Data Engineering - HEXAWARE

Assignment 1

1. Introduction to Data Warehousing

A Data Warehouse (DW) is a centralized system used for reporting and data analysis. Unlike transactional systems, DWs are optimized for read-heavy operations and long-term data storage.

Key Features:

- Subject-Oriented: Focuses on specific subjects like sales, finance, etc.
- Integrated: Combines data from multiple sources (e.g., ERP, CRM).
- Time-Variant: Stores historical data to enable trend analysis.
- Non-Volatile: Once data enters the warehouse, it is not changed.

Example:

A retail company collects sales data from all branches and stores it in the warehouse to analyze performance monthly or quarterly.

---

2. Purpose of a Data Warehouse

The main goal is to empower organizations with strategic decision-making through integrated and clean data.

Objectives:

- Centralize Data: Aggregates data from multiple systems (HR, Finance, Sales).
- Enable BI Tools: Makes it easy for tools like Power BI/Tableau to generate reports.
- Historical Analysis: Tracks performance over time.
- Data Consistency: Ensures uniform formats and standards across departments.
- Support OLAP Operations: Facilitates multi-dimensional queries and aggregations.

---

3. Data Warehouse Architecture

A typical 3-tier architecture includes the following layers:

a. Data Source Layer

- Sources: ERP systems, CRM platforms, flat files, web logs.

- Sends raw data to staging.

b. Data Staging Layer (ETL Layer)

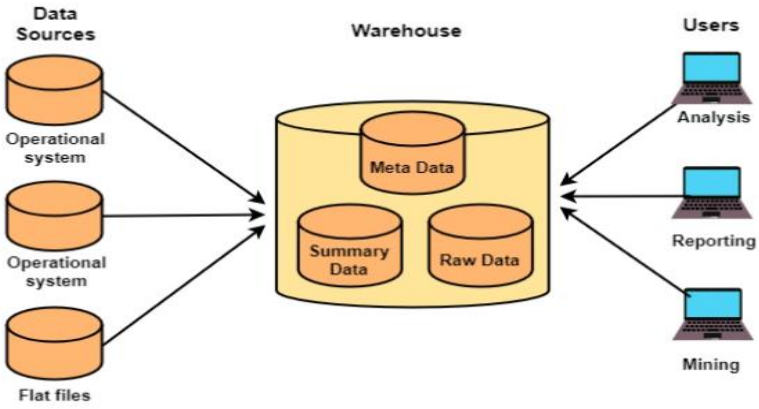
- ETL Process: Extract, Transform, Load.
- Cleans, validates, and formats the data.
- Tools: Informatica, Talend, SSIS.

c. Data Storage Layer

- Stores structured data.
- Schema design: Star or Snowflake schema.
- Contains Fact tables (measurable data) and Dimension tables (descriptive data).

d. Presentation/Access Layer

- Reporting via dashboards, OLAP cubes.
- Tools: Power BI, Tableau, Excel, MicroStrategy.



Flow:
Source → ETL → DW → BI Tools

---

4. Operational Data Store (ODS)

An ODS is an intermediate layer between the transactional system and DW. It holds real-time or near-real-time operational data.

Characteristics:

- Used for day-to-day reporting.
- Doesn't store historical data.

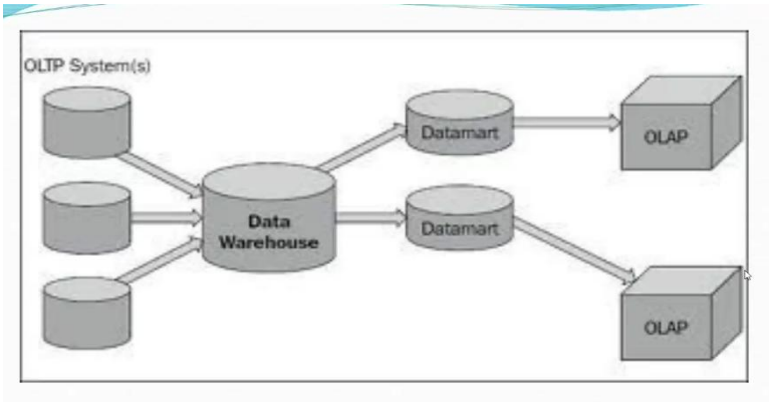
- Frequently updated (unlike DW which is periodically updated).
- Acts as a staging area before data moves to the DW.

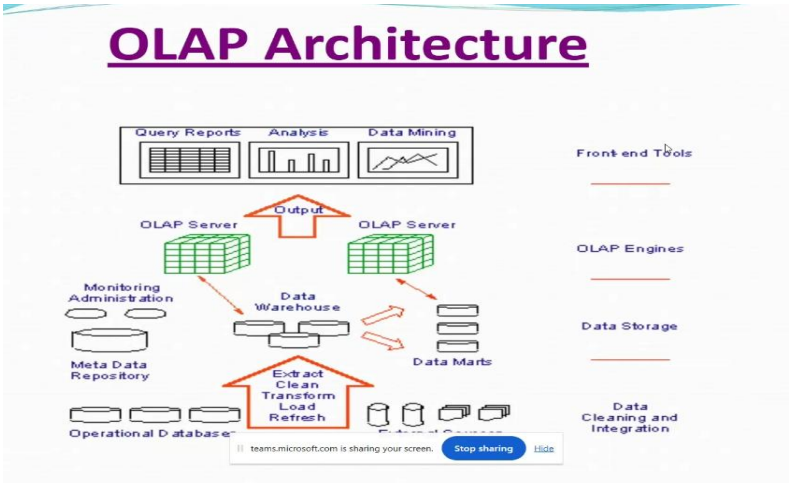
Example: A bank logs all customer transactions in an ODS for intraday analysis before archiving them into the DW.

---

5. OLTP vs Data Warehouse (OLAP)

Feature	OLTP (Online Transaction Processing)	Data Warehouse (OLAP)
Purpose	Run day-to-day operations	Analyze data
Data	Current, real-time	Historical, consolidated
Operations	Insert, update, delete	Select, aggregate, drill-down
Speed	Optimized for write-speed	Optimized for read-speed
Schema	Highly normalized	Denormalized (star/snowflake)
Examples	Banking system, e-commerce orders	KPI dashboards, trend analysis





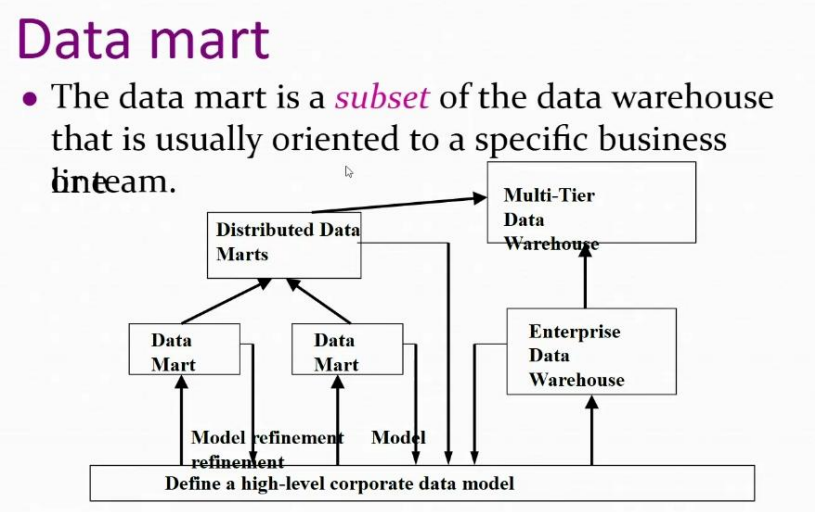
6. Data Marts

A Data Mart is a smaller, focused version of a data warehouse — usually specific to a department.

Characteristics:

- Stores data specific to a business line (e.g., Sales, HR, Finance).
- Easier to build and manage.
- Faster query response due to smaller data volume.
- Can be dependent (sourced from DW) or independent (from OLTP directly).

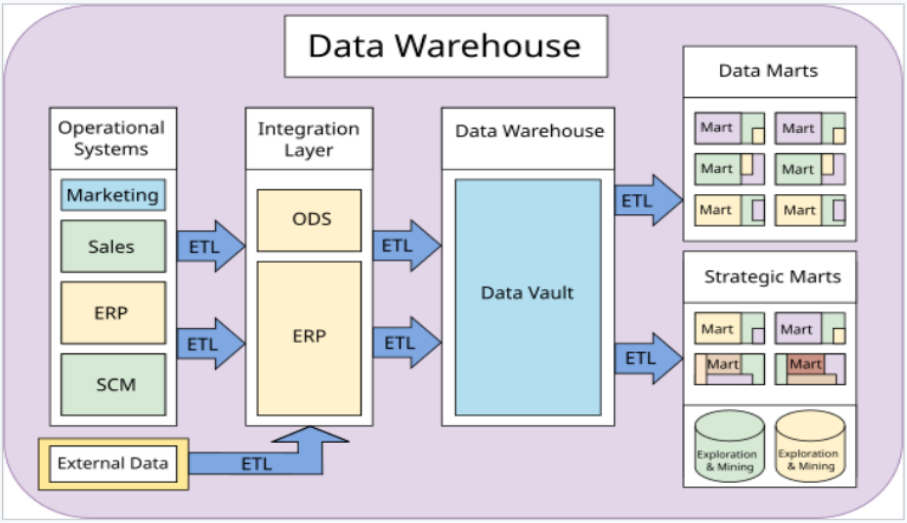
Analogy:
If a data warehouse is a full shopping mall, a data mart is a single store inside it.



---

7. Data Marts vs Data Warehouses

Feature	Data Mart	Data Warehouse
Scope	Department-level	Enterprise-wide
Size	Smaller	Larger
Data Sources	Few or one	Many diverse sources
Implementation	Faster, cost-effective	Time-consuming, expensive
Example	HR payroll analysis	Company-wide employee analytics



---

8. Data Warehouse Lifecycle

Lifecycle Phases:

1. **Requirement Gathering**
    - Identify business objectives, key performance indicators (KPIs), and user expectations.
2. **Design**
    - Logical design (schemas: star/snowflake).
    - Physical design (indexes, partitions).
3. **ETL Development**
    - Extract data from various sources.
    - Transform it into consistent formats.
    - Load it into DW.
4. **Testing**
    - Ensure data accuracy.
    - Validate ETL process and schema mappings.
5. **Deployment**
    - Make DW accessible to business users via BI tools.
6. **Maintenance**
    - Regular updates, bug fixes, and performance optimization.