|  |  |
| --- | --- |
| ***Roll No*** | ***22SW040 --> Section\_01*** |
| ***Subject*** | ***DWH*** |
| ***Self Study Assignment*** | ***5 Marks*** |
| ***Teacher*** | ***Sir Shehram*** |

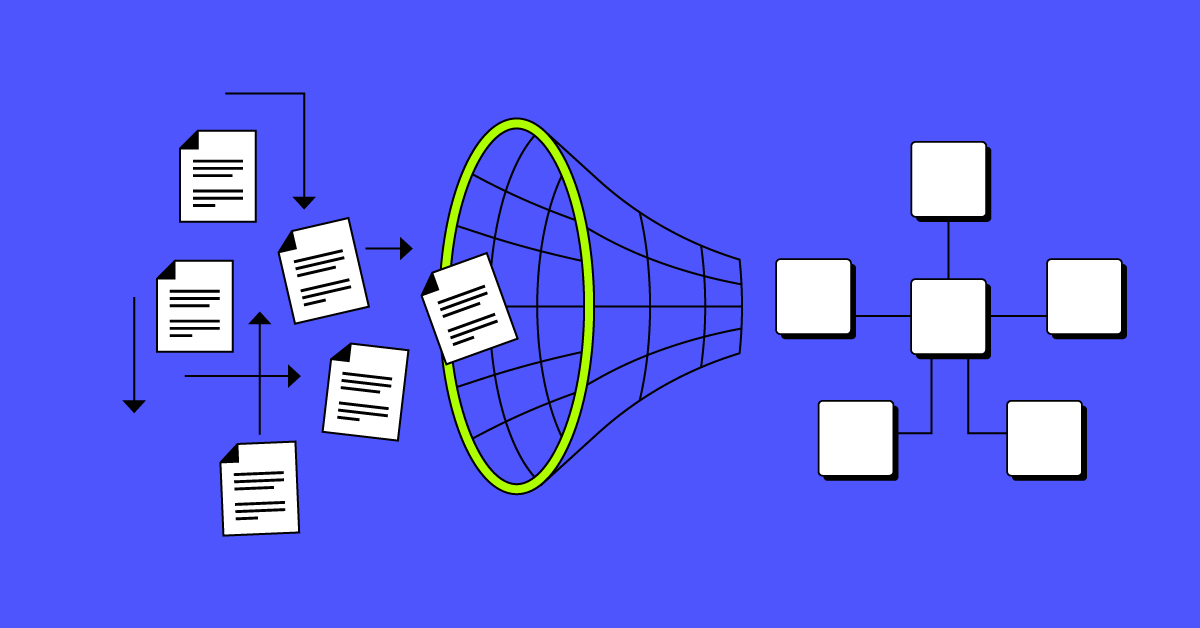


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Dimensional Modeling***

Dimensional modeling is a logical design technique used to structure the business dimensions and metrics that are analyzed in a data warehouse. It is primarily query-centric and focuses on simplifying the organization of data for easier retrieval and analysis.

**Ralph Kimball** developed the concept of dimensional modeling, which contrasts with entity-relationship modeling used in transactional systems.



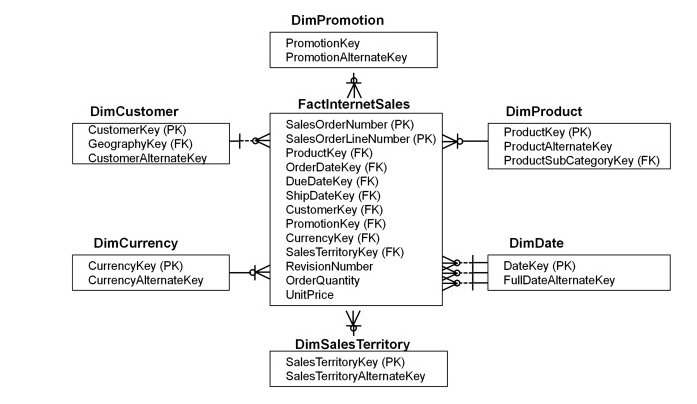
**Key features include**

**Facts:** These are the numeric values representing measurable business processes.

**Dimensions:** These are descriptive attributes related to the facts, allowing users to analyze facts from various perspectives, like **time, location, or product** categories.

**Purpose:** To facilitate high performance for queries, often optimized for decision-support systems (e.g., **sales, finance**, etc.).

**Example**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

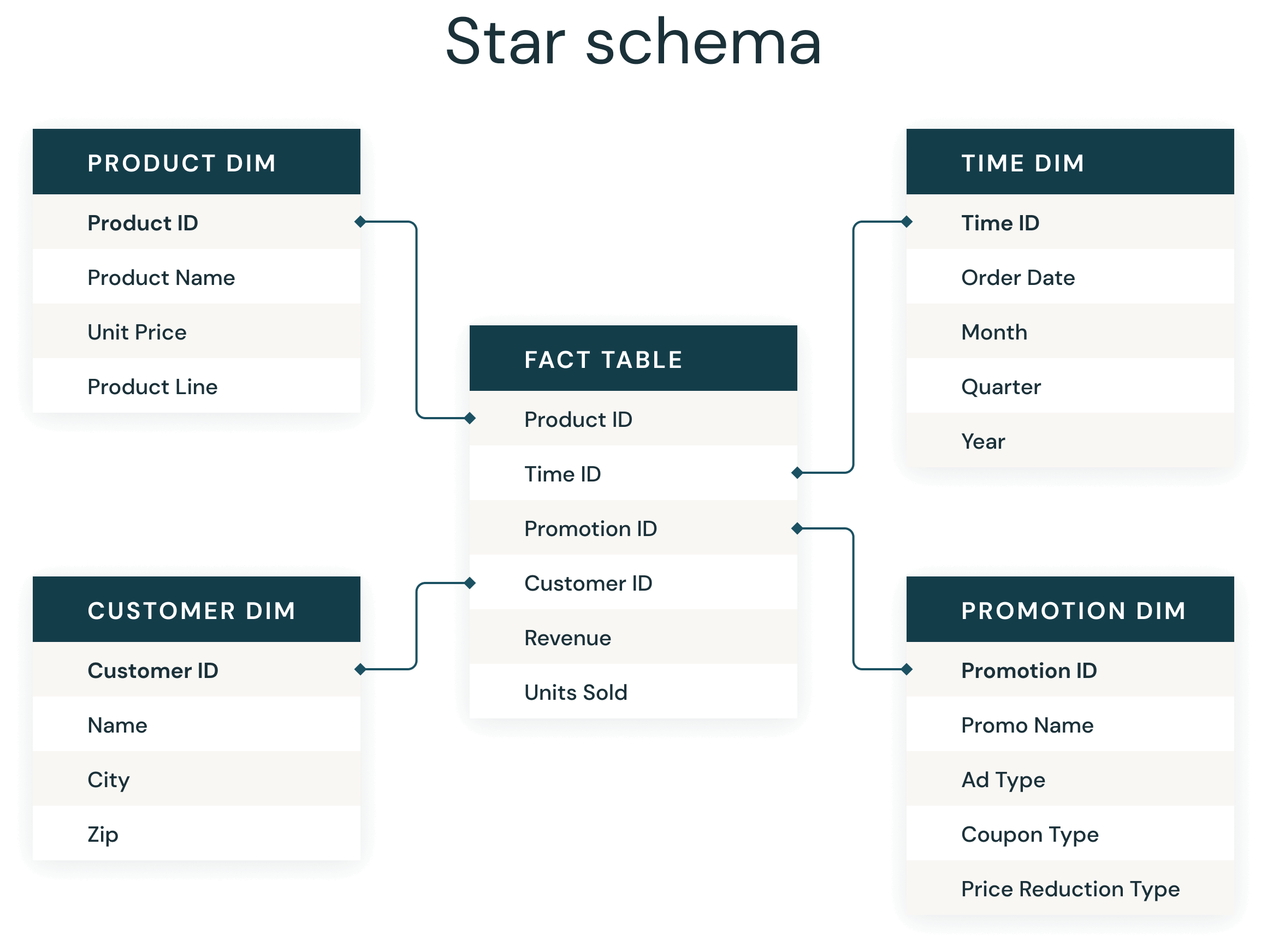
***Star Schema (with example)***

The **Star Schema** is a widely used and well-established form of dimensional modelling that organizes data into two primary categories one is **facts** and the other is dimensions. This schema is structured with a central **fact table** that contains quantitative data, such as sales figures or performance metrics. Surrounding this fact table are several **dimension tables** that provide descriptive attributes, such as product **names, customer details, and time periods.** The layout resembles a star, with the fact table at the center and the dimension tables radiating outward.

In a typical star schema, the relationships between the fact table and dimension tables are designed to facilitate easy access to the data for analytical purposes. The fact table typically includes foreign keys that reference the primary keys in the associated dimension tables. This design allows for efficient querying and reporting, enabling users to perform complex analyses on the data with minimal joins.

**Example**

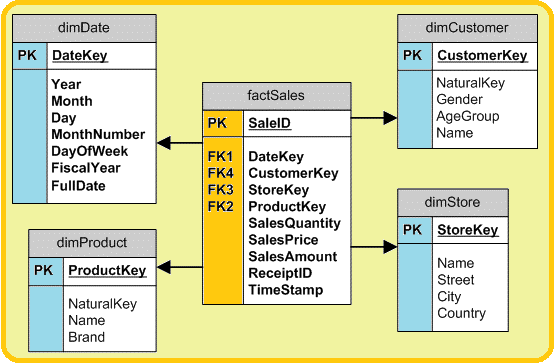
Imagine a star schema for a manufacturing company. The fact table could store measures like **"Revenue"** and **"Units Sold"**, while dimension tables could represent various attributes such as **"Product DIM"**, **"Customer DIM"**, **"Time DIM"** and **"Promotion DIM"**. Each of these dimensions provides context for the facts, enriching the data analysis experience.



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Fact Table***

A Fact Table stores the quantitative data or metrics of a business process. It is the central table in a star schema and contains numeric measures (such as **sales, revenue, or profit**). Each row in the fact table is identified by a concatenated primary key, made up of foreign keys from the related dimension tables.



**IMP to Note**

**Data Grain:** The level of detail captured in the fact table, which could represent individual transactions, **daily, weekly, or monthly aggregates**.

**Additive Measures:** Fully additive measures like sales can be aggregated across all dimensions (**time, product, customer,** etc.), while semi-additive measures can only be aggregated across some dimensions (e.g., **account balances across time**).

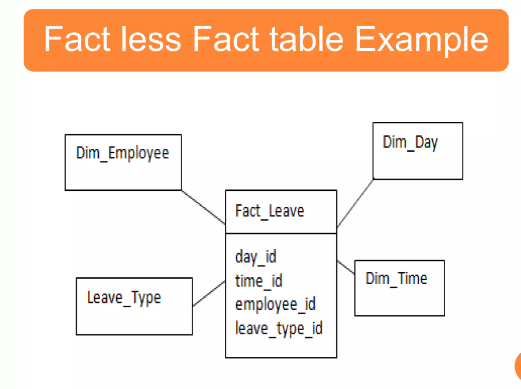
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Factless-Fact Table***

A Factless-Fact Table doesn’t contain any traditional numeric facts or measures. Instead, it records events or associations between dimensions where no explicit numeric data is required. The presence or absence of rows in the factless-fact table provides meaningful information.

**Example**

Tracking employee leaves. The factless fact table might have dimensions like day\_id, time\_id, employee\_id, and leave\_type\_id.



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Dimension Table***

A Dimension Table contains textual attributes that describe the business dimensions (e.g --> **product, customer, and store**). These attributes are used as filters or constraints in queries to slice and dice the facts stored in the fact table.

***Key Characteristics***

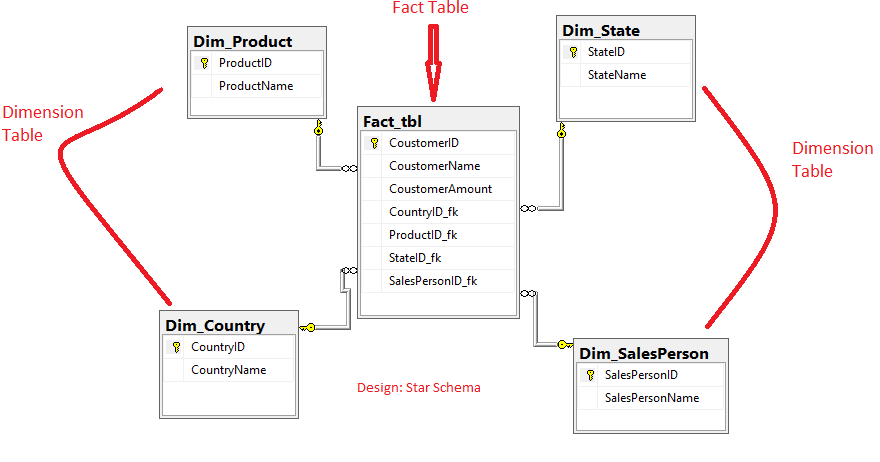
**Wide Tables:**Dimension tables often have many columns (attributes).

***Textual Attributes:*** Attributes are mostly descriptive, like **product name, category, or region**.

***Multiple Hierarchies:*** They can provide multiple hierarchies for drilling down (e.g --> grouping products by category, subcategory, and brand).

***Fewer Records:*** They generally contain fewer rows than the fact table but more attributes (e.g --> **hundreds of products vs. millions of sales transactions**).

**Example**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Schema Keys***

Primary and Foreign Keys are essential in the star schema to establish relationships between fact and dimension tables.

**Surrogate Keys:** These are system-generated keys used in dimension tables to avoid depending on production system keys, which might change or have built-in meanings.

**Foreign Keys:** The primary key from the dimension table becomes a foreign key in the fact table, enabling the relationship between the fact and dimension tables.

**Primary Key:** In a data warehouse, a **primary key** uniquely identifies each row in a dimension table. It ensures that each record is distinct and typically uses a surrogate key to avoid relying on changing business data. The primary key allows dimension tables to link with fact tables via foreign keys for accurate analysis.

**IMP to Note**

Three options exist for primary keys in the fact table:

1. A single compound key made of all dimension table keys.

2. A concatenated primary key.

3. A generated primary key, independent of the dimension tables, but with foreign keys from the dimension tables as additional attributes.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Advantages of Star Schema***

The star schema is widely used in data warehousing due to several benefits:

**Simplicity:** The schema structure is easy to understand, both for developers and users. It aligns closely with how users think about data, making it user-friendly.

**Optimized for Queries:** Star schemas are designed to facilitate efficient querying. By denormalizing the dimension tables, data retrieval is faster, as fewer joins are needed compared to a fully normalized schema.

**Easy Navigation:** The star schema simplifies the navigation through the database by reducing complex join paths. This makes query execution faster and easier to optimize.

**Supports Query-Centric Workloads:** The schema is optimized for decision-support queries, which involve filtering, aggregation, and analyzing large data sets.

**Drill-down and Roll-up:** It supports drilling down or rolling up along dimension hierarchies, such as analyzing sales data from yearly down to daily levels.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

