

Dimension and Fact Tables in Data Warehousing

Dimension tables and **fact tables** are fundamental components of data warehousing and star-schema design. They are used to structure large datasets in a way that enables fast, efficient data retrieval for reporting and analytics, especially in industries that rely on business intelligence (BI) tools and dashboards.

1. Fact Table:

A **fact table** contains **quantitative data** (facts) related to business processes and transactions. Each record in a fact table is a measurement or value associated with specific events, and it often contains foreign keys referencing the relevant dimension tables.

Key Characteristics:

- 1. Contains numeric values or metrics (e.g., sales, revenue, quantity).
- 2. Large volume of records.
- 3. Stores data at the lowest grain (level of detail).
- 4. Typically includes **foreign keys** to associated dimension tables.
- 5. Aggregates data for analysis (e.g., total sales, number of transactions).

Example Columns in a Fact Table:

- Sales Fact Table: SaleID, DateKey, ProductKey, StoreKey, SalesAmount, Quantity.

2. Dimension Table:

A **dimension table** contains **descriptive information** (attributes) that provide context for the facts in the fact table. Dimensions describe **who, what, when, where, and how** of the business process.

Key Characteristics:

- 1. Contains textual or descriptive attributes (e.g., customer name, product category).
- 2. Smaller volume compared to fact tables.
- 3. Provides filtering, grouping, and labeling context in queries.
- 4. Typically contains **primary keys** used in fact tables as foreign keys.

Example Columns in a Dimension Table:

- Product Dimension Table: ProductKey, ProductName, Category, Brand.
- Time Dimension Table: DateKey, Year, Month, Day.

Differences Between Fact and Dimension Tables:

Aspect	Fact Table	Dimension Table
Purpose	Stores measurable data related to business transactions.	Stores descriptive information used to provide context to fact table data.
Data Type	Numeric, quantitative (e.g., sales, revenue, quantity).	Textual, categorical (e.g., customer name, product category).

Granularity	Highly granular, representing individual transactions.	Less granular, stores descriptive data related to a specific category.
Size	Usually large, millions of rows.	Smaller, thousands or fewer rows.
Foreign Keys	Contains foreign keys to dimension tables.	Does not contain foreign keys.
Primary Use	Aggregation, calculations (e.g., sum, average).	Filtering, grouping, labeling (e.g., group by customer, filter by region).
Examples	Sales transactions, shipment quantities, customer interactions.	Products, customers, locations, time periods.

Industry-Specific Examples of Dimension and Fact Tables:

1. Retail Industry:

- **Fact Table (SalesFact):** Tracks sales transactions, including the total amount of each sale, the number of items sold, and the store location.
 - Columns: SaleID, ProductID, CustomerID, StoreID, DateID, QuantitySold, TotalSalesAmount.
- **Dimension Tables:**
 - **ProductDimension:** Stores product details like name, category, and brand.
 - Columns: ProductID, ProductName, Category, Brand.
 - **CustomerDimension:** Stores customer information such as age, gender, and location.
 - Columns: CustomerID, CustomerName, Age, Gender, Region.

Use Case: Track sales by product category, analyze which regions have the highest sales, and monitor sales trends over time.

2. Healthcare Industry:

- **Fact Table (PatientTreatmentFact):** Stores the number of treatments and associated costs for patients.
 - Columns: PatientID, TreatmentID, DateID, DoctorID, TotalCost.
- **Dimension Tables:**
 - **PatientDimension:** Stores patient demographic information like age and gender.
 - Columns: PatientID, PatientName, Gender, Age.
 - **DoctorDimension:** Stores doctor specialties and details.
 - Columns: DoctorID, DoctorName, Specialty.
 - **TreatmentDimension:** Describes various medical treatments.
 - Columns: TreatmentID, TreatmentType, TreatmentCategory.

Use Case: Analyze treatment costs by patient age group or by doctor specialization.

3. Finance Industry:

- **Fact Table (TransactionFact):** Stores individual financial transactions, including amounts and dates.
 - Columns: TransactionID, CustomerID, AccountID, DateID, Amount.
- **Dimension Tables:**
 - **CustomerDimension:** Describes customers and their demographics.
 - Columns: CustomerID, CustomerName, Age, IncomeLevel, Region.
 - **AccountDimension:** Stores bank account details.
 - Columns: AccountID, AccountType, BranchID.

Use Case: Monitor customer spending habits, track financial trends across regions, and analyze high-value transactions.

4. Education Industry:

- **Fact Table (StudentPerformanceFact):** Tracks student performance data, such as test scores and attendance records.
 - Columns: StudentID, CourseID, TestScore, AttendancePercentage, DateID.
- **Dimension Tables:**
 - **StudentDimension:** Describes students, including their grade levels and demographics.
 - Columns: StudentID, StudentName, GradeLevel, Gender.
 - **CourseDimension:** Describes course details.
 - Columns: CourseID, CourseName, CourseCategory.

Use Case: Track student performance over time, analyze test scores by course, and evaluate attendance trends.

5. Logistics Industry:

- **Fact Table (ShipmentFact):** Tracks shipment data, including costs and delivery times.
 - Columns: ShipmentID, ProductID, LocationID, DateID, Cost, DeliveryTime.
- **Dimension Tables:**
 - **ProductDimension:** Describes products being shipped.
 - Columns: ProductID, ProductName, Category.
 - **LocationDimension:** Describes shipping and receiving locations.
 - Columns: LocationID, LocationName, Region.

Use Case: Analyze shipping costs by product category or region and track delivery performance over time.

How to Create Fact and Dimension Tables:

In **MySQL** or other SQL-based database systems, you can create **fact** and **dimension** tables as follows:

1. Creating a Fact Table:

```
CREATE TABLE SalesFact (  
    SaleID INT PRIMARY KEY,  
    ProductID INT,  
    CustomerID INT,  
    StoreID INT,  
    DateID INT,  
    QuantitySold INT,  
    TotalSalesAmount DECIMAL(10, 2),  
    FOREIGN KEY (ProductID) REFERENCES ProductDimension(ProductID),  
    FOREIGN KEY (CustomerID) REFERENCES CustomerDimension(CustomerID),  
    FOREIGN KEY (StoreID) REFERENCES StoreDimension(StoreID),  
    FOREIGN KEY (DateID) REFERENCES DateDimension(DateID)  
);
```

2. Creating a Dimension Table:

```
CREATE TABLE ProductDimension (  
    ProductID INT PRIMARY KEY,  
    ProductName VARCHAR(255),  
    Category VARCHAR(100),  
    Brand VARCHAR(100)  
);
```

3. Populating Data:

Once the structure is created, data can be inserted into the dimension and fact tables, ensuring that the foreign keys align with the relevant dimension data.

Conclusion:

- **Fact tables** store transactional data and are used for analysis and aggregation.
- **Dimension tables** provide context to facts by storing descriptive attributes.
- Together, fact and dimension tables form the basis of star schemas in data warehousing, which are used across industries like retail, healthcare, finance, education, and logistics.
- Normalized dimension tables and large fact tables enable efficient querying and reporting, improving decision-making and operational analysis across industries.