IIT Madras

ONLINE DEGREE

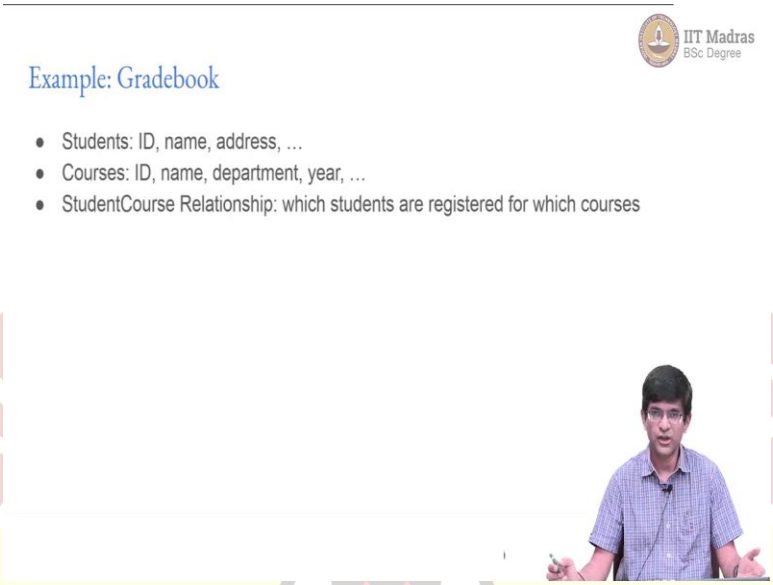
Modern Application Development - I
Professor Nitin Chandrachoodan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Persistent Storage

Hello everyone, and welcome to this course on modern application development. Today we are going to look at the topic of models. In other words, the M in the MVC paradigm.

(Refer Slide Time: 00:22)

To understand the requirement for a model or what we mean by a model, we first need to understand what we mean when we talk about persistent storage. As an example, we will take the Gradebook application. And as I walk you through, we will also see why it is that we need persistent storage. And as we move forward, we will look at ways by which this can be done.

(Refer Slide Time: 00:44)



So, as mentioned in the video on views, the running example that we will be using through the course is on implementing a student Gradebook. And the kind of data that we have in this Gradebook is a list of students for whom we will have information such as an unique ID number. In other words, each student has a unique ID,

We will have a name for each student. And this is actually where the ID becomes important because it is entirely possible that two students could have the same name. But on the other hand, it is guaranteed that two students cannot have the same ID. Students could also have additional information, maybe their address or if they are staying in a hostel, then what is the hostel, what is the room number.

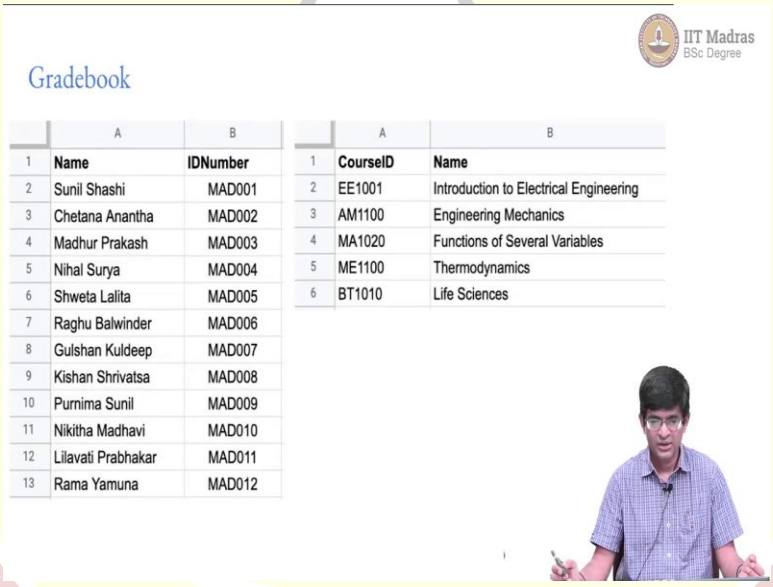
In addition to this, another sort of object or a piece of data that we have in regard to the Gradebook is a set of courses, a list of courses. Courses once again have an ID associated with them. And they have a name, for example, a course name, course number, or ID could be something like E1001. And the corresponding name might be introduction to electrical engineering. So, once again, you can see that the expectation is that the ID is unique.

In the case of courses, we generally expect the name also to be unique, although we might have situations where more than one course is called calculus. And then you have to sort of go by the course ID number in order to ensure that you have got right course. Courses could also have

additional information associated with them, they might be associated with certain departments, they might be offered in a particular year, and so on.

Now, as far as we are concerned, the main thing that we are trying to model over here is, there are also relationship between students and courses. In other words, I can talk about a student taking a course or I can ask for the list of, list of students who are registered for a given course. So, there is always one student and one course can be associated with each other, meaning that that student is registered for that course. So that is the sort of simple application that we are considering over here. And we will be using that in order to motivate our examples using the models.

(Refer Slide Time: 03:02)



	A	B
1	Name	IDNumber
2	Sunil Shashi	MAD001
3	Chetana Anantha	MAD002
4	Madhur Prakash	MAD003
5	Nihal Surya	MAD004
6	Shweta Lalita	MAD005
7	Raghu Balwinder	MAD006
8	Gulshan Kuldeep	MAD007
9	Kishan Shrivatsa	MAD008
10	Purnima Sunil	MAD009
11	Nikitha Madhavi	MAD010
12	Lilavati Prabhakar	MAD011
13	Rama Yamuna	MAD012

	A	B
1	CourseID	Name
2	EE1001	Introduction to Electrical Engineering
3	AM1100	Engineering Mechanics
4	MA1020	Functions of Several Variables
5	ME1100	Thermodynamics
6	BT1010	Life Sciences

So, this is one way by which we could represent the data that I was talking about. So, on the left-hand side, you have, you can see that there is a list of names, these are just made-up names, it turns out there is actually a random name generator somewhere on the internet. And the ID numbers will MAD Modern Application Development, I just gave it the name, 001, 2, 3, and so on. Typically, this would be some kind of a roll number associated with your college or institution or in the case of the online degree, the IIT Madras, which is actually granting the, which is giving you your roll numbers.

So, the ID number or roll number in the case of students has to be unique. On the other hand, names need not be unique, it is entirely possible that more than one person has the same name.

But then of course, we expect that they have different ID numbers. And there are some other ways of identifying who is who. Similarly, on the right-hand side, you can see that there is a list of courses. And the course ID is EE1001, AM1100, you know, just a few random names and ID.

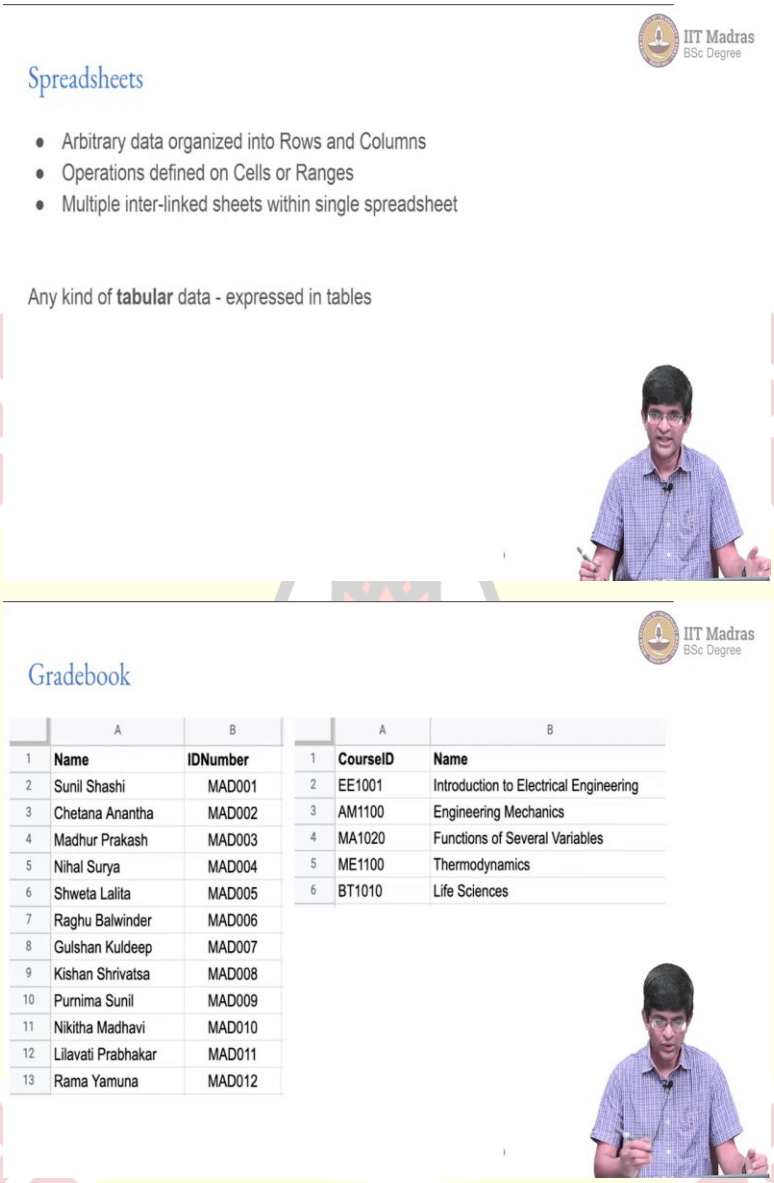
And the corresponding names, Introduction to Electrical Engineering, Engineering Mechanics, and so on. So, as you can see, one of the things that you can sort of notice as a pattern is, the course ID and similarly the ID number for the students are sort of clearly defined patterns, for the students it has like three letters MAD, followed by three digits. And this is very often done even in roll numbers in any college, you would have some kind of a pattern of this sort, simply because it makes it easy to validate whether the number that you have entered or the ID you have entered is a valid roll number.

Similarly, courses also tend to have systematic identification patterns. On the other hand, the names could be longer, they could have spaces in them, they could be in fact, even in a non-so-called Latin alphabet. So, that is where all the Unicode and the UTF 8 and so on things that we discussed earlier, could potentially come into play.

In our case, at least, we will be just restricting ourselves to the English alphabet, the 26 letters and maybe uppercase and so on. So, we do not really typically consider other alphabets, but there is nothing in principle preventing us from using those. So, with all of this in place, what we can see is that we have information which is represented in looking like it is in tabular form.

And you might already have recognized from the pictures that I have put over here, that these are essentially snippets from a spreadsheet. And what is this relationship with spreadsheets, why is it useful to use them to model data is also something that we will be looking at as we move forward.

(Refer Slide Time: 05:46)



So, spreadsheets in general, consist of arbitrary data, a spreadsheet is just a set of rows and columns. The intersection of a row and column is called a cell. Groups of cells are called ranges. All of this is terminology that we will not be getting into the purpose over here is not to really explain what spreadsheets are or how they work. But the important point is that some familiarity with how to use spreadsheets will definitely help you to sort of get a better picture of what is happening in general with structured data.

So, what do we mean by structured data, as you can see from the previous picture, the reason why I am calling the student information, for example, structured data is I know that the record

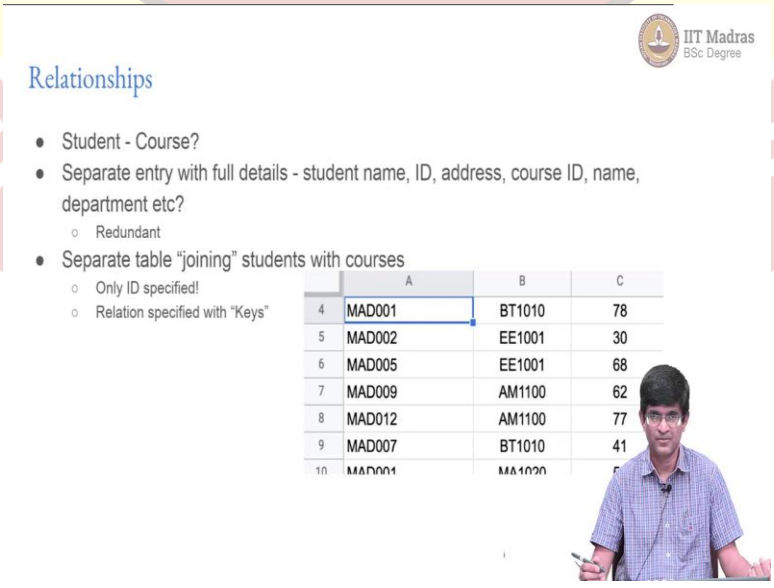
for every student is one row in the spreadsheet. And the row, the first cell in that row will be the name of the student, the second cell in that row will be the ID number. Similarly, for the courses, the first cell in the row for the courses, is going to be the course ID. And the second cell is going to be the name of the course.

So, any new course that I want to add has to follow the same structure, it has to be placed in the same sheet as the other courses. And it has to follow the same pattern. If I suddenly decided to switch things around and put the course ID in the second column and the name in the first column, it will mess things up. So, this is what we mean by structured data, or tabular data. Why tabular data, because it is expressed in the form of tables.

Now, a spreadsheet or a system of spreadsheets in general could consist of multiple linked sheets. So, today, you know everyone pretty much has access to Google Sheets, or you might have used Microsoft Excel, or LibreOffice Calc, all of them are examples of the same basic idea of spreadsheets, going all the way up to VisiCalc and Lotus in the 1980s.

And the fundamental idea over there is simply this, you have rows, columns, and cells. Cells contain data, that could be any kind of data. And the important point is, how do you sort of interact between the different cells and how do you extract data by combining cells in different ways, that is where spreadsheets really have their power.

(Refer Slide Time: 08:09)



So now, in this case, what I showed you was two tables, one corresponding to students and one corresponding to courses. And the question that we can then ask is, how do I indicate a student course relationship? Because after all, I have been saying that a student can take multiple courses, one course may have multiple students registered for it, how do I indicate all of this in some kind of a simple pattern?

One possibility is that I actually have one sheet or you know, some kind of a row or just a piece of text that basically gives the full details, it says, this is the student name, this is their address, this is the course ID for which they are registered, the name of the course which department it is, and it has all of this information listed out for every student and course combination.

Now, even the way I am saying it sounds unrealistic and unnecessary, because clearly, I do not want to be repeating this information, student names do not need to appear multiple times, course names do not need to appear several different places. After all, there is only one unique identifier for a course, one unique identifier for a student. Once I have those two pieces of information, I am actually done.

So, this in other words is redundant. And what we could do instead is just create another table, where we join these two, the students and courses. And in this new table, you can notice for example, that column A has a list of student IDs. And you would notice that it is neither in sorted order, nor is it unique, actually entry number 10 over here or row number 10 over here actually is again, MAD001, even though it is not fully visible.

Now, what does that indicate, it basically is saying that MAD001 and column B containing BT1010 is an indication that the student with ID number MAD001 has registered for the course with ID number BT1010, Life Sciences course.

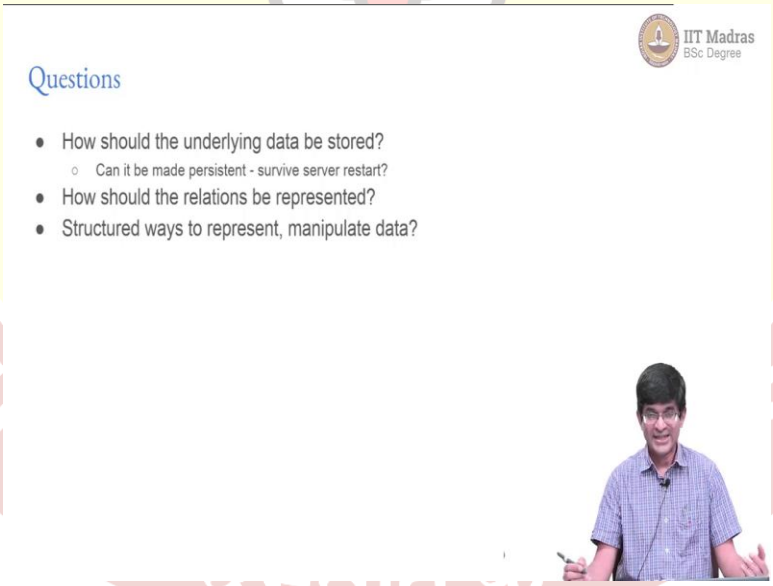
And in this case, column C also indicates the marks that they got in the course. So, it assumes that they have completed the course and they have got 78 marks out of whatever, it is just a number. Now the interesting thing you will notice is that only the IDs of the students have been specified, which is fine, because ultimately, we have guaranteed by the definition of the student table, that every student has a unique ID. If I had put names over here, I might run into trouble if two students have the same name.

But by putting the ID I guarantee that that problem is not possible. Similarly, I can also put course IDs in column B, and guarantee that they will never be that two course IDs will never be the same. And the interesting thing is because the student IDs and the course IDs are sort of in my control, I might even choose to just make them numbers. After all, why talk about MAD001, and so on, I could just choose to give it an integer value 1, 2, 3, 4.

And that 1, 2, 3, 4 is sufficient in order to indicate uniquely, which course or which student I am talking about, so a 1 in the student table would indicate whatever is in row number 1, which might corresponding to the student ID MAD001. Or it might correspond to something else, but point is it is unique.

So, this relationship between students and courses, in other words, has been specified by combining the keys from two different columns. And as you can see over here, by doing this, we have reasonably compactly managed to capture the entire joint information, which student, which course, how many marks.

(Refer Slide Time: 11:45)



So, all of this was just an example to show the kind of data that we might be interested in storing, it raises a few important questions that need to be answered. The first that we can think of is how should the underlying data out here be stored or saved? After I showed you some kind of spreadsheets, so how exactly do you save a spreadsheet, is there some standard data format for a

spreadsheet, is there some way that we can think about it, are there simple ways, complicated ways, what kind of information would we need to store?

And the reason why we want to save it, or store it somewhere, the primary reason is that there is a possibility that the server on which we have been running all our demonstrations, or our examples could crash, or restart. Now, what happens in such a situation, I do not want to lose all the data, I would like to see if I can retrieve the data and once again, make use of it? Or rather loaded up again. And once again, present it to the user and say, you know, continue where you left off.

The next question is, how should the relations be represented? I had given an example where you could use the students and courses and have a separate sheet out there in order to indicate the combination of relations between them, is that the best way, are there different ways by which this can be stored, how do you sort of efficiently put it so that it can be retrieved? All of those are interesting questions. And finally, it sort of raises the question of, you know, what kind of structured ways can we think of to represent or manipulate such data? So, as we move forward, we are going to look at some of these questions and try and answer it.