### DATA MODELS

A Database management system is a hardware/software system that consists of a database and necessary programs to carry out database management activities rolled in one. Users do not have to interact with the data files directly but through a language interface. Using a database management system for data storage and manipulation relieves the users from the bookkeeping responsibilities.

For simple data manipulation users do not have to write programs, instead the programs are already written in the database which can be invoked by user by issuing one or more command in a database language. For more complex manipulations programming interface is also included in a DBMS.

Most of the currently popular DBMS have more complex architecture. Moreover not only human users but other programs also can interact with the DBMS. In this arrangement the database is commonly known as back-end while the user program is called front-end. Front-end applications can be designed to suite any individuals or organizations specified data-interaction needs.

With Different types of DBMS can take different approaches manage the data? Each approach constitutes database.

Major database model types are:

### Hierarchical data model

### Network Database model

### Relational Database model

### Object oriented database model

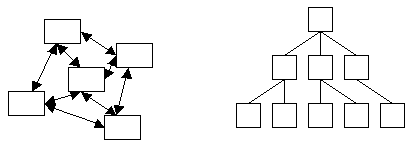
### Deductive database model

### Database systems can be based on different data models or database models respectively. A data model is a collection of concepts and rules for the description of the structure of the database.

Structure of the database means the data types, the constraints and the relationships for the description or storage of data respectively.

The most often used data models are:

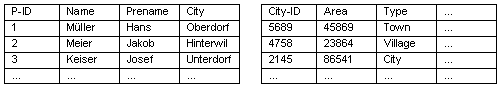
|  |  |
| --- | --- |
| Network Model and  Hierarchical Model | The network model and the hierarchical model are the predecessors of  the relational model. They build upon individual data sets and are able to express hierarchical or network like structures of the real world. |



Network Model and Hierarchical

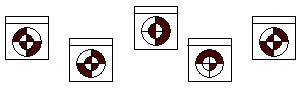
Model

|  |  |
| --- | --- |
| Relational Model | The relational model is the best known and in today’s DBMS most often  Implemented database model. It defines a database as a collection of tables (relations) which contain all data.  This module deals predominantly with the relational database model and the database systems based on it. |

Relational

Database Model

|  |  |
| --- | --- |
| Object-oriented Model | Object-oriented models define a database as a collection of objects with  features and methods. A detailed discussion of object-oriented databases follows in an advanced module. |

Schematic Representation of a Object-oriented

Database Model

### Database Schemes and Database Instances

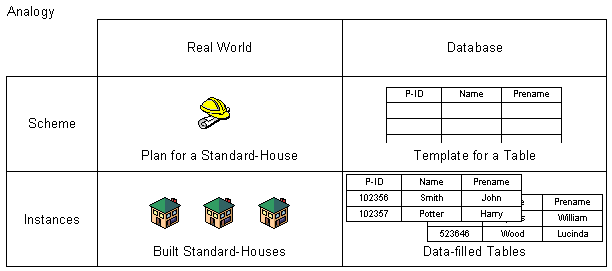
|  |  |
| --- | --- |
| Object-relational  Model | Object-oriented models are very powerful but also quite complex. With  the relatively new object-relational database model is the wide spread and simple relational database model extended by some basic object- oriented concepts. These allow us to work with the widely know relational database model but also have some advantages of the object- oriented model without its complexity. |

relational Database Model

Independent from the database model it is important to differentiate between the description of the database and the database itself. The description of the database is called **database scheme** or also [metadata.](http://www.gitta.info/DBSysConcept/en/html/DBSysConcept_glossary.html#d9e307) The database scheme is defined during the database design process and changes very rarely afterwards.

The actual content of the database, the data, changes often over the years. A database state at a specific time defined through the currently existing content and relationship and their attributes is called a **database instance**

The following illustration shows that a database scheme could be looked at like a template or building plan for one or several database instances.

Analogy Database Schemes and Building Plans

When designing a database it is differentiated between two levels of abstraction and their respective data schemes, the conceptual and the logical data scheme.

**Conceptual Data Scheme:**

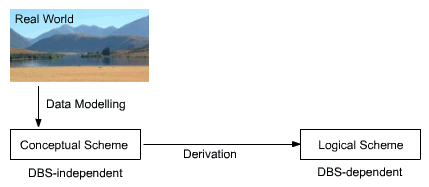
A conceptual data scheme is a system independent data description. That means that it is independent from the database or computer systems used. (Translated) ([ZEHNDER](http://www.gitta.info/DBSysConcept/en/html/DBSysConcept_bibliography.html#d9e333) 1998)

**Logical Data Scheme:**

A logical data scheme describes the data in a data definition language DDL of a specific database management system. (Translated) ([ZEHNDER](http://www.gitta.info/DBSysConcept/en/html/DBSysConcept_bibliography.html#d9e333)1998)

The conceptual data scheme orients itself exclusively by the database application and therefore by the real world. It does not consider any data technical infrastructure like DBMS or computer systems, which are eventually employed. [Entity relationship diagrams](http://www.gitta.info/DBSysConcept/en/html/DBSysConcept_glossary.html#d9e315) and relations are tools for the development of a conceptual scheme.

When designing a database the conceptual data scheme is derived from the logical data scheme (see unit [Relational Database Design](http://www.gitta.info/LogicModelin/en/html/index.html)). This derivation results in a logical data scheme for one specific application and one specific DBMS. A DB-Development System converts then the logical scheme directly into instructions for the DBMS.



The different schemas of three tier architecture

Knowing about the conceptual and the derived logical scheme (discussed in unit [Database Models, Schemes and Instances](http://www.gitta.info/DBSysConcept/en/html/unit_DataModSchem.html) this unit explains two additional schemes - the external scheme and the internal scheme - which help to understand the DBMS architecture.

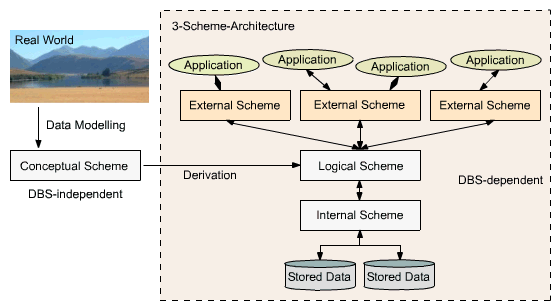
**External Scheme:**

An external data scheme describes the information about the user view of specificusers (single users and user groups) and the specific methods and constraints connected with this information. (Translated) ([ZEHNDER](http://www.gitta.info/DBSysConcept/en/html/DBSysConcept_bibliography.html#d9e333)1998)

**Internal Scheme:**

The internal data scheme describes the content of the data and the required service functionality which is used for the operation of the DBMS. (Translated) ([ZEHNDER](http://www.gitta.info/DBSysConcept/en/html/DBSysConcept_bibliography.html#d9e333) 1998)

Therefore, the internal scheme describes the data from a view very close to the computer or system in general. It completes the logical scheme with data technical aspects like storage methods or help functions for more efficiency.

****

**Three – tier Schemes Architecture**

The right hand side of the representation above is also called the three-scheme architecture: internal, logical and external scheme.

While the internal scheme describes the physical grouping of the data and the use of the storage space, the logical scheme (derived from the conceptual scheme) describes the basic construction of the data structure. The external scheme of a specific application, generally, only highlights that part of the logical scheme which is relevant for its application. Therefore, a database has exactly one internal and one logical scheme but may have several external schemes for several applications using this database.

The aim of the three-scheme architecture is the separation of the user applications from the physical database, the stored data. Physically the data is only existent on the internal level while other forms of representation are calculated or derived respectively if needed. The DBMS has the task to realize this representation between each of these levels.

### Data Independence

With knowledge about the three-scheme architecture the term data independence can be explained as followed: Each higher level of the data architecture is immune to changes of the next lower level of the architecture.

**Physical Independence:**

Therefore, the logical scheme may stay unchanged even though the storage space or type of some data is changed for reasons of optimization or re organization.

**Logical Independence:**

Also the external scheme may stay unchanged for most changes of the logical scheme. This is especially desirable as in this case the application software does not need to be modified or newly translated.