

Lecture Handout

Database Management System

Lecture No. 03

Reading Material

“Database Systems Principles, Design and Implementation” written by Catherine Ricardo, Maxwell Macmillan.	Section: 4.1.1, 4.1.2
--	-----------------------

Overview of Lecture

- Database Architecture
- External View of the database
- Conceptual view of the database

Database Architecture:

Standardization of database systems is a very beneficent in terms of future growth, because once a system is defined to follow a specific standard, or is built on a specific standard, it provides us the ease of use in a number of aspects.

First if any organization is going to create a new system of the same usage shall create the system according to the standards and it will be easier to develop, because the standards which are already define will be used for developing the system.

Secondly if any organization wants to create and application software that will provide additional support to the system, it will be an easier task for them to develop such system and integrate them into existing database applications.

Users which will be using the system will be comfortable with the system because a system built on predefined standards is easy to understand and use, rather than understanding learning and using an altogether new system which is designed and built without following any standards.

Expansion to systems which are not built on standards is very hard and needs lots of efforts.

Technical staff working on a system built on standard has no problem to learn the use and architecture of the system and whenever there is a need in change of staff new staff members can be hired and put to work without any prior training for the use of system.

Database standard proposed by ANSI SPARK in 1975 is being used worldwide and is the only most popular agreed upon standard for database systems.

The Three Level Schema architecture provides us a number of benefits. For accessing data at different levels we have a number of users because not all users have to access data in database at all the database levels. The 3 levels architecture allows us to separate the physical representation of data from the users' views of data.

In the database, same data is stored in a specific feasible format and is available to different users in different formats as desired by different users. For example, consider we have stored the DOB (Date of Birth) in the database in a particular format, like in the form of dd-mm-yyyy (for example, 28-03-1987). However, the users from different departments may require to view the date of birth in different forms; the examination department may ask it to be displayed as month-day-yyyy (like march-28-1987) the Registrar's office may ask to display date of birth as mm/dd/yyyy, still the Library may need the in the form of dd/mm/yy. The Three Level Schema allows us to access the data in different formats at the external level, which is stored in a specific format at the internal level.

The Three levels architecture is useful for hiding the details of internal systems; it in-fact hides the details of underlying system views from the users at other levels and restricts the access of data and the system from any unauthorized intervention. It is the mechanism which allows us to store the data in the system in such a way that it can be provided to all users in their desired formats and with unveiling other details and information stored in the database. Moreover if there is a change to be done to the data stored in the database subject to the requirements of a specific user it needs not be changed for that user specifically, we can create a change to the specific external view of that user and the internal details remain unchanged. Also if we want to change the underlying storage mechanism of the data stored on the disk we can do it without affecting the internal and conceptual view at the lowest level in the three levels architecture is the internal view or internal level which is shown below in the diagram and is illustrated in the coming lines.

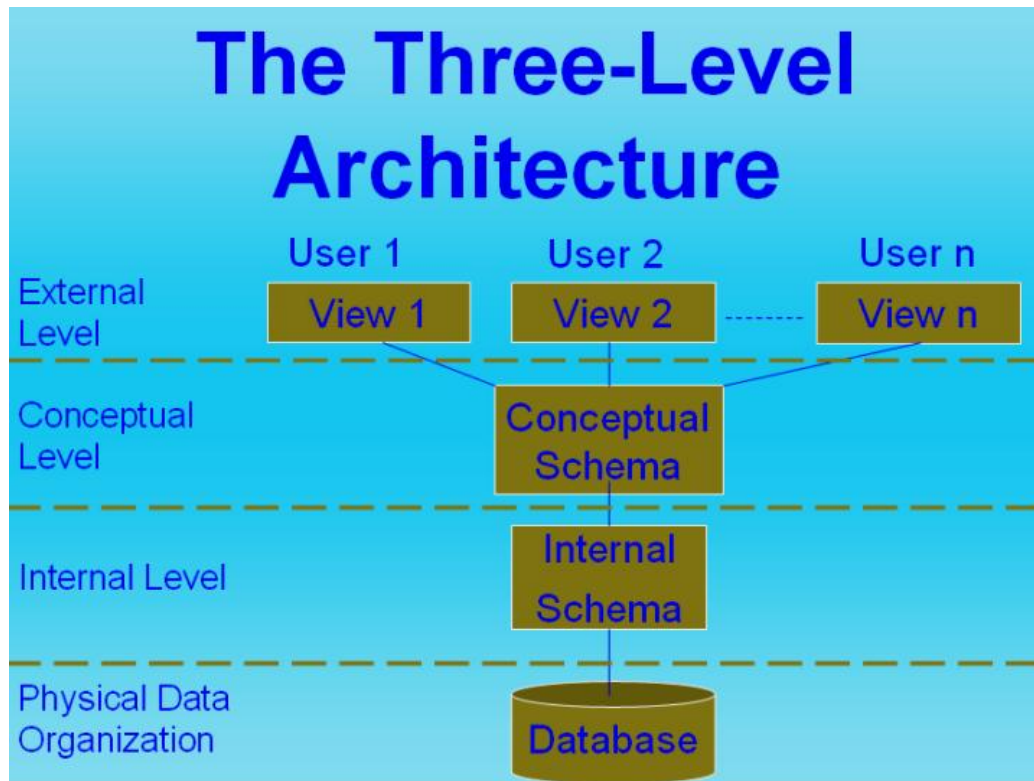


Fig. 1: Three level architecture of database

The Architecture:

The schemas as it has been defined already; is the repository used for storing definitions of the structures used in database, it can be anything from any entity to the whole organization. For this purpose the architecture defines different schemas stored at different levels for isolating the details one level from the other.

Different levels existing at different levels of the database architecture are expressed below with emphasis on the details of all the levels individually.

Core of the database architecture is the internal level of schema which is discussed a bit before getting into the details of each level individually. The internal level implements all the inner details and defines the intentions of the database. Internal schema or view defines the structures of the data and other data related activities in the database. For example it defines that for a student what data will be stored in terms of attributes of the student and it also defines how different values for these attributes will be stored, also it tells that who is allowed to make changes to the database and what changes he can make, etc.

These details give us the internal schema and are called the intention of the database. Intention for a database is almost permanent, because while designing the database it

is ensured that no information is left behind which is important enough to be stored in the database and what information is important to be stored in the database from the future point of view.

Once the intention of the database has been defined then it is undesirable to change the intention for any reason. Because any small change in the intention of the database may need a lot of changes to be made to the data stored in the database. Extension of the database is performed on the bases of a complete intention, i-e once a database has been defined it is populated with the data of the organization for which the database is created. This population of the database is also called as the extension of the database. Extension is always done according to the rules defined in the internal schema design or the intention of the database.

Effects of changes made to different levels of the database architecture:

We can make changes to the different levels of the database but these changes need very serious consideration before they are made, Changes at different levels of database architecture need different levels of users attention for example a change to the data made for the extension of data will effect only a single record whereas when we make a change to the internal level of the database the change effects all the stored records, similarly an invalid change in the extension of the database is not that fatal as a change in the intention of the database because a change in the extension of the database is not very hard to undo; incase of a mishap whereas a change of the same magnitude to the intention of the database might cause a large number of database errors (inconsistencies and data loss).

External View (Level, Schema or Model):

This level is explicitly an end user level and presents data as desired by the users of the database. As it is known that the database users are classified on two grounds

- Section of the organization
- Nature of Job of the users

The external level of the database caters to the needs of all the database users starting from a user who can view the data only which is of his concern up-to the users who can see all the data in the database and make all type of actions on that data.

External level of the database might contain a large number of user views, each user view providing the desired features and fulfilling requirements for the user or user

group for which it is intended. The restriction or liberty a user or user groups get in his rights is the external view of that user groups and is decided very carefully.

External views are also helpful when we want to display the data which is not place in the database or not stored at all. Example of the first case can be a customer Phone number stored in the database. But when contacting the person it might appear that the area code for that specific user is not stored in the database, in that case we can simply pick up the area or city id of the customer and find the area code for that city from the corresponding Area Codes table.

Another situation may arise when we want to get a student enrolled in an institution and want to make sure that the student qualifies for the minimum required age limit, we will look the database, for the students age but if we have stored only the date of birth of the student then the age of the student needs to be calculated at that very instance; this can be done very easily in the specific user view and age of the student can be calculated, even the user-view itself can tell use whether the student qualifies for the admission or not.

As the user view is the only entity or the interface through which a user will operate the database or use it so it must be designed in such a way that it is easy to use and easy to manage and self descriptive, also it is easy to navigate through. Also it should not allow the user to get or retrieve data which is not allowed to the user, so the user view should both be a facilitator and also a barrier for proper utilization of the database system.

As the system grows it is possible that a user view may change in structure, design and the access it provides to the users. SO External views are designed and create in way that they can be modified at a later stage without making any changes in the logical or internal views.

In the diagram below we can see two different users working as end users having their own external view; we can see that the same data record is displayed in two entirely different ways.

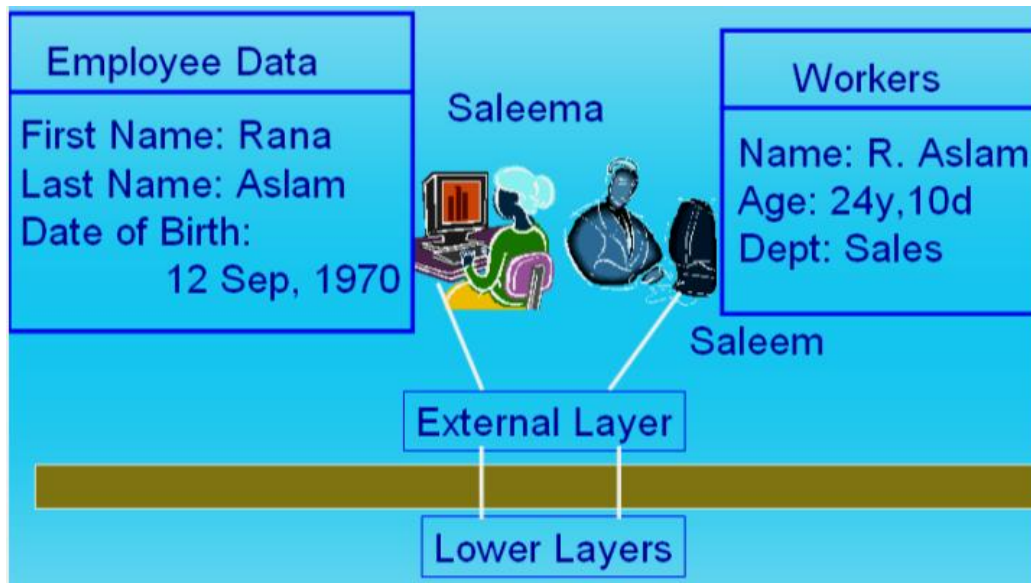


Fig. 2: Mapping between External layer and lower layers

Conceptual or Logical View:

This is the level of database architecture which contains the definition of all the data to be stored in the database and also contains rules and information about that structure and type of that data.

The conceptual view is the complete description of the data stored in the database. It stores the complete data of the organization that is why it is also known as the community view of the database. The conceptual view shows all the entities existing in the organization, attribute or characteristics associated with those entities and the relationships which exist among the entities of the organization.

We can take the example of the customers of a company. Now the conceptual schema will have all the details of the products of the company, retailing stores of the company, products present in the stock, products which are ready to be delivered, salespersons of the company, manager of the company and literally every other thing which is associated with the business of the company in any way.

Now after having all the information we know that the customers buy products from the outlets of the company, thus in such a case a specific customer has a relationship with that specific outlet of the company, or the customer may be represented as having association with the sales person which in-turn has association with the outlet., there may be a number of customers at a certain outlet and also to manage these salespersons there will be one or more managers. We can see from the above given

scenario that all the entities are logically related to each other in way or the other. The conceptual schema actually manages all such relationship and maps these relationships among the member entities. Conceptual schema along-with having all the information which is to be stored in the database stores the definition of the data to be stored. The definition may contain types of data, and constraints on data values etc. Conceptual schema is also responsible for holding the authorization and authentication information, means that only those people can make use of the database whom we have allowed to make these changes, so therefore it is the task of the DBMS to ensure be checking the conceptual schema that he is authorized to check the data or make any changes to the data.

Conceptual schema as it describes the intention of the database; it is not changed often, because to make a change to the conceptual schema of the database requires lots of consideration and may involve changes to the other views/levels of the database also.

As in the previous example we saw two database users accessing the database and we saw that both of them are having totally different user views. Here when we see in the logical view of the data we can see that the data stored in the database is stored only once and two users get different data from the same copy of data at the underlying concept



Fig. 3: External and conceptual layers

By summarizing it all we can say that the external view is the view of database system in which user get the data as they need and these database users need not to worry

about the underlying details of the data, all these users have to do is to provide correct requirement information to the DBA or the database designer whoever is designing the database for the system, so that the DBA or the database designer can create the database in such a way that they can fulfill the users requirements using the conceptual schema of the database.

Conceptual view/schema is that view of the database which holds all the information of the database system and provides basis for creating any type of the required user views and can accommodate any user fulfilling his/her requirements.

Exercise:

The data examples that you defined in the exercises of lecture 1, think of the different forms of data at the external and conceptual level. Also try to define mapping between them.

Thanks and good luck