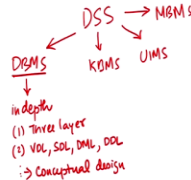


Computer Aided Decision Systems Industrial Practices using Big Analytics
Professor Deepu Philip, Professor Amandeep Singh
Department of Industrial & Management Engineering, Imagineering Laboratory
Indian Institute of Technology, Kanpur
Lecture 12
Entity Relationship

(Refer Slide Time: 0:23)



1

Good afternoon everyone. Welcome to the Web-Based Decision Support System course for decision makers and academicians. Today we will continue our discussions on the database component of the Decision Support System. We have seen 4 major aspects of the Decision Support System. So, if you think about it we were talking about DSS, which are 4 major components- DBMS (Database Management System), then KBMS (Knowledge Base Management System), then UIMS (User Interface Management System) and then what we did the last was MBMS (Model Based Management System) and we have seen all the 4 in brief and DBMS we will see in depth now.

We are looking into this component of the Decision Support System. So, in this aspect we have already seen what are the 3-layer architecture. We have seen the View Definition Language, Storage Definition Language, then the Data Manipulation Language, Data Definition Language etcetera. We have seen that aspect as well and we have also seen what is a Query Language etcetera, that we have already talked about and now we are in the process of the Conceptual Design, that is what we are trying to do today is how do we do the Conceptual Design of the database and

therefore that we need what we call as the Entity Relationship and from there we translate to what we call as an ER diagram as part of it.

(Refer Slide Time: 2:00)

with respect to DSS.

Main Terms

- ▶ Database application:
 - ↳ Specific database + application programs that implement the database queries and updates.
 - This course: MySQL → MariaDB (RDBMS)*
 - PHP → application programs*
- ▶ Entity Relationship (ER) model
 - ↳ popular high-level conceptual data model
 - *Extensively used by DSS designers.*
 - *Allows the design of complicated systems.*
 - *Easy to use (pen & paper)*
- ▶ ER Diagram
 - ↳ Pictorial representation of the ER model
- ▶ Data requirements
 - ↳ Concisely written set of users' requirements

Data requirements

↓

identify Entities

↓

identify Relations

↓

Create ER diagram.

▶ 2

So, without further delay let us look at the main terms that we need to understand today and to go from there. So, the first word that we hear about is:

➤ Database Application- again this is (with respect to DSS), the Database Application is the simplest term.

● It is a specific Database + Application programs that implement the database queries and updates. In this course, it is some specific database. It can be like an inventory database, it could be something and then an Application program that implements the queries and updates of the database.

For this course we are going to look into MySQL or what we better known as Maria DB is the RDBMS (Relational Database Management System) that will actually store the database or allow us to create and store the database and we will also look at what we call as PHP. It is a language that actually allows us to create dynamic hypertext pages (Dynamic HTML). So, this is the application program. So, the retrieval of the data, the updation, the representation of the results etcetera, all of this will be done with the help of PHP.

You can use other language like python and other kind of things but for the purpose of this course we will be dealing with PHP.

➤ Entity Relationship model (ER) model-

- The popular high level conceptual data model. We already seen what is a conceptual model and so high-level conceptual data model.

- We will use this because this extensively used by DSS designers (very popular) and allows the design of complicated systems

- Easy to use you can even do it with a pen and paper you can use software like UML and other kind of things but with just a pen and a paper also. You can use this ER model.

So, it is a high level conceptual data model. It allows us for the conceptual design but the more importantly lot of the DSS designers use this and it can help you to design complex systems or complicated systems and very easy because all you need to do is to have a pen and a paper. You can have suspect dedicated software like UML, Argo UML etcetera but then this is easy then comes what you call as the:

➤ ER diagram- So, your model is what we talk about then what is 'ER diagram' in a very simple way it is:

- The pictorial representation of the 'ER model'.

So, the 'Entity Relationship model' for us in this class we will be pretty much representing use instead of going through the extensive writing or other things. We will be using pen and paper to draw the 'ER diagram'. So, that will become the pictorial representation of the ER model.

➤ Data Requirements-

➤ Concisely written set of user's requirements. So, when the user requirements are given to you and lot of the time what happens, the rule is like this, the Data Requirements, identify Entities, then identify Relations, then create ER diagram. So, that is how we go. So, the ER diagram is the pictorial representation of the ER model. The Entities and Relations what we will see, what their definitions are and the Data Requirements are the concisely written statements of the user's requirements.

(Refer Slide Time: 7:47)

Entity

academic programs : B.Tech | Civil Engineering } Dual degree
 Program | Stream
 M.Tech | Computer Science }

- Basic Object of the ER model
 - ↳ usually it is a "thing" in the real world that has an independent existence.
 - ↳ recognizable object/concept/item, etc.
 - Eg: Student of IIT Kanpur (👤)
- Each entity has attributes
 - ↳ particular properties that describes the entity.
 - Eg: Student attributes → Roll no, first name, middle name, last name, date of birth, address, academic program, etc..
 - 1234 Ram Kumar Krishna 1-1-1985
- There are many types of attributes.
 - ↳ Simple attribute (atomic) vs. Composite attribute (sub-divisible)
 - ↳ Single valued (Single for a particular entity) vs. Multi-valued
 - ↳ Stored vs derived

Composite → Name: Ram Kumar Krishna
 Atomic
 Ram Kumar Krishna
 First Name Middle Name Last Name
 Date of Birth: 01-01-1980 ← Stored attribute.
 Age: (as of 01-01-2023) : 40 years ← derived attribute.

3

So, now let us talk about what we call as an 'Entity'. When we talk about the big term Entity, what does it actually mean? The simplest term or simplest way to think about Entity is:

- Basic object of the ER model.
- ❖ Usually it is "a thing" in the real world that has an independent existence. So, what we are saying is that it is a thing its usually something that you can recognize. So, this means Recognizable object/concept/item, etcetera. It does not have to be tree, it does not have to be a car or something, it can be also something like a project which is on a piece of paper and has money and finances and resources associated with it but there is you cannot really like you point a tree and say this is a mango tree that is where you cannot really point to and say that this is a project the project accessed on a piece of paper but you can recognize a particular project by project number of stuff like that. So, that is the Entity.

- Each Entity has attributes. What are attributes?
- ❖ They are Particular properties that describes the entity.

So, for an example, let us take an example as an entity. So, student of IIT Kanpur. So, that student can be a male/female. There is a physical you can point to a particular student. So, there is a Physical Accession. So, it is a thing in the real world. It is as an independent existence of it is own. So, what are some of the attributes of the student? So, let us take student attributes. Some of the

attributes will be the roll number of the student that will be the easiest thing for us to identify because the role number will be unique then there is the first name of the student. The student may or may not have a middle name. There is a last name, date of birth, address, academic program, etcetera. So, you can say that the roll number is let us say 11234 something. Like this first name is Ram then middle name is Kumar then last name is Krishna or something like that and the date of birth is 1.1.1985 and our address etcetera. So, these are all Attributes - the role number, the first name, last name, middle name. They all are Attributes of the particular student because it describes that student, whoever is called Ram Kumar Krishna, whatever, we just came up with the name. So, that is the attributes.

- There are many types of attributes. So, many of them are there and some of the important ones we are going to discuss now

- ❖ Simple attribute, it is also known as Atomic attribute, versus Composite attribute, this is also known as Sub-divisible. So, when you say the name: Ram Kumar Krishna, this is a Composite because you can subdivide this into Ram, Kumar and Krishna. So, this is your first name, middle name, last name (as shown in the slide). Now these 3 are Atomic attribute, because you cannot further subdivide it, similarly address is an example. So, if you say the address of the student is. So, all 3 IIT Kanpur. So, that can be subdivided into its component. So, Atomic is the simple; You cannot further divide it, subdivide it whereas composite. Attribute means you can further subdivide it. So, that is based on how granular the attribute is. So, some people call this as the granularity of the attribute as well, then there is something.

- ❖ Single Valued Attribute. Single for a particular entity versus Multi-valued. So, the example of the Single value versus Multi valued is the Academic Program, student is in B. Tech civil engineering. You can divide it into. This is sub divisible. It is into the program and this is the Stream or Department, whatever, you want to call it. So, you can subdivide this but a student can also have a minor in computer science. So, you can also have a M. Tec/Computer science. This becomes what you call as a Dual Degree student. So, now the Degree Academic Program can have multiple values depending upon the current status of the student. So, that is another example also. You can have something like address. You can have Multi-Valued Attributes of the residential address, permanent address etcetera. So, that is the other aspect.

❖ Stored versus Derived. So, the simple example of it is somebody's date of birth. So, the date of birth as 1.1.1980 is a Stored attribute. So, the age of the person as of 1.1.2023 that will be how much it will be 90,000. So, about 40 years, that will be the Derived Attribute. It is derived from the Stored attribute. So, that is why you rarely store the age of an individual in a database because age is dependent on the time. You actually store the date of birth and then the age is derived from the date of birth. It is a real world thing with an independent existence and each entity has an attribute. And attributes are particular characters or features or values that describe the entity and then the different type of attributes. We have already discussed the type of attributes at this point.

(Refer Slide Time: 17:43)

Null Values

First name	Middle name	Last name
DEEPU	Null	PHILIP

- Specific cases where a particular entity may not have an applicable value for an attribute
↳ not applicable
- Certain other cases where the value of a particular attribute might not be known or missing
↳ unknown.
- To handle both unknown & not-applicable cases ⇒ null value is used.

↳ Complex attributes : created by nesting a group of components.

Eg:-

Address	Room No 219, IME Building, Dept. of IME, IIT Kanpur, 208016, UP		
Room/Apt. No.:	_____	City _____	Country _____
Building:	_____	State _____	
Department:	_____	Pincode _____	
Landmarks:	_____		

Then we talk about what we call as Null Values and this is important because many times Null Values are very important in the database. For a simplest way of defining Null Values for us:

- Specific cases where a particular entity may not have an applicable value for an attribute. For example, if you have a database; you have first name, middle name, last name. These are the 3 aspects there. 3 things that you are storing and if my name has to be stored then I have a first name, I can write. I have a last name; I can write that as well. But I do not have a middle name. So, the middle name is not applicable to me. So, I would typically put a Null here, that means this middle name because I do not have a middle name. So, it is not applicable to me. So, such kind of

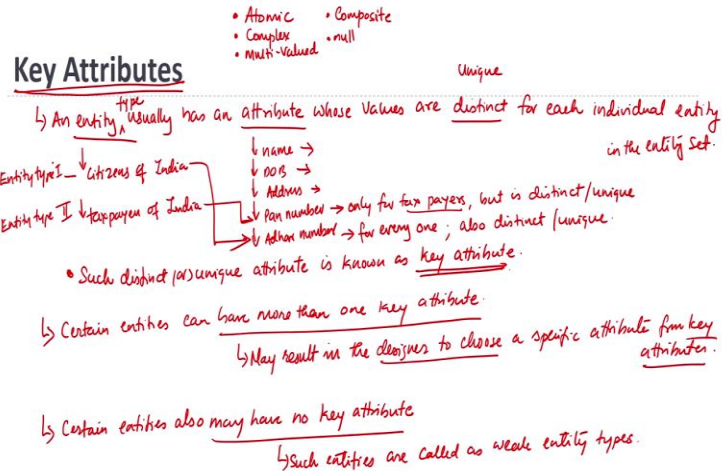
a case this is known as not applicable cases concept. So, that is one example of a Null Value when it is used.

- Certain other cases where the value of a particular attribute might not be known or missing. So, you may have a scenario, where you may not know or the value is actually missing. This scenario is known as Unknown cases. So, Unknown is different from not applicable because not applicable is their value, might be that you may not know. So, somebody asked me for my date of birth or something and if you do not know then that Null means there is a date of birth but you do not know about it or you are on you. So, that is unknown to you. So, Null can either describe not applicable or unknown

- To handle both unknown and not applicable cases- Null Value is used. So, whenever you see Null in a database it is either unknown or it is not applicable what it is you do not know. So, you have to figure that out. So, remember that there is another one that I want to add here which I forgot to mention in the previous case the single value stored another thing. So, there I want to add another stuff called as the Complex attribute. And Complex attribute is like these are; Created by nesting a group of components. Let us take an example; take my address. So, we will say room number 219, IME building, Department of IME, IIT Kanpur, 208016, UP, etcetera. It is a Complex Attribute because you can break this down into room/apartment number and then Building.

So, that is another one. Department can have another one. Landmark you can have another one. city, state, pin code and country, continent you can add whatever you want. So, all of these things put together, nesting them together, combining them together. You can create an address attribute one person, that type of an attribute is called as a Complex Attribute. I can now create a residential address also. And for address I have an official address and a residential address. Then that 2 addresses, two Complex Multi Valued Attributes, can be created out of it.

(Refer Slide Time: 24:03)



So, now let us talk about what we call as Attributes. We have been talking about major type of Attributes as part of this. So, now let us talk about what you call as the Key Attributes. We have discussed:

- Atomic attributes
- Complex attribute
- Multi-Valued attributes
- Composite
- Null

Now, let us talk about what is a Key Attributes? What is this concept called Key Attributes?

- An Entity Type usually has an attribute whose values are distinct for each individual entity. Let us call an Entity Type. We usually have an attribute whose values are distinct for each entity in the entity set. So, let us call the entity type as Citizens of India. So, if the entity type is Citizens of India then there are so many attributes and there are multiple attributes like they have name, they have date of birth, they have an address, there is a pan number and then there is a Adhar number etcetera.

So, let us assume that in this case the name could be same. Some people may have that may be multiple people with the name Ayush, Akash. You can find so many people with the same name.

Many of them will have the same date of birth also and some of them will have the same address as well, whereas, the pan number not everyone has it. So, let us say it is only for tax paying people but is distinct. We can use the word distinct or we can also call it as unique as well. Everybody may not have it. Let us assume that other number is for everyone. Everyone has it also distinct or unique.

So, if you are dealing with a scenario where you are only looking at the taxpayers then Adhar number is a good Key Attribute because it can uniquely identify every taxpayer. Two Adhar number will not be the same. Whereas if you are looking at the entire citizens also, one of the class we can think about is citizens of India for which Aadhar number could be the Key Attribute. Whereas, if we take another one as taxpayers of India, then pan card would be another attribute. So, this is Entity Type 1 and this is Entity Type 2, two different types of entity. Taxpayers of India is a smaller subset. They are also citizens of India. They will also have Aadhar card but they will definitely have a pan card. The citizens of India are a larger set. Not everybody is a taxpayer.

So, whoever is not a taxpayer they may not have a pan card. So, Aadhar card will become a much better Key Attribute.

- Such distinct or unique attribute is known as Key Attribute. So, when somebody says I am talking about a Key Attribute, then that is this we are talking about the unique. So, when in the database you will see here the term Primary key, Foreign key etcetera. We are talking about a distinct or a unique attribute of the entity that can be used to uniquely identify that. Other thing also you need to remember that:

- Certain entities can have more than one Key Attribute. So, if you add another attribute called the Ration card number or something, then that could be another or voter's ID card is another one. So, these are all different Key Attributes that means they are distinct for each individual entity in that set. So, when you take this, all the people of the citizens of India. There are so many people in that entity set. Citizens of India is an Entity type. So, individual citizen Ram, Sita, Gita they are all citizens and their Aadhar card or their voter's ID card, they are all Key Attributes because they are distinct or unique for each other all.

So, now the question is, if an entity can have more than one Key Attribute, what does this entails to?

May result in the designer to choose a specific attribute from Key Attributes. So, this may force you to say, I have so many Key Attributes. So, which one will I choose and you may have to end up choosing one and we need to know what that is?

- Certain entities also may have No Key Attribute. Such entities are called as Weak Entity types. So, like a person who we take the citizen of India and we define citizens of India or people of India or people who are living within the boundary of the country and we collect everybody's name and between we find one guy who has come from Pakistan to India without any Adhar card or anything of the sort, then that person would not have a Key Attribute, which is Adhar card, we have designed because that is the value for that. It is Null. There is no other cut for that individual.

So, if that happens then what that Entity, that particular individual will be the terrorists that crossed into India, will be a weak category type in that regard. So, I am just talking to you in real sense or real term so that you can understand or you can identify the in physically correlate- what is an Entity, what is an Attribute and what is a Key Attribute or when does an Attribute becomes a Key Attribute and when does an Entity becomes a Weak Entity, whether it depends on whether it has or does not have a Key Attribute.

Now, what we do is before we go forward, we will take a small break here and then we will talk about the Components of the ER diagram and so ER diagram is the Entity Relationship. So, Entity and its attributes how do you do, what are the conventions that you use to draw the diagram and etcetera. We will discuss in the coming lecture and we will solve take a Small Data Requirement statement from the customer and from there we will draw a ER diagram also and once that it is completed, we will go to the other topics of how to design and SQL aspects of it. Thank you very much.