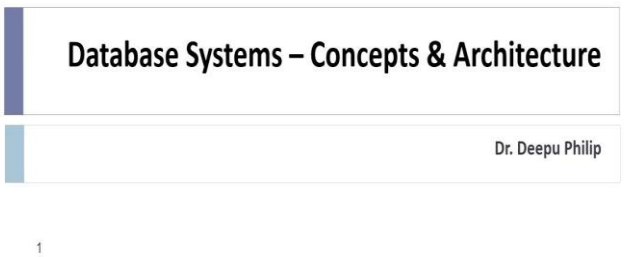
Computer Aided Decision Systems
Industrial Practices using Big Analytics
Professor Deepu Philip
Department of Industrial & Management Engineering
Professor Amandeep Singh
Imagineering Laboratory
Indian Institute of Technology, Kanpur
Lecture 11
Concepts & Architecture of Database Systems

Good afternoon, everyone. Welcome to the yet another lecture of Web Based Decision Support System for business decisions and practitioners. This is the continuation of the previous lecture and we were talking about details, we are delving into the details of Database Management System (DBMS), specifically, why the Database Management System is important to Decision Support System and what does it entail us to? We have already seen that part.

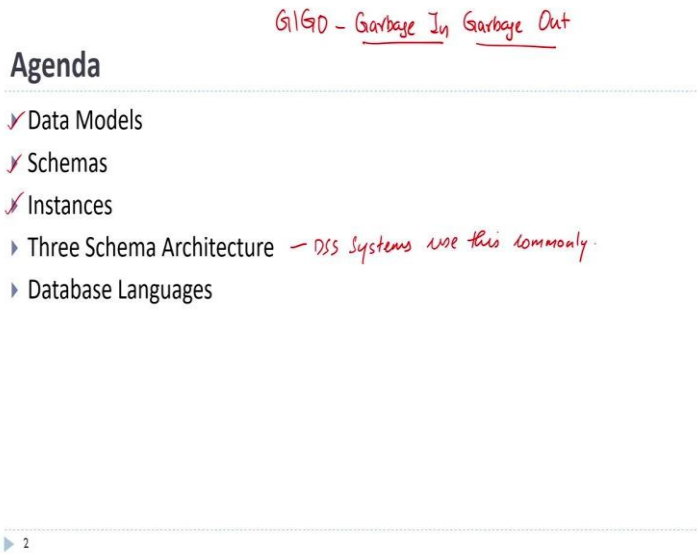
And we have also talked about some of the basic aspects of how the DBMS technology evolved, how the relational database systems, internet or HTML based database systems, object oriented database systems, all those things have been discussed in the previous class.

(Refer Slide Time: 1:07)



And today we are getting into what we call as the concepts and architecture, we are going to get into the nitty gritty details of how to get into the design of the database.

(Refer Slide Time: 1:14)

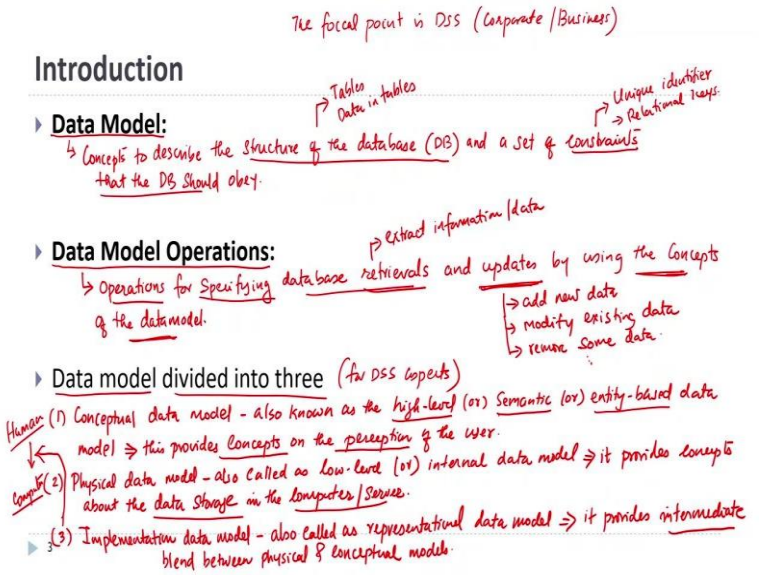


And this is important because if you do not have, so, the philosophy here we are going to do is Garbage In Garbage Out (GIGO). So, if you store garbage data, then you will get garbage results output. So, it is very critical that we store the correct data and today's presentation is all about discussion on what is:

i) Data Models
ii) Schemas
iii) Instances
iv) Three Schema Architecture

DSS systems use Three Schema Architecture commonly. So, that part and then also little bit a quick introduction of Database Languages and this will cover the background and allow us to go forward to the advanced topics and getting into ER Diagrams etc.

(Refer Slide Time: 2:13)



So, without further delay, let us see the some of the fundamentals. Remember, the focal point is DSS and that is corporate/business DSS. We are not talking about other DSS per se. So, the first one is we talk about something called as a 'Data Model'. A lot of people, with the DSS business you will say that, we are working on the Data Model. So what is a Data Model? So, this is (in simple terms).

- Concepts to describe the structure of the database (DB) and a set of constraints that the DB should obey. So that is what it calls the Data Model.

So, it describes the structure of the database, how the database will be like, what will be the major table? So, this will include the tables, data that goes into the table, etc. and the constraints like it includes what is the Unique identifier, Relational Keys etc. We will talk about what Relational Keys, etc. in the later thing. So, when you take the structure of the database and the set of constraints that the database should obey that completely both of them put together gives you the Data Model.

Then comes the second term, which is called as the Data Model Operations. So, what is that? So, this is the:

- Operations for specifying database retrieval and updates by using the concepts of the Data Model.

So, what we are saying here is operations or specific computer instructions that specifies how the database retrievals and updates, retrieval means 'extract information' or data from the Database. Updates include, add new data, modify existing data, remove some data, etc. So all

these things are aspects of what we call as updates, you can add new data, you can modify it, the database contains the latest information, but how do you do all these? How do you do the retrievals and updates? You have to use the concepts of the Data Model. So the Data Model concepts, whatever be the structure of the database and constraints, you use them and do not violate them to make sure that the retrievals and updates of the data happens.

So, such operations are called as Data Model Operations. And typically, this Data Model, what we talked about they are divided into three. There are many things but for the DSS aspects, we can take these three broad classifications. And the first one (the primary one is), what we call as the:

i) Conceptual Data Model- This is what primarily we will look into and this is also important to us and this is what the designer really focuses on.
   - ❖ It is also known as the High-level (or) some people also call it as the Semantic Model, where yet another name (or) Entity-based Data Model.

So Conceptual Data Model is sometimes people call it as a high-level model. It is in the designer level. It is mostly done by what human beings are, it is also called a Semantic Model or it can also be called as an Entity based Data Model. So, what does it do? What is the main purpose of it? This provides concepts on the perception of the user. Whoever be the user or person who creates the data, how do they perceive the data and their concepts on the data or how they perceive that data concept is what actually is captured in this Conceptual Data Model.

ii) Physical Data Model- This also called as Low-level model (or) Internal Data Model. Though people rarely called the Conceptual Data Model as External, but the Physical Data Model is also known as the Internal Data Model and what does it do? It provides concepts about the data storage. Where does this data storage happen? In the computer.

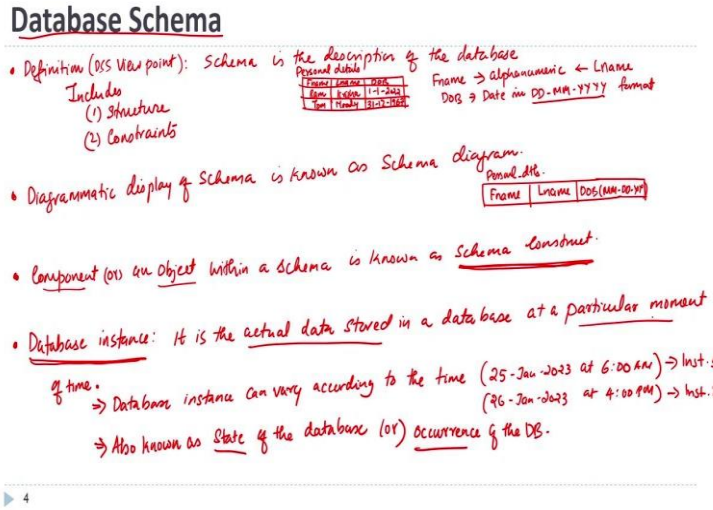
Or you can think about as or server or whatever you want to call it. Server is also a computer right? So the aim of the Physical Data Model is to give your ideas or concepts about how the data is being stored in a computer or a computer server. That is the second aspect of it.

iii) Implementation Data Model- Also known as the Representational Data Model. So, what does it does? It provides intermediate blend between Physical and Conceptual Models. So, what the idea here is that the Conceptual Model is in the mind of the human, the Physical Modal is in the computer. So, the Implementation Data Model is the one that

actually translates from human to computer translation, the Implementation Data Model is the intermediary in between it.

It provides you ways to translate, the intermediate translate or intermediate blender, what we want to call it between the Physical and Conceptual Model. Hopefully this is clear.

(Refer Slide Time: 11:37)



So, we move to the second concept that we call as ‘Database Schema’. So, there are so many definitions to this, but the simplest definition is, let us take it from a definition from a DSS standpoint or viewpoint. We can say that

- Schema is the description of the database.

So, when you describe the database, what does it includes? It includes two things:

i) Structure

ii) Constraints

So, these are the two points of Schema. So, this is the first aspect. Second aspect is:

- Diagrammatic display of Schema is known as Schema Diagram. So, one way to think about it is, Schema can give you the description of the data. So, sometimes let us say there is a table in the database, for example, there is a table like name or personal details. So, you have first name, last name, date of birth or something like this and you have Ram, Krishna, Tom, Moody, etcetera. And some date of birth like 1-1-2022, 31-12-1967, something like this. So, if you look into this, you can say some stuff like F name is alphanumeric, both F name and same is true with L name also (last name also). DOB is in DD/MM/YYYY format. So you can do this.

And this, some of them are the description, the structure, this is how it is and some of them are the constraints. You can also do the same thing in a fashion like this - F name, L name, DOB like MM/DD/YYYY, something like this. If you do this, and you call it as personal details, something like this, this diagrammatic representation of what you just talked about the structure and the constraints of the database is also known as the Schema Diagram. Then another aspect of it is:

- Component (or) an object within a Schema is known as Schema construct. So, if somebody says that I am working on a Schema construction, what they are doing is, they are either talking about one particular component or an object within the Schema - it can be a row, it can be column, they are talking about some specific aspect of the Schema. So, then you should remember this point also: the actual data that is stored in the database at a particular point of time, is the Database Instance.
- Database Instance- A lot of the time people confuse this with Schema construct. When somebody says Database Instance, it is the

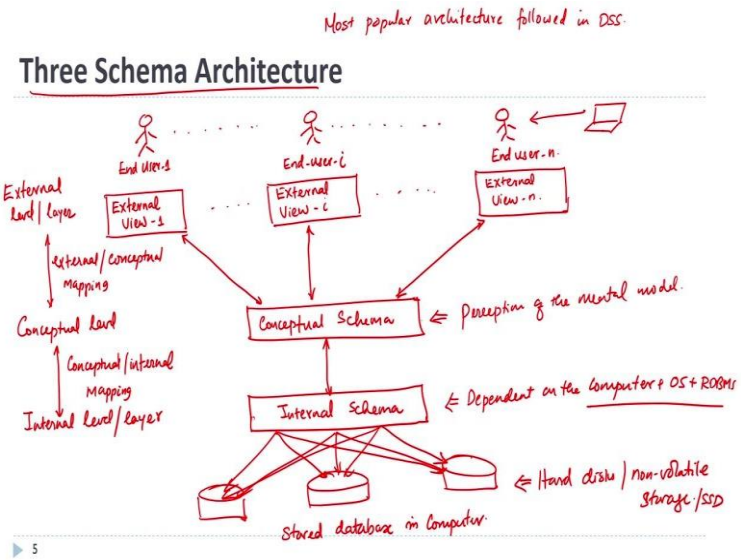
❖ Actual data stored in a database at a particular moment of time. Database Instance can vary according to the time.

So, if you say on (25th of Jan 2023 at 6am), what was the content of the database? And (26th January 2023 at 4pm), both these times, the database will have an instance 1 and this would be an instance 2, there will be some changes, some modification of the database. Database Instances, they are also known as State of the database (or) there is another name for it known

as the Occurrence of the database. So if somebody says State of the database or Occurrence of the database, they are all talking about a database instant.

So, remember Database Instance as we said earlier, the schema construct that means a specific component or something of the database or Schema, whereas the Database Instance is the whole collection, the actual data that is stored (the entire data that is stored) in the database at a given moment of time or a particular moment or instance of time (that is what it is).

(Refer Slide Time: 18:42)



So now without delay, let us go to the Three Scheme Architecture. This is the most popular architecture (the other architecture also) followed in DSS, corporate DSS, that are business DSS, the three Scheme architecture is actually followed. So, the main aspect of it is, let us we call it as, End user 1, End user i, End user n.

And each End user will have an ‘External View 1’ (for this particular End user). Here is another ‘External View I’, ‘External view n’. They can be the same, they can be different also, does not matter. So this is the External level or layer of the system. Then from there, we have known as the ‘Conceptual Schema’. So what happens is, each External View is created, the creation is facilitated using the Conceptual Schema. And, then from the Conceptual Schema you have something called as the Internal Schema. And, then under the Internal Schema, you have stored database in computer. So you have multiple databases that you are storing. It is also possible like this, as well. Now, this Conceptual Schema and Internal Schema, there is a mapping as

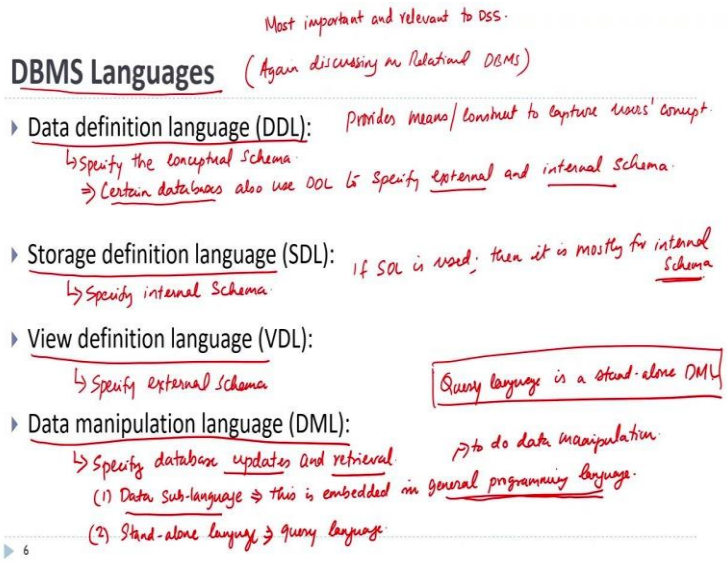
well. So, this Conceptual Schema is at the Conceptual level or layer and the Internal Schema is at the Internal level or layer.

Now, this transition, we call it as the 'External/Conceptual mapping'. And this is known as the 'Conceptual/Internal mapping'. So, the translation from the External layer, whatever the End user sees. So, the End user mostly will be, what we call as, using a computer. So, the End user will be using a computer or a display device to see an External View, how he or she perceives the data. That is translated to a Conceptual level where there is a Conceptual Schema.

So, that contains the Conceptual Schema and the Conceptual Schema is more like the High-level, Semantic entity based model. So, this is the perception of the designer. So, this is where the database Designer comes into picture, the perception of the mental model. It can be captured in the Relational Database Management System allow view mechanisms to capture them. And from there the Internal Schema, that means, it is dependent on the computer plus operating system plus Relational Database Management system (RDBMS), whatever be that particular choice, that you have, that translates this into and then this happens in the level of hard disks. Otherwise, non-volatile storage, it can be a Solid State Device (SSD) or whatever you want to call it, some way of permanent storage is, what we talk about at this point.

So, this is the typical Three Schema Architecture, that is followed in the Decision Support System.

(Refer Slide Time: 24:26)



So, now, let us talk about a little bit of the DBMS Languages, the major DBMS Languages, that we use in the system. And there are so many of them, but we are only talking about the

most important and relevant to DSS. So many but let us just define with these four. So the first thing that we are going to deal with this is:

- Data Definition Language (DDL):
- This specify the Conceptual Schema. So, this DDL allows or it provides (provides means constructs to capture users' concept). So, the DDL is the language that allows you means to kind of capture this. Also remember this:
- Certain databases also use DDL to specify External and Internal Schemas. So, remember this, the External and Internal Schemas are all DDL can, certain cases can also be used to (certain database systems can use) DDL to specify this. So, keep that in mind.
- Storage Definition Language (SDL):
- Specify Internal Schema. So, if SDL is used then it is mostly for Internal Schema or how the internal storage happens. So, in certain cases, this Internal Schema and External Schema, certain DBMS uses DDL. But if somebody uses SDL (the Storage Definition Language) works in this Internal Schema aspect. If you are using SDL, then mostly DDL is defined with the External or Conceptual aspect of it.

- View Definition Language (VDL): So, these are all languages that are mostly associated with what we call as the 'RDBMS'. This again discussing on Relational Database.

- Specifying External Schema. So, then what happens is, if you look into this diagram, the VDL is associated exactly with the External. So, if somebody is using VDL, NSDL then the DDL, remains in the Conceptual Schema format, but certain database systems, DDL can also be here. So, certain languages or Database Management Systems use DDL throughout, certain cases they have DDL for the Conceptual, VDL for External and Storage Definition Language for the Internal Schema.

- DML (Data Manipulation Language). So, this is a tricky one. So, we have already cleared it up. So, where is this DML? where is a space for DML? So DML is to, what it does is, it:
- Specify database updates and retrievals. So how do you update the data or how do you retrieve the stuff? It is basically driven by DML. So you have to think about two sub parts of it:

i) Data Sub Language- This is embedded in general programming language.

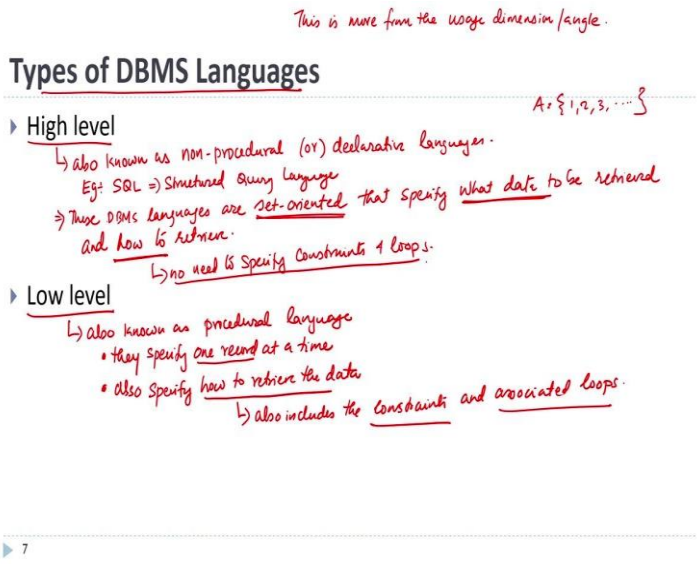
So, for each programming language (when we talk about a specific programming language), let us say, C ++, Fortran, Python, Perl, PHP, whatever the programming language, then the

Data Sub Language means whatever the specific sub language associated with that programming language to do Data Manipulation. And what is Data Manipulation? Data Manipulation is updates and retrieval of the data.

ii) Stand-alone Data Manipulation Language- This (most of the time) is also known as the Query Language. So when somebody says SQL. When somebody says you the word Query Language, that means it is a 'Stand-alone Data Manipulation Language'.

So, if you are using a construct of C++ or Python or something like that, that is okay, then that will come in the Data Sub Language, which is embedded and clearly associated with the programming language. So, each time you learn a programming language, you have to learn its associated Data Sub Language. Otherwise, if you learn SQL, which is Stand-Alone and independent of any programming language, then you learn SQL and that gives you ways and means to deal with updating and retrieving the information of data from the database. Hopefully this is clear to you. And, remember, this is the Three Schema Architecture, how the various database, DBMS languages find applications in that areas, what we discussed.

(Refer Slide Time: 32:07)



Then there is another type of DBMS language. We talked about the four types of DBMS language here, as the DDL, SDL, VDL, DML. So, there is another classification. This is more from the usage dimension or angle. Again related to the DSS. So, the first one is called:

- High level language (or) High level DBMS language-

- They are also known as ‘Non-procedural’ or ‘Declarative languages’. So, if somebody says ‘Non-procedural’ or ‘Declarative’ or ‘High level language’, we are talking about the same. The main example SQL, stands for Structured Query Language. So Structured Query Language, as I said earlier, is a Data Manipulation Language. It is a Stand-alone Data Manipulation Language. So what is the important thing?
- These DBMS languages are set-oriented. It is oriented towards a set. Set means, you remember what a set is. Set A is equal to {1, 2, 3,….} it is something like this is a numerical set and stuff like this, it is a set-oriented language that specifies that what data to be retrieved and how to retrieve? It specifies what is the data needs to be retrieved and how do you want the data to be retrieve, from the database. So all those aspects come in the part of the High level language.

- Low level language-

- These are also known as ‘Procedural Languages’. The other one is Non-Procedural Language, this is called as a Procedural Language. There are no other names for it. What do they do? There are certain aspects to them.
- They specify one record at a time. So, remember major difference is that other one is a set, this is ‘set-oriented’. So it retrieves the entire set of data, but here it receives only one record at a time.
- Also specify how to retrieve the data. But this is how further. It also includes the constraints and associated loops. So it tells you what are the constraints and associated loop (a while loop) or whatever it is, whereas the other one because you are retrieving a set, it tells what data is to be retrieved and how to retrieve the data, but in high- level languages there is no need to specify constraints and loops.

So this way, the High level languages are much more easy for the human being or the programmer, whereas Low level language you have to pretty much deal with everything. The reason why you go with a Low level language is, because it is quicker and faster compared to the High level language and the computing overheads are low. So that is one of the reasons why people prefer High level.

If anybody uses Low level language, that is to expedite or speed up the process. Ho, that pretty much finishes off our current lecture, the basics of the relation database, before we get into ‘how do we do the conceptual design of a database’, which is the next part of the course. And then from there, we will go all the way up to (ER Diagram) and how to create an ER Diagram.

And then after that, we will learn a little quickly how to translate ER diagram to be a Schema Diagram. And then we will look into Database Normalization quickly, not all aspects, relevant aspects of Normalization. And, then once it is done, we will learn a little bit of SQL quickly, how to Create a Table, Load the Table with the data, Manipulate the Table. And then I will give you a brief primer of what is Big Data.

And, then we will stop there for the other instructor to come forward and give you other information about it. So, once these fundamentals are clear to you, then the advanced aspects of managing the database is not a big deal. So thank you very much for your patient hearing. And we will continue on more topics about the database in the upcoming class. Thank you very much.