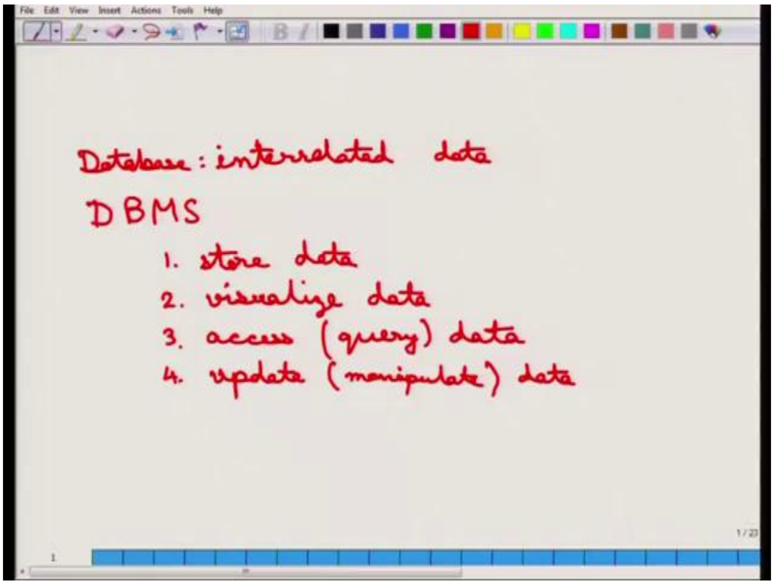
Fundamentals of Database Systems

Prof. Arnab Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 01
Introduction to Databases

Welcome all of you to this course on Fundamentals of Database Systems. This is mostly an Undergraduate course for Computer Science and Engineering as well as for Information Technology students. So, let us start off with what do we mean by a database. So, a database is essentially a collection of data.

(Refer Slide Time: 00:32)



Database : interrelated data

DBMS

1. store data
2. visualize data
3. access (query) data
4. update (manipulate) data

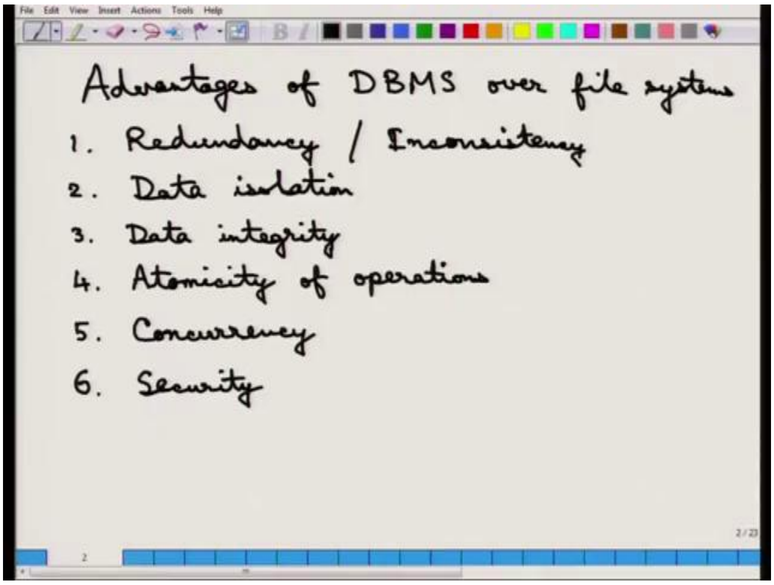
But, very importantly, it is not any data, it is a collection of inter related data. So, the data fields or the data points must have some connections with them. So, it is a set of inter related data. For example, I mean all have some idea of what databases are and where it can be used, for example, in an institute, the entire records about an institute constitutes one database.

So, it will have different entities such as faculty members, students, staff, etcetera as well as other kinds of things, such as courses, etcetera and everything constitutes one database. So, this is what a database and what is a database management system or a DBMS, it is a system that provides an environment to handle a database. So, it must be efficient and convenient to use and essentially it contains some programs and interface to store data.

So, the first point is for store data; that is number 1, number 2 is, it must be able to visualize data, then access data or access or what is also known as querying the data and the 4th one is update or manipulate the data. So, these are the four tasks of a database management system. So, this is over the database, the question comes, then is that the normal file system can also do all of these things, it can of course, stored data.

So, we store files etcetera, it can visualize data. So, well it is not really clear, what does visualization mean, we will see that later, but you can at least see, what the files and folders are etcetera. You can of course, access data, you can access a particular file within a particular folder etcetera and it let us you update the contents of the file, contents of the folder, etcetera. So, how is a database more important or how it is first of all different from a file, how is it more important than a file system. So, there is different advantage about the file system that let us go through.

(Refer Slide Time: 02:54)



Advantages of DBMS over file systems

1. Redundancy / Inconsistency
2. Data isolation
3. Data integrity
4. Atomicity of operations
5. Concurrency
6. Security

So, advantages of DBMS over file systems, so the first one is that, it reduces the data redundancy and inconsistency problems, so redundancy stroke inconsistency. So, let me go over each of these points a little bit. So, what does data redundancy mean? Essentially data redundancy means the following, is that there is a same piece of data, the same information that is stored in multiple places. And database, I mean a well designed database does not really allow that.

So, if there is an employee or if there is information about a student, the name of the student for example, the roll number etcetera is generally should be stored in only one place. This is the big difference about file systems, where most of us replicate files, copy files from here to there, etcetera, etcetera. What does that copying also does is that it runs into this problem of inconsistency.

So, what does inconsistency actually mean is that, you have two versions of a file and in one version, you have changed something and you have inadvertently forgot to make the same modification to the other file or there are two other two people making two different versions of the file etcetera. So, the information in these files two files may be inconsistent. So, one of them may not be correct, because the other one has been updated and that update has not yet come to this piece.

So, database by virtual storing each piece of information only once and in one place reduces this inconsistency. So, there is no redundancy and there is no inconsistency; that is the first point. The second point is, it is a very related point, it is called data isolation. So, here what happens is that, the data is isolated in the form that there is only one. So, the data is stored in an internal format and there is only one interface to access the data.

So, the essentially the data is stored in some kind of a binary format, which is not the headache of the final user. The user wants the piece of data and there is an interface, the database interface will let that user access the data in a particular format. So, the other important advantage of this is that, because the data is isolated and stored in a format that the database handles, the problem of formatting of data is not there.

And this, we often run into with file systems, for example you may be working with an excel file, which is a spreadsheet and once, you take it to for example, Linux system or Mac system that excel file may not open correctly. Because, it is not in that open document format and all those things and the programs to open them is does not behave consistently; that is the problem, but the database isolates that it.

The third important thing is called data integrity. So, what does data integrity mean essentially is that, there are many pieces of data, where there is schematics of data or there is a correctness condition. For example, is that the CPI or the Cumulative Point Average or your marks etcetera, must be the grade point average, whatever you call it, they are generally constant to be between 0 to 10 in our Indian systems at least.

But, suppose you are storing all of these information in a file or in an excel spreadsheet or whatever, now there is no check. So, if you for example, by chance entered somebody’s CPI as minus 1 or 12, there is no check in when you try to put in the data. However, a database will let you put in some integrity constants, so whenever you define the CPI field, you can say that, it is between 0 to 10 and so, any attempts to enter our data value, for example, minus 1 will result in a nil error and it will not let you enter.

The logic is in the database system itself not by the access. So, once you define the database of that CPI, you can say that, it is between 0 to 10 and that logic is within the database system. So, this is the 3rd point, the 4th point is called the atomicity of operations. So, this is probably one of the most famous uses of database and as an example, let us takes the bank transfer case.

So, certain amount of money, so let us say 100 rupees is transferred from person A’s account to person B account, the atomicity of operations. So, the atomicity of transactions defines that either the money is transferred completely or no money is transferred at all. So, what do we mean by that is that, it cannot happen that rupees 100 has been deducted from A’s account, but it has not been credited to B’s account or the other way around that rupees 100 has been credited to B’s account, but it has not been debited from A’s account.

And you can see, what problems will happen if this I mean inconsistent credits and debits can take place. So, a database system encodes that entire operation of debiting from A’s account and crediting to B’s account as one operation or one transaction and we will see more about transactions later in the course and one transaction and it mandates the atomicity of it. So, either this has happened completely.

So, either 100 has been debited and credited both or it has not been debited credited either. So, this is the other important part. Let us come to the 5th point, the 5th point is concurrency. So, very simply concurrency means multiple operations can take place together in the database; the database is I mean generally takes care of those things as long as there are no conflicts in that particular data item.

So, if A is transferring money to B and C is transferring money to D, they do not need to wait for each other and both the transactions can happen together. In file systems, it may not happen if all the transactions are opened in the same file. So, one file, the same file needs to be first edited for A to B, then saved and then C to D should happen, but it depends also on the file system that is being used.

Probably, the last important thing about databases or why it is useful over file system is that of security. So, security of the data is paramount in databases and the database or the data fields can be made secure in multiple ways. So, first of all for example, the employee database is secured in the sense that unauthorized users may not gain access to any of this. So, that is one part of the security, the other and probably the more important part of the security is that an employee.

So, there are different access control levels in the database for different fields and different roles. For example, an employee may be able to see her own record, but not records of others. It may happen that way, it may also happen that student may see a CPI of herself, but not of others, but a student may change her address home address, but not her CPI.

So, even if the record is about the student, so even within a record, there can be attribute level security attribute level access privileges, etcetera. And the security mandates that nobody, I mean, unless it is allowed by the database system and these roles have to be defined very carefully while defining the fields of the database will be allowed to change that. These are the six important points for the advantages of DBMS over the file systems.

Just to go over it, the first one is redundancy or inconsistency, then it is data isolation, then it is data integrity, the 4th one is atomicity of operations, 5th one is concurrency and the 6th one is security. So, well that ends the first part of the module, in the next module, we will cover, we will start on with the relational data models.