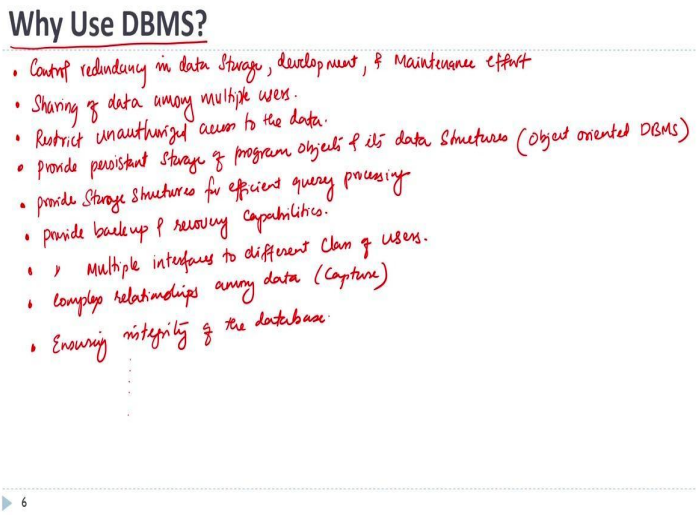
Computer Aided Decision Systems
Industrial Practices using Big Analytics
Professor Deepu Philip
Department of Industrial & Management Engineering
Professor Amandeep Singh
Imagineering Laboratory
Indian Institute of Technology, Kanpur
Lecture 10
Reasons behind using DBMS

Good evening, all. Welcome back again to the lecture on the database design part. We initially saw the Database Management System component of DSS. Now we are into the details of how to design and the need of a database and when to use a database, when not to use a database, etc. so that we can move into how to design a database.

(Refer Slide Time: 0:39)

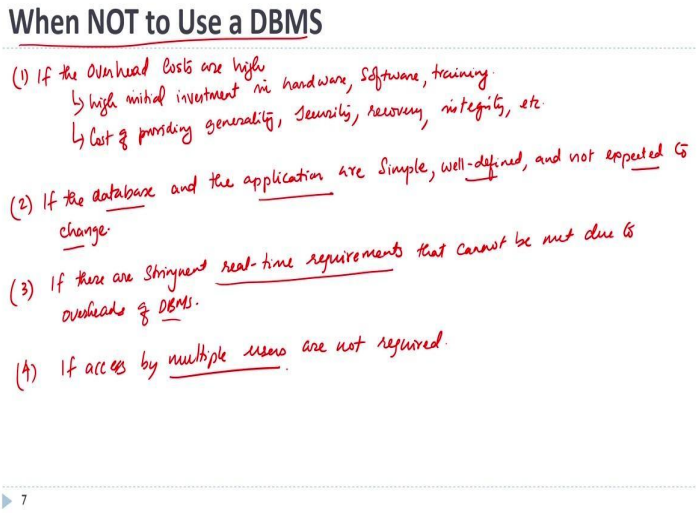


So, let us look into the slide today without any further delay. Why use a DBMS? When do you use a Database Management System again from a DSO standpoint? The first point is

- Control redundancy in data storage, development and maintenance efforts'. Avoid redundancy and duplication. So, you do not want to store it all over the place, you just store it at one particular location and then take care of it.
- Sharing of data among multiple users. You have many users and you want to share the data among them.
- Restrict unauthorized access to the data. So, you do not want anybody accessing the data without any authorization.

- Provide persistent storage of program objects and its data structures ‘Object-Oriented DBM. Persistent storage of program objects and its data structures.
- Provide storage structures for efficient query processing. When you want to retrieve the data or you want to query the data to extract, whatever is relevant to you, how can you do it quickly and efficiently?
- Provide backup and recovery capabilities. So, if you want the database or system crashes, can you recover it as part of it?
- Provide multiple interfaces to different classes of users.
- Complex relationship among data Capture.
- Ensuring integrity of the database. So, there are so many other uses, but from a DSO’s standpoint, these are the major reasons why we end up or why we like to use a database or a DBMS.

(Refer Slide Time: 4:42)



Now, the next question is, when not to use a DBMS? So, there are many again but let us take four major aspects.

1) If the overhead costs are high. So, what do you mean by high overhead cost?

- High initial investment in hardware, software, training. So you need to think if you are putting a very large investment in this, is it worth it?

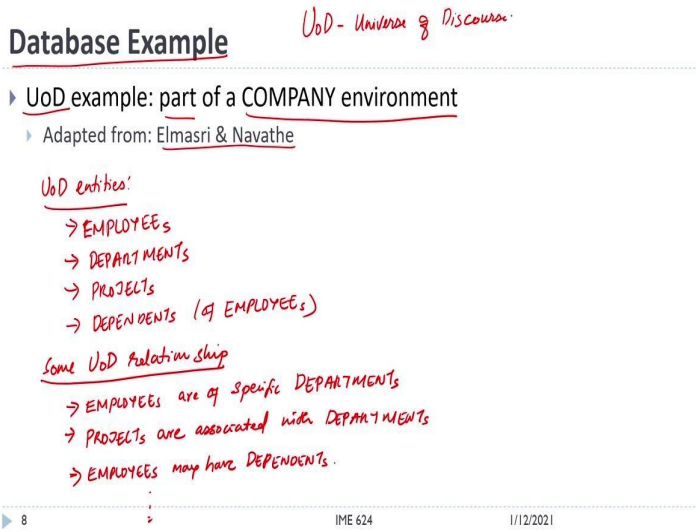
- Cost of providing generality, security, recovery, integrity, etc. These are nice things – generality, security, recovery, etc. are all nice, but what is the cost of doing it? That also you need to consider. So, if these overhead costs are extremely high compared to the benefits you are going to get out of it, then you may not want to use a DBMS.

2) If the database and the application are simple, well defined and not expected to change. So, if it is a very simple thing, the database and application is also very simple, it is very well defined, and you do not expect it to change frequently or do not even expect it to change, then maybe you do not need a database, you can just embed all the thing into the program or you can put a file and then access it with the program.

3) If there are stringent real-time requirements that cannot be met due to overheads of DBMS. So, DBMS because it is a program, it will take its own time to run and memory etc. And the real time requirements of the application are so important that you may not want to use the DBMS, like an autopilot in aircraft. That is a good example of this.

4) If access by multiple users is not required. It is only a single user, and there is no multiple user or anything of the sort, then it is probably better that you probably do not need a DBMS, you can just get away with a simple file.

(Refer Slide Time: 8:17)



So now let us talk about a database example. And we will be using this example. This is an example adapted from the textbook of “Elmasri and Navathe”, which is one of the mandatory textbooks in this class. And what we are going to do is remember ‘UOD’. We mentioned what UOD is. UOD stands for Universe of Discourse. The part of the universe that we are interested in and that is a part of a company environment is what we are interested in it.

So, for us, there are some UOD entities. The major UOD entities are:

- EMPLOYEES
- DEPARTMENTS
- PROJECTS
- DEPENDENTS (of EMPLOYEES).

So, these are some of the major entities. We will talk about what entities are and other things. Some UOD relationships: So, remember we talked about relational database, RDBMS right. So what are the major relationships?

- EMPLOYEES are of specific departments. This is one relation we want to capture.
- PROJECTS are associated with DEPARTMENTS
- EMPLOYEES may have DEPENDENTS.

So, in our case, we can talk about four entities - Employees, Departments, Projects and Dependents of employees (as four)… and then many relationships do exist between the entities. One way to think about these entities is something that you can physically identify. We will talk about what is an entity and those kinds of definitions in the coming class, but it is better to have an example in front of us and discuss or talk about their example. And we will also talk about an Entity Relationship Diagram in the coming class. So, this discussion will lead into Entity Relationship Diagram’ popularly known as ER diagram, which is a tool to do database design.

So, we will talk about how this database design and etc. happen, but keep this universe of discourse example in your mind because we will be referring to this example through the next few classes so that you can understand how we design the database, how do we identify entity, how do we identify relationships and translate that into a relational database design?,

that in which you will store the data which can be created by different processes and access it by different models of the decision support system through the help of user interface and then appropriate decisions can be made. So, this concludes the presentation. Thank you for your patient hearing. We will continue in the next class on the advanced topics of this. Thank you very much.