A Multi-Dimensional Data Model
• A Data Warehouse is based on a Multidimensional data model which views data in the form of a data cube

• A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
  – Dimension tables, such as item (item name, brand, type), or time (day, week, month, quarter, year)
  – Fact table contains measures (such as dollars) and keys to each of the related dimension tables
Conceptual Modeling of Data Warehouses

- **Modeling data warehouses: dimensions & measures**
  - **Star schema**: A fact table in the middle connected to a set of dimension tables
  - **Snowflake schema**: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - **Fact constellations**: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation
Example of Star Schema

**Time**
- time_key
- day
- day_of_the_week
- month
- quarter
- year

**Location**
- location_key
- street
- city
- province_or_street
- country

**Item**
- item_key
- item_name
- brand
- type
- supplier_type

**Branch**
- branch_key
- branch_name
- branch_type

**Sales Fact Table**
- time_key
- item_key
- branch_key
- location_key
- units_sold
- dollars_sold
- avg_sales

**Measures**
Example of Snowflake Schema

- **time**
  - time_key
  - day
  - day_of_the_week
  - month
  - quarter
  - year

- **branch**
  - branch_key
  - branch_name
  - branch_type

- **Sales Fact Table**
  - time_key
  - item_key
  - branch_key
  - location_key
  - units_sold
  - dollars_sold
  - avg_sales

- **item**
  - item_key
  - item_name
  - brand
  - type
  - supplier_key

- **supplier**
  - supplier_key
  - supplier_type

- **location**
  - location_key
  - street
  - city_key

- **city**
  - city_key
  - city
  - province_or_street
  - country
Example of Fact Constellation

Sales Fact Table
- time_key
- item_key
- branch_key
- location_key
- units_sold
- dollars_sold
- avg_sales

Measures
- item_key
- item_name
- brand
- type
- supplier_type

Branch
- branch_key
- branch_name
- branch_type

Location
- location_key
- street
- city
- province_or_street
- country

Shipping Fact Table
- time_key
- item_key
- shipper_key
- from_location
- to_location
- dollars_cost
- units_shipped

Shipper
- shipper_key
- shipper_name
- location_key
- shipper_type
The Classic Star Schema

- Multi-dimensional data models are needed for the creation of data warehousing or OLAP application, in other words, for analytical applications.

- The classic star schema, is the most frequently used multi-dimensional model for relational databases.

- This database schema classifies two groups of data: **Facts** (sales or quantity, for example) and **Dimension attributes** (customer, material, time, for example).

- **Facts** are the focus of the analysis of a business’ activities.

- The Fact data (values for the facts) are stored in a highly normalized **fact table**.
• The values of the **Dimension attributes** are stored in various denormalized **Dimension tables** (from a semantical point of view: The Dimensions)

• Here, logically related dimension attributes are stored as hierarchy (parent-child relationships) within the dimension table.

• The dimension tables are linked relationally with the central fact table by way of foreign or primary key relationships.

• The dimensional attribute with the finest level of detail of the corresponding dimension table is a foreign key in the fact table.

• In this way, all data records in the facts table can be identified uniquely.
The following section explains the classic star schema in greater detail using the sales example from the above graphic.

**Dimension Tables**

Customer Dimension table

<table>
<thead>
<tr>
<th>CUSTOMER_ID</th>
<th>Customer Name</th>
<th>City</th>
<th>Region</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>K100</td>
<td>Jogensen</td>
<td>Oslo</td>
<td>North</td>
<td>...</td>
</tr>
<tr>
<td>K200</td>
<td>Fourier</td>
<td>Paris</td>
<td>West</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MATERIAL_ID</th>
<th>Material Name</th>
<th>Material Group</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1111</td>
<td>Monitor</td>
<td>Hardware</td>
<td>...</td>
</tr>
<tr>
<td>M2222</td>
<td>Keyboard</td>
<td>Software</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAY_ID</th>
<th>Month</th>
<th>Quarter</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>03.01.2002</td>
<td>01.2002</td>
<td>Q1/2002</td>
<td>2002</td>
</tr>
<tr>
<td>05.08.2002</td>
<td>08.2002</td>
<td>Q3/2002</td>
<td>2002</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Figure 9:** Sales example: Dimension Tables
In a dimension table, any number of semantically-related dimension attributes are stored in a hierarchy (parent-child relationship as a 1:N-relationship).

The time dimension tables are made up the dimension-attributes ‘Year’, ‘Quarter’, ‘Month’ & ‘Day’. If a M:N relationship exists between dimension attributes, they are stored in different dimension tables.

A dimension attribute can possess any number of described attributes, also called non-dimension-attributes.

They can be used as supplemental information sources. Described attributes always have a 1:1 relationship with the dimension attribute. In the first graphic, ‘material name’ is the described attribute for the dimension attribute ‘material’ in the material dimension table.
A dimension attribute/described attribute consists of any number of values.

- For example, ‘hardware’ and ‘software’ are assigned to the dimension attribute ‘material group’, and the values ‘monitor’ and ‘keyboard’ are assigned to the described attribute ‘material name’. Here, these values are uniquely assigned to the dimension attribute ‘material group’ and are not allowed to be assigned to additional dimension attributes.

- Semantically speaking, the dimension tables in the classic star schema are often referred to as **dimensions**.

- A dimension (perspective) described a possible user’s (decision-maker’s) view of the facts.
Each classic star schema consists of one or more dimension tables.

Each dimension table has a primary key, called the **dimension key**. This key is determined by the dimension attribute with the highest granularity. In the graphic (*For more information, see Figure 9: Sales example: Dimension table*), the dimension attribute ‘DAY’(DAY_ID) is the primary key in the time dimension table. The dimension tables are linked relationally with the central fact table by way of foreign or primary key relationships.

The dimension tables are fully denormalized.

**Note:** From the OLTP perspective, the values of the dimension attributes or described attributes correspond to master data.
## Fact Table

<table>
<thead>
<tr>
<th>DAY_ID</th>
<th>CUSTOMER_ID</th>
<th>MATERIAL_ID</th>
<th>Sales Volume</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>03.01.2002</td>
<td>K100</td>
<td>M1111</td>
<td>50.000</td>
<td>100</td>
</tr>
<tr>
<td>03.01.2002</td>
<td>K100</td>
<td>M2222</td>
<td>3.000</td>
<td>60</td>
</tr>
<tr>
<td>03.01.2002</td>
<td>K200</td>
<td>M1111</td>
<td>100.000</td>
<td>250</td>
</tr>
<tr>
<td>03.01.2002</td>
<td>K2000</td>
<td>M2222</td>
<td>10.000</td>
<td>250</td>
</tr>
<tr>
<td>05.08.2002</td>
<td>K100</td>
<td>M1111</td>
<td>25.000</td>
<td>50</td>
</tr>
<tr>
<td>05.08.2002</td>
<td>K200</td>
<td>M2222</td>
<td>300</td>
<td>6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Combination of Values for Dimension Keys**

**Fact Data**

**Figure 10: Sales Example: Fact Table**
Each classic star schema is made up of exactly one fact table.

The fact table contains the fact data. It contains the facts ‘sales’ with the fact data (50,000, 3,000, 100,000…) and ‘quantity’ with the fact data (100,60,250,…)

The central fact table is connected relationally with the surrounding dimension tables via a unique key.

The primary key of the fact table is made up of all dimension keys (=foreign keys). In the above graphic, the primary key of the fact table is made up of the dimension keys ‘DAY_ID’, ‘CUSTOMER_ID’ and ‘MATERIAL_ID’.

The result is that all data records (and so all fact data) in the fact table can be identified uniquely.
**Sales Example:** The fact data (50,000, 3,000) is uniquely identified with the value combination (03.01.2002, K100, M1111) of the dimension key.

 luyện số: The fact table is highly normalized.

**Note:**

From the OLTP perspective, fact data corresponds to transaction data.

The graphic shows how the dimension tables and fact tables are arranged in a star formation.

It also shows the connections between the denormalized dimension tables and the highly normalized fact table.
### Customer Dimension Table

<table>
<thead>
<tr>
<th>CUSTOMER_ID</th>
<th>Customer Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>K100</td>
<td>Jogensen</td>
</tr>
<tr>
<td>K200</td>
<td>Fourier</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Material Dimension Table

<table>
<thead>
<tr>
<th>MATERIAL_ID</th>
<th>Material Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1111</td>
<td>Monitor</td>
</tr>
<tr>
<td>M2222</td>
<td>Keyboard</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Fact Table

<table>
<thead>
<tr>
<th>DAY_ID</th>
<th>CUSTOMER_ID</th>
<th>MATERIAL_ID</th>
<th>Sales Volume</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>03.01.2002</td>
<td>K100</td>
<td>M1111</td>
<td>50.000</td>
<td>100</td>
</tr>
<tr>
<td>03.01.2002</td>
<td>K100</td>
<td>M2222</td>
<td>3.000</td>
<td>60</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Time Dimension Table

<table>
<thead>
<tr>
<th>CUSTOMER_ID</th>
<th>Customer Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>K100</td>
<td>Jogensen</td>
</tr>
<tr>
<td>K200</td>
<td>Fourier</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

---

*Figure 11: Sales Example: Classic Star Schema*
Storing data in the form of the classic star schema is optimized for reporting. It allows the user to view facts from a variety of perspectives (Dimensions). A user may be interested in getting answers to the following questions:

Who have we sold to?

What have we sold?

How much have we sold?

When did we sell it?

From this request, the system generates a three-dimensional results structure, which can be depicted graphically as a three-dimensional (data) cube.
The structure of this kind of data cube is determined by the number of dimensions, the values of the individual dimensions attributes and the assigned cube cells.

The dimension attribute values represent the coordinates via which the cells can be accessed uniquely.

The cells only contains one entry for a particular fact. In the above graphic, the “selected” cell is addressed uniquely via the value combination (North+South+West+EAST), Hardware, 2002).

This cell contains the fact data 4,000,000(sales) and 3,000(quantity).

Multi-dimensional analysis techniques (OLAP functions) can be used to define a variety of views on the data cube/multi-dimensional data structure.

Not all decision-makers have or need the same view of the data.
The SAP BW Star Schema

• The multi-dimensional model in SAP BW is based on the SAP BW star schema, which was developed as an enhanced (refined) star schema as a response to problems experienced with the classic star schema.

• The following graphic shows the crossover between the classic star schema and the SAP BW star schema, using the same sales example from the first diagram.

• For the time being, only components relevant to the modeling view are taken into consideration.
<table>
<thead>
<tr>
<th>CUSTOMER_ID</th>
<th>Customer Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>City</td>
<td>Region</td>
</tr>
</tbody>
</table>

Customer

**Dimension Table**

<table>
<thead>
<tr>
<th>DIM_ID_CUSTOMER</th>
<th>SID_CUSTOMER</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Material Group</th>
</tr>
</thead>
</table>

**Material Dimension Table**

<table>
<thead>
<tr>
<th>DIM_ID_MATERIAL</th>
<th>SID_MATERIAL</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Material Group</th>
</tr>
</thead>
</table>

**External Hierarchy**

**Fact Table**

<table>
<thead>
<tr>
<th>DIM_ID_TIME</th>
<th>SID_DAY</th>
<th>SID_MONTH</th>
<th>SID_QUARTER</th>
<th>SID_YEAR</th>
</tr>
</thead>
</table>

Sales Volume
Quantity

**Time Dimension Table**

**Attribute Table**

<table>
<thead>
<tr>
<th>MATERIAL_ID</th>
<th>Material Name</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>MATERIAL_ID</th>
<th>Material Group</th>
</tr>
</thead>
</table>

**Text Table**

**Attribute Table**

Figure 14: The SAP BW Star Schema
• This graphic shows how the SAP BW-star schema is an enhancement of the classic star schema.
• The enhancement comes from the fact that the dimension tables do not contain master data information.
• This master data information is stored in separate tables, called master data tables.
• This section firstly explains the SAP BW-Star Schema in detail.
• At the end of the section, both star schemas are compared in terms of their advantages and disadvantages.
• In the SAP BW-star schema, the distinction is made between two self-contained areas:
  – InfoCube
  – Master Data Tables/Surrogate ID (SID-) Tables
InfoCube

- InfoCubes are the central objects of the multi-dimensional model in SAP BW. Reports and analyses are based on these. From a reporting perspective, an InfoCube describes a self-contained data set within a business area, for which you can define queries.

- An InfoCube (BasicCube) consists of a number of relational tables that are combined on a multi-dimensional basis. In other words, it consists of a central **fact table** and several surrounding **dimension tables**.

**Hint:** There are various types of InfoCube in BW. The InfoCube with type BasicCube is the InfoCube relevant for modelling, since only physical objects (objects that contain data) are considered in the modeling within the SAP BW-data model. For this reason, InfoCube always refers to BasicCube in this section. (You can find additional information about other cube types in the Virtual Cubes lesson).
Figure 15: InfoCube
SAP BW Star Schema

- In the SAP BW-star schema, the facts in the fact table are referred to as **key** figures and the dimension attributes as **characteristics**.
- The dimension table are linked relationally with the central fact table by way of foreign or primary key relationships.
- In contrast to the classic star schema, characteristics are not components of the dimension tables, in other words, the characteristic values are not stored in the dimension tables.
- A numerical **SID** key is generated for each characteristic.
- This foreign key replaces the characteristic as the component of the dimension table.
- Here, SID stands for **surrogate ID** (replacement key).
• In the graphic above, these keys are given the prefix `SID_`. For example, ‘SID_MATERIAL’, is the SID key for the characteristic ‘MATERIAL’ (‘MATERIAL_ID’).

• Each dimension table has a generated numerical ‘primary key’, called the dimension key.

• In the graphic above, this dimensions key is denoted with the prefix `DIM_ID_`.

• Here, ‘DIM_ID_MATERIAL’ is the dimension key for the material dimension table. As in the classic star schema, the primary key of the fact table is made up dimension keys (‘DIM_ID_DATENPAKET’, ‘DIM_ID_ZEIT’, ‘DIM_ID_EINHEIT’, ‘DIM_ID_KUNDE’, ‘DIM_ID_MATERIAL’).
• **Master Data Tables/SID Tables**
• Additional information about characteristics is referred to as master data in the SAP BW. A distinction is made between the following master data types:
  • **Attributes**
  • **Texts**
  • **(External) Hierarchies**
Master Data Tables/SID Tables

- Master data information is stored in separate tables, that is independent of InfoCube, in what are called **Master data tables** (separately for attributes, texts and hierarchies).

- In the following graphic, for example, the attribute ‘material group’ is stored in the **attribute table**, the text description for ‘material name’ is stored in the **text table** and the material hierarchy is stored in the **Hierarchy table** for the characteristic ‘MATERIAL’.

- In this way, the characteristic ‘MATERIAL’ is the primary key for the master data tables belonging to this characteristic.

- As was already mentioned in the *InfoCubes* section above, precisely one numerical SID key is assigned to each characteristic.
• This assignment is made in a SID table for the respect characteristic, whereby the characteristic becomes the primary key in the SID table.

• In the following graphic, the SID key ‘SID_MATERIAL’ is assigned to the characteristic ‘MATERIAL’ in the SID table for characteristic ‘MATERIAL’.

• The SID table is connected to the associated master data tables via the characteristic key.

• **Hint:** By using the term ‘Hierarchy’, we usually mean an arrangement of objects having a 1:N relationship to each other.
• In this sense, there are hierarchies in the dimension-, attribute-and hierarchy tables in BW.
• This ‘hierarchy’ term is strongly connected with the ‘drilldown’ term (pre-defined drilldown path) in data warehousing terminology.
• However, in the SAP BW, the term “drilldown’ can also be used without referring to a hierarchy.
• In SAP BW, under “external hierarchies”, we mean presentation hierarchies, which are stored in what are called hierarchy tables as a structure for characteristic values.
Connecting Master Data Tables to an InfoCube

- Master data tables are connected to an InfoCube (and thus to the key figures of the fact table) by way of the SID tables.
- The excavation of master data from the dimension tables using SID technology allows you to use the master data with different InfoCubes.
- In other words, the master data is InfoCube-independent, and can be used by several InfoCubes at the same time.
### Classic Star Schema in Comparison with the SAP BW-Star Schema

Firstly, let us compare terminology for the two schemas.

<table>
<thead>
<tr>
<th>Classic Star Schema</th>
<th>BW Star Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fact</td>
<td>Key Figure</td>
</tr>
<tr>
<td>Dimension Attribute</td>
<td>Characteristic</td>
</tr>
<tr>
<td>Described Attribute</td>
<td>Attribute Tex</td>
</tr>
<tr>
<td></td>
<td>External Hierarchies</td>
</tr>
<tr>
<td>Dimension Tables</td>
<td>Dimension Tables (contain no master data)</td>
</tr>
<tr>
<td>Dimension = Dimension Table</td>
<td>Dimension Table, SID Tables (optional)Master Data Tables</td>
</tr>
</tbody>
</table>
Advantages and Disadvantages of the Classic Star Schema

Advantages

- Data access runs performantly due to the small number of join operations (there are only join operations between the fact tables and the involved dimension tables).

Disadvantages

- Redundant entries exist in the dimension tables.
- In contrast to the historization of fact data (the same reference is given implicitly via the time dimension table), historization of dimensions (→ “slowly changing dimension”) is not easy to model.
- The multi-lingual capability is cumbersome.
- Modeling some hierarchy types (parallel and imbalanced hierarchies for example) in a dimension can lead to anomalies.
- Query performance is also made worse, since aggregates and
  
  Basis fact data stored in the same table (fact table).
Advantages of the SAP BW Star Schema

- Advantages.
  - The use of automatically generated INT4 keys I (SID keys, DIMID keys) enables faster access to data than via long alpha-numeric keys.
  
  - Thanks to the excavation of master data from the dimension tables using the SID technique, the following modeling possibilities exist:
Historizing dimensions

Multi-lingual capability

Cross-InfoCube use of master data (→”shared dimensions”)

- The query performance is improved here as aggregated key figures can be stored in their own fact tables.

Hint: Another enhancement vis-à-vis the classic star schema is the excavation of aggregated key figures in their own fact tables using the construction of aggregates that were previously not taken into account in the SAP BW-star schema. You can find additional information about aggregates in the Administering InfoCubes & Aggregates unit.
Multidimensional Data

- Sales volume as a function of product, month, and region

Dimensions: Product, Location, Time
Hierarchical summarization paths

- Industry
- Region
- Year
- Category
- Country
- Quarter
- Product
- City
- Month
- Office
- Week
- Day
A Sample Data Cube

Total annual sales of TV in U.S.A.
Cuboids Corresponding to the Cube

- 0-D(apex) cuboid
- 1-D cuboids
- 2-D cuboids
- 3-D(base) cuboid
Browsing a Data Cube

- Visualization
- OLAP capabilities
- Interactive manipulation
Metadata Repository

- Meta data is the data defining warehouse objects. It has the following kinds
  - Description of the structure of the warehouse
    - schema, view, dimensions, hierarchies, derived data definition, data mart locations and contents
  - Operational meta-data
    - data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (warehouse usage statistics, error reports, audit trails)
  - The algorithms used for summarization
  - The mapping from operational environment to the data warehouse
  - Data related to system performance
    - warehouse schema, view and derived data definitions
  - Business data
    - business terms and definitions, ownership of data, charging policies
• **Data warehouse**
  - A subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management’s decision-making process

• **A multi-dimensional model of a data warehouse**
  - Star schema, snowflake schema, fact constellations
  - A data cube consists of dimensions & measures

• **OLAP operations:** drilling, rolling, slicing, dicing and pivoting

• **OLAP servers:** ROLAP, MOLAP, HOLAP

• **Efficient computation of data cubes**
  - Partial vs. full vs. no materialization
  - Multiway array aggregation
  - Bitmap index and join index implementations

• **Further development of data cube technology**
  - Discovery-drive and multi-feature cubes
  - From OLAP to OLAM (on-line analytical mining)
End of Multi Dimensional Model