Delta-exam-topics-ETL

1. Why Transformation is Needed?

🔹 Data from different sources (databases, APIs, files) comes in different formats and structures.

🔹 We **transform** data to make it **consistent, clean, and usable** for analysis.

📌 **Example:**

- A sales database stores `Date` as `YYYY-MM-DD`, but another system uses `DD/MM/YYYY`.
- To analyze data properly, we need to **standardize the date format** during transformation.

2. ETL vs. ELT

🔹 **ETL (Extract → Transform → Load)**

- Data is **transformed first**, then loaded into the warehouse.
- Best for **structured** data.

🔹 **ELT (Extract → Load → Transform)**

- Data is **loaded first**, then transformed inside the warehouse.
- Best for **Big Data & Cloud** environments.

Feature	ETL	ELT
Processing	Before loading	After loading
Storage	Smaller data	Requires large storage
Speed	Slower	Faster
Best for	Traditional Data Warehouses	Big Data, Cloud

📌 **Example:**

- A **bank** may use **ETL** to clean and load financial data.
- A **cloud-based system** (like Google BigQuery) may use **ELT** to store raw data first and process it later.

---

3. ELT Procedures

🔹 **ELT Procedures** define the steps for:
✅ **Extracting** data from multiple sources
✅ **Loading** data into the warehouse
✅ **Transforming** it inside the warehouse

📌 **Example Steps:**
1️⃣ **Extract** sales data from an e-commerce website and a CRM system.
2️⃣ **Load** raw data into a cloud data warehouse (like Snowflake).
3️⃣ **Transform** the data to standardize customer names, dates, and sales formats.

4. SCDs (Slowly Changing Dimensions)

🔹 **SCDs** handle changes in dimension data (e.g., customer address, job title).

📌 **Example:**

- A customer **moves to a new city** or **gets promoted at work**.
- How do we store the old and new values in a data warehouse?

5. Types of SCDs

Slowly Changing Dimensions (SCDs) are techniques used in data warehousing to manage changes in dimension data over time. They are categorized into different types (SCD 0 to SCD 6), based on how changes are tracked and stored. Let's break them down with simple explanations and examples.

SCD Type 0 (Retain Original) – No Change

👉 **No changes are allowed in the dimension table.** Data remains the same even if the source system updates it.

🔹 **Example:** A product table where historical prices must never change, even if the company updates them.

Product_ID	Product_Name	Price
101	Laptop	50000
102	Phone	20000

🔹 If the price of "Laptop" changes to 55000, it **won't** be updated in the dimension table.

SCD Type 1 (Overwrite) – Keep Only Latest Data

👉 **Old data is replaced with new data, and history is lost.**

🔹 **Example:** A customer table where only the latest address is stored.

Customer_ID	Name	Address
201	John	Pune

- If John moves to Mumbai, the table updates:

Customer_ID	Name	Address
201	John	Mumbai

Old address is lost.

SCD Type 2 (Versioning) – Maintain History with New Rows

👉 **A new row is added for each change, maintaining historical records.**
👉 **Usually, start and end dates or version numbers are used.**

- **Example:** Employee salary history

Emp_ID	Name	Salary	Start_Date	End_Date	Is_Current
301	Alice	50000	2023-01-01	2024-01-01	No
301	Alice	55000	2024-01-02	NULL	Yes

- If Alice's salary increases to 60000, a new row is added with updated dates.

SCD Type 3 (Add Column) – Keep Only One Previous Value

👉 **Stores the previous value in a separate column, but doesn't keep full history.**

- **Example:** Product price changes

Product_ID	Name	Current_Price	Previous_Price
401	TV	40000	35000

- If the price updates to 45000, the table updates:

Product_ID	Name	Current_Price	Previous_Price
401	TV	45000	40000

Only the last change is kept.

SCD Type 4 (Separate History Table) – Archive Old Data

👉 **Current data stays in the main table, and historical data is moved to a separate history table.**

🔹 **Example:**

Main Table (Current Data):

Customer_ID	Name	Address
501	Sam	Mumbai

History Table (Old Data):

Customer_ID	Name	Old_Address	Changed_Date
501	Sam	Pune	2024-02-01

If Sam moves again, the old address moves to the history table.

SCD Type 5 (Hybrid – Type 1 + Type 4)

👉 **Uses both Type 1 (overwrite current data) and Type 4 (history table).**
👉 **Adds a surrogate key to track changes.**

🔹 **Example:**

Current Table (Type 1 Overwrite):

Customer_SK	Customer_ID	Name	Address
1001	601	Raj	Delhi

History Table (Type 4 Archive):

Customer_SK	Customer_ID	Name	Old_Address	Changed_Date
1000	601	Raj	Jaipur	2024-02-15

SCD Type 6 (Hybrid – Type 1 + Type 2 + Type 3)

👉 **Combines Type 1 (overwrite), Type 2 (new row for changes), and Type 3 (store previous value).**

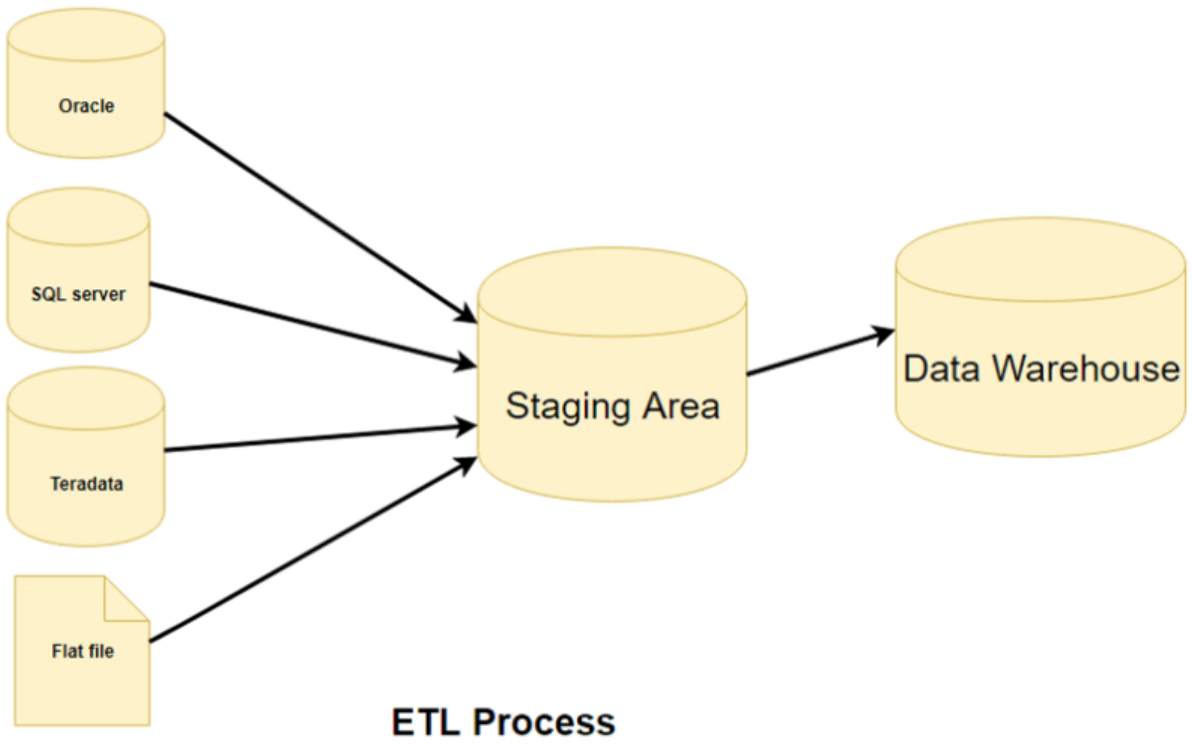
🔹 **Example:**

Emp_ID	Name	Salary	Previous_Salary	Start_Date	End_Date	Is_Current
701	Mike	70000	NULL	2023-01-01	2024-02-01	No
701	Mike	75000	70000	2024-02-02	NULL	Yes

🔹 If Mike gets a raise, a new row is added (Type 2), the current row is updated (Type 1), and the old salary is stored in a separate column (Type 3).

6. Data cleaning steps in ETL

7. What is ETL



ETL (Extract, Transform, Load) is a process used in data warehousing to collect data from multiple sources, clean and transform it, and then load it into a data warehouse for analysis. Here's a breakdown of each step:

Step 1: Extraction

- Extracts data from multiple sources like databases, legacy systems, flat files, ERP systems, etc.
- Uses a **staging area** to avoid impacting source system performance and allow validation before loading.
- Data mapping is done to define the relationship between source and target.

Extraction Methods:

1. **Full Extraction** – Extracts all data each time.
2. **Partial Extraction (without update notification)** – Extracts only changed data, but changes are identified manually.
3. **Partial Extraction (with update notification)** – Extracts only changed data using timestamps or database triggers.

Validations during Extraction:

- Ensure data integrity and remove duplicates.
- Validate data types and formats.
- Filter out irrelevant or incorrect data.

Step 2: Transformation

- Converts raw extracted data into a usable format by applying business rules, cleansing, and data integration.
- Includes processes like:
  - **Data Cleansing** – Removing duplicates, handling missing values.
  - **Data Mapping** – Aligning data structures from different sources.
  - **Aggregation** – Summarizing data (e.g., calculating total sales).
  - **Joining & Splitting** – Merging data from multiple sources or splitting complex fields.

Example: Combining `First Name` and `Last Name` into a single `Full Name` field.

Step 3: Loading

- Moves transformed data into the **data warehouse** for analytics and reporting.
- Needs to be optimized for performance, ensuring integrity and consistency.

Types of Loading:

1. **Initial Load** – First-time full data population.
2. **Incremental Load** – Only updates changed or new data.
3. **Full Refresh** – Deletes existing data and reloads from scratch.

Load Verification:

- Ensure primary keys and relationships are correct.
- Validate BI reports and ensure aggregated values are accurate.
- Check historical data updates in slowly changing dimensions (SCD).

ETL plays a crucial role in Business Intelligence (BI) and decision-making by ensuring high-quality, structured data is available for analysis. 🚀