



www.juergen-konicek.de

Design Methodologies for Data Warehouses (DWHs)

This section introduces into the fundamental data warehousing objects and design concepts as well as multi-dimensional data models.

Abbreviations:

APIs: Application Programming Interfaces

BI: Business Intelligence

CASE: Computer Aided Software-Engineering

CDIF: Case Data Interchange Format

CLI: Call Level Interface

CWH: Common Warehouse Model

CORBA: Common Object Request Broker Architecture

DBMS: Database Management System

DB2 UDB: DB2 Universal Database

DC: Direct Costs

DDL: Data Definition Language

DOM: Document Object Model

DTD: Document Type Definition

DWH: Data Warehouse

ECMA: European Computer Manufacturers Association

EIA: Electronic Industries Associations

ERM: Entity-Relationship Model

ETL: Extraction Transformation Loading

ERP: Enterprise Resource Planning

GIF: Graphics Interchange Format

IDL: Interface Definition Language

IRDS: Information Resource Dictionary System

ISO: International Organization for Standardization

MDC: Meta Data Coalition

MDIS: Metadata Interchange Specification

MDL: Model Definition Language

In the area of analysis-orientated database systems and DWH design, there are various modelling concepts for representing multidimensional schemas, namely:

- the **star schema**,
- the **snowflake schema**,
- the hybrid form: **starflake schema**,
- the **galaxy schema**.

Each DWH schema is build on fact tables and dimension tables. The fact tables are the core tables in the middle of a dimension schema. A fact table is structured by numeric facts (variables or measures [BuG01, page 200]) and foreign keys to join dimension tables as displayed in [figure 2](#). The facts are usually historical and quantitative business transactions for analysis such as turnover or sales volume. In general, fact tables provide numeric, additive fields and can be aggregated by arithmetical operations [ORA03, Data Warehousing Objects] and they are associated with multiple dimension tables. Dimension tables, also called "lookup or reference tables" [ORA03, DataWarehousing Objects], stores usually textual and qualitative values for specification the measures in the fact tables. In general, they are composed of a primary key and dimension attributes (see [figure 2](#)). Dimension attributes are organised in certain logical structures, which enables to drill down or up fact data along a defined dimension hierarchy "to view different levels of granularity" [ORA03, Data Warehousing Objects]. For instance, an example Time Dimension table could be contain attributes in the following hierarchical order: Year - Quarter - Month - Day (see [figure 1](#)).

An overall DWH and also appropriate data marts are main parts of BI architectures. They can be implemented by different relational or multidimensional schemas as mentioned previously.

- MOF:** Meta Object Facility
- mUML:** Multidimensional UML
- MS:** Microsoft
- OC:** Overhead Costs
- OCL:** Object Constraint Language
- ODBC:** Open Database Connectivity
- ODBMS:** Object-Orientated Database Management System
- OLAP:** Online Analytical Processing
- OMS:** Object Management System
- OOD:** object-oriented design
- OMG:** Object Management Group
- PCTE:** Portable Common Tool Environment
- RPCS:** Remote Procedure Calls
- SQL:** Structured Query Language
- SVG:** Scalable Vector Graphics
- UML:** Unified Modeling Language
- UDFs:** User Defined Functions
- WWW:** World Wide Web
- W3C:** World Wide Web Consortium
- XMI:** XML Meta Data Intechange
- XML:** Extensible Markup Language

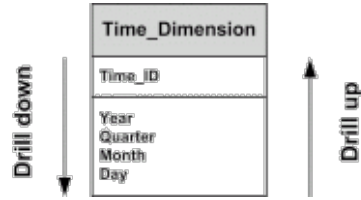


Figure 1: Example for a dimension hierarchy and navigation along a dimension table.

In the following, there is a short outline of the common used and aforementioned multidimensional data schemas:

- The **star schema** is a star-shaped representation of denormalised dimension tables around a fact table [BuG01, page 201] (see figure 2a). In general, the design of dimension tables and fact tables are based on 1:n relationships. The denormalised star schema provides an optimised query performance.
- The **snowflake schema** is an extended star schema that includes further normalised dimensions to prevent possible occurrence of update anomalies [BuG01, page 201] and to eliminate redundancy. In other words, the dimension tables are broken up into subdimension tables instead of one large table which causes a reduced query performance. Figure 2b) presents a graphical representation of a snowflake schema.

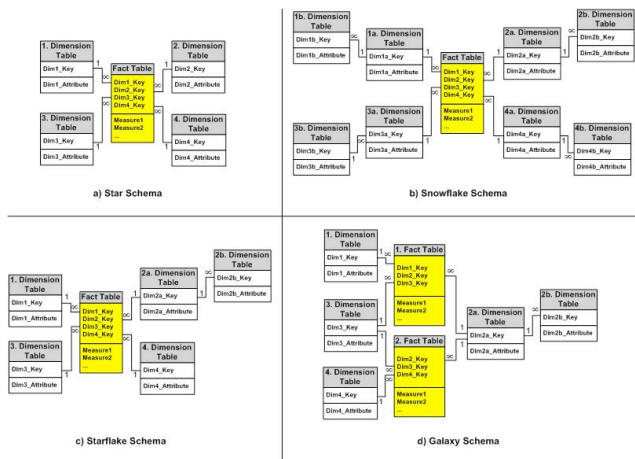


Figure 2: Variety of common schema models for data warehousing (derived from [BuG01, page 201]).

References:

[CBS02]

T. Connolly, C. Begg and A. Strachan: Datenbanksysteme; Eine praktische Anleitung zu Design, Implementierung und Management; Addison-Wesley (2002).

[Cog02]

Cognos BI Guide, Der Business Intelligence-Leitfaden; Cognos GmbH, Frankfurt am Main; 3. Auflage (2002)

[Kon04]

Jürgen Konicek: Metadata within Business Intelligence Solutions, Thesis - University of Applied Science of Ulm (Commercial Information Systems) in cooperation with the Napier University, Edinburgh (UK) (School of Computing) TERM 2003/04 (2004).

[ORA03]

Oracle9i Database CD-Rom Documentation, Release 2 (9.2)

- The **starflake schema** is a hybrid of the star and snowflake schema [BuG01, page 204] (see [figure 2c](#)). The starflake is composed of a central fact table and a set of constituent denormalised and normalised dimension tables. In contrast to the denormalised tables, the normalised tables are further broken up into subdimension tables.

- The **galaxy schema** or "multiple fact tables schema" [BuG01, page 205] is composed of multiple fact tables, which are associated partially with the same dimension tables [BuG01, page 204] as shown in [figure 3d](#).

For a general description of dimension tables in particular with regard to the *Time_Dimension* in [figure 1](#), a formal notation is introduced in the following. According to the notation in [BuG01], the data cube in [figure 4](#) can be stated here in the form given in [BuG01, page 222-223]:

- In general a dimension D with m dimension values can be defined formally as a m tuple (a sorted list of m values):

$$D = \{x^{D_1}, x^{D_2}, \dots, x^{D_m}\}$$

- Consequently, $|D| = m$ defines the number of dimension values of the dimension D . For instance the dimension Time Dimension (see [figure 2](#)) with five dimension values (*Time ID*, *Year*, *Quarter*, *Month* and *Day*) can be generally characterised as:

$$|D_{Time\ Dimension}| = 5.$$

By Jürgen Konicek