Data Modeling

Data modelling is the process of creating a blueprint of how data is stored, connected, and retrieved in a system.

Data modeling is the process of designing a visual and logical representation of how data is stored, organized, and related in a system—usually before building databases, data warehouses, or analytics solutions.

Key Points

- **Purpose**: To define what data is needed, how it is structured, and how different data entities relate to each other.
- **Outcome**: A blueprint (diagram + rules) that database engineers and analysts use to implement efficient, consistent, and accurate data storage and retrieval.

Components

1. **Entities / Tables**
   Real-world objects or concepts to store (e.g., *Customer*, *Product*, *Order*).
2. **Attributes / Columns**
   Properties of each entity (e.g., *Customer_Name*, *Order_Date*).
3. **Relationships**
   How entities connect (e.g., one customer places many orders).

Common Modeling Patterns

- **Relational (OLTP)**: Normalized tables for transactional systems.
- **Dimensional (OLAP)**: Star or snowflake schemas for analytics.
  - **Star schema**: Fact table at center (e.g., Sales) linked to dimension tables (e.g., Date, Product).
  - **Snowflake schema**: Dimensions further normalized into sub-dimensions.

(Data modeling is like creating an architectural plan for a database or data warehouse, ensuring that data is stored efficiently and relationships are well defined before actual implementation).

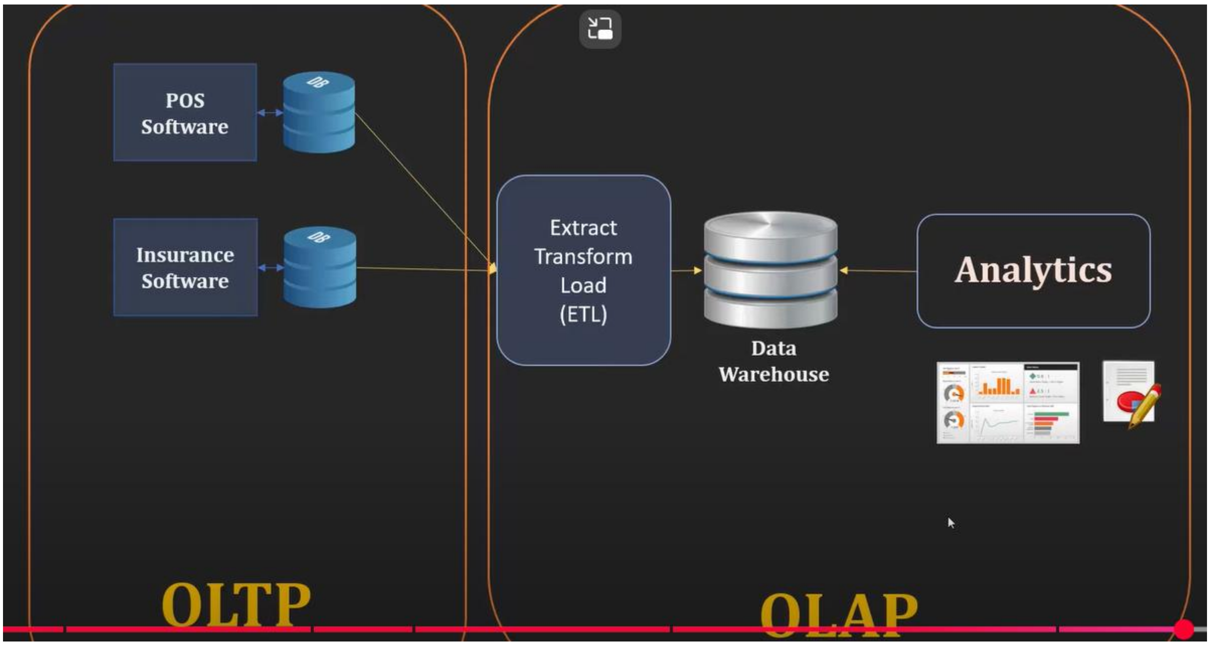
Data modelling is a critical step in designing a database as it provides a structured framework for organizing, storing, and accessing data efficiently. It serves as a blueprint that defines the data elements, their relationships, and the rules governing them, ensuring clarity and consistency throughout the database lifecycle.

As a Data Engineer, it’s not just about pipelines and tools – you need to design data structures that support reporting, scalability and performance.

Makes data easier to understand and use.

Improves query performance

Helps build scalable system

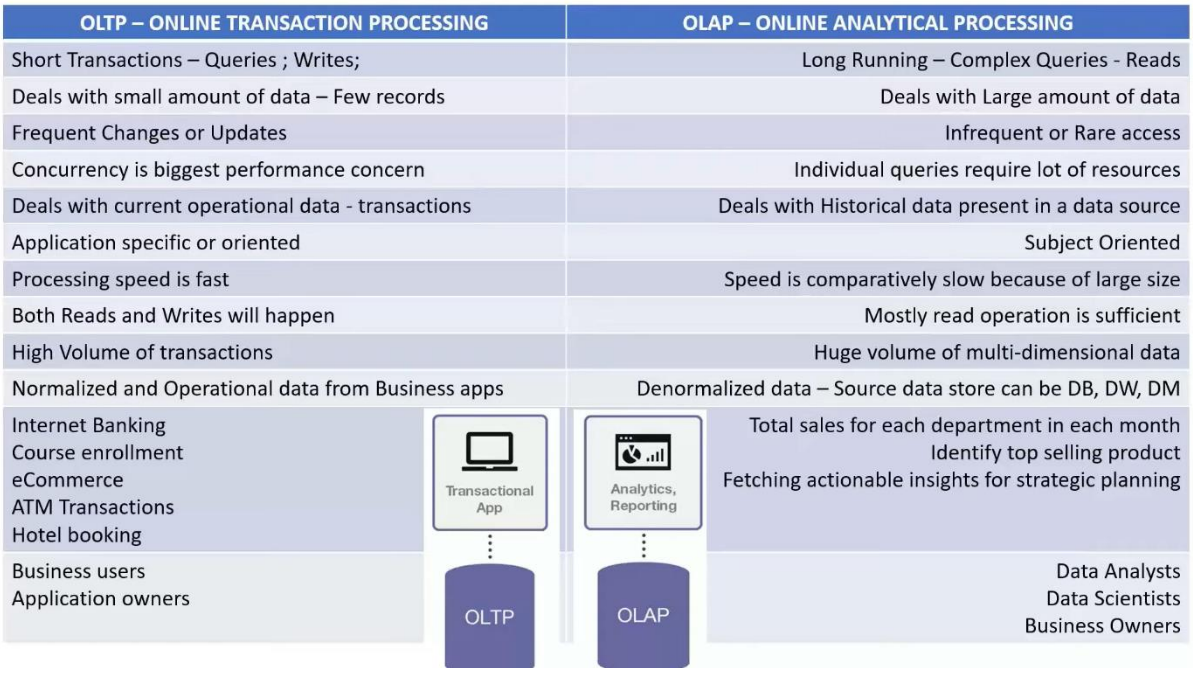


OLTP – Online Transactional Processing

- Handle day-to-day operations and fast inserts/updates.
- **Data type:** Current, detailed, transactional.
- **Users:** Application users, front-end systems, customer-facing apps.
- **Workload:** Lots of short, frequent transactions (e.g., placing an order, updating inventory).
- **Schema style:** Highly **normalized** relational schema (3NF) to avoid redundancy and ensure integrity.
- **Examples:**
  - Banking system processing deposits/withdrawals
  - E-commerce order management
  - Ride-hailing app backend

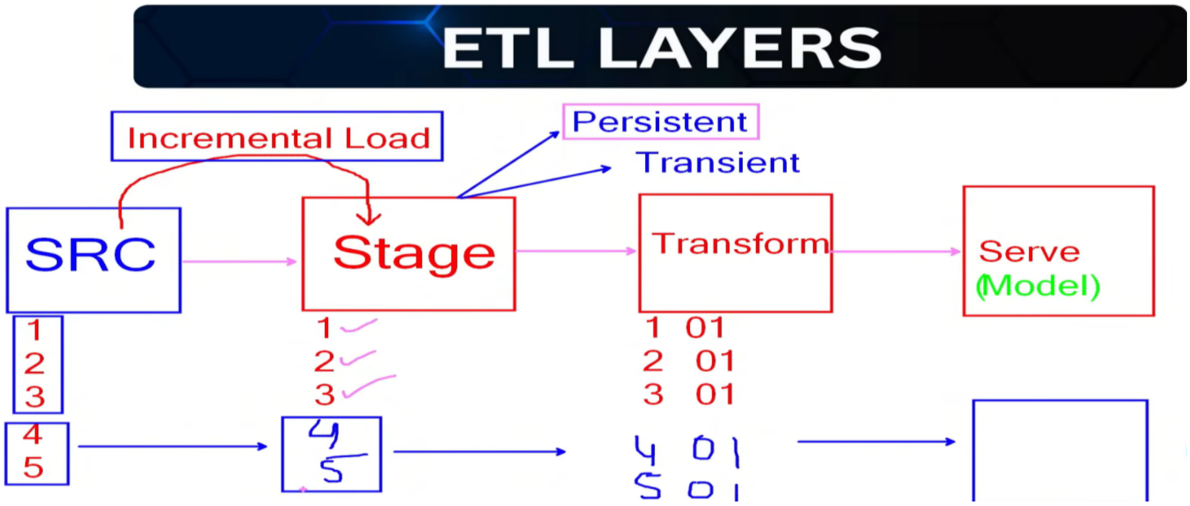
OLAP – Online Analytical Processing

- Support complex queries for analysis, reporting, and business intelligence.
- **Data type:** Historical, aggregated, read-heavy.
- **Users:** Data analysts, business intelligence tools, data scientists.
- **Workload:** Fewer transactions but long, complex queries (e.g., “total sales by region over 5 years”).
- **Schema style: Star or Snowflake** schemas, often denormalized, stored in a **data warehouse**.
- **Examples:**
  - Sales performance dashboards
  - Market trend analysis
  - Executive reporting systems



OLTP – ONLINE TRANSACTION PROCESSING	OLAP – ONLINE ANALYTICAL PROCESSING
Short Transactions – Queries ; Writes;	Long Running – Complex Queries - Reads
Deals with small amount of data – Few records	Deals with Large amount of data
Frequent Changes or Updates	Infrequent or Rare access
Concurrency is biggest performance concern	Individual queries require lot of resources
Deals with current operational data - transactions	Deals with Historical data present in a data source
Application specific or oriented	Subject Oriented
Processing speed is fast	Speed is comparatively slow because of large size
Both Reads and Writes will happen	Mostly read operation is sufficient
High Volume of transactions	Huge volume of multi-dimensional data
Normalized and Operational data from Business apps	Denormalized data – Source data store can be DB, DW, DM
Internet Banking Course enrollment eCommerce ATM Transactions Hotel booking	Total sales for each department in each month Identify top selling product Fetching actionable insights for strategic planning
Business users Application owners	Data Analysts Data Scientists Business Owners

ETL Layer



Persistent – if new data comes it will keep on append with old data

Transient - every new data comes it will overwrite (old data is deleted and new data is overwritten)

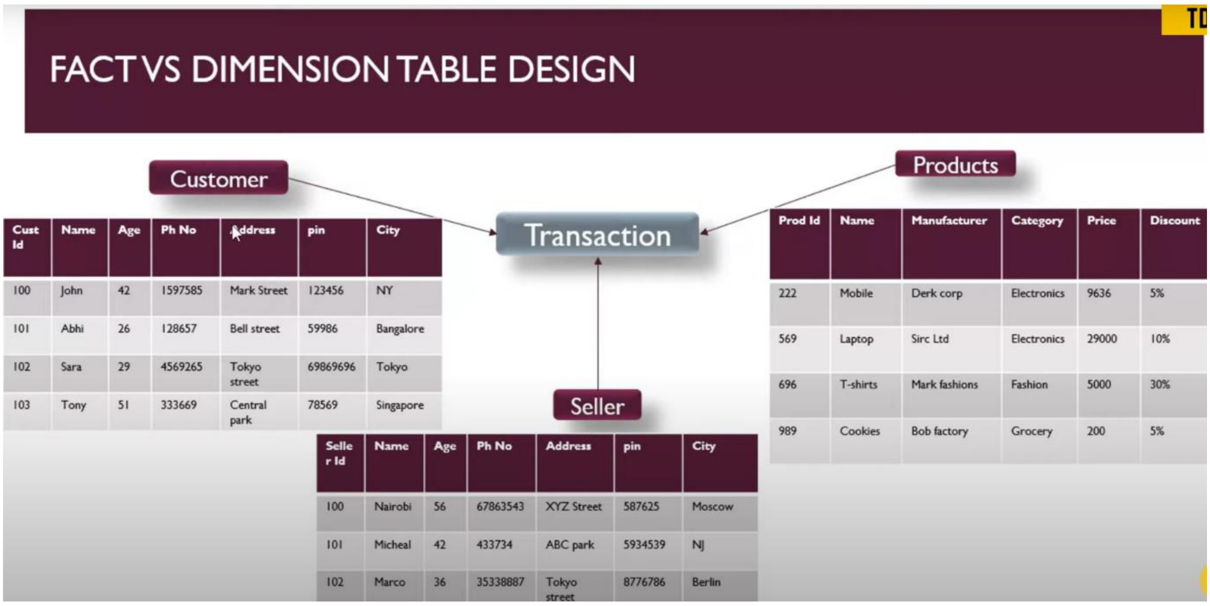
Fact table

Fact table is a table that stores the measurements, metrics, or facts related to a business operation.

It is located at the center of a star or snowflake schema and is surrounded by dimension tables.

- A fact table has two types of columns: those that contain the facts and those that serve as foreign key linking to dimension tables.

- The primary key of a fact table is often a composite key made up of all of the foreign keys in the table.
- Fact tables can hold various types of measurements, such as additive, non-additive, and partly additive measures, and store important information in the data warehouse.



FACT VS DIMENSION TABLE DESIGN

Customer

Cust Id	Name	Age	Ph No	Address	pin	City
100	John	42	1597585	Mark Street	123456	NY
101	Abhi	26	128657	Bell street	59986	Bangalore
102	Sara	29	4569265	Tokyo street	69869696	Tokyo
103	Tony	51	333669	Central park	78569	Singapore

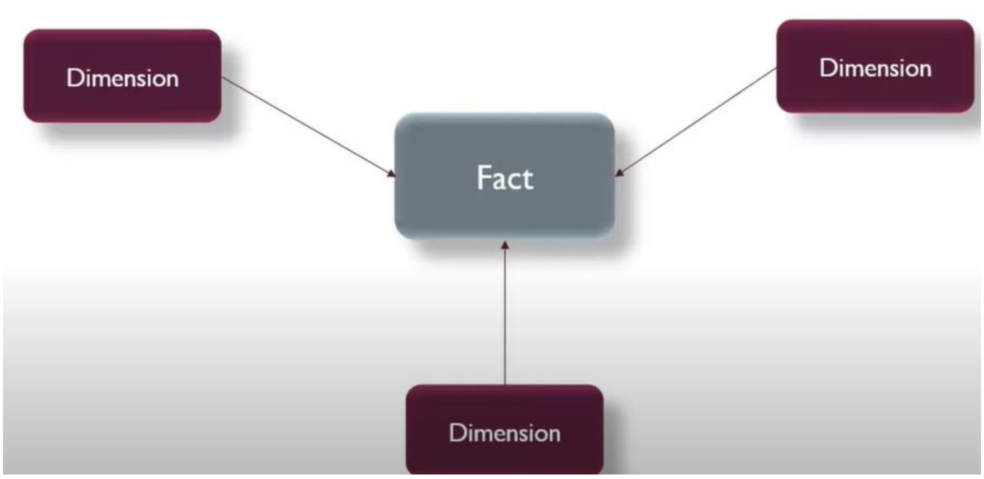
Transaction

Products

Prod Id	Name	Manufacturer	Category	Price	Discount
222	Mobile	Derk corp	Electronics	9636	5%
569	Laptop	Sirc Ltd	Electronics	29000	10%
696	T-shirts	Mark fashions	Fashion	5000	30%
989	Cookies	Bob factory	Grocery	200	5%

Seller

Seller Id	Name	Age	Ph No	Address	pin	City
100	Nairobi	56	67863543	XYZ Street	587625	Moscow
101	Micheal	42	433734	ABC park	5934539	NJ
102	Marco	36	35338887	Tokyo street	8776786	Berlin

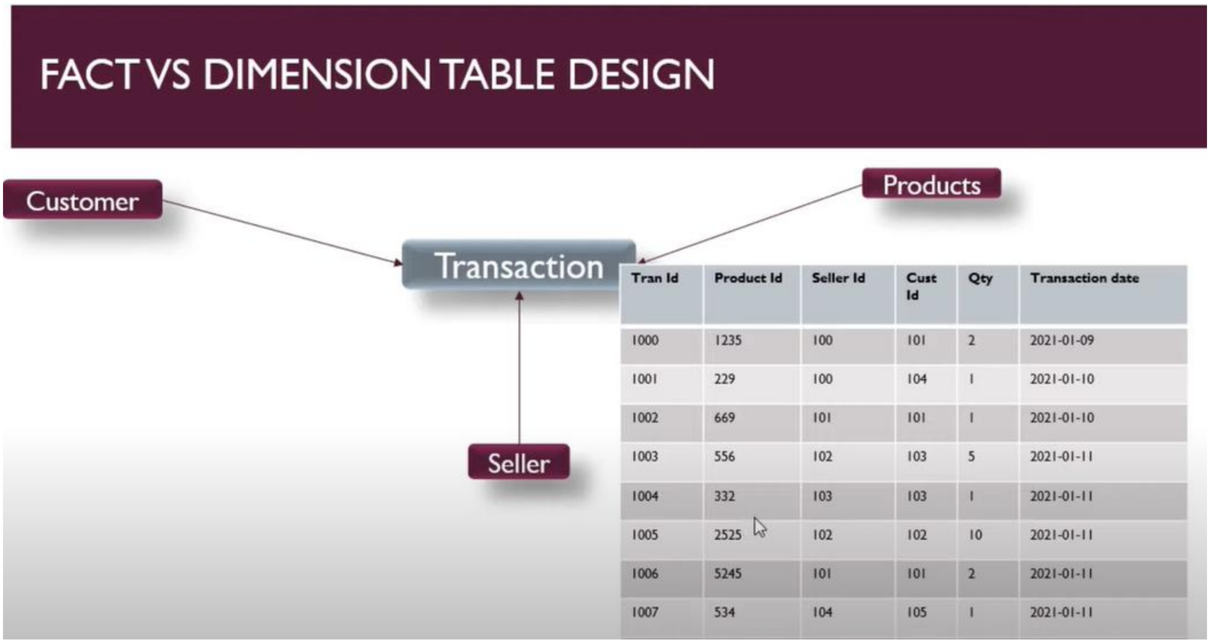


Dimension

Dimension

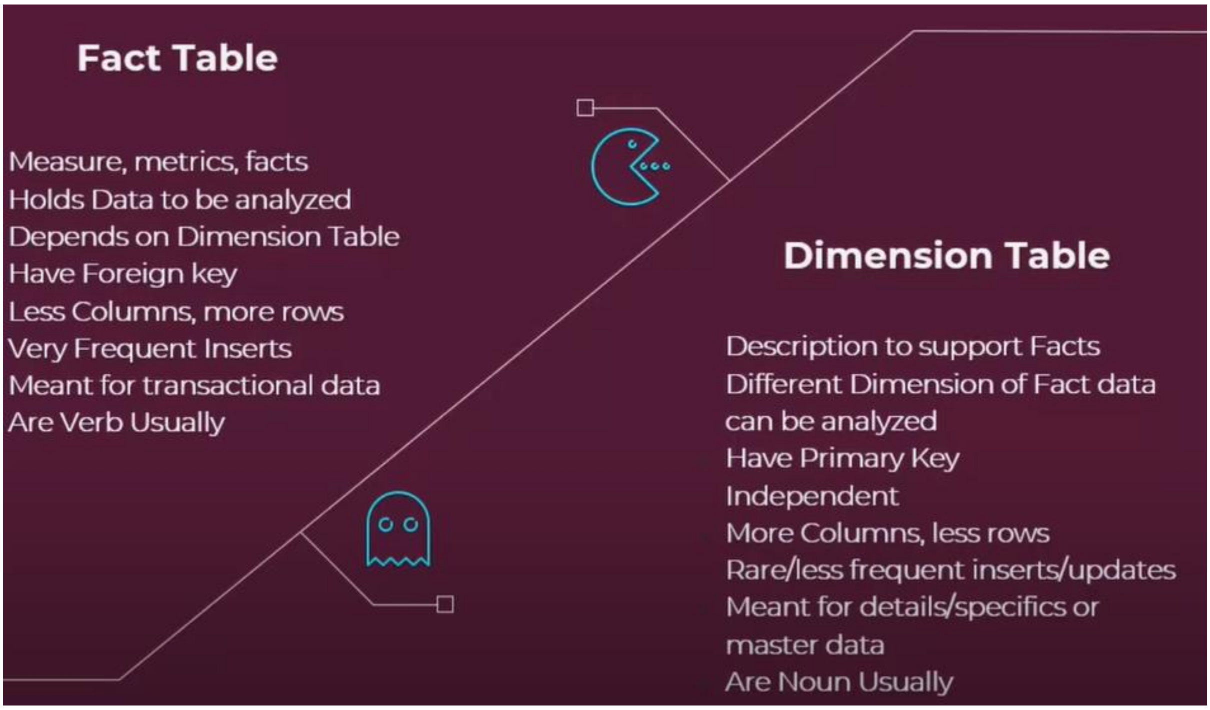
Fact

Dimension



FACT VS DIMENSION TABLE DESIGN

Tran Id	Product Id	Seller Id	Cust Id	Qty	Transaction date
1000	1235	100	101	2	2021-01-09
1001	229	100	104	1	2021-01-10
1002	669	101	101	1	2021-01-10
1003	556	102	103	5	2021-01-11
1004	332	103	103	1	2021-01-11
1005	2525	102	102	10	2021-01-11
1006	5245	101	101	2	2021-01-11
1007	534	104	105	1	2021-01-11



Fact Table

Measure, metrics, facts
Holds Data to be analyzed
Depends on Dimension Table
Have Foreign key
Less Columns, more rows
Very Frequent Inserts
Meant for transactional data
Are Verb Usually

Dimension Table

Description to support Facts
Different Dimension of Fact data can be analyzed
Have Primary Key
Independent
More Columns, less rows
Rare/less frequent inserts/updates
Meant for details/specifics or master data
Are Noun Usually

Star Schema

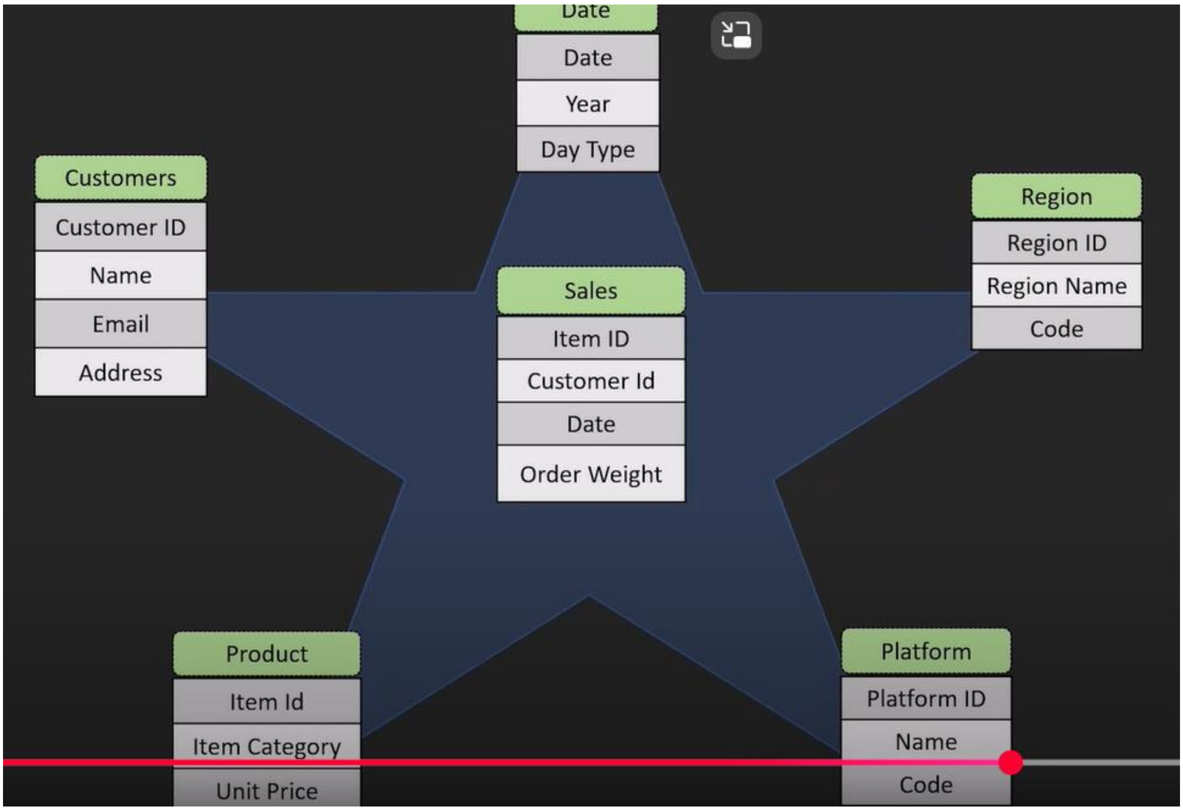
Think of a star:

- **Center (Fact table):** Holds measurable, numeric business data (“facts”)—for example, sales_amount, order_count, quantity.
- **Points (Dimension tables):** Surrounding tables that describe the facts—like date_dim, customer_dim, product_dim, store_dim.

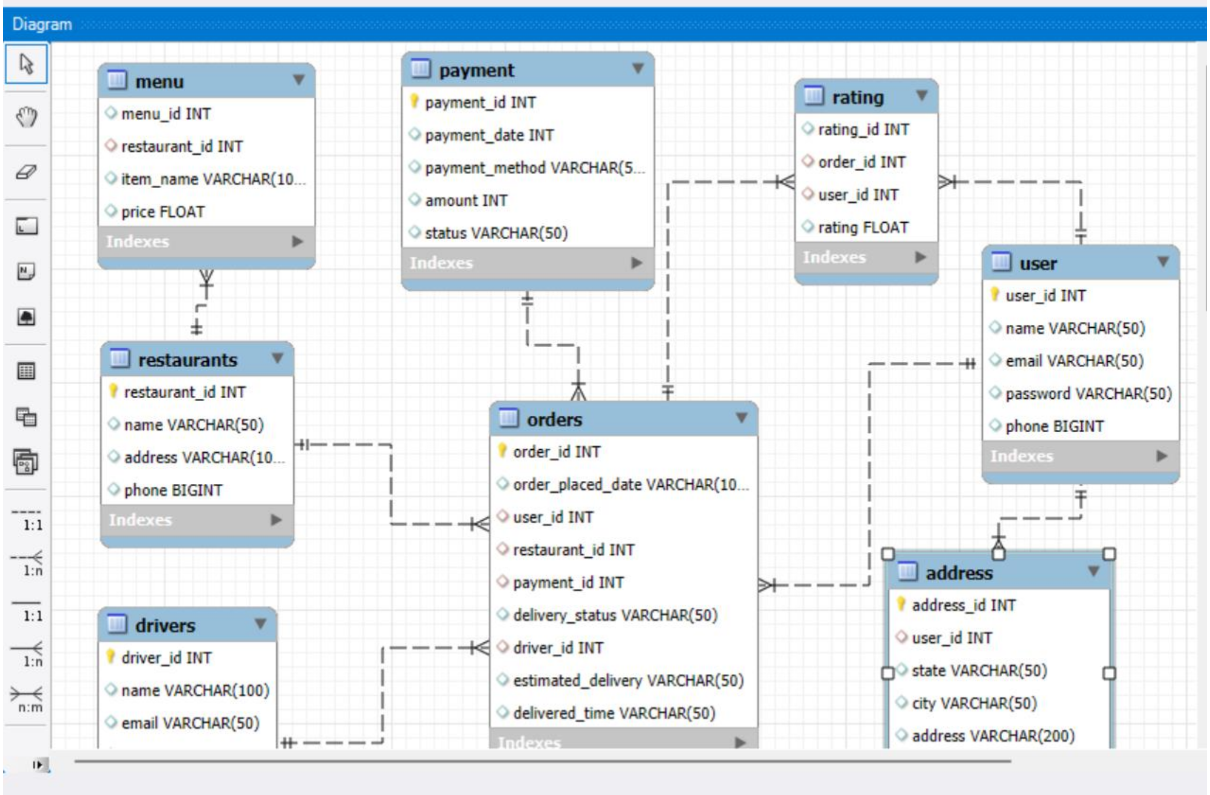
Characteristics

- Dimensions are **denormalized** (all descriptive columns stored in one wide table).

- Simple joins: fact ↔ dimension.
- Best for fast query performance and easy understanding.



In mysql (Zomato dataset)



https://youtu.be/w-0IWyAeZ3M?si=di_WQPvI20De2X16

Airbnb Data Warehouse Schema - Data Engineering Mock Interview

Snowflake Schema

A snowflake has branches.

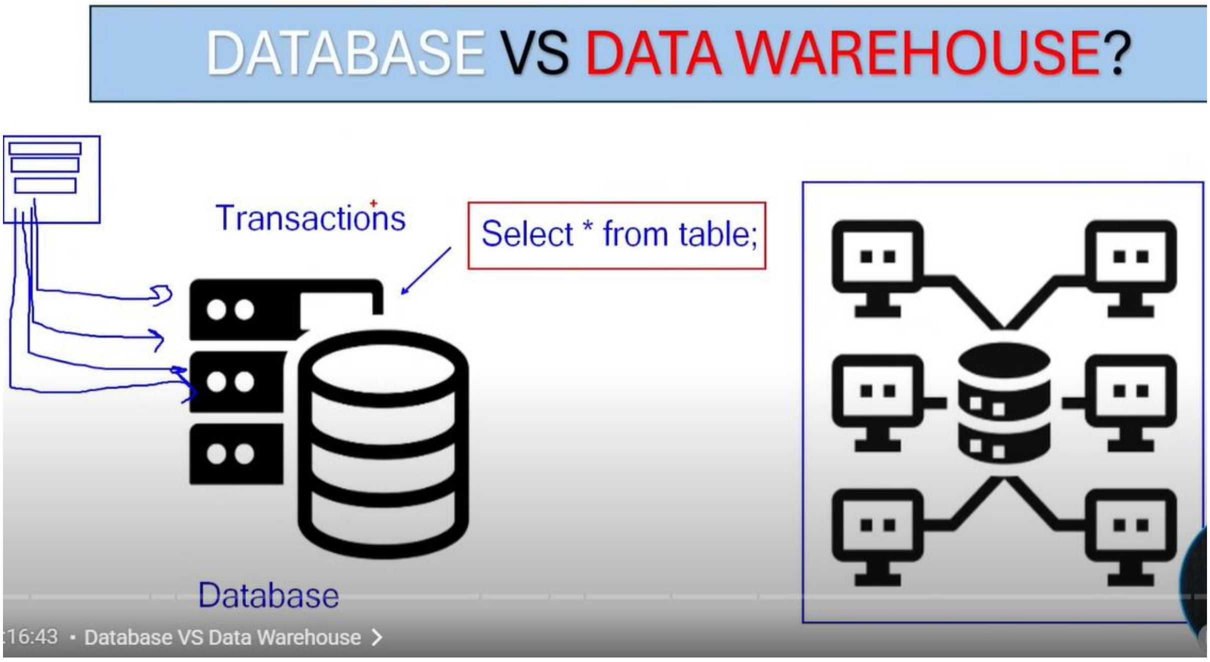
- Starts like a star but **dimension tables are normalized** into multiple related tables.
- For example, instead of a single product_dim, you might have:
  - product
  - product_category
  - product_subcategory

Characteristics

- Reduces data redundancy and storage.
- More complex joins; slightly slower queries.
- Useful when dimensions are very large or change frequently.

Date warehouse

Central location to store every single data.

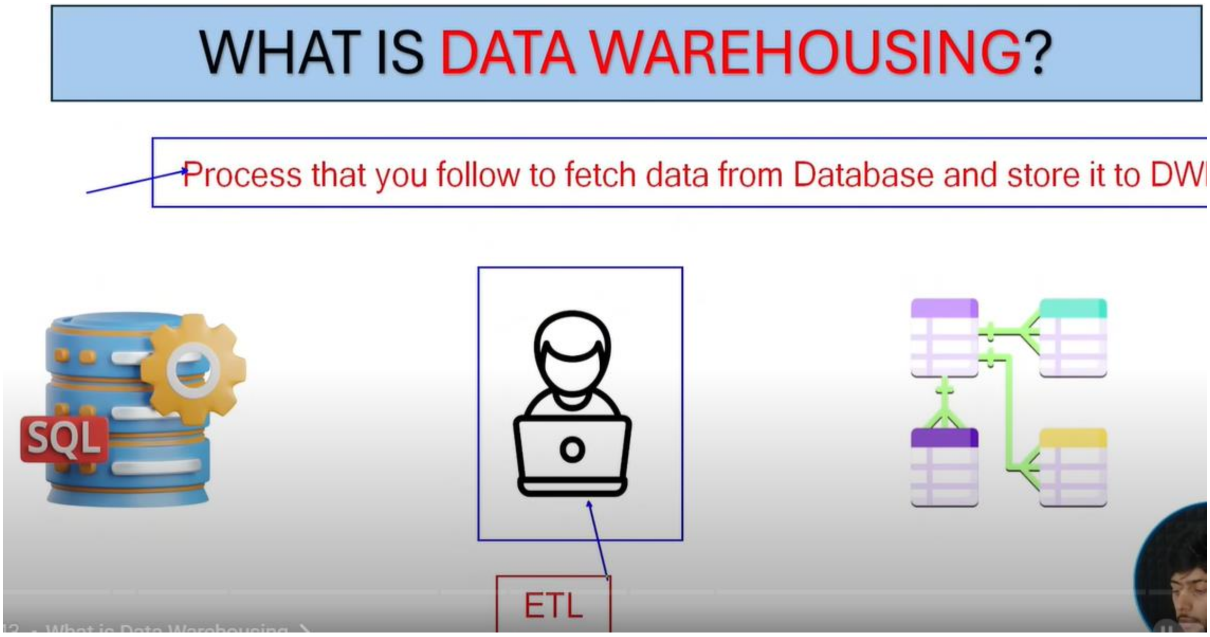

DATABASE VS DATA WAREHOUSE?
Transactions
Select * from table;
Database
16:43 • Database VS Data Warehouse >

Datebase – is continuously receive data from the user, so we are not supposed to disturb the database.

Datawarehouse – from here we can pick the data for the ETL process (for the analysis things).

What is Data warehousing?

Process that you follow to fetch data from Database and store it to Data warehouse.

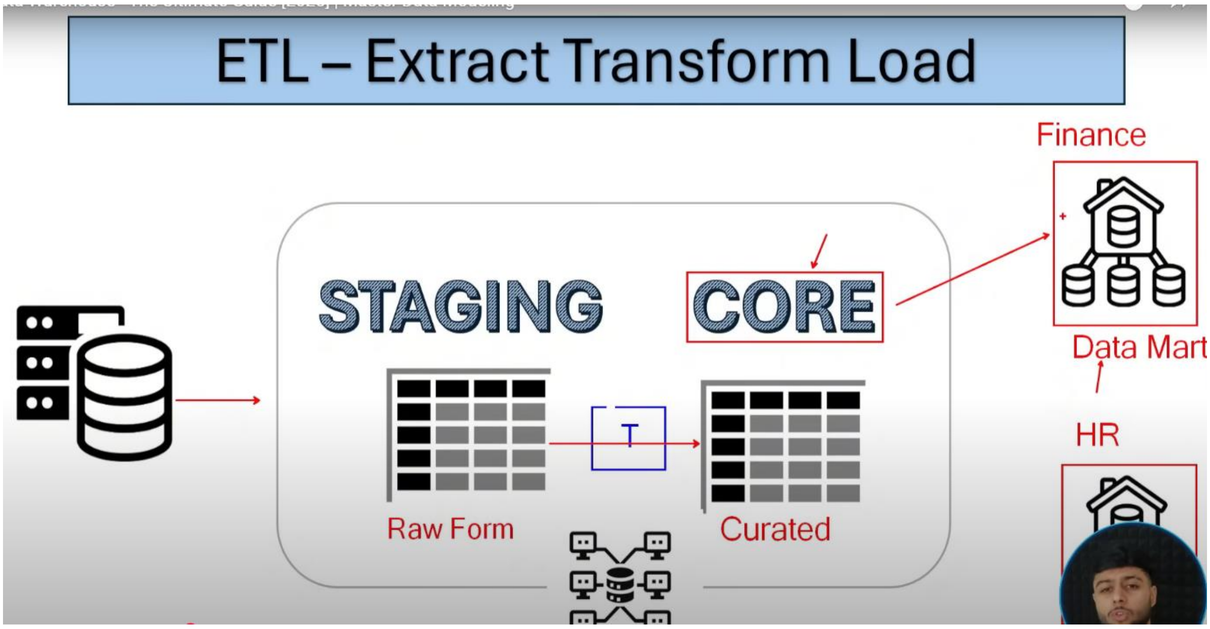


WHAT IS DATA WAREHOUSING?

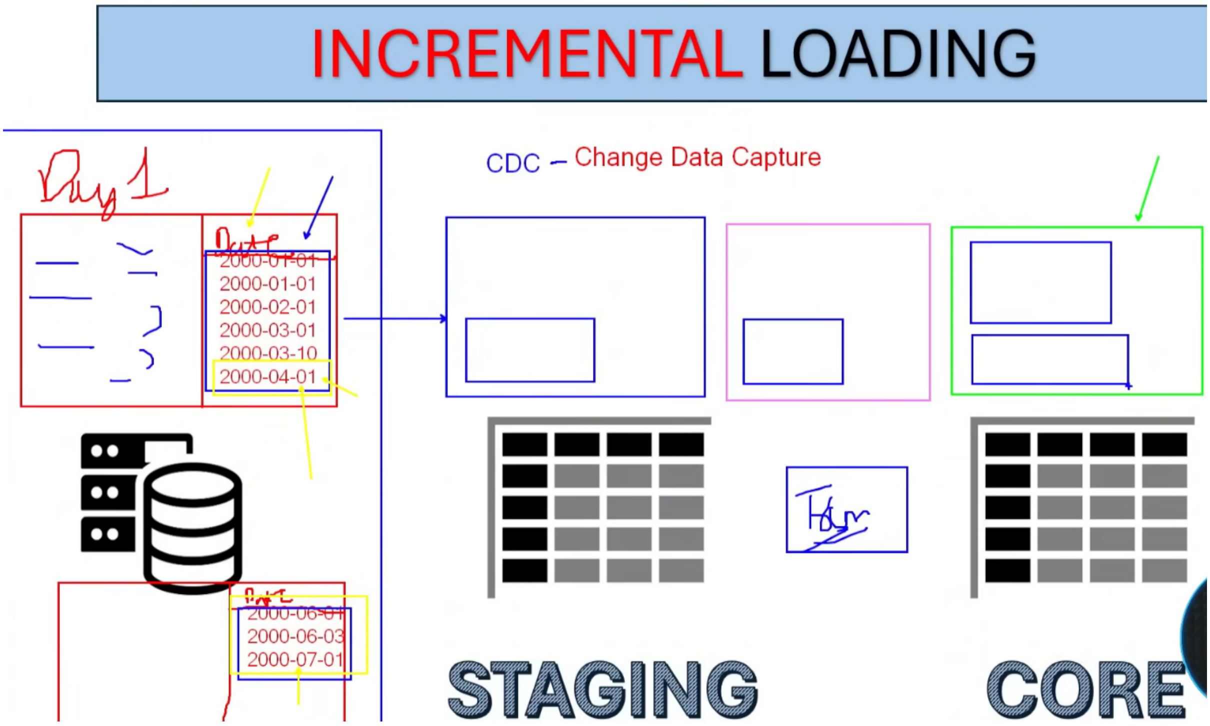
Process that you follow to fetch data from Database and store it to DW

Inside data warehouse – Staging area (here raw form of data is stored) → Transformation → Core area (from here data is given to the analysis)

Data Mart – is sub part of Data warehouse (is a single unit or department inside warehouse)



ETL – Extract Transform Load



INCREMENTAL LOADING

CDC – Change Data Capture

2000-01-01
2000-01-01
2000-02-01
2000-03-01
2000-03-10
2000-04-01

2000-06-01
2000-06-03
2000-07-01

STAGING

CORE