Designing a Data Warehouse

There are two different approaches for Relational Data Warehouse design: one that applies **dimensional modeling** techniques, and another that bases mainly in the concept of **materialized views**.

The components of dimensional modeling are **facts**, **dimensions** and **measures**:

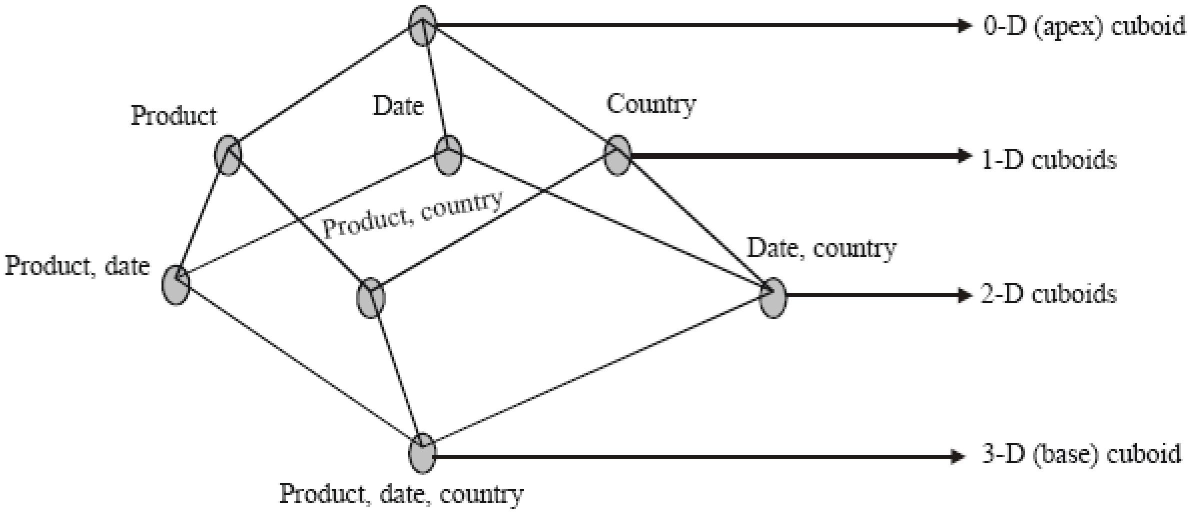
A **fact** is a collection of related data items, consisting of measures and context data. It typically represents business items or business transactions.

A **dimension** is a collection of data that describe one business dimension. Dimensions determine the contextual background for the facts; they are the parameters over which we want to perform OLAP.

A **measure** is a numeric attribute of a fact, representing the performance or behavior of the business relative to the dimensions.

Cube is an important tool of the dimensional model if the data warehouse is designed to support the way users wants to query data.

Example of a cube is as given below:

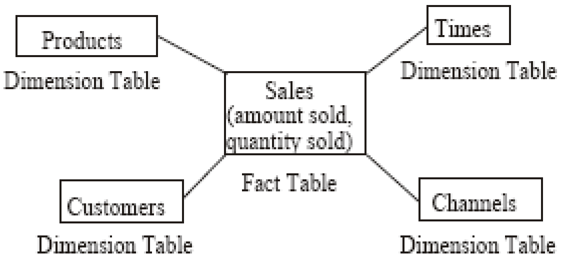


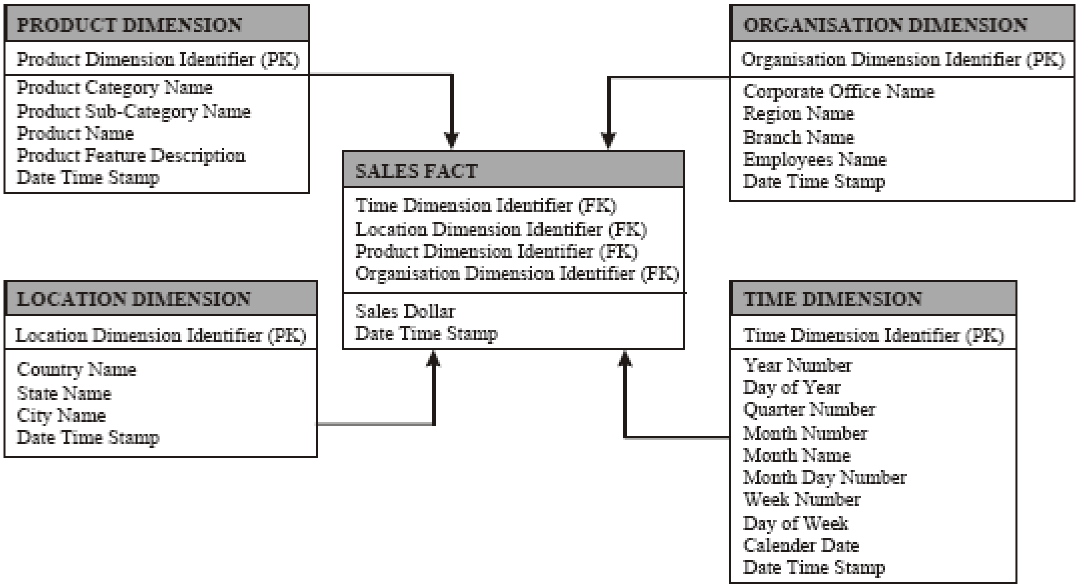
Some of the important components of a dimensional modeling schema are defined below:

Fact tables are the large tables in a warehouse schema that store business measurements. Fact tables typically contain facts and foreign keys to the dimension tables. Fact tables represent data, usually numeric and additive, that can be analyzed and examined. Examples include sales, cost, and profit. A fact table typically has two types of columns: those that contain **numeric facts**, and those that are **foreign keys to dimension tables**. A fact table contains either detail-level facts or facts that have been aggregated. Fact tables that contain aggregated facts are often called SUMMARY TABLES. A fact table usually contains facts with the same level of aggregation. Facts can be **additive**, **semi-additive** and **non-additive**.

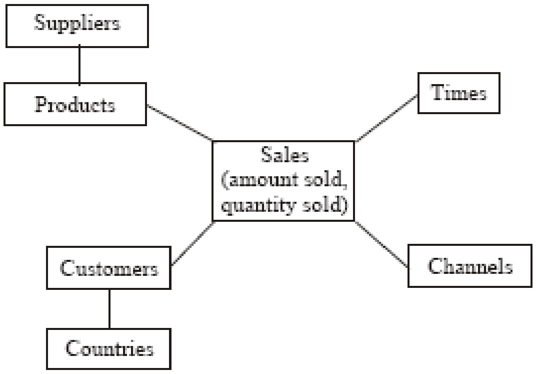
Dimension tables, also known as **lookup** or **reference tables**, contain the relatively static data in the warehouse. Dimension tables store the information you normally use to contain queries. Dimension tables are usually textual and descriptive and you can use them as the row headers of the result set. Examples are customers, Location, Time, Suppliers or products.

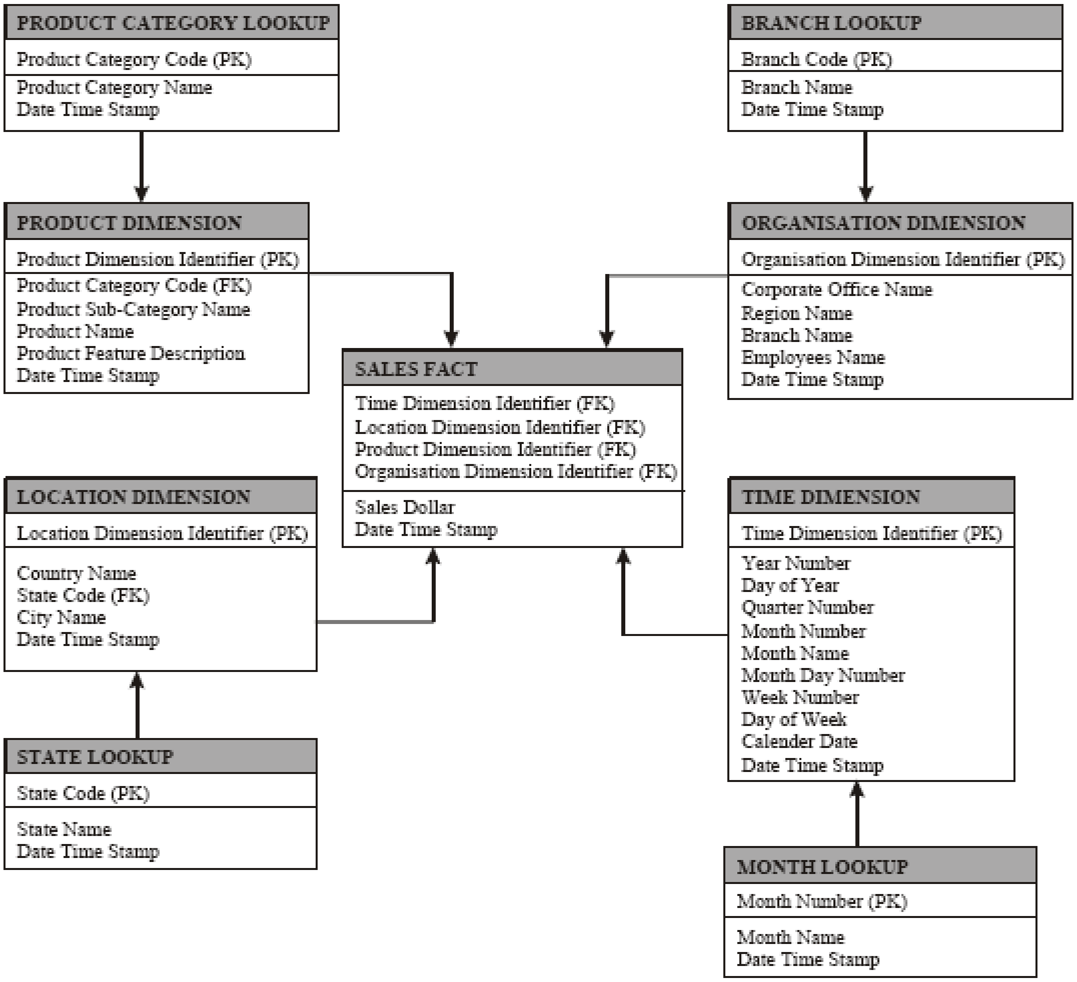
Star schema





Snowflake schema





PRODUCT CATEGORY LOOKUP

Product Category Code (PK)

Product Category Name
Date Time Stamp

BRANCH LOOKUP

Branch Code (PK)

Branch Name
Date Time Stamp

PRODUCT DIMENSION

Product Dimension Identifier (PK)

Product Category Code (FK)
Product Sub-Category Name
Product Name
Product Feature Description
Date Time Stamp

ORGANISATION DIMENSION

Organisation Dimension Identifier (PK)

Corporate Office Name
Region Name
Branch Name
Employees Name
Date Time Stamp

SALES FACT

Time Dimension Identifier (FK)
Location Dimension Identifier (FK)
Product Dimension Identifier (FK)
Organisation Dimension Identifier (FK)

Sales Dollar
Date Time Stamp

LOCATION DIMENSION

Location Dimension Identifier (PK)

Country Name
State Code (FK)
City Name
Date Time Stamp

TIME DIMENSION

Time Dimension Identifier (PK)

Year Number
Day of Year
Quarter Number
Month Number
Month Name
Month Day Number
Week Number
Day of Week
Calender Date
Date Time Stamp

STATE LOOKUP

State Code (PK)

State Name
Date Time Stamp

MONTH LOOKUP

Month Number (PK)

Month Name
Date Time Stamp

Star schema vs Snowflake schema

Star and snowflake are most common types of dimensional modeling. Always a debating question in the data warehousing context is which one works better? You will here arguments favouring both sides; however the question is incomplete without mentioning the system/business. The decision whether to employ a star schema or a snowflake schema should consider the relative strengths of the database platform in question and the query tool to be employed.

Star Schemas

The star schema is the simplest data warehouse schema. It is called a star schema because the diagram resembles a star, with points radiating from a center. The center of the star consists of one or more fact tables and the points of the star are the dimension tables.

Snowflake Schema

The snowflake schema is a variation of the star schema used in a data warehouse. The snowflake schema (sometimes called snowflake join schema) is a more complex schema than the star schema because the tables which describe the dimensions are normalized.

Star vs Snowflake

	Snowflake Schema	Star Schema
Which Data warehouse?	Good to use for small datawarehouses/datamarts	Good for large datawarehouses
Normalization (dimension table)	3 Normal Form	2 Normal Denormalized Form
Ease of Use	More complex queries and hence less easy to understand	Less complex queries and easy to understand
Ease of maintenance/change	No redundancy and hence more easy to maintain and change	Has redundant data and hence less easy to maintain/change
Query Performance	More foreign keys-and hence more query execution time	Less no. of foreign keys and hence lesser query execution time