

Extract – Load – transform

Elt – is more flexible and faster since the data is already in our db

Kimball Dimensional Modelling is a methodology for designing data warehouses, developed by **Ralph Kimball**. It focuses on making data **easier to access and analyse** by structuring it into a **star schema** or **snowflake schema**. The approach is widely used for **business intelligence (BI) and reporting**.

The objective is to deliver understandable data, and deliver fast query performance.

Prioritize user understandability and query over non redundant data 3f

Other approaches – bill Inmon, Data Vault

**Key Concepts of Kimball Dimensional Modeling:**

1. **Star Schema** – The primary structure of Kimball modeling, where a **central fact table** is linked to multiple **dimension tables**.
2. **Fact Table** – Stores **measurable business events** (e.g., sales, revenue, transactions).
3. **Dimension Tables** – Contain **descriptive attributes** (e.g., customer details, product categories, time periods).
4. **Denormalization** – Data redundancy is **accepted** to improve **query performance**.
5. **Surrogate Keys** – Unique keys in dimension tables that replace natural keys for consistency.
6. **Slowly Changing Dimensions (SCDs)** – A method to handle **historical changes** in dimension data.
7. **Grain** – The level of detail stored in a fact table (e.g., daily sales vs. monthly sales).
8. **Conformed Dimensions** – Shared dimensions across multiple fact tables for **data consistency**.

**Advantages of Kimball Dimensional Modeling:**

✅ **Optimized for Query Performance** – Faster and easier retrieval of data.  
✅ **Simplified for End Users** – Business analysts can understand and use the model easily.  
✅ **Historical Data Tracking** – Effective handling of changes in business data over time.  
✅ **Scalability** – Can be expanded by adding new fact tables and conformed dimensions.

**Example: Sales Data Warehouse**

* **Fact Table (Sales\_Fact)**: Contains sales amount, quantity sold, and keys to dimensions.
* **Dimension Tables**:
  + **Customer\_Dim** (Customer ID, Name, Location)
  + **Product\_Dim** (Product ID, Name, Category)
  + **Time\_Dim** (Date, Month, Year)

What is dbt (Data Build Tool)?

dbt (Data Build Tool) is an open-source data transformation tool that helps analysts and engineers write, test, and manage SQL-based transformations inside a data warehouse. It is widely used for modern data engineering in cloud-based environments like Snowflake, BigQuery, Redshift, and Databricks.

Why Use dbt?

✅ Modular SQL Development – Allows you to write reusable SQL models instead of complex ETL scripts.

✅ Version Control – Works with Git, making it easy to track changes.

✅ Automated Testing – Built-in data quality tests for validating transformations.

✅ Dependency Management – Handles model dependencies and ensures execution in the right order.

✅ Documentation & Lineage – Generates data lineage graphs and detailed documentation.

**How Does dbt Work?**

1. **Extract & Load Data (ELT)** – Data is loaded into a **cloud data warehouse** (e.g., Snowflake, BigQuery).
2. **Transform Data with dbt** – You write **SQL models** to clean, aggregate, and prepare data.
3. **Run & Test Transformations** – dbt compiles SQL models, executes them, and validates data quality.
4. **Deploy & Monitor** – Use dbt Cloud or a scheduler (like Airflow) for production workflows.
5. **Show analytics** – Use looker, tableau, Sisense, power bi