

## Data Independence

Database systems keep data in order so that people can extract information from it easily. As we have said this implies that the users must have a mental picture of the sort of data the computer stores and a language for communicating with the computer about the data. Last week we covered the users' view of how the data is stored. In practice this need not have anything to do with the way the data is really stored provided the database system is capable of dealing with the translation. And given that the computer can deal with this dichotomy, there is no reason to stop at one view of the data. Many different users may have different views of the same data at the same time. This difference between how data is really stored and how its users like to think it is stored provides **data independence** and is one of the fundamental motivations for using any database system.

The way database systems achieve data independence is usually illustrated by the diagram on the next page which represents the **three level architecture**.

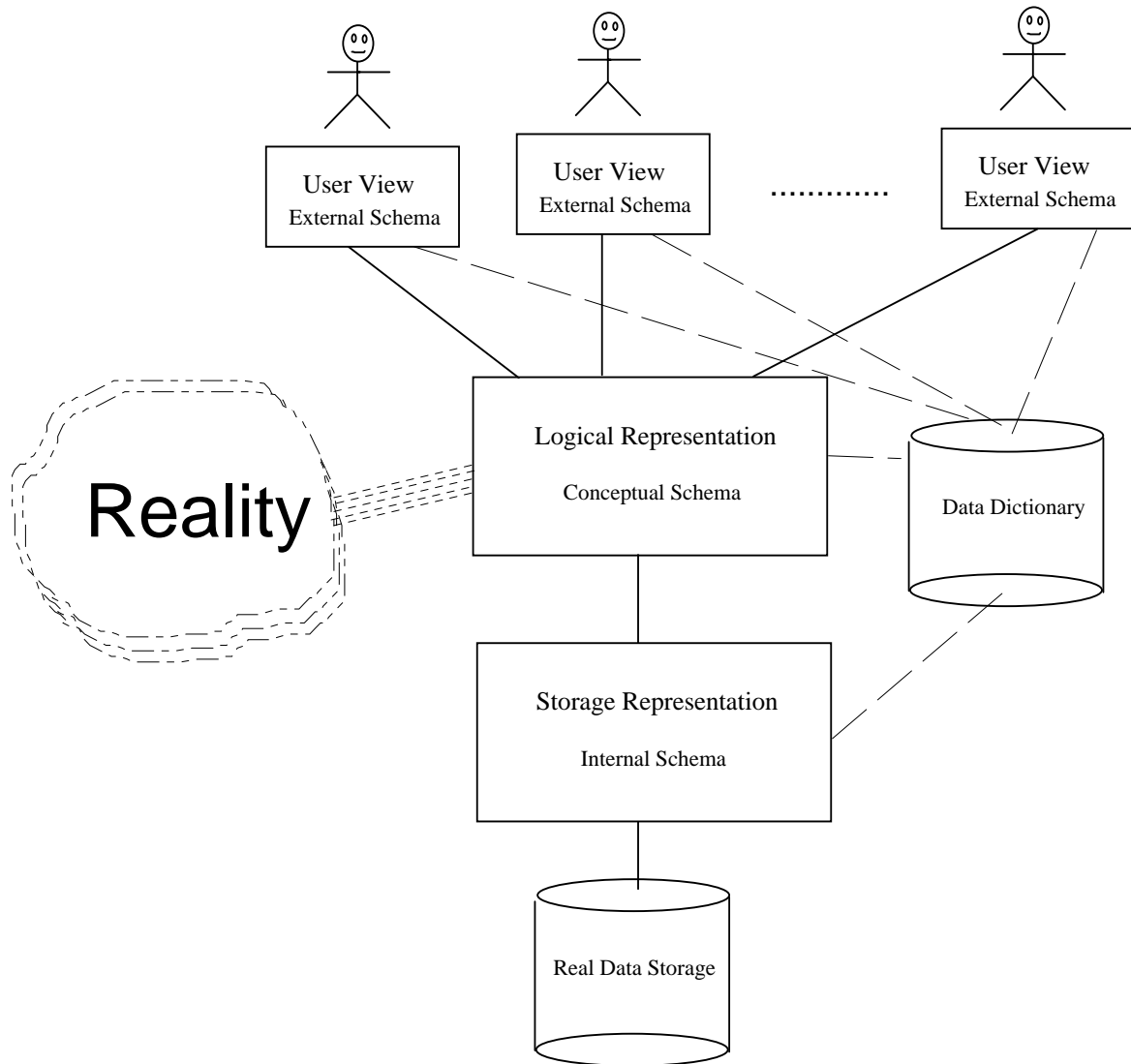
In any database system the data stored is a **model** of things (or processes) that exist (or are happening) in the real world. Computers do not provide a very realistic representation of reality but the **logical data model** is the best that can be achieved. It is a global view of all the data stored by an organization and, to a certain extent, what it means. The logical data model is described in a **Data Definition Language (DDL)**; a language that describes what is stored on the computer. The description of the data is called the **schema**.

Not all users need to know about all the data. In fact, to a large extent, the less the users know (and can change) the better. The users therefore have their own individual **views** (described by **sub schemas**) of the data. Different users can see different subsets of the data, they can identify it in different ways and they can have permission to change different bits of it. The users read and change the data using a language called a **Data Manipulation Language (DML)**. The separation between what the users see as the database and the underlying logical data model provides **logical data independence**.

At the bottom level there is the real data storage. This is conventionally illustrated by a cylinder but the actual storage medium is irrelevant. In fact one of the main points of the three level architecture is to keep the way the database works independent of the details of the machine it works on. This means that it is theoretically possible to replace all the storage by a completely different sort of storage without affecting the users of the data. This property is known as **physical data independence**. There is one other important distinction between data storage in a database system and that used in a conventional program; the data in a database will survive a power failure intact.

To the right of this diagram is the **Data Dictionary**. A data dictionary is a permanent store for **meta-data**; data about data. It can be used for reference purposes by all the levels of the database system and often by the system's users too.

## The Three-Level Architecture



This diagram does not represent the way the database system works, just how it separates different levels of data. The way that different database systems actually process the data varies; newer systems are based on the client-server model with some of the processing done on user's own machine whereas others are centralized and the user's machines simply display data and accept input.