INDIAN INSTITUTE OF TECHNOLOGY MADRAS

सिद्धिर्भवति कर्मजा

IIT Madras

ONLINE DEGREE

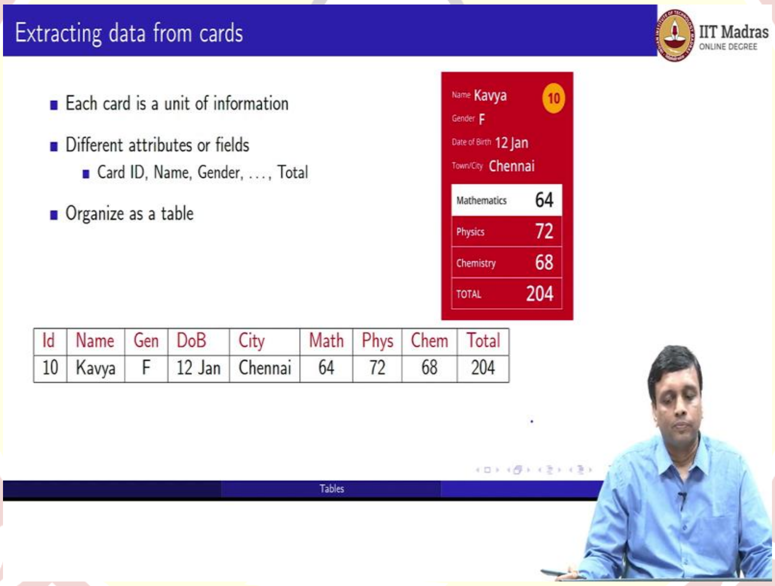
Computational Thinking

Prof. Madhavan Mukund
Prof. G. Venkatesh
Department of Computer Science
Chennai Mathematical Institute
Indian Institute of Technology, Madras

Lecture – 3.1
Presentation of datasets in the form of a table

So, we have had in our all our lectures now we have had the data on cards which is not very convenient for a program to read. So, let us see how we can present the cards in a different way using tables.

(Refer Slide Time: 00:28)



Extracting data from cards

- Each card is a unit of information
- Different attributes or fields
  - Card ID, Name, Gender, . . . , Total
- Organize as a table

Id	Name	Gen	DoB	City	Math	Phys	Chem	Total
10	Kavya	F	12 Jan	Chennai	64	72	68	204

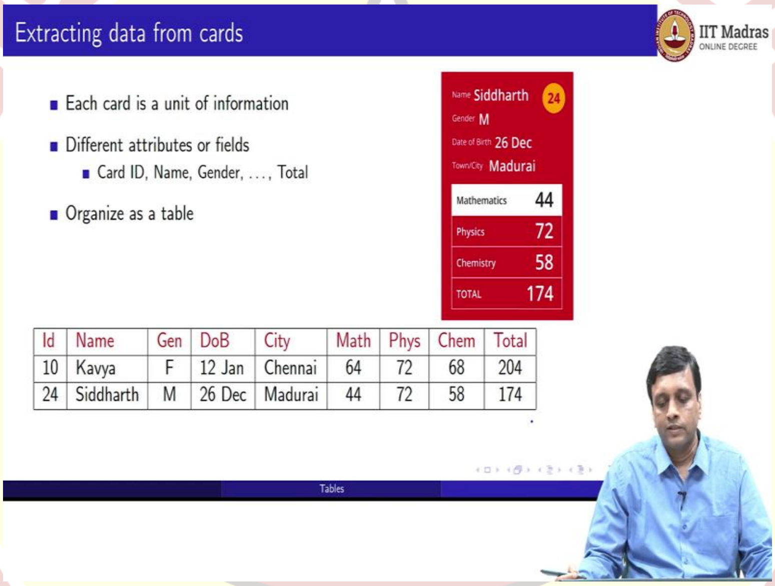
So, for us, each card has been a unit of information. For instance, when we had grade cards, each card was a unit of information about one student, and it had various attributes.

For instance, there is the Card ID which identifies uniquely which grade card it is, and then we have these different fields like name, gender, date of birth with the different subjects and the total. So, what we need to do, when we look at such a card as a human being is to identify all these important attributes and the values of them.

So, for example, we look at it and we can see that the physics mark is 72. Now, this is not a very convenient format in which to present the information to an algorithm or a procedure. So, better ways to organize it as a table and in the table, what we will do is, we will convert each of these attributes into a column. So, each card now becomes a row in this table.

For instance, this particular card and every card in the grade card set has precisely 9 attributes. It has a Card ID, the name of the student, gender, the date of birth, the city, the maths marks, physics marks, chemistry marks, and the total marks. So, if we now make a table with 9 columns then we can write out these values for each card as a rows for instance, Kavya's card has ID 10, Name Kavya and so on.

(Refer Slide Time: 01:50)



Extracting data from cards

IIT Madras
ONLINE DEGREE

- Each card is a unit of information
- Different attributes or fields
  - Card ID, Name, Gender, . . . , Total
- Organize as a table

Name Siddharth 24
Gender M
Date of Birth 26 Dec
Town/City Madurai

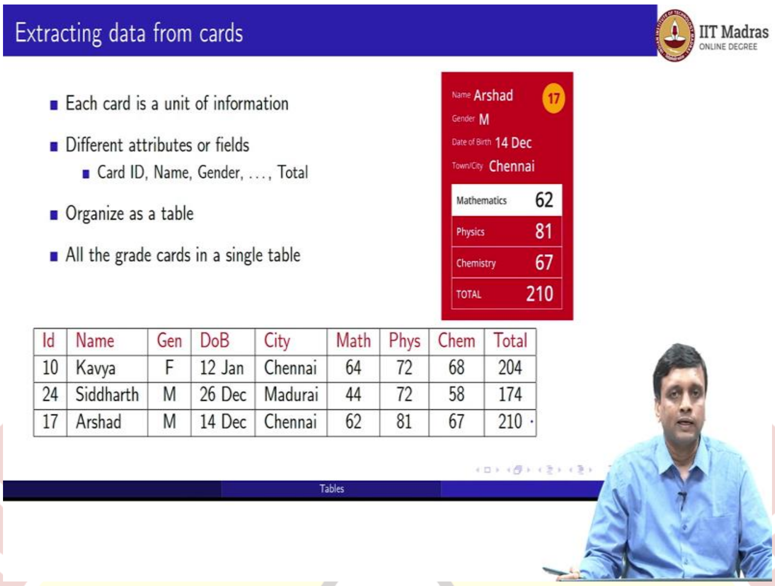
Mathematics	44
Physics	72
Chemistry	58
TOTAL	174

Id	Name	Gen	DoB	City	Math	Phys	Chem	Total
10	Kavya	F	12 Jan	Chennai	64	72	68	204
24	Siddharth	M	26 Dec	Madurai	44	72	58	174

Tables

So, if we pick up a different card, for instance, the card for Siddharth, then this becomes a new row in this table. So, in this way, row by row, we can enter the details for each card in this table. And finally, we have a single table which represents the entire deck of cards.

(Refer Slide Time: 02:05)



Extracting data from cards

- Each card is a unit of information
- Different attributes or fields
  - Card ID, Name, Gender, . . . , Total
- Organize as a table
- All the grade cards in a single table

Name Arshad 17
Gender M
Date of Birth 14 Dec
Town/City Chennai

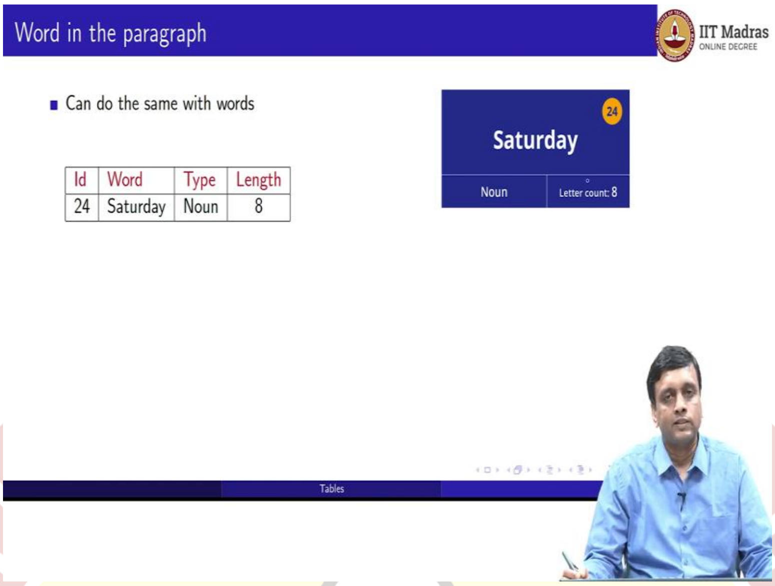
Mathematics	62
Physics	81
Chemistry	67
TOTAL	210

Id	Name	Gen	DoB	City	Math	Phys	Chem	Total
10	Kavya	F	12 Jan	Chennai	64	72	68	204
24	Siddharth	M	26 Dec	Madurai	44	72	58	174
17	Arshad	M	14 Dec	Chennai	62	81	67	210

So, this way we have a single compact data structure which allows us to look at all the information across all the cards. And at the same time by looking at the row and the column we can precisely get for instance in Card 17, we can look at the city and say that it is Chennai.

So, it is very easy to extract information from this table by looking at the row Id and the column Id. And most importantly we now have a single structure in which all the data is. So, we can pass it around just like we have been passing around other things to our procedures, we can pass this around from one procedure to another.

(Refer Slide Time: 02:45)



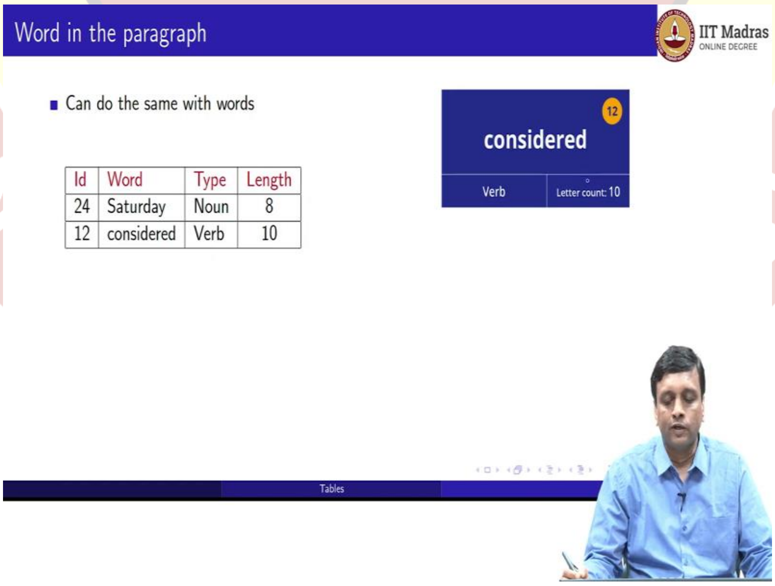
Word in the paragraph

- Can do the same with words

Id	Word	Type	Length
24	Saturday	Noun	8

So, the other kind of data set that we had were words in a paragraph. So, once again in this we have some attributes, we have the card ID, we have the word itself, then we have the part of speech and then we have the letter count. So, we can now represent this as a 4 column table which has these 4 values.

(Refer Slide Time: 03:08)



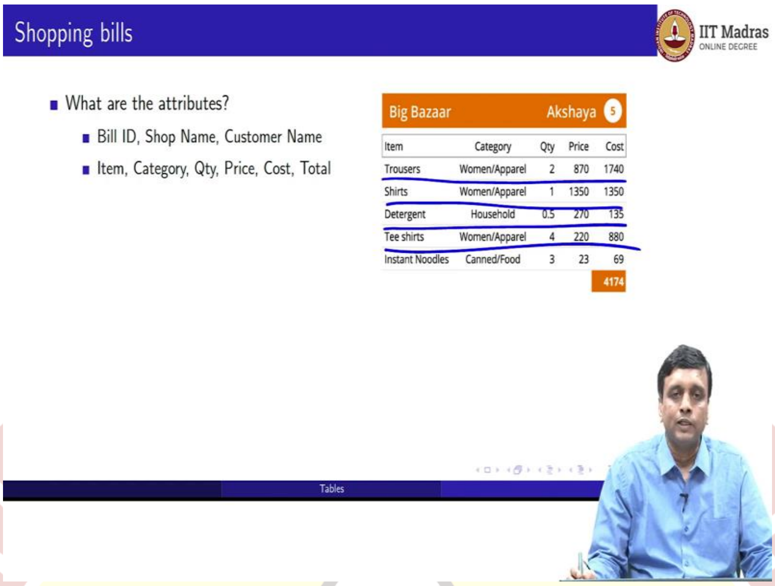
Word in the paragraph

- Can do the same with words

Id	Word	Type	Length
24	Saturday	Noun	8
12	considered	Verb	10

And if we take a different word for instance considered, then the Id is different, word is different. Now, the type is a verb, and its length is 10. So, so far so good.

(Refer Slide Time: 03:23)



Shopping bills

- What are the attributes?
  - Bill ID, Shop Name, Customer Name
  - Item, Category, Qty, Price, Cost, Total

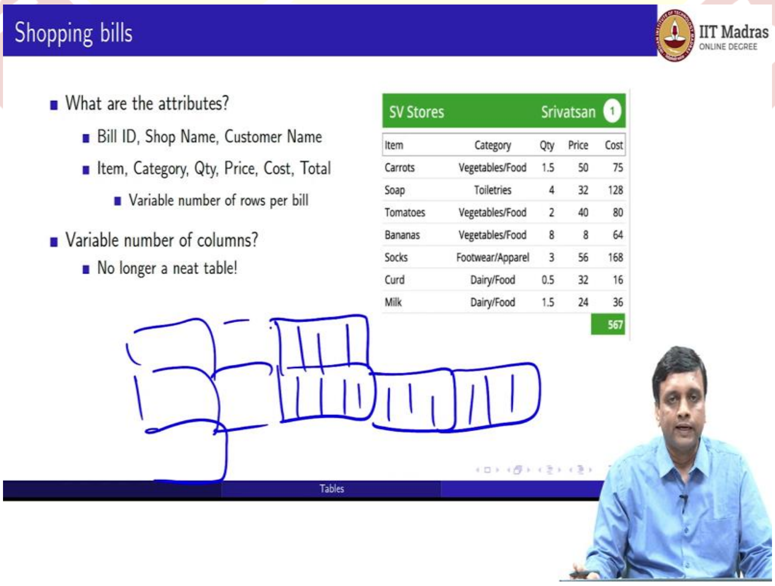
Big Bazaar Akshaya 5

Item	Category	Qty	Price	Cost
Trousers	Women/Apparel	2	870	1740
Shirts	Women/Apparel	1	1350	1350
Detergent	Household	0.5	270	135
Tee shirts	Women/Apparel	4	220	880
Instant Noodles	Canned/Food	3	23	69
				4174

Now, let us look at the third artifact that we had which was the shopping bill data. So, what are the attributes in the shopping bill? So, we have the Shop Name, we have the Bill ID, and we have the Customer Name.

So, these are single values which occur once on the card. We also have a total which occurs once on the card. But what we have on the other hand is these items which correspond to the individual bill items which are bought by the customer, and there are multiple rows of this in each card.

(Refer Slide Time: 03:55)



Shopping bills

- What are the attributes?
  - Bill ID, Shop Name, Customer Name
  - Item, Category, Qty, Price, Cost, Total
    - Variable number of rows per bill
- Variable number of columns?
  - No longer a neat table!

SV Stores Srivatsan 1

Item	Category	Qty	Price	Cost
Carrots	Vegetables/Food	1.5	50	75
Soap	Toiletries	4	32	128
Tomatoes	Vegetables/Food	2	40	80
Bananas	Vegetables/Food	8	8	64
Socks	Footwear/Apparel	3	56	168
Curd	Dairy/Food	0.5	32	16
Milk	Dairy/Food	1.5	24	36
				567

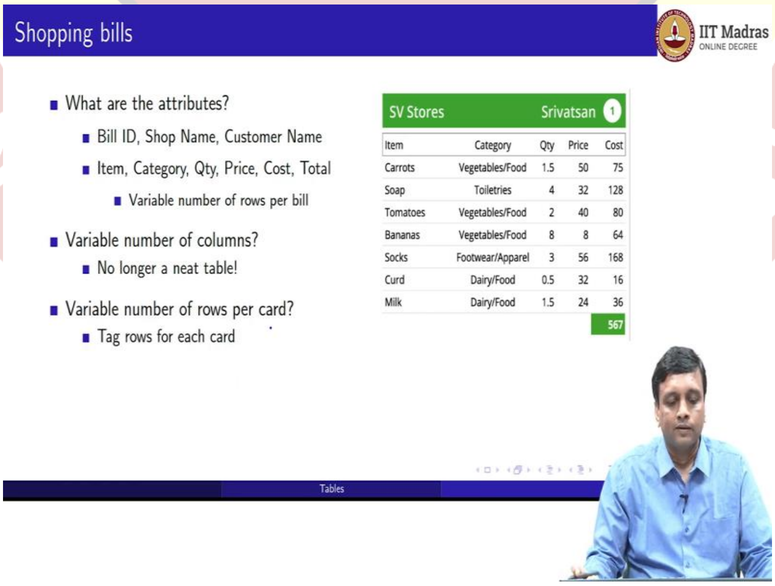
And the problem is that the number of rows is not the same. So, for instance, if you look at Akshaya’s bill, it has 5 items; and if you look at Srivatsan’s bill, it has 7 items. So, how do we keep track of this in a table?

Because if these are our attributes, if item, quantity, price, cost are also our attributes, then we have to store them as columns, but the number of columns will depend on the number of different copies of this which are there as rows in the shopping bill itself. So, this is a variable quantity. So, we could have a variable number of columns, right?

So, for each so we could have you know one block of columns corresponding to the first row of this, then another block corresponding to second row and so on. But then what we will have is that we will have Akshaya’s for example, Akshaya’s row will end here; Srivatsans row has two more items, so it will end here; somebody else might have a much smaller thing and so on.

So, we will have a very ugly structure which is not a table at all. Each row will have a different length. So, this is not a table at all, and this is not something that we want. So, how do we squeeze this information into a table format which has a rigid structure of having a fixed number of columns for every item in our data set?

(Refer Slide Time: 05:04)



Shopping bills

IIT Madras
ONLINE DEGREE

- What are the attributes?
  - Bill ID, Shop Name, Customer Name
  - Item, Category, Qty, Price, Cost, Total
    - Variable number of rows per bill
- Variable number of columns?
  - No longer a neat table!
- Variable number of rows per card?
  - Tag rows for each card

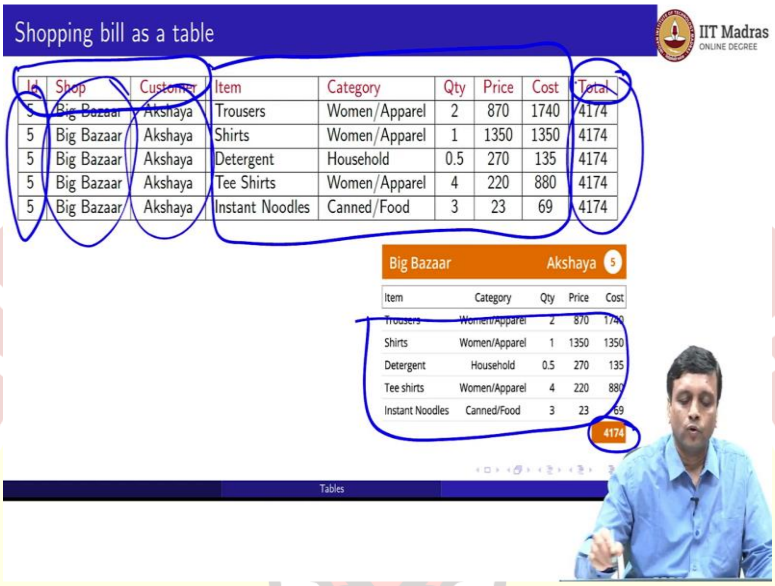
SV Stores Srivatsan 1

Item	Category	Qty	Price	Cost
Carrots	Vegetables/Food	1.5	50	75
Soap	Toiletries	4	32	128
Tomatoes	Vegetables/Food	2	40	80
Bananas	Vegetables/Food	8	8	64
Socks	Footwear/Apparel	3	56	168
Curd	Dairy/Food	0.5	32	16
Milk	Dairy/Food	1.5	24	36
				567

Tables

So, one solution is to have not one row per card, but multiple rows per card. Of course, then we have to make sure that all the rows for a given card are tagged in some way, so that we know that this whole thing corresponds to a single card.

(Refer Slide Time: 05:20)



Shopping bill as a table

IIT Madras ONLINE DEGREE

Id	Shop	Customer	Item	Category	Qty	Price	Cost	Total
5	Big Bazaar	Akshaya	Trousers	Women/Apparel	2	870	1740	4174
5	Big Bazaar	Akshaya	Shirts	Women/Apparel	1	1350	1350	4174
5	Big Bazaar	Akshaya	Detergent	Household	0.5	270	135	4174
5	Big Bazaar	Akshaya	Tee Shirts	Women/Apparel	4	220	880	4174
5	Big Bazaar	Akshaya	Instant Noodles	Canned/Food	3	23	69	4174

Big Bazaar Akshaya 5

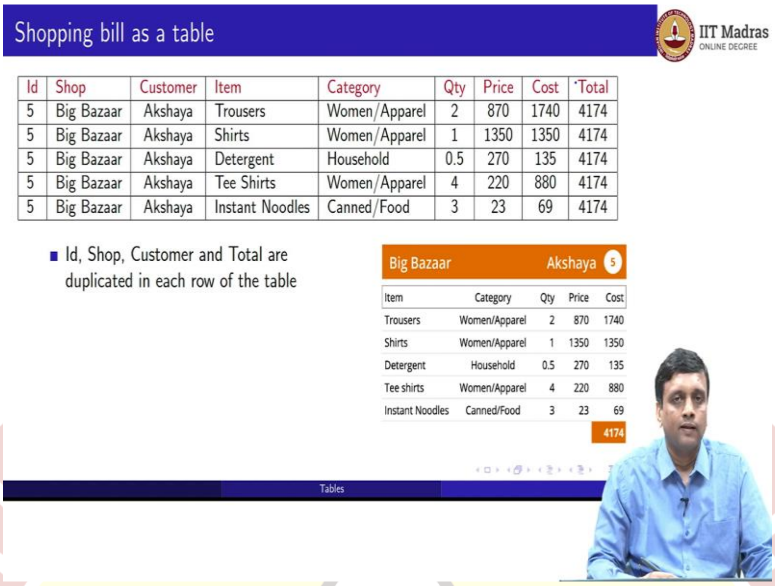
Item	Category	Qty	Price	Cost
Trousers	Women/Apparel	2	870	1740
Shirts	Women/Apparel	1	1350	1350
Detergent	Household	0.5	270	135
Tee shirts	Women/Apparel	4	220	880
Instant Noodles	Canned/Food	3	23	69
				4174

So, let us see how to do this. So, here is Akshaya’s card again. And here is now how we will represent it in a table with a fixed number of columns. So, we have these three columns which are of course not a problem at all, because we have only one Id, one shop, one customer on this card.

This column is also not a problem because we have only one total at the bottom of the card. But the other columns correspond to the different rows in the cards. So, we have these 5 rows in our table, and we have correspond 5 rows in our card correspondingly we have 5 rows in the table.

And how do we know that these are all in the same bill? Well, in particular, they all have the same bill Id. So, they must be from the same bill, so that is the real information that we need. But because we are forced to write down this information on every row, we also have 5 copies of the name of the shop, 5 copies of the name of the customer, and 5 copies of the total right.

(Refer Slide Time: 06:12)



Shopping bill as a table

Id	Shop	Customer	Item	Category	Qty	Price	Cost	Total
5	Big Bazaar	Akshaya	Trousers	Women/Apparel	2	870	1740	4174
5	Big Bazaar	Akshaya	Shirts	Women/Apparel	1	1350	1350	4174
5	Big Bazaar	Akshaya	Detergent	Household	0.5	270	135	4174
5	Big Bazaar	Akshaya	Tee Shirts	Women/Apparel	4	220	880	4174
5	Big Bazaar	Akshaya	Instant Noodles	Canned/Food	3	23	69	4174

- Id, Shop, Customer and Total are duplicated in each row of the table

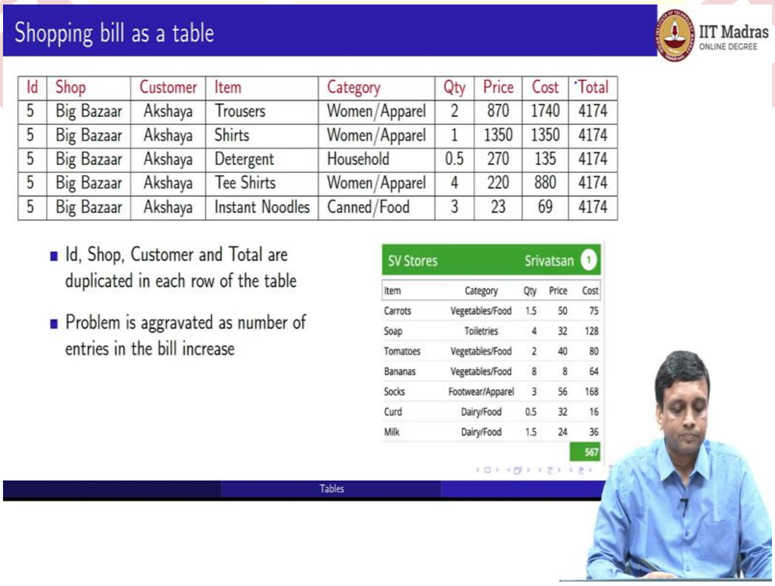
Big Bazaar Akshaya 5

Item	Category	Qty	Price	Cost
Trousers	Women/Apparel	2	870	1740
Shirts	Women/Apparel	1	1350	1350
Detergent	Household	0.5	270	135
Tee shirts	Women/Apparel	4	220	880
Instant Noodles	Canned/Food	3	23	69
				4174

So, we have this unnecessary duplication of shop, customer, and table and total across the table. We have to duplicate Id because if we do not duplicate the id, how would we know that these 5 rows belong to the same bill.

So, we need some way of connecting them by having some value which is common to all of them to say that they are the same bill, but we need not and we do not want to unnecessarily duplicate or we should not ideally unnecessarily duplicate because we are wasting a lot of space in keeping track of this data.

(Refer Slide Time: 06:41)



Shopping bill as a table

Id	Shop	Customer	Item	Category	Qty	Price	Cost	Total
5	Big Bazaar	Akshaya	Trousers	Women/Apparel	2	870	1740	4174
5	Big Bazaar	Akshaya	Shirts	Women/Apparel	1	1350	1350	4174
5	Big Bazaar	Akshaya	Detergent	Household	0.5	270	135	4174
5	Big Bazaar	Akshaya	Tee Shirts	Women/Apparel	4	220	880	4174
5	Big Bazaar	Akshaya	Instant Noodles	Canned/Food	3	23	69	4174

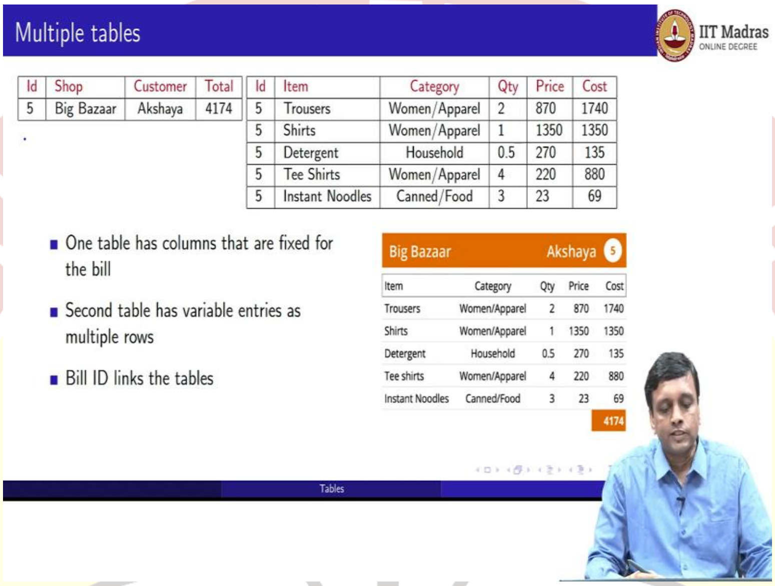
- Id, Shop, Customer and Total are duplicated in each row of the table
- Problem is aggravated as number of entries in the bill increase

SV Stores Srivatsan 1

Item	Category	Qty	Price	Cost
Carrots	Vegetables/Food	1.5	50	75
Soap	Toiletries	4	32	128
Tomatoes	Vegetables/Food	2	40	80
Bananas	Vegetables/Food	8	8	64
Socks	Footwear/Apparel	3	56	168
Curd	Dairy/Food	0.5	32	16
Milk	Dairy/Food	1.5	24	36
				567

And now if we move to a larger bill, for example, Srivatsan’s bill, then the duplication would become 7 rows instead of 5 rows. So, this is one unavoidable consequence of keeping multiple rows corresponding to variable length entries in a card. So, when we have data like the shopping bill data, we have a small problem translating it to tables because of having to make these kind of adjustments.

(Refer Slide Time: 07:12)



Multiple tables

Id	Shop	Customer	Total
5	Big Bazaar	Akshaya	4174

Id	Item	Category	Qty	Price	Cost
5	Trousers	Women/Apparel	2	870	1740
5	Shirts	Women/Apparel	1	1350	1350
5	Detergent	Household	0.5	270	135
5	Tee Shirts	Women/Apparel	4	220	880
5	Instant Noodles	Canned/Food	3	23	69

- One table has columns that are fixed for the bill
- Second table has variable entries as multiple rows
- Bill ID links the tables

Big Bazaar Akshaya 5

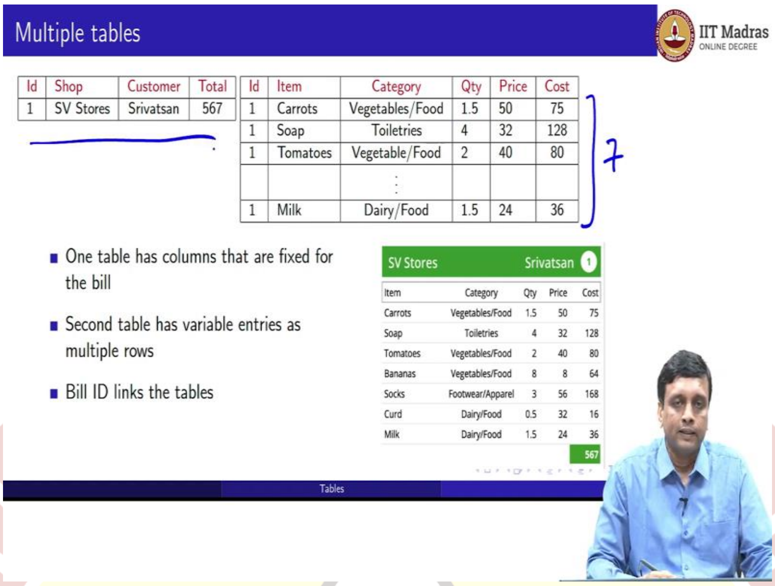
Item	Category	Qty	Price	Cost
Trousers	Women/Apparel	2	870	1740
Shirts	Women/Apparel	1	1350	1350
Detergent	Household	0.5	270	135
Tee shirts	Women/Apparel	4	220	880
Instant Noodles	Canned/Food	3	23	69
				4174

There is a slight improvement that we can do which is to split this actually into two separate tables. One table corresponding to the values which are fixed for a card, and one table corresponding to the variable rows that we need to represent multiple columns in the card.

So, for instance, in this case, we will table these three constant value shop, customer total, customer and total value, and put it into a separate table tagged by this bill id. And separately, we will have another set of 5 rows tagged by the bill Id which has these 5 columns, which change from row to row.

So, now, together we can merge these two to form the total bill if we need it, right. So, we have one table for the fixed columns, and the second table for the variable entries. Now, the advantage is that the only duplication that we have is the absolute minimum duplication which is that we need to record the bill Id for each row in the second table, so that we can link this bill Id to that bill Id and make sure that together all of these constitute the same bill.

(Refer Slide Time: 08:18)



Multiple tables

Id	Shop	Customer	Total
1	SV Stores	Srivatsan	567

Id	Item	Category	Qty	Price	Cost
1	Carrots	Vegetables/Food	1.5	50	75
1	Soap	Toiletries	4	32	128
1	Tomatoes	Vegetable/Food	2	40	80
		⋮			
1	Milk	Dairy/Food	1.5	24	36

7

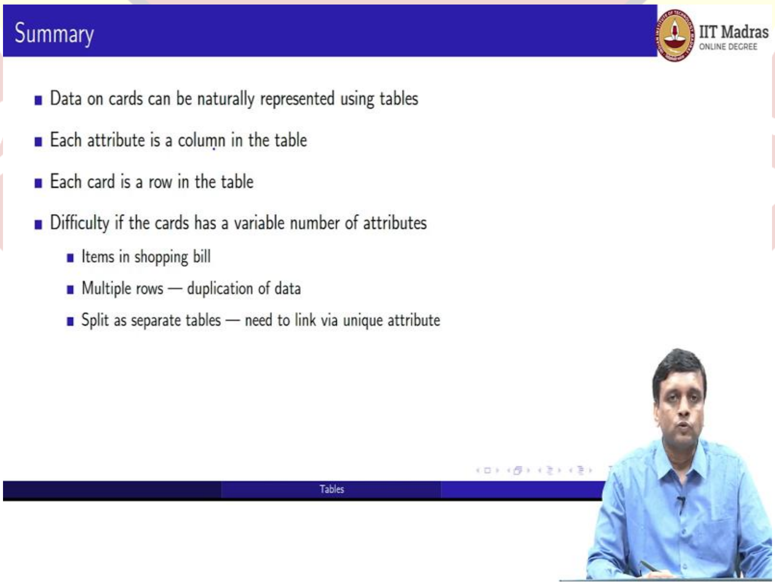
- One table has columns that are fixed for the bill
- Second table has variable entries as multiple rows
- Bill ID links the tables

SV Stores Srivatsan 1

Item	Category	Qty	Price	Cost
Carrots	Vegetables/Food	1.5	50	75
Soap	Toiletries	4	32	128
Tomatoes	Vegetables/Food	2	40	80
Bananas	Vegetables/Food	8	8	64
Socks	Footwear/Apparel	3	56	168
Curd	Dairy/Food	0.5	32	16
Milk	Dairy/Food	1.5	24	36
				567

So, for instance, if we look at Srivatsan’s bill, then this would have now 7 rows here because there are 7 items in the bill, but it has only one copy of the customer information recorded with the bill that is the name of the shop, who the customer was, and what the total amount was. So, this is another way of representing this kind of variable data in a table.

(Refer Slide Time: 08:38)



Summary

- Data on cards can be naturally represented using tables
- Each attribute is a column in the table
- Each card is a row in the table
- Difficulty if the cards has a variable number of attributes
  - Items in shopping bill
  - Multiple rows — duplication of data
  - Split as separate tables — need to link via unique attribute

So, to summarize, it is convenient to take the data that we have been working with on cards and put it into a table, because tables are much easier to represent, and they also keep all the data in one single data structure. So, in the table, each column represents an attribute and each row represents a card. But we saw that with the shopping bill kind of data, if you have some variation in the attributes across cards, then you will have a variable number of columns.

So, one solution for this is to have multiple rows for each of the variable quantities in the card, but then you have to duplicate all the fixed quantities. The other solution which is more complicated is to somehow split it into two separate tables. One of them has the fixed quantities; one of them has the variable quantities.

And then you need to use some unique attribute like the bill Id which is not shared across different data items to connect these two tables, and make sure that all the data for a single bill can be recovered from these tables.