***Task -2 & 3 2023/03/15***

1. ***Data Marts:*** A data mart is a subset of a data warehouse (DWH) that is designed for a specific business unit, department, or functional area within an organization. It contains a set of data that is tailored to meet the specific needs of the particular group of users. Data marts are often designed and maintained separately from the DWH to provide faster access and more targeted analysis for end-users. Data marts can be created either by extracting data from a DWH or by integrating data from various sources.
2. ***Data Lakehouse:*** A data lakehouse is a hybrid data storage architecture that combines the advantages of both data lake and data warehouse technologies. A data lakehouse allows organizations to store structured and unstructured data in a single repository and enables users to perform analytical queries and real-time analytics on that data. This architecture eliminates the need for ETL (Extract, Transform, Load) processes that are used in traditional data warehouses and allows for more flexibility and agility in data processing.
3. ***Data Mesh:*** Data Mesh is a modern data architecture that emphasizes decentralization and domain-driven design. In this approach, data is treated as a product and each domain is responsible for the data they own. Data Mesh emphasizes the use of microservices and APIs to enable data sharing and standardization across different domains. It aims to empower data consumers and encourages collaboration between different teams within an organization.
4. ***DWH vs Data Lake:*** A Data Warehouse (DWH) is a centralized repository that stores structured data from various sources for analysis and reporting. It is designed for business intelligence and decision-making purposes. A Data Lake, on the other hand, is a storage repository that can store both structured and unstructured data in its native format. It is designed for storing large volumes of raw data that may be used for multiple purposes. The main differences between DWH and Data Lake are the types of data they store, the level of data processing they perform, and the way they are designed and maintained.
5. ***OLTP vs OLAP:*** OLTP (Online Transaction Processing) is a type of database that is designed for transactional processing such as inserting, updating, and deleting data. OLTP databases are typically used for transactional applications such as e-commerce, banking, and inventory management systems. OLAP (Online Analytical Processing), on the other hand, is a type of database that is designed for analytical processing such as querying and reporting data. OLAP databases are typically used for business intelligence and decision-making purposes.
6. ***Answers to the questions:***
7. ***Can a database be used as DWH?*** Yes, a database can be used as a DWH if it is designed and structured to support analytical processing and reporting. However, using a database as a DWH may not be as efficient as using a dedicated data warehousing system because databases are optimized for transaction processing and not for analytical processing.
8. ***Major differences between structured and unstructured data.*** Structured data refers to data that is organized in a predefined format with a well-defined schema. Examples of structured data include data stored in relational databases, spreadsheets, and XML files. Unstructured data, on the other hand, refers to data that does not have a predefined format or schema. Examples of unstructured data include emails, social media posts, images, and videos. The main differences between structured and unstructured data are the way they are stored, processed, and analyzed.
9. ***What are the duties of a data engineer? (high-level)*** Data engineers are responsible for designing, building, and maintaining the data infrastructure of an organization. Some of the main duties of a data engineer include:

* Designing and implementing data storage solutions
* Building and maintaining data pipelines and ETL processes
* Developing and managing databases
* Ensuring data quality and integrity
* Monitoring