

Data Abstraction

The one was entered

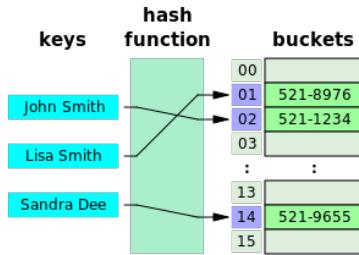


figure: Data structure

In the last presentation, we have seen about the three **tier architecture**

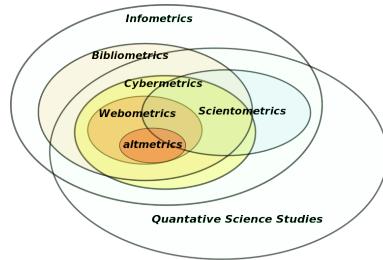


figure: Information science

In this presentation, we will focus on the view of **data**



figure: Mountain View, California

In the last presentation i told you. We have multiple views at the presentation level. Why do we need multiple views of **data**

title	release_year	length	replacement_cost
<code>dvdrental=# select title, release_year, length, replacement_cost from film dvdrental=# where length > 120 and replacement_cost > 29.50 dvdrental=# order by title desc;</code>			
West Lion	2006	159	29.99
Virgin Daisy	2006	179	29.99
Underworld: Stilts	2006	170	29.99
Truck Cider	2006	142	29.99
Song, Hedwig	2006	165	29.99
Slacker Liaisons	2006	179	29.99
Sassy Packer	2006	154	29.99
River of Time	2006	149	29.99
Right Cranes	2006	153	29.99
Quest Nussotini	2006	177	29.99
Poseidon Forever	2006	159	29.99
Eastwood's City	2006	140	29.99
Lawless Vision	2006	181	29.99
Jingle Sagebrush	2006	124	29.99
Jericho Mulan	2006	171	29.99
Jipperson	2006	155	29.99
Gilmores Boiled	2006	163	29.99
Floating Garden	2006	145	29.99
Fantasia Park	2006	131	29.99
Everyone's a Winner	2006	122	29.99
Everyone's Craft	2006	163	29.99
Dirty Aces	2006	147	29.99
Clyde Theory	2006	139	29.99
Clockwork Paradise	2006	143	29.99
Ballroom Rockingbird	2006	173	29.99
(25 rows)			

figure: Database

The answer for this will be revealed in this presentation before we step into **various views** of **data** we need to understand. What is data **abstraction**



figure: Data

I will explain **data abstraction** now we know basically, the primary goal of any **data base** is to store and retrieve the data conveniently and effectively and obviously. We need storage for doing this and this storage may involve simple or even complex **data structures**



figure: Moderat

We have already seen about this in the second presentation of this chapter

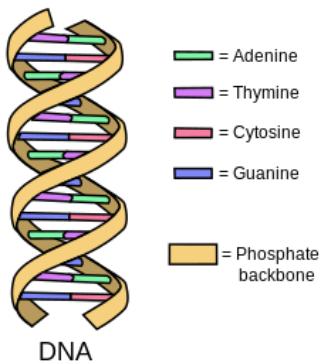


figure: Structure

If you want more clarity, i request you to watch my previous lecture titled bim's definition

And now we will focus on the **data**. Abstraction means hiding the complexity why we need to hide the complexity

In order to understand the **data abstraction** that is hiding the complexity. Let us take an analogy for better understanding. Let us assume. We have subscribed to a milk vendor and every day we are getting milk packets delivered to our home in time by the milk vend

Let us imagine, there is some delay in the delivery of milk packets for a particular day, and we are interested to know the reason for the delay. In the delivery of the milk packet. Let us say we are enquiring the milk delivery agent regarding the delay

The reply obviously from the milk delivery agent will be the reason for the delay. And if he gives the reason that his alam was not working, and that is why he was not able to wake up on time that would be sufficient for us instead. If the milk delivery agent replies as that he has gone to the bed last night at this time, and he has woken up twice and slept again, alam, which he has said, has disappointed he still continued sleeping. It was six a m already for him. But still he was sleeping, it was six one a m and still he was sleeping. And

Who needs all these information?

We wanted only the reason for the delay. In other words, we want a short and sweet answer to be precise. We want abstract **data**, not the complete data.

Though we may have encountered complexities, but we want only **abstract information**

So we are focusing on hiding the complexity and providing only the essential details. That is what **data abstraction** is all about in order to do that we have several levels of abstraction. We know basically in a three **tier architecture**, we have three levels likewise here, we have three levels. The first level is the physical level, the second level is the logical level and the third level is the view level which is at the topmost level the user level.

In this presentation, we are going to see all the levels, one by one

So we assure that there are three levels of **data abstraction**

One is the physical level, the other one is the logical level and finally, we have the view level multiple views are there. Let us see all the levels one by one in detail

And the topic now is the **various view** of **data**. And we know basically there are three levels. The top level is the view level followed by the middle level, the logical level followed by the bottom. Most level the physical level

Let us start with the first eleven the physical level

What this physical level is all about this physical level is actually the lowest level of **abstraction**. Now we know this physical level is actually the **data base** level. The physical storage is involved. We know data basis are actually stored in the physical medium. It may be hardest or ass any physical medium. And that is why it is the lowest level of abstraction. And it deals with how the data are stored. Because the data are actually stored in this level. The database actually exist in this level. And that is why this level deals with how the data are stored. When we talk about how the data are stored, it deals with the **data structures**. And this storage may be involving **simple data structures** or even **complex data structures**. But please noted that these uses the **low level data structures**. In case, we are storing multimedia data in the data base. Obviously the **data structure** would be complex data structures

Any way we will talk about the **data** structures elaborately in the coming presentation. And we know this is actually the **data base** level. We know data stored in the data basis, and these **data bases** are actually stored in this tourage medium.

And that is why this level deals with storage

We are done with the bottom. Most level, the physical level. Let us now move on to the next level.

We know physical level deals with how the **data** are stored

And this logical level deals with what **data** are actually stored and what is the relationship among the data store so logical level described what data are actually to be stored in the **data base**. And what is the relationship among the data that are stored in the database? nobody is when we see relational database e r models we will understand about the relationship among the data so now just understand this level. The logical level deals with what data are actually stored the physical level deals with how the data are stored, logical level deals with what data are stored and the relationship among the data

And what about the **data** structures?

For example, if we are storing entire database this entire database may be involved in simple **data** structures or even **complex data structures**, but storing the data physically at this level at the physical level involves **more complexity**. But at the logical level, it may use **only simple data structures**. Just see at the logical level, we use only simple data structures, but at the physical level, we use complex data structures. How this simple data structures interact with complex data structures

That is what this point. There is physical **data** independence. What we mean by this and we know this logical level is at the application level, this at the application. Now, the application actually do not depend on the physical level, or the physical schema do not worry about the term schema. In the next lecture, i will explain about schema elaborately now just understand the structure of the data basis referred as schema. So when we have the existing structure at the physical level, and when we make some changes in the logical level, it does not mean that we need to rewrite the entire structure in the physical level. Simply application do not depend on the physical schema. So we need not rewrite the **physical structure** if there is a change in the logical structure because these logical structures are going to use only simple **data structures**. And there should exist physical data independence. So that if any changes are made in the logical level, it will not affect the physical level. And obviously who is going to write all the logic

The **data** base administrator. What to store how to store? everything is done by the data base administrator. Of course, he is involved in all the levels. But the actual logic is involved in the logical level.

We are done with the logical level. Let us now move on to the topmost level, the view level. And we know there are multiple views existing and we know this is the highest level of **abstraction**. It means it involves **more complexity** and we are going to hide most of the complexity at this level

Because we need not show the complexities to the user. Say, if user wants to transfer one from one account to another account, he should be given with the option how to do the transfer. At the back end, at the logical level or at the physical level. Multiple complexities are involved. We need not show that complexity to the user

And that is why this level deals with the users and the access to the users so granting privileges. Everything will be coming under this level. And we know, this is the view level where user is going to interact. So obviously the interaction with the system is done at this level. How the user is going to interact with the system through the interface. It may be a web interface or a mobile interface simply g the graphical user interface. And also this level is going to interact with the next level. That is the logical level where application programs are written and as mentioned multiple views exist in this level. And of course, we need to focus on the security of multiple views

How multiple views exist in this level. We know all **data** are stored in a central place. Let us take banking. Environment a normal user will be able to access his own data

A taylor may access some more **data** than the normal user, but he will be able to give assistance to the customers or users with the basic details. Think about the bank manager

He will be obviously having more privileges when compared to other users in its branch

And what about the regional manager he, or she will have more privileged when compared to the branch manager

It does not mean that **data** are stored separately for every user. All data are stored in the same **data base**. But there exist multiple views. I will give you one more. Example. Just assume you are going to a machine for drawing your money.

When you insert your atm car, only your details are fetched. This is view one when i go and insert my atm car. The database system will provide only my delays. This is another view.

See. This is how multiple views are existing. And also there is security that strongly enforce that i can access only my **data** where i am not given with other data, that security is also achieved at the view level. This is just an example. Anyway, when the course progresses, we will be able to understand even more things about multiple views and security guise i hope. In this presentation, we have seen **various views** of data in the next presentation. I will teach us what is schema and instance

I hope the session is informative and thank you for watching

A aaaaaaaaargh

Related Links

[https://en.wikipedia.org/wiki/Abstraction_\(computer_science\)](https://en.wikipedia.org/wiki/Abstraction_(computer_science))

<https://en.wikipedia.org/wiki/Abstraction>

https://en.wikipedia.org/wiki/Multitier_architecture

https://en.wikipedia.org/wiki/Semi-structured_data

https://en.wikipedia.org/wiki/Abstract_data_type

https://en.wikipedia.org/wiki/Data_structure

https://en.wikipedia.org/wiki/Data_structure

https://en.wikipedia.org/wiki/Data_structure

https://en.wikipedia.org/wiki/Information_science

https://en.wikipedia.org/wiki/Implicit_data_structure

https://en.wikipedia.org/wiki/Data_structure

https://en.wikipedia.org/wiki/Mountain_View,_California

<https://en.wikipedia.org/wiki/Database>

<https://en.wikipedia.org/wiki/Data>

https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos

<https://en.wikipedia.org/wiki/Database>

<https://en.wikipedia.org/wiki/Database>

<https://en.wikipedia.org/wiki/Moderat>

https://en.wikipedia.org/wiki/Data_independence

<https://en.wikipedia.org/wiki/Complexity>

<https://en.wikipedia.org/wiki/Structure>