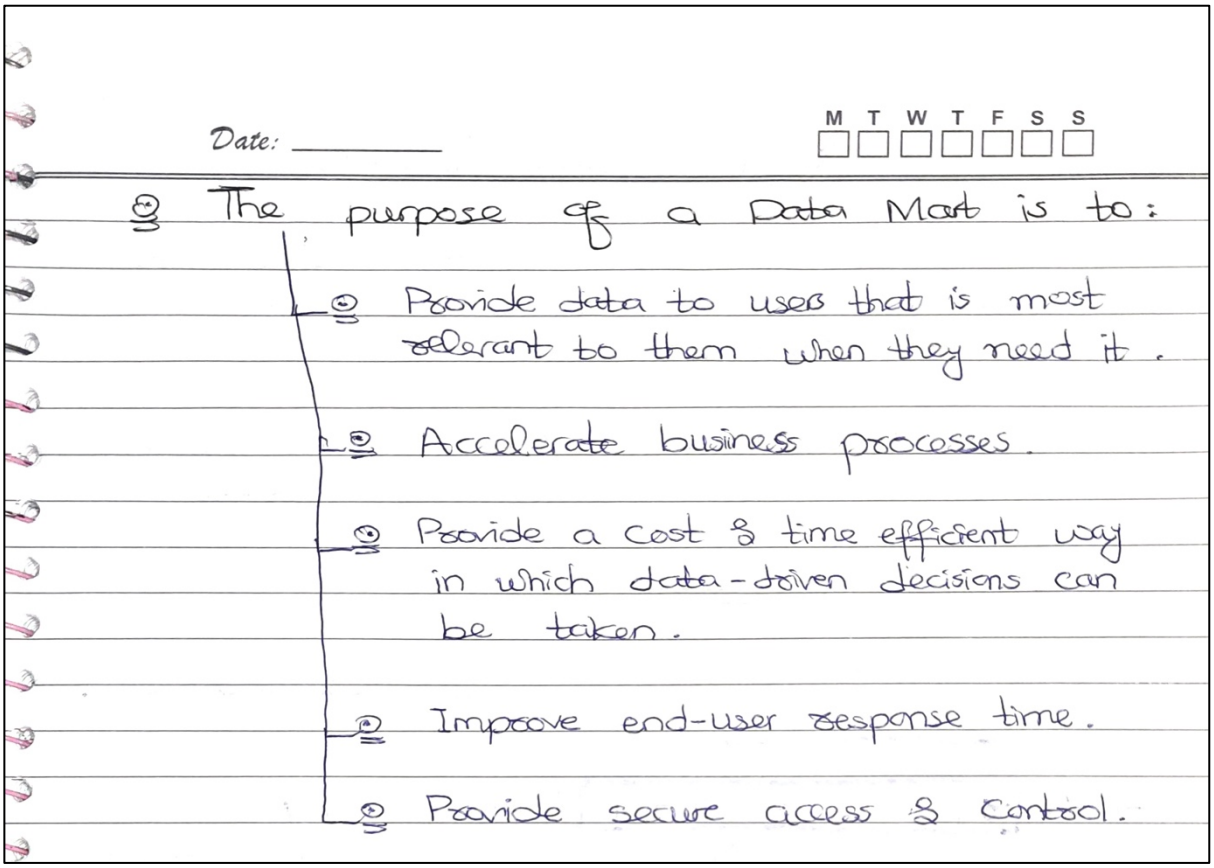
Date: _____ M T W T F S S



- The difference also lies in how data is extracted from the source systems, the transformation that need to be apply, & how the data is transported into the mart.



NADEEM



2.DATA LAKEHOUSE:

- A Data Lakehouse is a **new data storage architecture** that combines the best features of both data lakes and data warehouses.
- It's a **modern approach to data management** that enables companies to store and analyse massive amounts of data in a flexible and cost-effective way.
- In a Data Lakehouse, data is stored in a central repository, which is typically a cloud-based object store like Amazon S3, Google BigQuery or Microsoft Azure Blob Storage. The data is stored in its

raw form, without any transformation or pre-processing. The Data Lakehouse also includes a processing layer that provides compute resources for running analytical queries on the data.

Comparing Data Lakehouse to Data Warehouse & Data Lake:

- To ensure that your business can meet its present needs while being able to expand in the future, your data architecture should evolve accordingly. Fortunately, you have access to better options than either a low-cost data swamp that doesn't provide sufficient data governance or a rigid data ingestion machine lacking artificial intelligence capabilities.
- Data lakes are not equipped to provide the necessary data governance capabilities to securely manage large data sets, while data warehouses only support structured data ingestion and lack the flexibility and scalability required by growing businesses. These solutions were not designed to address the current data challenges faced by organizations.
- In response to these limitations, data lakes were developed to handle all types of data, with cost-effective storage and support for data science and machine learning applications. However, they lack a critical component found in data warehouses: the

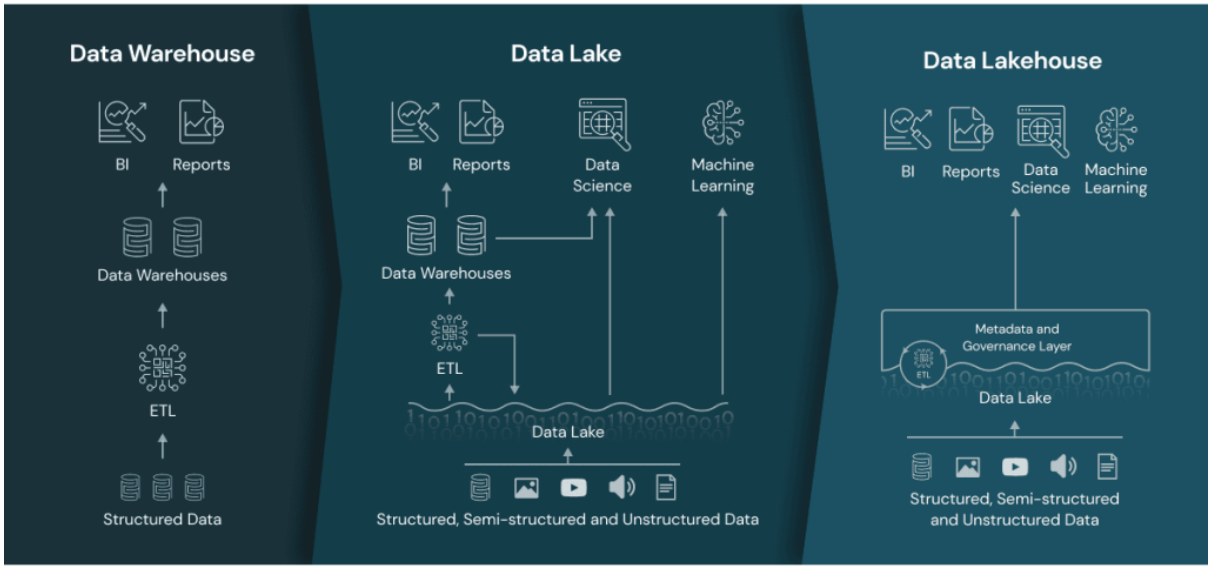
ability to support ACID transactions or enforce data quality and governance, which can make working with data cumbersome and time-consuming.

- *The key difference between a Data Lakehouse and a Data Warehouse is that a Data Warehouse is typically optimized for high-performance queries on structured data, whereas a Data Lakehouse is optimized for storing and querying both structured and unstructured data. A Data Lake, on the other hand, is typically used for storing large amounts of unstructured data in its raw form, without any processing or transformation.*

Summary:

- Organizations need to use a Data Lakehouse because it enables them to store and analyze massive amounts of data in a flexible and cost-effective way.
- Organizations can avoid the cost and complexity of traditional ETL (Extract, Transform, Load) processes by implementing a Data Lakehouse. In a traditional ETL process, data is extracted from source systems, transformed to fit a predefined schema, and loaded into a target system such as a data warehouse. This process is often time-consuming, costly, and requires a significant amount of effort to maintain.

- With a Data Lakehouse, organizations can store their data in a flexible, schema-on-read format, meaning that data is not transformed before it is loaded into the lakehouse. This eliminates the need for a separate transformation step, reducing the complexity and cost of the process.



This diagram shows the comparison of Data Warehouse, Data Lake and Data Lakehouse Architecture.

3.DATA MESH:

Data Mesh is a new approach to data engineering and management that seeks to address the challenges of managing data at scale in a complex, distributed ecosystem. The core idea of Data Mesh is to treat data as a product and to enable teams to create, manage, and share data as a service across the organization.

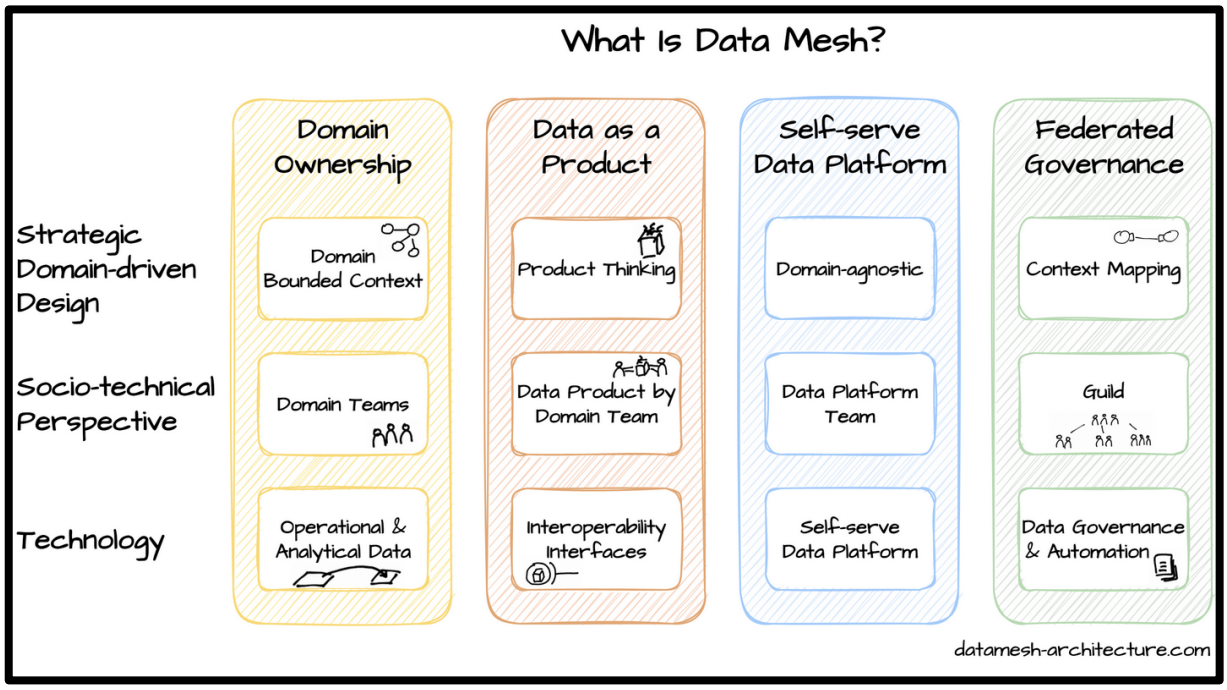
- In a Data Mesh architecture, data is decentralized and owned by the teams that create and use it. These teams are responsible for the quality, governance, and lifecycle management of their data products. This approach empowers teams to make decisions about the data they create and use, and reduces the burden on centralized data teams.

Example:

To illustrate the concept of Data Mesh, let's consider a simple example.

- **Imagine a retail company** that wants to improve its customer experience by analyzing customer data. Traditionally, the company **might have a centralized data** team responsible for collecting, cleaning, and analyzing customer data. However, with a Data Mesh approach, the company would create a

customer data product that is owned and managed by a cross-functional team of product managers, data scientists, and engineers. This team would be responsible for defining the data schema, creating data pipelines to ingest and transform data, and ensuring the quality and governance of the data product. Other teams across the organization could then access this data product as a service to build customer-facing applications, run analyses, or perform experiments.

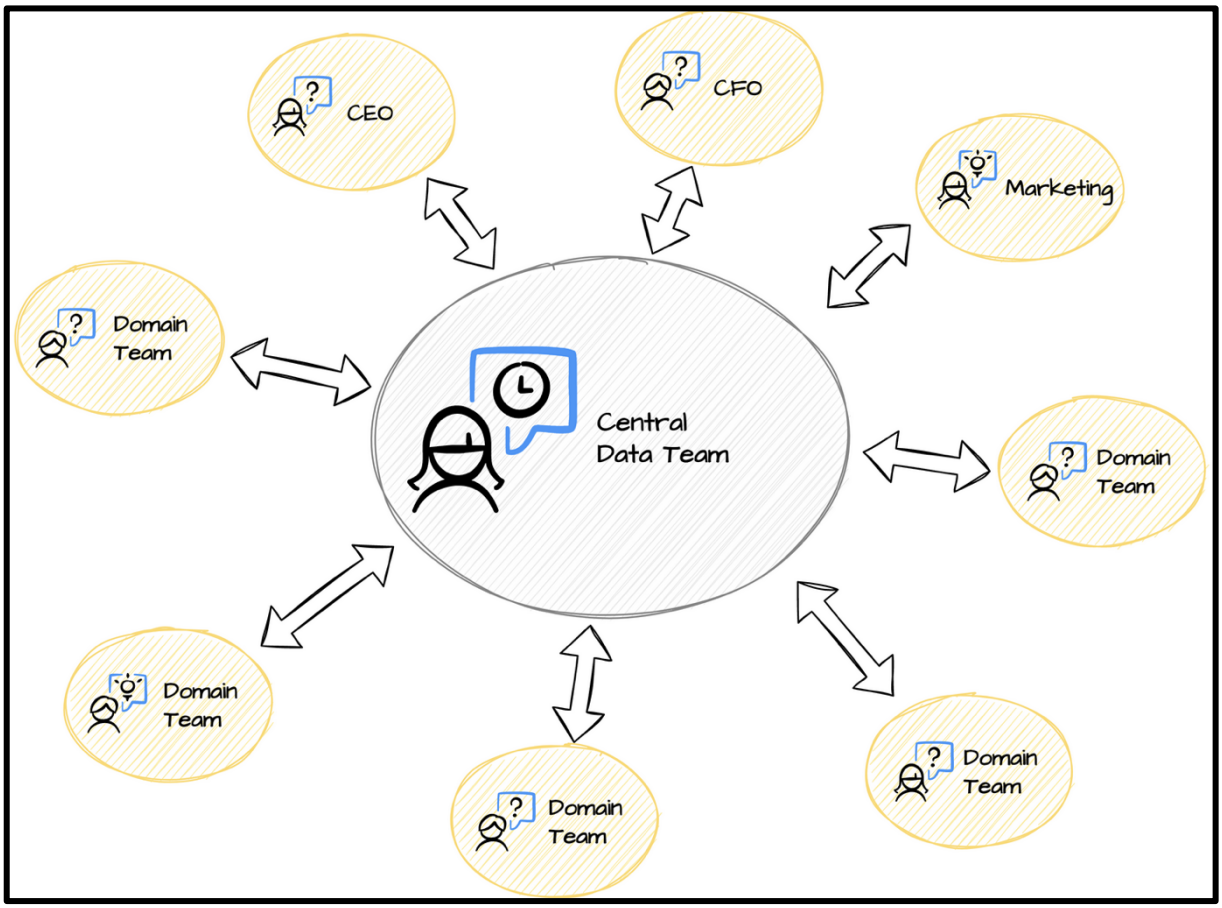


Some of the key principles of Data Mesh include:

- **Data as a product:** treating data as a first-class citizen and managing it like a product, with well-defined interfaces, quality standards, and ownership.

- **Domain-oriented decentralized architecture:** organizing data around autonomous domains that are responsible for managing their own data and providing well-defined APIs for other domains to access it.
- **Self-serve data infrastructure**: providing a self-serve data infrastructure that enables teams to easily discover, access, and use data from other domains.
- **Federated governance**: decentralizing data governance across domains and creating a federated governance model that enables teams to manage their own data while still adhering to overall organizational standards.

Why You May Need a Data Mesh:



- Many organizations have invested in a central data lake and a data team with the expectation to drive their business based on data. However, after a few initial quick wins, they notice that **the central data team often becomes a bottleneck**. The team cannot handle all the analytical questions of management and product owners quickly enough. This is a massive problem because making timely data-driven decisions is crucial to stay competitive. For example: Is it a good idea to offer free shipping during Black Week? Do customers accept longer but more

reliable shipping times? How does a product page change influence the checkout and returns rate?

- The data team wants to answer all those questions quickly. In practice, however, they struggle because they need to spend too much time fixing broken data pipelines after operational database changes. In their little time remaining, **the data team has to discover and understand the necessary domain data**. For every question, they need to learn domain knowledge to give meaningful insights. Getting the required domain expertise is a daunting task.
- On the other hand, organizations have also invested in domain-driven design, autonomous domain teams (also known as stream-aligned teams or product teams) and a decentralized microservice architecture. These **domain teams own and know their domain**, including the information needs of the business. They design, build, and run their web applications and APIs on their own. Despite knowing the domain and the relevant information needs, the domain teams have to reach out to the overloaded central data team to get the necessary data-driven insights.
- With the eventual growth of the organization, the situation of the domain teams and the central data team becomes worse. A way out of this is to shift the responsibility for data from the central data team to the domain teams. This is the core idea

behind the data mesh concept: **Domain-oriented decentralization for analytical data**. A data mesh architecture enables domain teams to perform cross-domain data analysis on their own and interconnects data, similar to APIs in a microservice architecture.

Data Mesh has gained popularity in recent years as a way to address some of the challenges associated with traditional centralized data architectures, such as data silos, data quality issues, and slow time-to-insights. By decentralizing data ownership and governance, Data Mesh enables organizations to build more agile, scalable, and resilient data platforms that can better support their business needs.

4.DATA WAREHOUSE VS DATA LAKE:

Criteria	Data Warehouse	Data Lake
Purpose	Designed for structured data with a focus on querying and analysis.	Designed to store and process large volumes of structured and unstructured data for various use cases.

Data Type	Structured Data	Structured, Semi Structured, and Unstructured.
Schema	Follows a rigid schema.	Schema-on-read; Schema is flexible and can be modified at any time.
Data Processing	Uses ETL (Extract, Transform, Load) process to integrate and transform data.	Uses ETL (Extract, Transform, Load) process to load data and then perform transformation.
Data Quality	Emphasizes on data accuracy, consistency, and completeness.	Emphasizes on data agility, accessibility, and scalability.
Storage	Optimized for fast read access and query performance.	Provides cost-effective and scalable storage for data.

Tools & Technologies	Examples: Amazon Redshift, Google BigQuery, Microsoft Azure Synapse Analytics.	Examples: Amazon S3, Apache Hadoop, Apache Spark, Microsoft Azure Data Lake Storage.

Here is an example to illustrate the difference between Data Warehouse and Data Lake:

- **Data Warehouse:** Suppose a retail company wants to analyze its sales data to identify trends and insights. The company's Data Warehouse would store data from various sources, such as point-of-sale systems and online orders, and transform it into a structured format. The data would be stored in a predefined schema that supports efficient querying and analysis. The company could then use tools like Amazon Redshift or Microsoft Azure Synapse Analytics to analyze the data and generate reports.
- **Data Lake:** If the retail company wanted to store all of its sales data, including unstructured data like social media mentions and customer feedback, it could use a Data Lake. The Data Lake would store all of the data in its native format, without enforcing any schema or structure. The data could be stored in

a cost-effective and scalable storage solution like Amazon S3 or Microsoft Azure Data Lake Storage. The company could then use tools like Apache Spark or Apache Hadoop to extract, load, and transform the data, and use it for various use cases, such as machine learning and predictive analytics.

5.OLTP VS OLAP:

OLTP	OLAP
OLTP stands for Online Transaction Processing.	OLAP stands for Online Analytical Processing
Used for Day-to-day operations such as order processing, inventory management, & customer service.	Used for data analysis, decision making, an business intelligence.
Data is highly normalized and stored in relational databases.	Data is denormalized and stored in Data Warehouse.

Transaction are small, simple, & atomic.	Queries are complex and involve aggregations, calculations, and data mining.
Tools and technologies used: Relational databases (Oracle, SQL Server, MySQL), ERP systems (SAP, Oracle E-Business Suite).	Tools and technologies used: Data warehouses (Teradata, IBM InfoSphere, Amazon Redshift), Business Intelligence tools (Tableau, Power BI, QlikView).
Example: A bank ATM transaction, a customer placing an online order, a supermarket checkout.	Example: Analyzing sales data to identify trends, predicting customer behavior based on purchase history

- **OLTP systems are used in operational environments** where fast and accurate transaction processing is required. They are typically used by front-line employees such as customer service representatives, salespeople, and clerks. The data generated by OLTP systems is used to run the day-to-day operations of the business.

- **OLAP systems are used in analytical environments** where data analysis, decision-making, and business intelligence are required. They are typically used by managers, executives, and analysts who need to analyze large volumes of data to make strategic decisions. The data generated by OLAP systems is used to support decision-making, identify trends and patterns, and gain insights into business performance.

Bytewise is startup and provides Data services to its clients:

- As a data engineering company, Bytewise Limited would likely be involved in designing and implementing data solutions for their clients. Both OLTP and OLAP technologies would be relevant for different stages of the data engineering process.
- **OLTP systems would be used for building transactional systems that handle day-to-day operations for clients**. For example, if a client needs a system to handle online orders, Bytewise Limited would design an OLTP system to manage the transactions and data associated with the order processing. The OLTP system would likely be built using a relational database management system (RDBMS) such as Oracle, SQL Server, or MySQL.

- On the other hand, **OLAP systems would be used for building analytical systems that provide insights and decision-making capabilities for clients**. For example, if a client needs to analyze sales data to identify trends and make informed decisions, Bytewise Limited would design an OLAP system that aggregates and analyzes the data. The OLAP system would likely be built using a data warehouse such as Teradata, IBM InfoSphere, or Amazon Redshift, along with business intelligence tools such as Tableau, Power BI, or QlikView.
- In summary, Bytewise Limited would use OLTP systems for building transactional systems that handle day-to-day operations for clients, and OLAP systems for building analytical systems that provide insights and decision-making capabilities for clients. The specific technologies and tools used would depend on the requirements and preferences of each client.

Answering the Questions:

1. Can a database be used as DWH?

While it is possible to use a database as a data warehouse, it may not be the most efficient or scalable solution for large-scale analytical

processing. Databases, such as Oracle, SQL Server, and PostgreSQL, can be used as data warehouses with the implementation of data warehousing features such as partitioning, compression, and parallel query processing.

- However, **databases are primarily optimized for transactional processing**, **while data warehouses are optimized for analytical processing**. A database is designed to support rapid insertion, update, and retrieval of individual records, while a data warehouse is designed to support querying and analysis of large volumes of data. To adapt a database for data warehousing, **techniques such as partitioning, indexing, and materialized views can be implemented**.

Example:

- For example, a company with a database containing transactional data could create a data warehouse within the same database by using partitioning to divide the data into smaller, more manageable chunks, creating indexes on data warehouse tables, and using materialized views to precompute frequently accessed queries. With these techniques in place, the company could run complex queries against the data

warehouse to gain insights into sales trends, customer behaviour, and inventory management.

- Overall, while a database may not be the optimal choice for a data warehouse, it is possible to adapt it to support analytical processing with the right design and implementation techniques.

2. Major differences between Structured & Unstructured data.

Structured data refers to information that is organized in a **specific format** with **predefined categories and fields**. It can be easily stored, managed, and analyzed using database management systems. Examples of structured data include data in spreadsheets, relational databases, and machine-readable formats such as XML and JSON.

Unstructured data refers to data that has no specific format or organization. It is typically made up of free-form text, images, videos, and audio files, and can be difficult to process and analyze using traditional methods. Examples of unstructured

data include social media posts, emails, customer reviews, and sensor data.

- The main difference between structured and unstructured data is the level of organization and predictability. Structured data is highly organized and predictable, while unstructured data is more chaotic and difficult to organize. Structured data is easier to process and analyse using automated tools and algorithms, while unstructured data requires more sophisticated methods such as natural language processing and machine learning.
- However, both structured and unstructured data have their advantages and disadvantages depending on the application. Structured data is useful for transactional systems and business intelligence applications, while unstructured data is valuable for understanding customer sentiment and social media trends. As such, organizations must develop strategies to manage and utilize both types of data to gain a comprehensive understanding of their operations and customers.

Date: _____ M T W T F S S

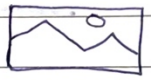
Topic :- Types of Data :-

- Data is unorganized information that is processed to make it meaningful.

- Generally Data comprises of: ~~facts~~

Facts
Observations
Perceptions

Numbers
characters
Symbols



Images

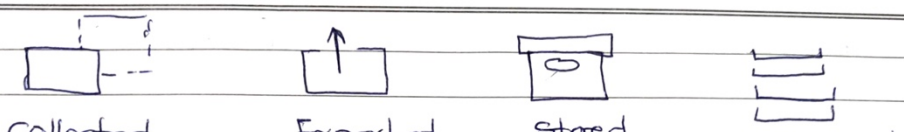
that can be interpreted to derive meaning.

- Data categorized based on its structure: → level of rigidity of

1. **Structure:**
   - Has well-defined structure
   - Can be stored in well-defined schemas
   - Can be represented in a tabular manner with rows & cols.

- Structured data are objective FACTS & Numbers that can be: ~~collected~~

Date: _____ M T W T F S S



in typical databases.

Sources of Structured Data:

↳ sources includes:

- SQL Databases
- Online Transaction Processing
- Spreadsheets (Excel, Google)
- Online forms (Google form)
- Sensors GPS & RFID
- Network & Web server logs

- You can easily examine Structured Data with standard data analysis methods & tools.

Date: _____ M T W T F S S

Unstructured Data :-

- Does not have easily identifiable structure.
  - ↙ Therefore
- Cannot be organized in a mainstream relational database in the form of rows & cols.
- Does not follow any particular format, sequence, semantics, or rules.

(A) : Sources :

It includes:

- Web Pages
- Social media feeds
- Images in varied file formats (JPG, PNG, JPEG, GIF)
- Video & audio files
- Documents & PDF files
- Powerpoint Presentations
- Media logs
- Surveys

NADEEM

Date: _____

M T W T F S S

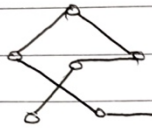
Unstructured Data can be stored in :

for

Manual Analysis

Files/Documents
(Word doc)

or



for

NoSQL Databases
(Cassendra / Mongo DB)

Analysis Tools
(NoSQL have its own tools to examine the data.
Example Mongo DB
- charts
- Studio 3T)

NADEEM

3. What are the duties of a Data Engineer? (High Level)

Some of the key duties of a Data Engineer may include:

1. Designing, building, and maintaining data pipelines: Data Engineers create robust and scalable pipelines to collect, store, and process data from various sources such as databases, APIs, and data streams.

2. Developing and implementing data storage solutions: Data Engineers design and implement data storage solutions such as data warehouses, data lakes, and NoSQL databases, that can efficiently store and manage large amounts of data.

3. Ensuring data quality and consistency: Data Engineers ensure that the data being processed is accurate, consistent, and clean by implementing data validation, cleansing, and normalization processes.

4. Developing and maintaining ETL processes: Data Engineers design, develop, and maintain Extract, Transform, and Load (ETL) processes that extract data from source systems, transform it into a format suitable for analysis, and load it into the target system.

5. Collaborating with data scientists and analysts: Data Engineers work closely with data scientists and analysts to understand their data needs and help them access and use the data effectively.
6. Monitoring and optimizing data performance: Data Engineers monitor the performance of data processing systems and optimize them to ensure fast and efficient data processing.
7. Ensuring data security and privacy: Data Engineers are responsible for ensuring that the data being processed is secure and meets privacy regulations by implementing appropriate security measures and access controls.