

Basics of SAP Business Information Warehousing



Satisfying Customers Beyond Expectations

Table of Contents

1 DATA MODELING	4
1.1 INTRODUCTION.....	4
1.1.1 <i>Some Theories on Data Modeling</i>	4
1.1.2 <i>Conceptual Approaches to Modeling</i>	4
1.1.3 <i>Designs Based on the Entity Relationship Model</i>	5
1.1.4 <i>Designs Based on the Object-Oriented Model</i>	6
1.1.5 <i>Modeling (Conceptual and Physical Schema)</i>	7
1.1.6 <i>Dimensioning of Key Figures</i>	7
1.1.7 <i>Physical Data Model: Determining the Objects for Reporting</i>	7
1.1.8 <i>Golden Rules of Dimensional Modeling</i>	8
1.2 INFOPROVIDERS	9
1.2.1 <i>Introduction</i>	9
1.2.2 <i>InfoProviders</i>	9
1.2.3 <i>Structure (Types of Info Provider)</i>	10
1.2.4 <i>Information Provider Interface</i>	10
1.2.5 <i>InfoProvider View (Administrator Workbench)</i>	11
1.2.6 <i>InfoArea</i>	11
1.2.6.1 <i>Create InfoArea</i>	12
1.2.7 <i>Hybrid Providers</i>	13
1.3 MULTI PROVIDER.....	13
1.3.1 <i>Introduction</i>	13
1.3.2 <i>Structure</i>	13
1.3.2.1 <i>Characteristics</i>	14
1.3.2.2 <i>Key Figures</i>	14
1.3.3 <i>Categories</i>	14
1.3.3.1 <i>Homogenous MultiProviders</i>	14
1.3.3.2 <i>Heterogeneous MultiProviders</i>	14
1.3.4 <i>Integration</i>	15
1.3.5 <i>MultiProviders and InfoSets</i>	15
2 DETERMINING ALL THE REQUIRED OBJECTS (CHARACTERISTICS, ATTRIBUTES, AND KEY FIGURES)	16
2.1 CHARACTERISTICS.....	16
2.2 KEY FIGURE (TYPE OF INFO PROVIDER).....	16
2.3 ATTRIBUTES	16
2.3.1 <i>Display Attribute</i>	17
2.3.2 <i>Navigational Attribute</i>	17
2.3.3 <i>Time Dependent Attribute</i>	18
2.3.4 <i>Compound Attribute</i>	18
2.3.5 <i>Transitive Attribute</i>	18
3. TRACKING HISTORY	19
3.1 STRUCTURE OF DIMENSIONAL TABLES	19
3.1.1 <i>Tables of Dimensional Model</i>	20

3.1.1.1 Dimension Tables	20
3.1.1.1.1 Dimension Details.....	21
3.1.1.1.2 Dimension Keys	21
3.1.1.1.3 Dimension Surrogate Keys.....	21
3.1.1.1.4 Dimensions Types.....	22
3.1.1.2 Fact Tables:.....	22
3.1.1.2.1 Fact Table Types	22
3.1.1.2.1.1 Transactional Fact Tables.....	23
3.1.1.2.1.2 Snapshot Fact Tables	23
3.1.1.2.1.3 Factless Fact Tables.....	23
3.1.1.2.2 Measures Classified	23
3.1.1.2.2.1 Aggregations.....	23
3.1.1.2.2.2 Additive Nature.....	23
3.1.1.2.2.3 Fact Table Granularity.....	24
3.1.2 Dimensional normalization VS Transactional normalization	24
3.2 DIMENSION MODELING	24
3.2.1 Introduction	24
3.2.2 What is a dimensional model?.....	24
3.2.2.1 Fact table characteristics.....	25
3.2.2.2 Dimension table characteristics	26
3.2.3 Types of dimensional models.....	27
3.2.3.1 Star model	27
3.2.3.2 Snowflake model.....	27
3.2.3.3 Multi-star model.....	28
3.3 STAR SCHEMA.....	29
3.3.1 Introduction	29
3.3.2 Components of a Star Schema Diagram	29
3.3.2.1 Fact table.....	30
3.3.2.1.1 Building a fact table	30
3.3.2.2 Dimension table	31
3.3.3 Star Schema Example.....	31
3.3.4 Design of star Schema.....	32
3.3.5 Types of Star Schema	32
3.3.5.1 Classic Star Schema	32
3.3.5.2 Extended Star Scheme/BW Star Schema.....	33
3.3.5.3 Advantages of SAP BW Star Schema over classic star schema	33
3.3.6 OLAP and Star Schema.....	33

1 Data Modeling

1.1 Introduction

Data modeling is the process of creating and extending data models which are visual representations of data and its organization. The objects are displayed in modeling in a tree structure. And sorted with respect to hierarchical criteria. The Data Modeling process helps to model various Info Providers. These Info Providers include InfoObjects, DataStore objects, InfoCubes, Virtual Providers, InfoSets and MultiProviders. Effective data modeling results in transforming data into an enterprise information asset that is consistent, comprehensive and current.

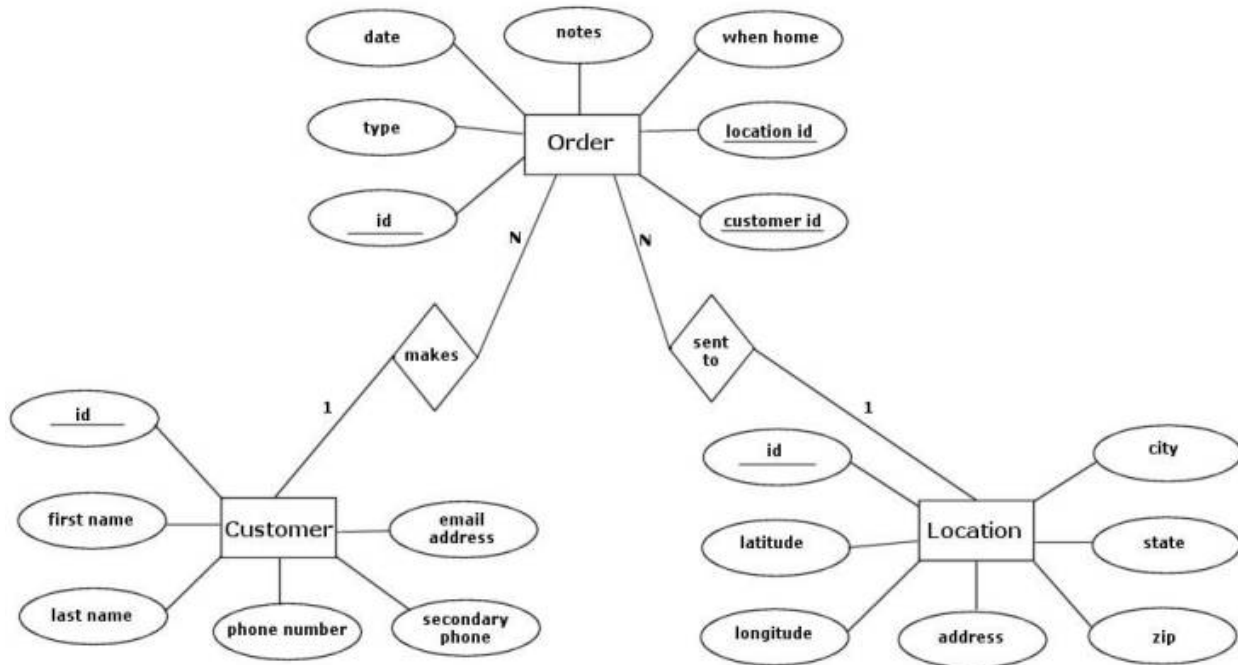
1.1.1 Some Theories on Data Modeling

Data modeling concepts is strongly recommended for the projects requiring a standard means of defining and analyzing data within an organization. Data modeling is also a technique for detailing business requirements for a database. Data modeling may be performed during various types of projects and in multiple phases of projects.

- The **Conceptual Data Model** describes data from a high level. It defines the problem rather than the solution from the business point of view. It includes entities and their relationships.
- The **Logical Data Model** describes a logical solution to a data project. It provides more details than the conceptual data model and is nearly ready for the creation of a database. These details include attributes, the individual pieces of information that will be included.
- The **Physical Data Model** describes the implementation of data in a physical database. It is the blueprint for the database.

1.1.2 Conceptual Approaches to Modeling

Conceptual data models provide languages to describe conceptual schemas. Conceptual schemas are used to describe the classes of objects that occur in an application area, their properties, their relationships, and the constraints that hold with respect to those classes of objects.



1.1.4 Designs Based on the Object-Oriented Model

The object-oriented model is based on a collection of objects, like the E-R model. The objects contain objects to an arbitrarily deep level of nesting. An object also contains bodies of code that operate on the object which are called as **methods**. Objects that contain the same types of values and the same methods are grouped into **classes**. A class may be viewed as a type definition for objects. The only way in which one object can access the data of another object is by invoking the method of that other object. This is called sending a message to the object.

Object-Oriented Model

Object 1: Maintenance Report

Date	
Activity Code	
Route No.	
Daily Production	
Equipment Hours	
Labor Hours	

Object 1 Instance

01-12-01
24
1-95
2.5
6.0
6.0

Object 2: Maintenance Activity

Activity Code	
Activity Name	
Production Unit	
Average Daily Production Rate	

1.1.5 Modeling (Conceptual and Physical Schema)

A data model *instance can be either Conceptual or Physical.*

Conceptual schema describes the semantics of a domain, being the scope of the model. Specifically, it describes the things of significance to an organization (entity classes), about which it is inclined to collect information, and characteristics of (attributes) and associations between pairs of those things of significance (relationships). A conceptual schema specifies the kinds of facts or propositions that can be expressed using the model. For example, it may be a model of the interest area of an organization or industry. This consists of entity classes, representing kinds of things of significance in the domain, and relationships assertions about associations between pairs of entity classes.

Physical schema describes the physical means by which data are stored. This is concerned with partitions, CPUs, tablespaces. Physical data model includes all required tables, columns, relationships, database properties for the physical implementation of databases. Database performance, indexing strategy, physical storage and denormalization are important parameters of a physical model.

1.1.6 Dimensioning of Key Figures

There are three fixed dimensions datapackage, Time and Unit. So if you include a time characteristic in the cube it is directly assigned to the time dimension. Same is the case with the units. Only chars needs to be specifically modeled and assigned to the created dimensions. In the case of multiprovider you have the additional option of infoprovider. Time chars are automatically assigned to time dimensions by the system. You will have this dimension always present in the cube. Key figures will never be assigned to the dimensions.

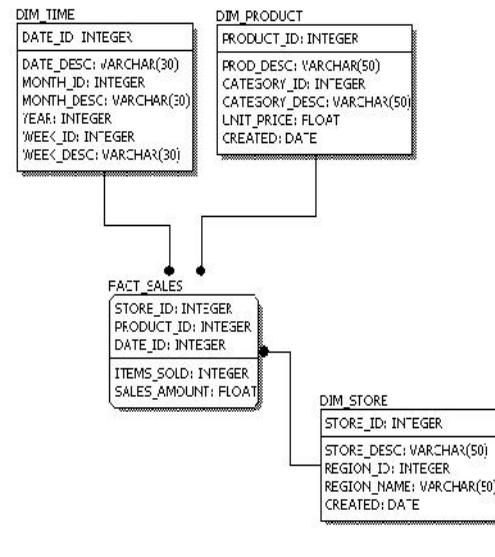
1.1.7 Physical Data Model: Determining the Objects for Reporting

Physical data model represents how the model will be built in the database. A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables.

Physical considerations may cause the physical data model to be quite different from the logical data model. Features of a physical data model includes: Specification all tables & columns, Foreign keys are used to identify relationships between tables, Denormalization may occur based on user requirements.

The steps for physical data model design are as follows: Convert entities into tables. Convert relationships into foreign keys. Convert attributes into columns. Modify the physical data model based on physical constraints / requirements.

The figure below is an example of a physical data model.



1.1.8 Golden Rules of Dimensional Modeling

Following are the rules of Dimensional Modeling to ensure the granular data, flexibility and a future-proofed information resource.

- Load detailed atomic data into dimensional structures.
- Structure dimensional models around business processes.
- Ensure that every fact table has an associated date dimension table.
- Ensure that all facts in a single fact table are at the same grain or level of detail.
- Resolve many-to-many relationships in fact tables.
- Resolve many-to-one relationships in dimension tables.
- Store report labels and filter domain values in dimension tables.
- Make certain that dimension tables use a surrogate key.
- Create conformed dimensions to integrate data across the enterprise.
- Continuously balance requirements and realities to deliver a DW/BI solution that's accepted by business users and that supports their decision-making.

1.2 InfoProviders

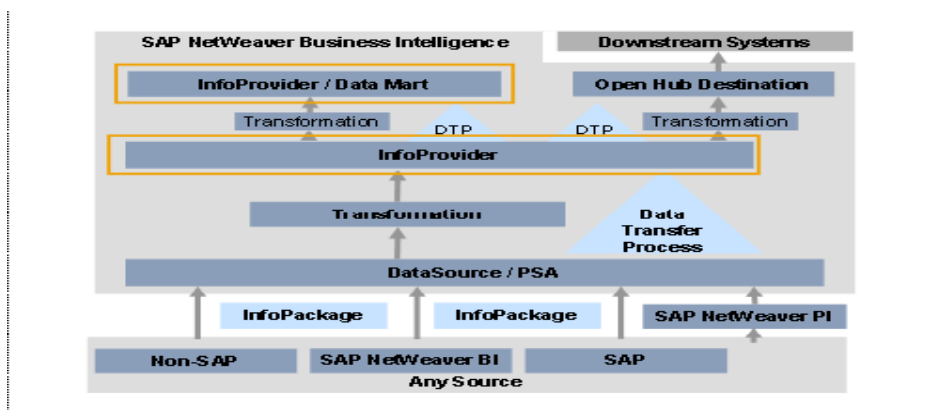
1.2.1 Introduction

SAP's BW information model is based on the core building block of InfoObjects which are used to describe business processes and information requirements. They provide basis for setting up complex information models in multiple languages, currencies, units of measure, hierarchy, etc. The key elements in the SAP's BW information model are:

- InfoProviders
- DataSource
- InfoSources
- ODS Objects
- InfoCubes
- MultiProviders

1.2.2 InfoProviders

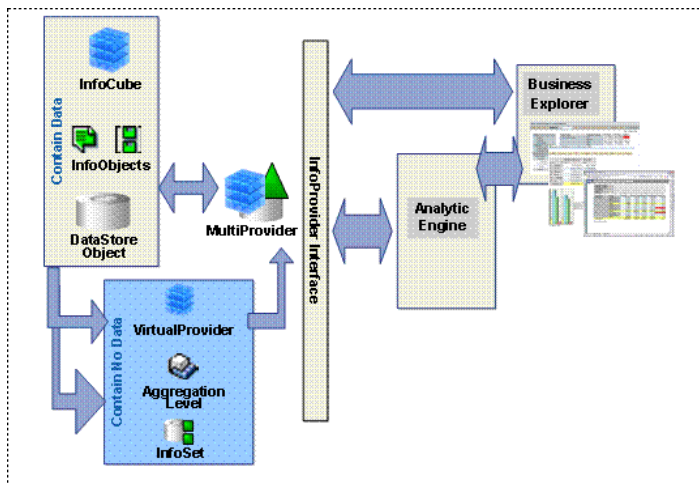
- InfoProviders refer to all physical or virtual data objects that are present in the SAP BW systems. These include all the data targets viz. InfoCubes, ODS objects and master data tables along with Info sets, remote Infocubes and MultiProviders.
- InfoProviders are the basis of most of the reporting you will be doing in the real time. They are the objects that serve your reports with data. Hence the name InfoProviders.
- An Infoproviders represents a view of data that can be used to define queries. At runtime, the query instructs the InfoProvider to supply data. The query defines the data that is needed and the selection criteria that are used.
- Infoproviders are objects for which you can create and execute queries in SAP BI
- Data Target refers to an object where there is a physical storage of data. Info-provider refers to a logical object where there is no physical storage of data.



1.2.3 Structure (Types of Info Provider)

There are 2 types:

1. Data targets that contain physical data.
 - Infocubes
 - DataStore Objects
 - InfoObjects as InfoProviders
2. Targets displays but no physical storage.
 - InfoSets
 - Virtual Providers(Remote Cubes)
 - MultiProviders
 - Aggregation Levels



1.2.4 Information Provider Interface

- With SAP BW 3.0 the Information Provider interface has been introduced to generalize access to data available in SAPBW.
- The Information Provider interface allows access to physical and virtual InfoProviders. Physical InfoProviders include basic InfoCubes, ODS objects, master data tables, and InfoSets physically available on the same system.
- Access to physical InfoProviders is handled by the storage services layer.
- VirtualInfoProviders include MultiProviders and remote InfoCubes. Access to virtual Info-Providers requires analyzing the request and routing the actual access to a remote

system (in case of remote InfoCubes) or accessing several physical objects through the storage services layer (in case of MultiProviders).

1.2.5 InfoProvider View (Administrator Workbench)

InfoProvider	Tech. Name	M ...	Execute F
↳ ZINM_PPA	ZINM_PPA		Change
↳ T_05A18	T_05A18		Change
↳ YTE_AREA	YTE_AREA		Change
↳ viswanath1	VISWANATH1		Change
↳ leena_infarea	INFOARF41		Change

- InfoProviders that contain real-time InfoCubes provide the data basis for BI Integrated Planning
- InfoCubes are filled with data from one or more InfoSources or other InfoProviders. They are available as InfoProviders for analysis and reporting purposes.
- InfoProviders are used by Bex. However, in the BEx Query Designer, they are seen as uniform objects.
- Aggregation levels are a type of virtual InfoProvider and are created on the basis of a real-time InfoCube, or a MultiProvider that contains InfoCubes of this type. Aggregation levels are specifically designed so that you can plan data manually or change it using planning functions.
- InfoProviders are different metaobjects in the data basis that can be seen within query definition as uniform data providers. Their data can be analyzed in a uniform way. The type of data staging and the degree of detail or "proximity" to the source system in the data flow diagram differs from InfoProvider to InfoProvider.
- You can flag an InfoObject of type characteristic as an InfoProvider if it has attributes. In the InfoObject maintenance on the Master Data/Texts tab page, you set the With Master Data indicator. The data is then loaded into the master data tables using the transformation rules.

1.2.6 InfoArea

- An InfoArea is a directory of InfoProviders and InfoObject catalogs used in the same business context. Every InfoProvider or InfoObject catalog belongs to exactly one singleInfoArea.
- InfoAreas in the same way as InfoObject catalogs help organize project work in large SAP BW implementations
- A characteristic can be an InfoProvider if it has master data and is assigned to an InfoArea. By default, Characteristics are not configured as InfoProviders. They must be explicitly defined as an InfoProvider in the InfoObject maintenance.
- In BW, InfoAreas are the branches and nodes of a tree structure. InfoCubes are listed under the branches and nodes

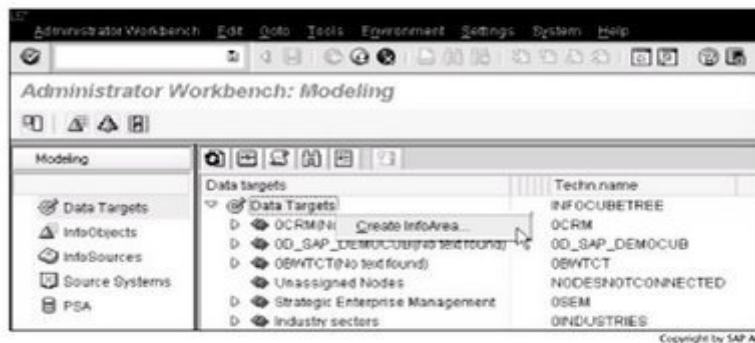
1.2.6.1 Create InfoArea

1. After logging on to the BW system, run transaction RSA1, or double-click Administrator

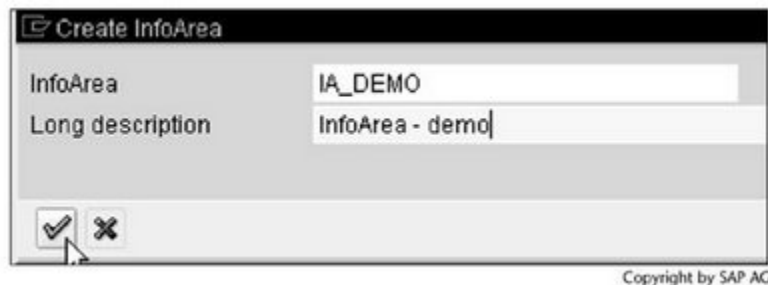


Workbench.

2. In the new window, click Data targets under Modeling in the left panel. In the right panel, right-click InfoObjects and select Create InfoArea....



3. Enter a name and a description for the InfoArea, and then click the right mark to continue



1.2.7 Hybrid Providers

The Hybrid Provider is a combination of a DataStore object and an InfoCube. Hybrid Providers are part of the operational data store layer. This layer should only contain data that is relevant to operational transactions. It should not contain the entire history of the data.

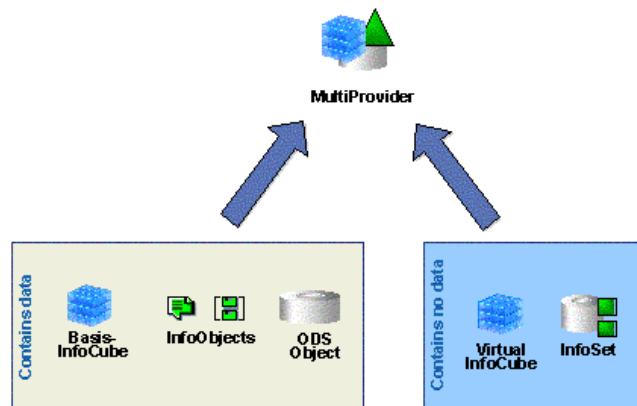
1.3 Multi provider

1.3.1 Introduction

A MultiProvider is a type of InfoProvider that combines data from a number of InfoProviders and makes it available for reporting purposes. The MultiProvider does not itself contain any data. Its data comes entirely from the InfoProviders on which it is based. These InfoProviders are connected to one another by a union operation. InfoProviders and MultiProviders are the objects or views that are relevant for reporting.

1.3.2 Structure

A MultiProvider can consist of different combinations of the following InfoProviders: InfoCube, ODS object, InfoObject, and InfoSet. A union operation is used to combine the data from these objects into a MultiProvider. Here, the system constructs the union set of the data sets involved. In other words, all values of these data sets are combined. As a comparison: InfoSets are created using joins. These joins only combine values that appear in both tables. In contrast to a union, joins form the intersection of the tables.



1.3.2.1 Characteristics

In a MultiProvider, every characteristic in each of the InfoProviders involved must correspond to exactly one characteristic or navigation attribute (wherever these are available). If it is not clear, at the MultiProvider definition stage, you have to specify to which InfoObject you want to assign the characteristic of the MultiProvider. The MultiProvider contains the characteristic 0COUNTRY and an InfoProvider contains the characteristic 0COUNTRY as well as the navigation attribute 0CUSTOMER__0COUNTRY. In this case, select exactly one of these InfoObjects in the assignment table.

1.3.2.2 Key Figures

Select a key figure contained in a MultiProvider from at least one of the InfoProviders involved. In general, exactly one of the InfoProviders provides the key figure. However, there are cases where it is better to select from more than one InfoProvider. If the 0SALES key figure is stored redundantly in more than one InfoProvider (meaning that it is contained completely in all value combinations of the characteristics) you must select from exactly one of the InfoProviders involved (otherwise the duplicated value is added incorrectly in the MultiProvider). However, if 0SALES is stored as an actual value in one InfoProvider and as a planned value in another InfoProvider so that there is no overlap between the data records (in other words, sales are divided separately between several InfoProviders) then it makes sense to make a selection across several InfoProviders.

1.3.3 Categories

- Homogenous MultiProviders
- Heterogeneous MultiProviders:

1.3.3.1 Homogenous MultiProviders

These consist of technically identical InfoProviders, such as InfoCubes with exactly the same characteristics and key figures, where one InfoCube contains the data for 2001, for example, and a second InfoCube contains data for 2002. Homogenous MultiProviders can be used to partition on the modeling level of the InfoProvider.

1.3.3.2 Heterogeneous MultiProviders

These are made up of InfoProviders that only have a certain number of characteristics and key figures in common. Heterogeneous MultiProviders can be used to simplify the modeling of scenarios by dividing them into sub-scenarios. Each sub-scenario is represented by its own InfoProvider. An example is a sales scenario made up of the sub-processes order, delivery and payment. Each of these sub-processes has its own (private) InfoObjects (delivery location and invoice number, for example) as well as a number of cross-process objects (such as customer or order number). It makes sense here to model each sub-

process in its own InfoProvider and then combine these InfoProviders into a MultiProvider. It is possible to:

- Model all sub-scenarios in an InfoProvider or
- Create an InfoProvider for each sub-scenario, and then combine these InfoProviders into a single MultiProvider.

The latter option usually simplifies the modeling process and can improve system performance when loading and reading data.

1.3.4 Integration

MultiProviders only exist as a logical definition. The data continues to be stored in the InfoProviders on which the MultiProvider is based. A query based on a MultiProvider is divided internally into subqueries. There is a subquery for each InfoProvider included in the MultiProvider. These subqueries are usually processed in parallel. A query based on a MultiProvider is divided internally into sub-queries. A sub-query is generated for each InfoProvider associated with the MultiProvider.

Technically there are no restrictions with regard to the number of InfoProviders that can be included in a MultiProvider. However, SAP recommends that you include no more than 10 InfoProviders in a single MultiProvider, because any more than this number and splitting the MultiProvider queries and reconstructing the results from the individual InfoProviders takes a substantial amount of time and is generally counterproductive. Modeling these types of MultiProviders is also highly complex.

1.3.5 MultiProviders and InfoSets

An InfoProvider is a BI Content object for which BI queries can be created or executed in the BEx. InfoProviders are the objects or views that are relevant for reporting. A MultiProvider as well as an InfoSet do not physically store data, but display logical views.

A MultiProvider builds up a data union of basic InfoProviders. The complete data of all basic InfoProviders are available for reporting. A MultiProvider is interpreted at runtime as independent BI queries on each basic InfoProvider where the results are merged into a single result set.

An InfoSet builds up a data join of basic InfoProviders. The valid combination of records from the basic InfoProviders is determined by the join condition of the InfoSet.

2 Determining All the Required Objects (Characteristics, Attributes, and Key Figures)

2.1 Characteristics

Characteristic is a type of InfoObject represented by master data. A characteristic has a technical name. Characteristics are sorting keys, such as company code, product, customer group, fiscal year, period, or region. They specify classification options for the dataset and are therefore reference objects for the key figures. The characteristics determine the granularity (the degree of detail) at which the key figures are kept in the InfoCube. In general, a data target contains only a sub-quantity of the characteristic values from the master data table. The master data includes the permitted values for a characteristic which are known as the characteristic values. Characteristics have attributes, texts, or hierarchies at their disposal then they are referred to as master data-bearing characteristics.

A hierarchy is always created for a characteristic. This characteristic is the basic characteristic for the hierarchy which does not refer other characteristics. Time characteristics are characteristics such as date, fiscal year, and so on. Technical characteristics are used for administrative purposes only within BW. An example of a technical characteristic is the request number in the InfoCube. This is generated when you load a request as an ID and helps you locate the request at a later date.

Characteristics are descriptions of key figures, such as Customer ID, Material Number, Sales Representative ID, Unit of Measure, and Transaction Date.

2.2 Key Figure (type of Info provider)

Key Figures are characterized by a multiplicity of characteristics that are arranged in different dimensions. These Key figures represent facts. It is a type of InfoObject which is a numeric field like account debit amount. The key figure can contain a formula, which is displayed in full with all of the operators and objects used. You can navigate in the Key Figure Definition dialog box using the context menu. Key figures can be any of these types:

- Key figures of type amount are always assigned a currency key for conversion
- Key figures of type quantity are assigned a unit of measurement for conversion
- Key figures of type number don't require a conversion

2.3 Attributes

Attributes are InfoObjects that exist already, and that are assigned logically to the new characteristic. In the Attributes list, specify directly in the fields that are ready for input the name of an InfoObject that you want to use as an attribute. If the InfoObject that you want to use does not yet exist, you have the option of creating a new InfoObject at this point. Any new InfoObjects that you create are inactive. They are activated when the existing InfoObject is activated. Use F4 Help for the fields that are ready for input in the Attributes of the Characteristic list, to display all the InfoObjects. Choose the attributes that you need. It is possible to choose attributes from the Attributes of the Assigned DataSources list. Attributes are defined within entities. An attribute contains values that help to describe the member they're related to attributes can also reference members of other entities defined within the same model. Attributes can be of five types which are used in real time scenarios:

- Display attribute
- Navigational attribute
- Time dependent navigational attributes
- Compound attributes
- Transactional attributes

2.3.1 Display Attribute

Display attribute does not support any user interaction but a navigational attribute (email, website, or any ID on clicking of which a new screen opens with data). Display attributes used to display the properties of the characteristic in the report. (Simply saying display purpose) but navigational attributes can be used for Drill down/drill across in the result area in reports. It means it behaves like display attribute as well as characteristic. A display attribute provides supplemental information to a characteristic.

2.3.2 Navigational Attribute

A Navigational attribute indicates a characteristic-to-characteristic relationship between two characteristics. It provides supplemental information about a characteristic and enables navigation from characteristic to characteristic during a query. The naming convention for a navigational attribute is: InfoObject_Attribute, such as Ocustomer_Ocountry. The basic navigation functions are:

Filtering a characteristic according to characteristic values or hierarchy nodes

- Drilling down according to a characteristic and changing the drilldown status
- Filtering a characteristic, and drilling down according to a different characteristic
- Distributing the characteristics and key figures along the row axes and the column axes of the query

- Changing the sequence of characteristics and key figures on an axis
- Expanding a hierarchy
- Hiding and displaying key figures

2.3.3 Time Dependent Attribute

With time-independent master data attributes it is possible to avoid inconsistency, although the information may not be perfectly accurate. This inaccuracy arises from characteristic values relevant to when the system processed the data in SAP NetWeaver BI, not when it created the transaction or when the transaction took place in the source system. In BI only master data can be defined as a time-dependent data source. Two additional fields/attributes are added to the characteristic. As of SAP NetWeaver 2004s, you can specify a date or use a time characteristic with DataStore objects and InfoCubes to describe the validity of a record. These InfoProviders are then interpreted as time-dependent data sources. DataStore objects and InfoCubes that are being used as InfoProviders in the InfoSet cannot be defined as time dependent.

2.3.4 Compound Attribute

A compound attribute differentiates a characteristic to make the characteristic uniquely identifiable. For example, if the same characteristic data from different source systems mean different things, then we can add the compound attribute OSOURSYSTEM (source system ID) to the characteristic; OSOURSYSTEM is provided with the Business Content. In real time, typically in an organization the employee id are allocated in serial like say 10001, 10002 and so on. Lets your Organization comes out with a new employee id scheme where the employee id for each location would start with 101. So the employee id starting for India would be India/101 and for US would be US/101. Now note that the employee India/101 and US/101 are different. Now if someone has to contact employee 101 he needs to know the location without which he cannot uniquely identify the employee. Hence in this case location is the compounding attribute.

2.3.5 Transitive Attribute

Transitive attribute is really an "attribute of an attribute" or a "second level attribute". For example, an InfoProvider has an InfoObject and a Cost Center, which has a navigational attribute of a CompanyCode, which in turn has its own attribute of the company. Transitive attributes are defined as display attributes. The company is the transitive attribute that you could make a navigational attribute. If a characteristic was included in an InfoCube as a navigation attribute, it can be used for navigating in queries. This characteristic can itself have further navigation attributes, called transitive attributes. These attributes are not automatically available for navigation in the query. As described in this procedure, they must be switched on.

3. Tracking History

SAP refers to four different business scenarios when dealing with tracking history: historical truth, current, time dependent, and comparable. Historical truth means displaying information from when events happened, but not reflecting any subsequent changes. The historical truth business scenario works by associating master data attributes with transactional data. Routines or master data attribute lookups perform this in transformations. Historical truth scenario involves looking up master data attributes, looking up values from any object, such as DataStore objects or custom tables. Historical truth is required when tracking performance over time.

3.1 Structure of Dimensional Tables

Combination of normalization and denormalization often called Dimensional Normalization. It can be used for full data warehouse design as well as for data mart designs. Logical model is easy to understand the Standard framework and business model for end user apps. This model can be done independent of expected queries and handle changes easy such as adding new dimensional attributes. Optimized for high performance “browsing” across the attributes. Strategy to handling aggregates, leveraging summary tables or OLAP aggregation technologies. It’s logical redundant with base table to enhance query performance and OLAP engines can make strong assumptions on how to optimize. Keeps Historical tracking of information. Manages Strategies for handling changing dimensions. Fact design allows high volume snapshots and transaction Tracking.

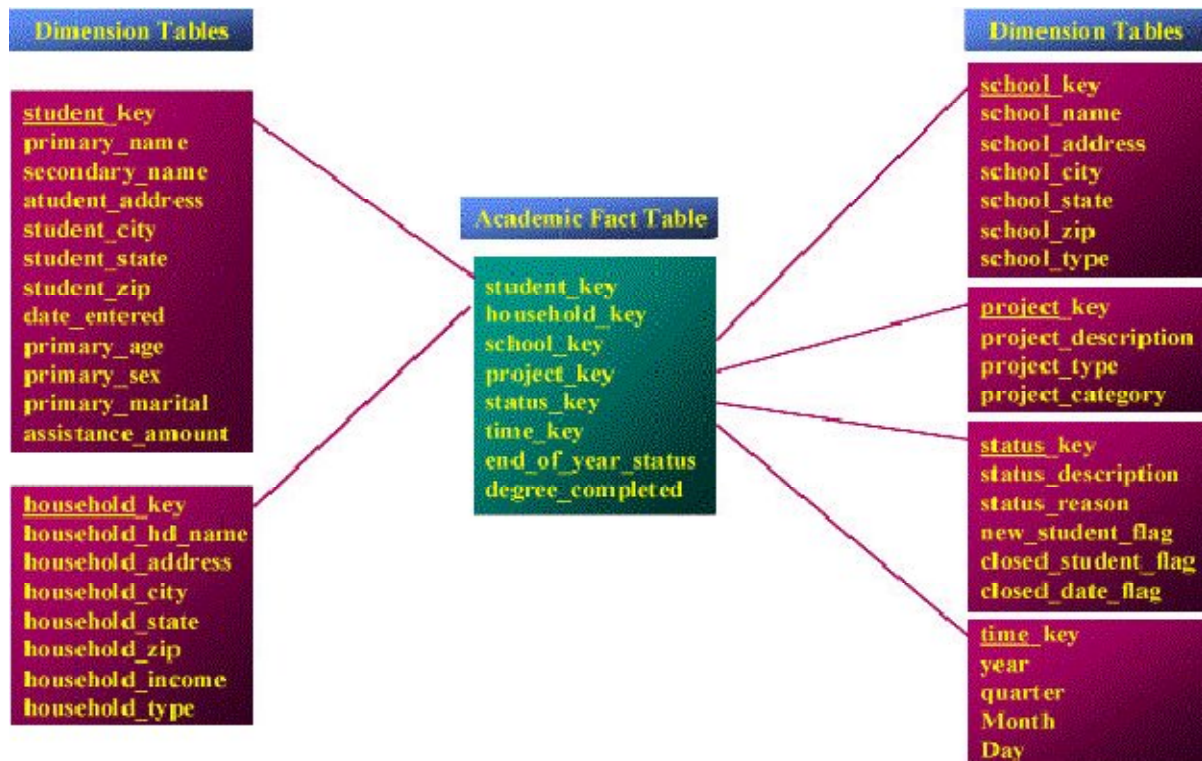


Figure One -- A Dimensional Model Star Schema of A Student Academic Fact Database

3.1.1 Tables of Dimensional Model

There are two types of tables in Dimensional model.

- Dimension tables
- Fact tables

3.1.1.1 Dimension Tables

It consists of attributes related to business entities like Customers, vendors, employees, Products, materials, even invoices (attributes), Dates and sometimes time (hours, mins.). Often employs surrogate keys to the defined the dimensional model. It is highly de-normalized to reduce joins.

Organized hierarchies of categories, levels, and members to “slice” and query within a cube. Business perspective from which data is looked upon for collection of text attributes that are highly correlated (e.g. Product, Store, and Time).

3.1.1.1.1 Dimension Details

➤ Attributes

- ✚ Descriptive characteristics of an entity
- ✚ Building blocks of dimensions, describe each instance
- ✚ Usually text fields, with discrete values
- ✚ e.g., the flavor of a product, the size of a product

➤ Dimension Keys

- ✚ Surrogate Keys
- ✚ Candidate Business Keys

➤ Dimension Granularity

- ✚ Granularity in general is the level of detail of data contained in an entity
- ✚ A dimensions granularity is the lowest level object which uniquely identifies a member
- ✚ Typically the identifying name of a dimension

3.1.1.1.2 Dimension Keys

➤ Dimension Business Key

- ✚ Column or columns that identify a unique instance of the business record
- ✚ Used in the ETL process to tie fact records with dimension members

➤ Dimension Record Surrogate Key

- ✚ Defines the dimension’s primary key
- ✚ Relates to the fact table foreign key field
- ✚ Numeric data type, typically integer (2,4,8 byte)

3.1.1.1.3 Dimension Surrogate Keys

Surrogate Key consolidates multi-value business keys, allows tracking of dimension history, standardizes dimension tables and limits fact table width for optimization

Surrogate Key Design Practices avoids smart keys, production keys .The Company may acquire a competitor and thereby change the key building rules changed record, but deliberately not changed key.

3.1.1.1.4 Dimensions Types

- Basic Dimension Types
 - ✚ Standard Star dimension
 - ✚ Snowflake dimension
 - ✚ Parent-Child dimensions
- Advanced Types
 - ✚ Degenerate
 - ✚ Profile or Junk dimensions
 - ✚ Role-playing and Outriggers

3.1.1.2 Fact Tables:

Contain numbers and other business metrics and defines the basic measures users want to analyze in business process. Numbers are then aggregated according to related dimensions. A fact table contains dimension keys and defines relationship between measures and dimensions using surrogate keys. Typically they consists of narrow tables, but often very large in number.

The fact is a measure that is being tracked for quantity, count, amount, and percent. Mostly numerical, continuous values e.g., price of a product, quantity sold, and number of products in inventory, budget value, count of customers, and count of sales, account balance.

3.1.1.2.1 Fact Table Types

- Transactional Fact Tables
- Snapshot Fact Tables.
- Factless Fact Tables.

3.1.1.2.1.1 Transactional Fact Tables

Track the occurrence of events, each detailed event is captured into a row in the fact table. Measures are typically additive across all dimensions. Common transactional fact table types – Sales, Visits, Web-page hits, Account Transactions.

3.1.1.2.1.2 Snapshot Fact Tables

Snapshots inventory level fact tables in time of quantities in stock or balances of accounts. Time dimension used to identify grain. On additive measures across time, but typically additive across all other dimensions. Common transactional fact table types Inventory levels, Event booking levels, Chart of account balance levels.

3.1.1.2.1.3 Factless Fact Tables

Are useful to describe events and coverage the information that something has or has not happened. Often used to represent many-to-many relationships. It contains only dimension keys and few common factless fact tables are Class attendance, Event tracking, Coverage tables, and Promotion or campaign facts.

3.1.1.2.2 Measures Classified

- Aggregations
- Additive nature
- Fact table Granularity

3.1.1.2.2.1 Aggregations

A summarization of base-level fact table records and aggregations need to account for the additive nature of the measures, created on-the-fly or by pre-aggregation. Common aggregation scenarios are category product aggregate by store by day, District store aggs by product by day. Common aggregations are Sum, Count, Distinct Count, Max, Min, Average, Semi-additive (Last Child, Last Non-empty Child)

3.1.1.2.2.2 Additive Nature

Additive Facts that can be summed up/aggregated across all of the dimensions in the fact table (e.g., quantity sold, dollars sold).Semi-Additive: Facts that can be summed up for some of the dimensions in the fact table, (e.g., account balances, inventory level, distinct counts).

Non-Additive: Facts that cannot be summed up for any of the dimensions present in the fact table (e.g., measurement of room temperature).

3.1.1.2.3 Fact Table Granularity

It's the level of detail of data contained in the fact table. Description of a single instance can be a record of the fact table. Typically includes a time level and distinct combinations of other dimensions e.g. Daily item totals by product, by store, Weekly snapshot of store inventory by product.

3.1.2 Dimensional normalization VS Transactional normalization

Logical design technique that presents data in an intuitive way allowing high-performance access. The Targets decision support information and focused on easy user navigation and high performance design. Logical design technique to eliminate data redundancy, to keep data consistency, and storage efficiency. Makes transactions simple and deterministic. ER models for enterprise are usually complex often containing hundreds, or even thousands, of entities/tables.

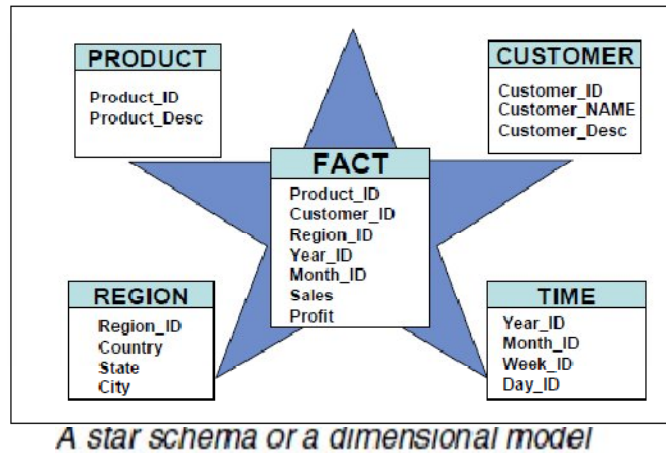
3.2 Dimension Modeling

3.2.1 Introduction

To overcome performance issues for large queries in the data warehouse, we use dimensional models. The dimensional modeling approach provides a way to improve query performance for summary reports without affecting data integrity. However, that performance comes with a cost for extra storage space. A dimensional database generally requires much more space than its relational counterpart. However, with the ever decreasing costs of storage space, that cost is becoming less significant.

3.2.2 What is a dimensional model?

A dimensional model is also commonly called a star schema. This type of model is very popular in data warehousing because it can provide much better query performance, especially on very large queries, than an E/R model. It consists, typically, of a large table of facts (known as a fact table), with a number of other tables surrounding it that contain descriptive data, called dimensions. When it is drawn, it resembles the shape of a star, therefore the name.



The dimensional model consists of two types of tables having different characteristics. They are:

- Fact table
- Dimension table

3.2.2.1 Fact table characteristics

The fact table contains numerical values of what you measure. For example, a fact value of 20 might mean that 20 widgets have been sold. Each fact table contains the keys to associated dimension tables. These are called foreign keys in the fact table. Fact tables typically contain a small number of columns. Compared to dimension tables, fact tables have a large number of rows. The information in a fact table has characteristics, such as: It is numerical and used to generate aggregates and summaries. Data values need to be additive or semi-additive, to enable summarization of a large number of values. All facts in Segment 2 must refer directly to the dimension keys in Segment 1 of the structure, as you see in This enables access to additional information from the dimension tables.

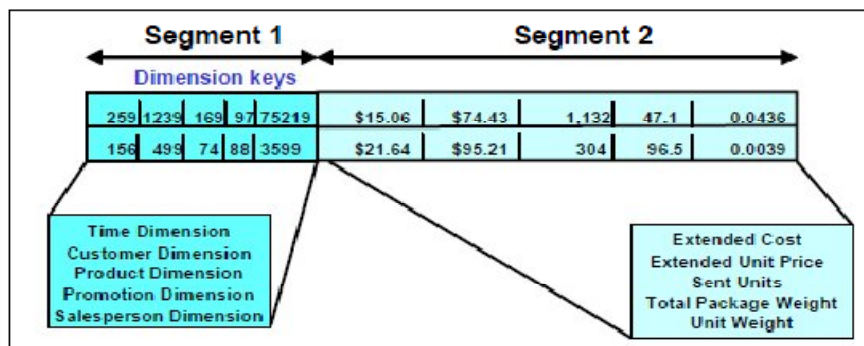


Fig: Fact Table Structure

We have depicted examples of a good fact table design and a bad fact table design. The bad fact table contains data that does not follow the basic rules for fact table design. For example, the data elements in this table contain values that are:

- Not numeric. Therefore, the data cannot be sum
- Not additive. For example, the discounts and rebates are hidden in the unit price.
- Not directly related to the given key structure, which means they cannot be not additive.

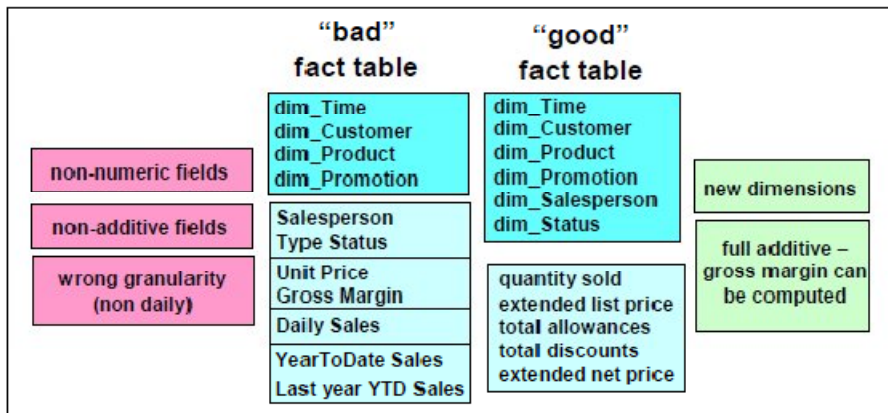


Fig: Good and

Bad Fact table

3.2.2.2 Dimension table characteristics

Dimension tables contain the details about the facts. That, as an example, enables the business analysts to better understand the data and their reports. The dimension tables contain descriptive information about the numerical values in the fact table. That is, they contain the attributes of the facts. For example, the dimension tables for a marketing analysis application might include attributes such as time period, marketing region, and product type. Since the data in a dimension table is denormalized, it typically has a large number of columns. The dimension tables typically contain significantly fewer rows of data than the fact table. The attributes in a dimension table are typically used as row and column headings in a report or query results display. For example, the textual descriptions on a report come from dimension attributes.

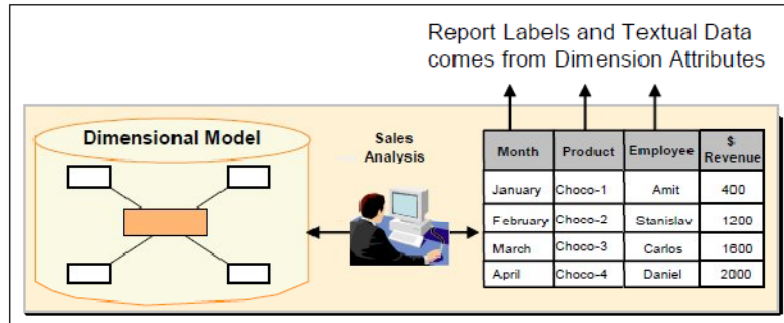


Fig: The textual data in the report comes from dimension attributes

3.2.3 Types of dimensional models

There are three basic types of dimensional models, and they are:

- Star model
- Snowflake model
- Multi-star model

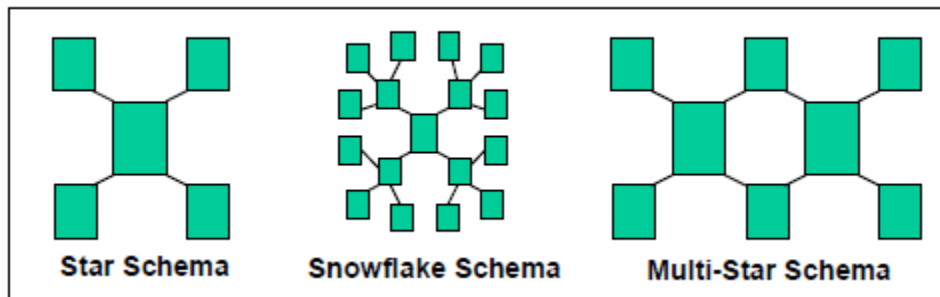


Fig: Types of Dimension Modeling

In the following section, we give a brief summary of the three types of dimensional models:

3.2.3.1 Star model

Star schemas have one fact table and several dimension tables. The dimension tables are not denormalized.

3.2.3.2 Snowflake model

Normalization and expansion of the dimension tables in a star schema result in the implementation of a snowflake design. A dimension is said to be snowflaked when the low-cardinality columns in the dimension have been removed to separate normalized tables that then link back into the original dimension table.

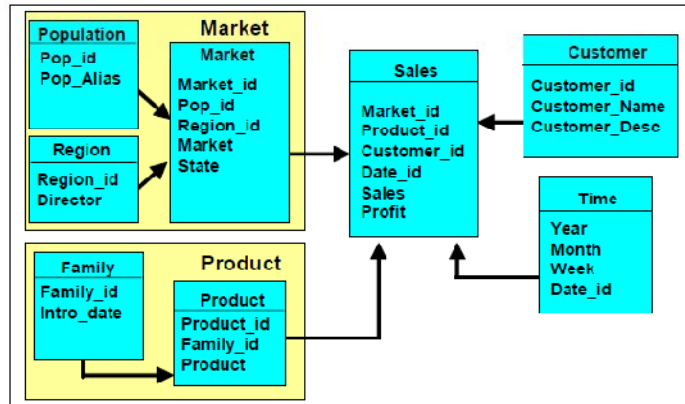


Fig: Snowflake Schema

In this example, we expanded (snowflaked) the Product dimension by removing the low-cardinality elements pertaining to Family, and putting them in a separate Family dimension table. The Family table is linked to the Product dimension table by an index entry (Family_id) in both tables. From the Product dimension table, the Family attributes are extracted by, in this example, the Family Intro_date. The keys of the hierarchy (Family_Family_id) are also included in the Family table. In a similar fashion, the Market dimension was snowflaked.

3.2.3.3 Multi-star model

A multi-star model is a model that consists of multiple fact tables, joined together through dimensions.

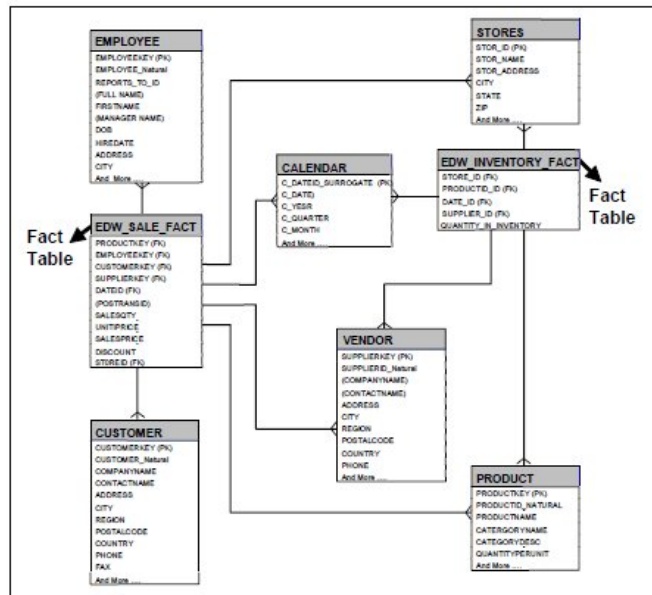
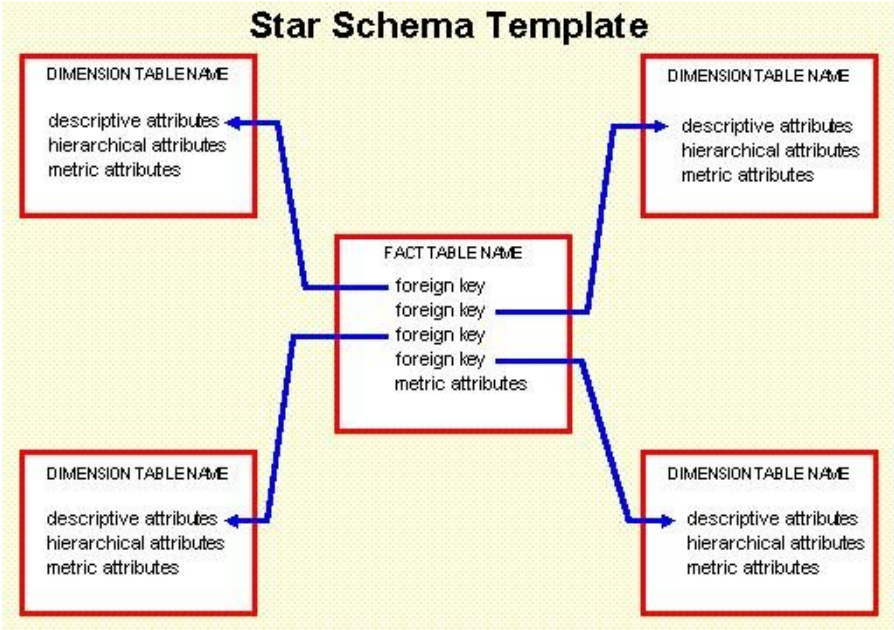


Fig: Multi Star Model

3.3 Star Schema

3.3.1 Introduction

Star schema architecture is the simplest data warehouse design. The main feature of a star schema is a table at the center, called the fact table and the dimension tables which allow browsing of specific categories, summarizing, drill-downs and specifying criteria. The data in data warehouses and data marts is accessed by end-users. The information contained in the data warehouse/data mart must be easy for the end-user to use and access. Denormalized designs are easier for end-users to use than highly normalized designs; however these designs are more difficult to design and maintain. The Star Schema diagram graphically models the end-user's view of how the information is accessed.



3.3.2 Components of a Star Schema Diagram

The Start Schema consists of three main components:

- Fact Table and its contents: metric attributes and the foreign keys necessary to join to the dimension tables.

- Dimension Tables and their contents: reference attributes, hierarchical attributes, and metric attributes. The dimension tables are highly denormalized.
- The lines that link the Dimension Tables to the Fact Table.

3.3.2.1 Fact table

The fact table is not a typical relational database table as it is de-normalized on purpose - to enhance query response times. The fact table typically contains records that are ready to explore, usually with ad hoc queries. Records in the fact table are often referred to as events, due to the time-variant nature of a data warehouse environment. The primary key for the fact table is a composite of all the columns except numeric values / scores (like QUANTITY, TURNOVER, exact invoice date and time).

Typical fact tables in a global enterprise data warehouse are (usually there may be additional company or business specific fact tables):

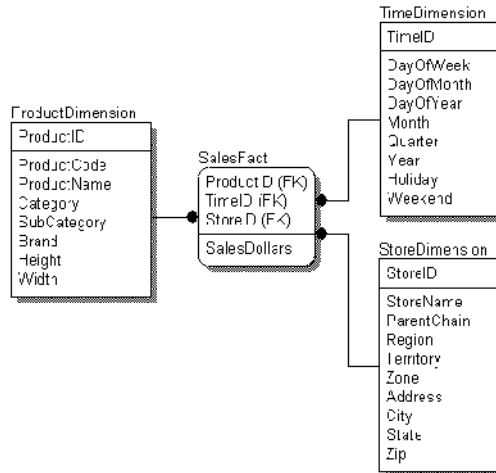
- Sales fact table - contains all details regarding sales
- Orders fact table - in some cases the table can be split into open orders and historical orders. Sometimes the values for historical orders are stored in a sales fact table.
- Budget fact table - usually grouped by month and loaded once at the end of a year.
- Forecast fact table - usually grouped by month and loaded daily, weekly or monthly.
- Inventory fact table - report stocks, usually refreshed daily

3.3.2.1.1 Building a fact table

The Fact Table holds the measures, or facts. The measures are numeric and additive across some or all of the dimensions. For example, sales are numeric and users can look at total sales for a product, or category, or subcategory, and by any time period. The sales figures are valid no matter how the data is sliced.

While the dimension tables are short and fat, the fact tables are generally long and skinny. They are long because they can hold the number of records represented by the product of the counts in all the dimension tables.

Shown below diagram is the example for simplified star schema:



3.3.2.2 Dimension table

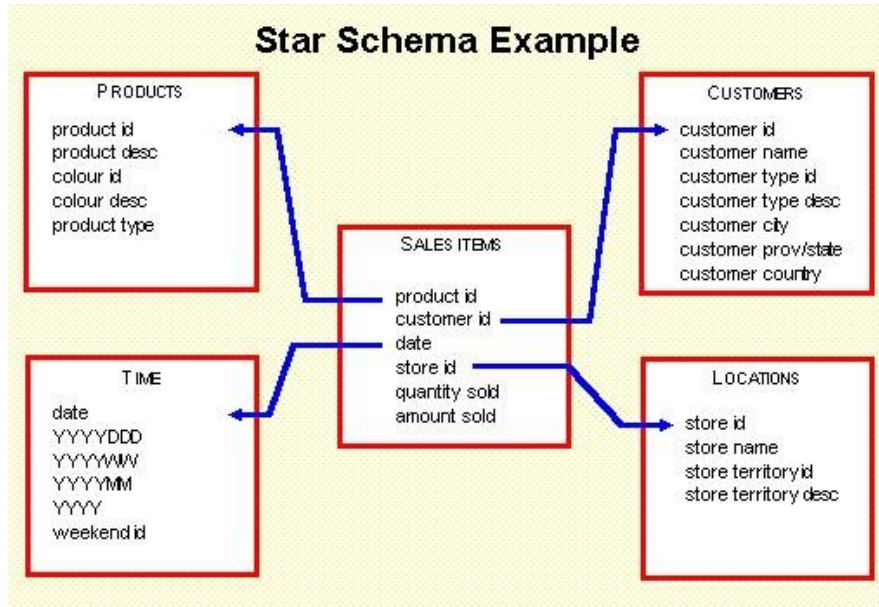
Nearly all of the information in a typical fact table is also present in one or more dimension tables. The main purpose of maintaining Dimension Tables is to allow browsing the categories quickly and easily. The primary keys of each of the dimension tables are linked together to form the composite primary key of the fact table. In a star schema design, there is only one de-normalized table for a given dimension. A dimensional modeling technique in which a detail fact table is linked to dimension tables.

Typical dimension tables in a data warehouse are:

- Time dimension table
- Customers dimension table
- Products dimension table
- Key account managers (KAM) dimension table
- Sales office dimension table

3.3.3 Star Schema Example

The following is an example of a star schema for sales items.



3.3.4 Design of star Schema

The process of building a data warehouse, it is important to understand how to move from a standard, on-line transaction processing (OLTP) system to a final star schema. The "by" conditions are called dimensions. There is almost always a time dimension on anything. Like sales by category or by product. These "by" conditions will map into dimensions: there is almost always a time dimension, and product and geography dimensions are very common as well. Therefore, in designing a star schema, the first order of business is usually to determine what people want to see (the measures) and how they want to see it (the dimensions).

3.3.5 Types of Star Schema

There are two types of Star Schema

- Classical Star Schema
- BW Star Schema

3.3.5.1 Classis Star Schema

The traditional star schema to successfully query, analyze and build reports in SAP BW for in any reporting tool. It forms the basis of a multi-dimensional model. It is designed with a Fact Table (with Fact's) in the center with primary keys pointing to Dimensions, which are extended around and they are connected to the Fact Table with dimension keys. Essentially it has Fact's in the middle with dimension attributes surrounding it. The dimensions and the fact table are linked to one another using abstract identification numbers (dimension IDs) which are contained in the key part of the particular database table. Characteristics that logically belong

together are grouped together in a dimension. The dimension attribute that will give the finest detail of information in the dimension form the primary in the fact table. For Ex: “Customer ID” in the Fact table will be the primary key for the Fact Table and is known as DIM ID in customer Dimension. Facts table contains facts/data. The fact table and dimension tables are both relational database tables.

3.3.5.2 Extended Star Scheme/BW Star Schema

It is an enhanced version of the classic star schema because the dimensional tables do not contain master data information. This master data information is stored in separate tables, called master data tables. The facts in the fact table are referred to as key figures and dimension attributes as characteristics. Characteristics are not the components of dimension table as in classic star schema. Schema is extended beyond the dimension table for each characteristics and a numeric SID key is generated for each characteristics called as SID ID (Surrogate ID). Each SID had reference to Master (attribute), Texts, and Hierarchies. An attribute should reside in a dimension table and thus in the InfoCube or in a master table or even both. Data is loaded separately into the master data (attributes), text & hierarchy tables. The fact table can contain 16 dimensions and 3 are predefined Time, Unit and Package. The Time dimension holds the time characteristics needed for analysis.

3.3.5.3 Advantages of SAP BW Star Schema over classic star schema

- SID keys reference enables faster data access
- “Slowly changing dimension” capable
- Multi language capability
- Master data is used by several InfoCube

3.3.6 OLAP and Star Schema

OLAP stands for Online Analytical Processing. OLAP is a term that means many things to many people. Here, the term OLAP and Star Schema are basically interchangeable. The assumption is that a star schema database is an OLAP system. An OLAP system consists of a relational database designed for the speed of retrieval, not transactions, and holds read-only, historical, and possibly aggregated data.

While an OLAP/Star Schema may be the actual data warehouse, most companies build cube structures from the relational data warehouse in order to provide faster, more powerful analysis on the data.