| **The Sales History schema is used** to store and manage transactional data related to sales activities within an organization. This schema encompasses tables that capture detailed information about sales, products, customers, channels, promotions, and financial metrics. The schema is optimized for different uses, primarily supporting analytical reporting and strategic decision-making (OLAP). | | |
| --- | --- | --- |
| **With this data**, organizations can analyze their sales performance by tracking sales volumes, revenues, and profits over time. They can also identify trends and patterns in customer purchasing behavior, enabling data-driven decisions regarding inventory management, marketing strategies, and product development. Additionally, businesses can measure the effectiveness of promotions and marketing campaigns by linking sales data with promotional activities. By segmenting customers based on demographics, geography, and purchasing behavior, organizations can target their marketing efforts more effectively. The schema also enables the calculation of costs, revenues, and profit margins across different products, regions, channels, and time periods, which provides insights into profitability. Overall, the Sales History schema supports strategic planning by offering detailed reporting and analysis that highlights growth opportunities, potential risks, and areas for improvement. Through this schema, organizations can leverage historical data to optimize sales strategies, enhance customer satisfaction, and drive revenue growth. | | |
| **Table/View Name** | **Used For** | **Additional Notes** |
| **COUNTRIES** | Stores geographical data (e.g., country names, regions). | A dimension table; used for geographic analysis. |
| **CUSTOMERS** | Contains customer information like name, demographics, and contact details. | Dimension table for customer segmentation or behavior analysis. |
| **CHANNELS** | Describes sales channels (e.g.,direct- indirect sales, others). | Dimension table to analyze performance across sales channels. |
| **TIMES** | Stores time-related data (e.g., day, month, year). | Dimension table for temporal analysis (e.g., seasonal trends). |
| **PRODUCTS** | Holds product-related data such as name, category, and price. | Dimension table for analyzing sales by product or category. |
| **PROMOTIONS** | Stores details about promotions, discounts, and offers. (categories of promotions e.g Tv commercial, loyal customer discount) | Dimension table to assess promotion effectiveness. |
| **COSTS** | Contains cost-related metrics for transactions.(Unit price, unit cost, promo id etc) | Fact table to calculate profitability. Often used with SALES and PROFITS. |
| **SALES** | Stores sales transaction data (e.g., revenue, units sold). | Fact table central to the schema, used for analyzing revenue and performance. |
| **Supplementary Demographics** | Contains additional data about the customers (e.g., education level, occupation level, household size, comments) | It is a dimension table, which contains descriptive information (e.g., income levels, education, household size), which are characteristics or attributes rather than measurable metrics. |
| **PROFITS** | Contains calculated profits for transactions. | Derived from SALES and COSTS. Represents profitability metrics. It is a view, a virtual table that does not store data itself but provides a way to encapsulate and reuse queries. It can be also be considered as a fact table, as aggregates data and provides measurable KPIs. |

**Schema Design**

1. **Fact Tables**:
   * **SALES**: Core fact table for analyzing revenue, units sold, and other sales data.
   * **COSTS** and **PROFITS**: Additional fact tables used for financial performance analysis.
2. **Dimension Tables**:
   * **COUNTRIES**, **CUSTOMERS**, **CHANNELS**, **TIMES**, **PRODUCTS**, and **PROMOTIONS** provide contextual data for analyzing the facts.
3. **Relationships**:
   * Fact tables (e.g., SALES) have foreign keys linking them to each dimension table (e.g., CUSTOMERS, PRODUCTS, TIMES).
   * This creates a **star schema** with SALES at the center and dimensions as its spokes.

**Schema Design Considerations**

* **OLAP**:
  + OLAP systems are designed to perform complex analytical queries that involve aggregations, joins, and filtering across large datasets. The design is optimized for querying, summarizing, and analyzing data over time, usually with a focus on historical data. The purpose of this schema is primarily for analysis, reporting, and trend analysis (e.g., sales performance over time, promotion effectiveness).
  + Fact tables are designed to handle aggregations (e.g., total sales, average costs).
  + Dimension tables allow filtering and grouping by attributes (e.g., sales by region or product).
* **OLTP :**
  + OLTP systems are designed for handling transactional data. They are optimized for speed, reliability, and handling a high volume of short-lived transactions. The focus is on real-time data entry, updates, and retrieval, which
  + Fact and dimension tables don't exist in OLTP systems. The database employs a highly normalized data structure, with multiple tables linked through relationships (foreign keys). This normalization is optimal for data integrity and ensuring consistency but not for transaction speed and efficiency. In OLTP systems, the focus is on minimizing the number of tables and relationships to speed up transaction processing. Highly normalized schemas, like the one described here, can lead to more complex queries and slower performance when inserting, updating, or deleting records.
  + The use of normalization up to 3NF (Third Normal Form) is typical in OLTP systems. This normalization level is aimed at reducing redundancy and ensuring that each table stores a single type of data. However, in our schema, the use of multiple dimension tables (e.g., COUNTRIES, CUSTOMERS, CHANNELS) and the profits view indicates a more complex structure suited for analytical processing rather than real-time transactional handling.
  + The purpose of the schema is primarily for analysis, reporting, and trend analysis, while OLTP systems are generally not equipped for this level of detailed data mining and data manipulation. They are designed to handle transactions at scale, often with lower latency but limited analytical capabilities.