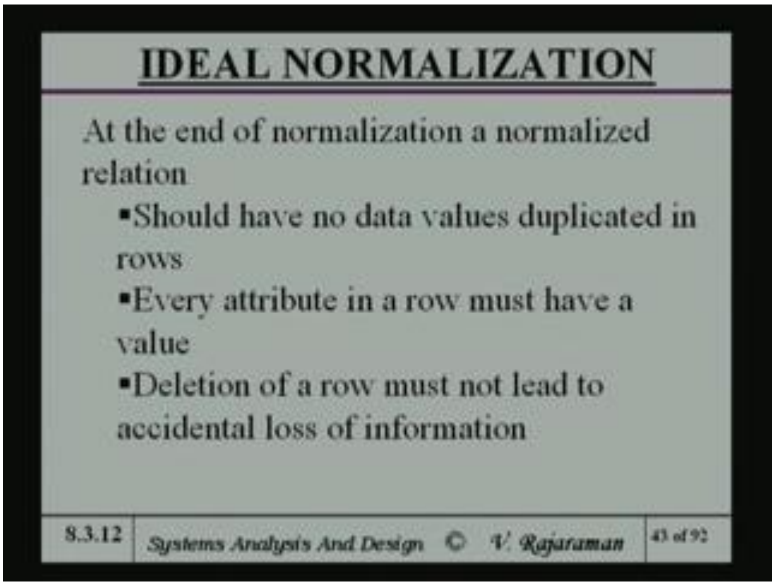
System Analysis and Design
Prof. V Rajaraman
Department of Super Computer Education & Research
Indian Institute of Science, Bangalore

Lecture - 24

Last time, we were discussing the general idea of normalization. And I pointed out a very important part of relation data base theory is the idea of normalization.

(Refer Slide Time: 01:35)



IDEAL NORMALIZATION

At the end of normalization a normalized relation

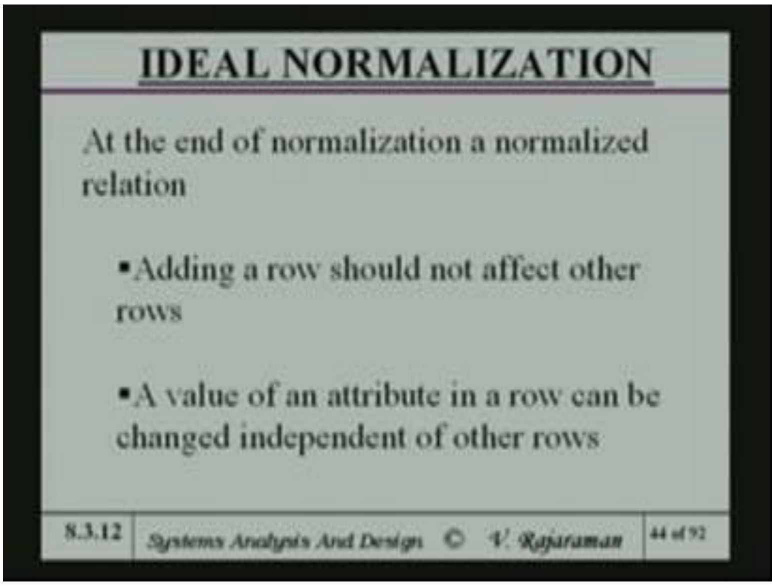
- Should have no data values duplicated in rows
- Every attribute in a row must have a value
- Deletion of a row must not lead to accidental loss of information

8.3.12 Systems Analysis And Design © V. Rajaraman 43 of 92

The whole region, why we normalize is that at the end of the normalization step, should have no data value duplicated in many rows. that is more than two rows are not be identical. Also, very not to duplicate the same data in more than one row. Every attribute the row must have a value that is little many blanks, in any attribute. And deletion of the row must not lead to accidental loss of information. Like, last time we saw that if you delay in order, the price of our item was lost in that process.

And later on we want to order that items even know the price in order to get a gain the price from the vendor.

(Refer Slide Time: 02:32)



IDEAL NORMALIZATION

At the end of normalization a normalized relation

- Adding a row should not affect other rows
- A value of an attribute in a row can be changed independent of other rows

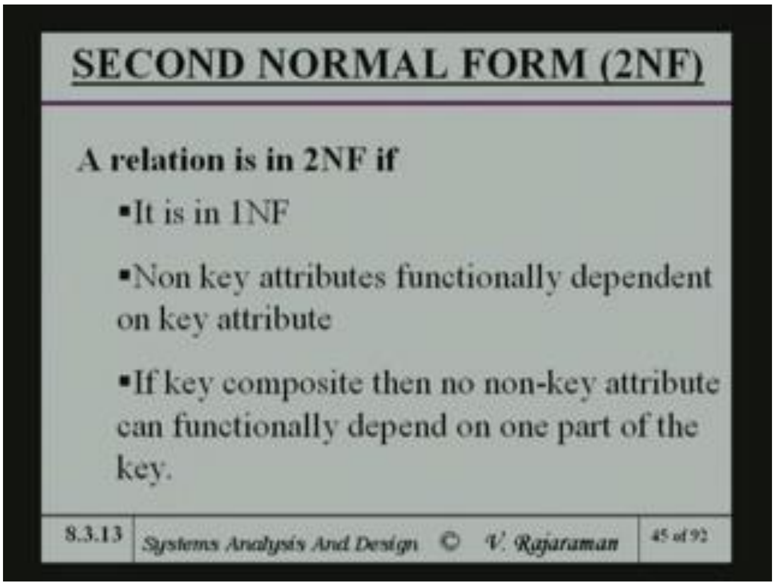
8.3.12 Systems Analysis And Design © V. Rajaraman 44 of 92

Further, add your row, there is updating the relation should not affect other rows. In other words, whatever you add should be orthogonal to whatever already exists. A value of an attribute in a row can be changed independent of other rows. So, in other words, attribute values any particular arbitrary attribute value even to be updated. You should be able to update it without disturbing. The entire there is ((Refer Time: 03:10)) the entire relation database might say.

Now, we look at the first normal form. First normal form is really nothing but, converting an arbitrary kind of a format to a format, which is really a properly define other relation. That means, every row has an n-triple. And there is a plot file whereas, there is no particular attribute is got multiple values of that, you do not get a variable record light from the pair lines file design. All the records are equal length.

So, end up duplicating some of the rows to be able to get the first normal form relation. That is beginning. In fact, it is not even normalized might say. Just see, 0 state, at which we start the normalization step. The first real normalization step is the, what is known as a second normal form.

(Refer Slide Time: 04:29)



SECOND NORMAL FORM (2NF)

A relation is in 2NF if

- It is in 1NF
- Non key attributes functionally dependent on key attribute
- If key composite then no non-key attribute can functionally depend on one part of the key.

8.3.13 Systems Analysis And Design © V. Rajaraman 45 of 92

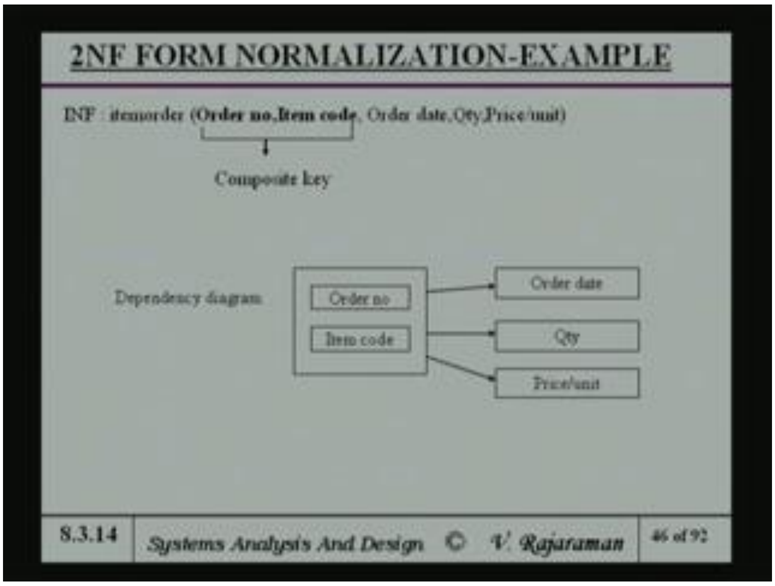
Commonly, written as 2NF or the relation is said to 2NF provided already in 1NF. Nevertheless, we always said we already said that, normalization goes step after step after step. You start from the first normal form, go to second normal form go to the third normal form. Then you go to the Boyce codd normal form, then fourth normal form and fifth normal form and so on. And each, higher normal form is the refinement, over a earlier form based on some difficulties, we will encounter in the normalization process.

And so, the normalization process as I said, we always start with first normal form. Then you go to the second normal form. Before, you go to the third normal form, you should already been second normal form. So, in other words, to a second normal form is contained in the third normal form might say. And then, when you go to the BCNF, then unless the relation already in the third normal form, you not go to the next higher step. So, you go to higher and higher step, from lower and lower steps.

So, a relation is said to be in second normal form, you are already in first normal form. And no non key attribute is functionally dependent on key attributes. Otherwise, the problem which arose the form, when you deleted was that the item code price are related. And so when we removed on item from the order the price was lost. So, that means this case, non key attribute namely priced per unit was depended on a key attribute.

If key composite then non-key attribute can be functionally depended can one part of the key. Then no non key attribute. In other words, there is a composite attribute the composite attribute together should determine the other attributes. And not a portion of part of the composite attribute. So, let us go back to example.

(Refer Slide Time: 07:07)



2NF FORM NORMALIZATION-EXAMPLE

1NF : itemorder (Order no,Item code, Order date,Qty,Price/unit)

Composite key

Dependency diagram

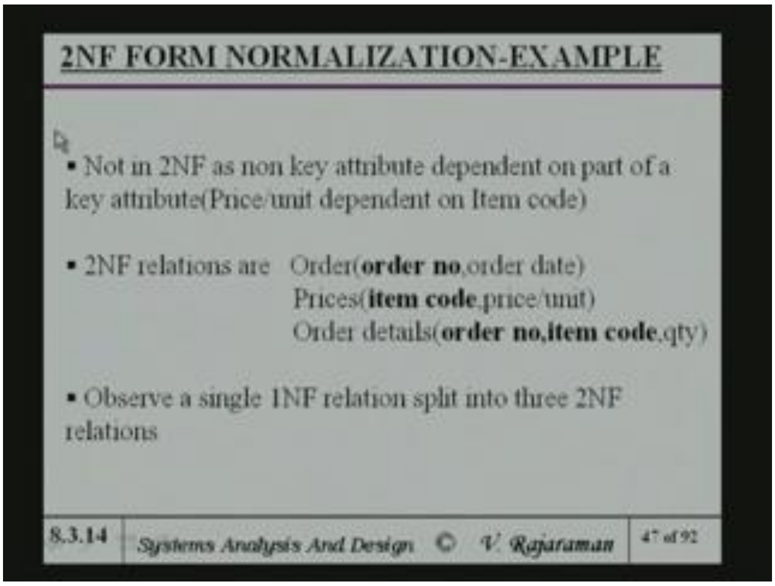
S.3.14 Systems Analysis And Design © V. Rajaraman 46 of 92

We had order number item code. So, complex number attribute and we have order date quantity price per unit, which is original first normal form relation, which is we started. And I look at the dependency diagram. And order no item code is together and as I shown in this diagram. Order date quantity and price per unit are shown as the, this fully dependent on the composite attribute.

But, it is not really so. Because, the price per unit depends on the, if you look at this the item code the price per unit is dependent on item code. And so in fact, this line should be really extended to that point. So, in the future I extend that whenever there is a dependency of the non key attribute to a part of a composite key attribute. This the region I shown this way is that the tendency, if you look at this ((Refer Time: 08:16)) blindly like this. And which is in fact, in error.

Because, we would know that from a waste statement that price per unit is depended on the item code.

(Refer Slide Time: 08:34)



2NF FORM NORMALIZATION-EXAMPLE

- Not in 2NF as non key attribute dependent on part of a key attribute(Price/unit dependent on Item code)
- 2NF relations are Order(**order no**,order date)
  Prices(**item code**,price/unit)
  Order details(**order no,item code**,qty)
- Observe a single 1NF relation split into three 2NF relations

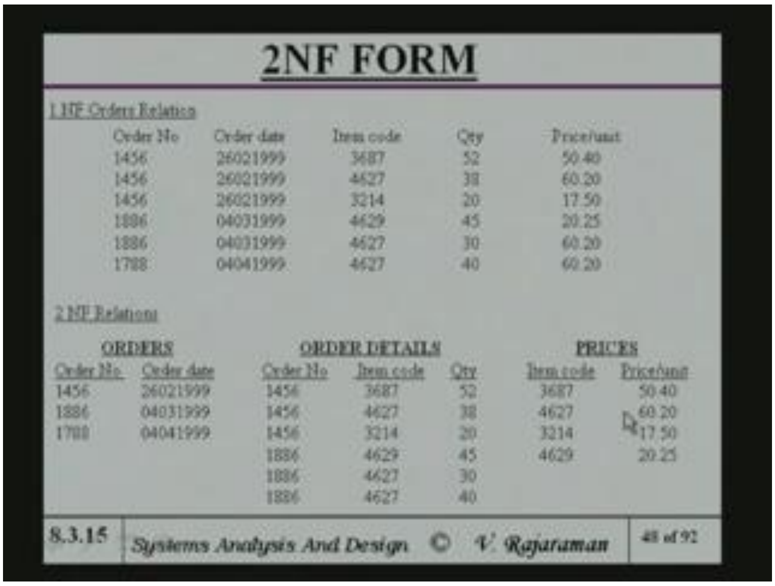
8.3.14 Systems Analysis And Design © V. Rajaraman 47 of 92

So, it is not in second normal form as non key attribute dependent on part of the key attribute, price per unit dependent on item code. So, you break up that first normal form relation into 3 second normal form relations. In other words ((Refer Time: 08:51)) you go under the one non single relation you break it up. So, that this problem does not arrive in that. So, order number order date is one relation, which is completely self contain there is order date is depended on the order number and nothing else.

And prices item code price per unit, price per unit depends only on item code. And order details, for a given order order number and item code together declaiming the quantity. And so you could actually, put the order details as this. Observe a single INF is making the three 2NF relations, which is in some sense you might say, I seem to be actually making the result of the normalization more complex by putting more relations or adding more relations.

But, it is not really true. So, because when I add this relation, I am really making sure that no problem arises. When I update or when I delete or when I do certain operations like addition and. so on. In other words, suppose no items is going to be added, which is not already added.

(Refer Slide Time: 10:22)



2NF FORM

1 NF Orders Relation

Order No	Order date	Item code	Qty	Price/unit
1456	26021999	3687	52	50.40
1456	26021999	4627	38	60.20
1456	26021999	3214	20	17.50
1886	04031999	4629	45	20.25
1886	04031999	4627	30	60.20
1788	04041999	4627	40	60.20

2 NF Relations

ORDERS

Order No	Order date
1456	26021999
1886	04031999
1788	04041999

ORDER DETAILS

Order No	Item code	Qty
1456	3687	52
1456	4627	38
1456	3214	20
1886	4629	45
1886	4627	30
1886	4627	40

PRICES

Item code	Price/unit
3687	50.40
4627	60.20
3214	17.50
4629	20.25

8.3.15 Systems Analysis And Design © V. Rajaraman 48 of 92

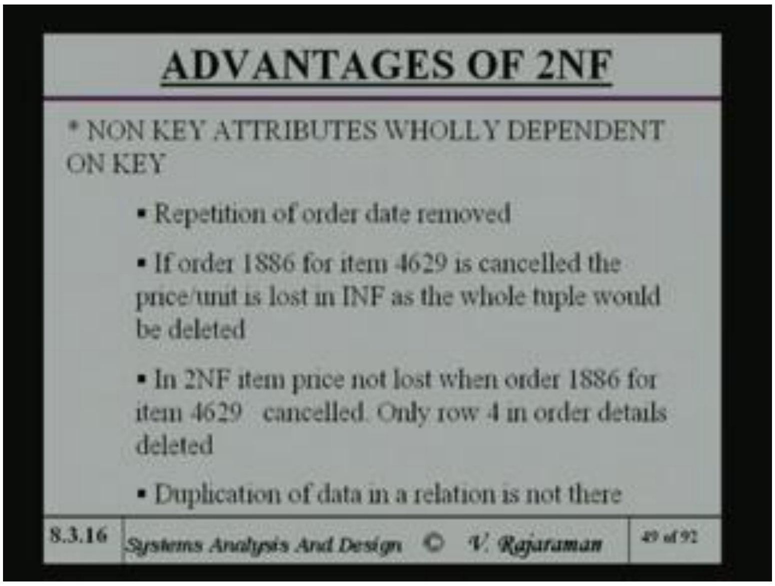
This case, I want to add an item code to this price in the prices list. Without any others, because order is not there and order details the order is not placed. So, item will not appear here. And the because the order is not placed it have a other prior. Only this will be updated, I put new item code and put a new price. So, updating is kind of independent. Suppose, order number 1456 is cancelled. Then, if I cancel this, this and this these three. And similarly, this will be cancelled. There is in other words, it has been deleted.

And in that process, I will not touch this one. Because, this one is not related the in a given order. So, the point I am trying to make is that there is a ((Refer Time: 11:20)) kind of independence with the updating as of this. For instance suppose, I change the date the order. But, retain the item code order number and quantity then only date can be changed on this without disturbing any other relations. So, what is really happens is that relation, addition changes I want to change the price of an item 4627 like 68 20 the price has gone to 62 rupees 20 pica or something like that.

When I just change this here, I want any of these things. So, similarly as I said the date of the order is changed only the step this. I will not touch any of them here. So, where is great advantage of the normalization to second normal form. In other words, the I am kind of making the relations, even though the multiple relations in such a way, that any updation, any deletion or any addition does not cause unnecessary difficulties or ambiguities in the result.

We will essentially, what the whole idea is of normalization. Of course, there are same operations, like joint project and so on. Which we are not going to talk about in this course, which will allow for any kind of applications to be able to combine more than one relation together, like combined these two for some purposes. But, I am not concerned about it now. I am only looking at the question of normalization. So, we have come to second normal form.

(Refer Slide Time: 13:23)



ADVANTAGES OF 2NF

* NON KEY ATTRIBUTES WHOLLY DEPENDENT ON KEY

- Repetition of order date removed
- If order 1886 for item 4629 is cancelled the price/unit is lost in INF as the whole tuple would be deleted
- In 2NF item price not lost when order 1886 for item 4629 cancelled. Only row 4 in order details deleted
- Duplication of data in a relation is not there

8.3.16 | Systems Analysis And Design © V. Rajaraman | 49 of 92

Now, the next advantage of second normal form which of course, already said that let me just relate to it. Non key attributes are only dependent on the key. That is what the make sure. So repetition of order date is removed. Because, there is separate particular relation, if an order number for particular item is cancelled price per unit is not lost. And then if previous case price is lost for a particular item cancelled the orders are cancelled. And also, pointed out that in a price changes. I can change that relation not such any of them.

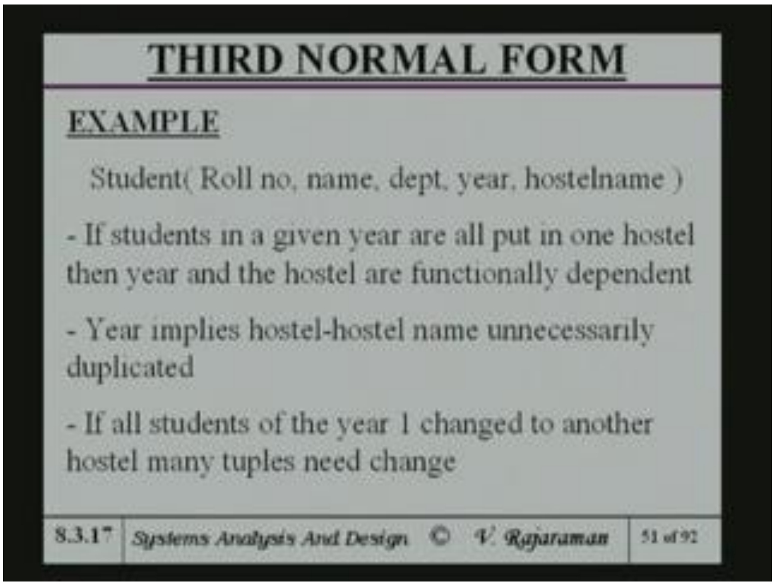
Duplication of data is not there. And it is kind of reduced. You might of course, ((Refer Time: 14:15)). And there is some duplication same that the order number is seem to appear again and again. But, then the reason is there otherwise, if I put these together then this will become and one become plat file, it not been first format at all.

So, the reason this get repeated ((Refer Time: 14:37)) for that I need to have a section of the plat file or relation their every attribute. Every nth flow of equal size, there is 3 triple, 3 triple, 3 triple in this case. So, they are all 3 triples and that is the reason why.

I make sure, that in are the even though it looks like a repetition. But, you can see here. One more repetition here, there is no repetition here. And this repetition is a kind of the quart repetition. So, you do not start with second normal form. We said that, second normal form has certain advantages. That is care about, because of the fact that we would normalization first normal form. But this is not the end of the game. And the reason why, there is higher normal forms.

Particularly, a third normal form, you should understand very well. Because, in many situations. In real practice if I do the day after third normal form.

(Refer Slide Time: 16:07)



THIRD NORMAL FORM

EXAMPLE

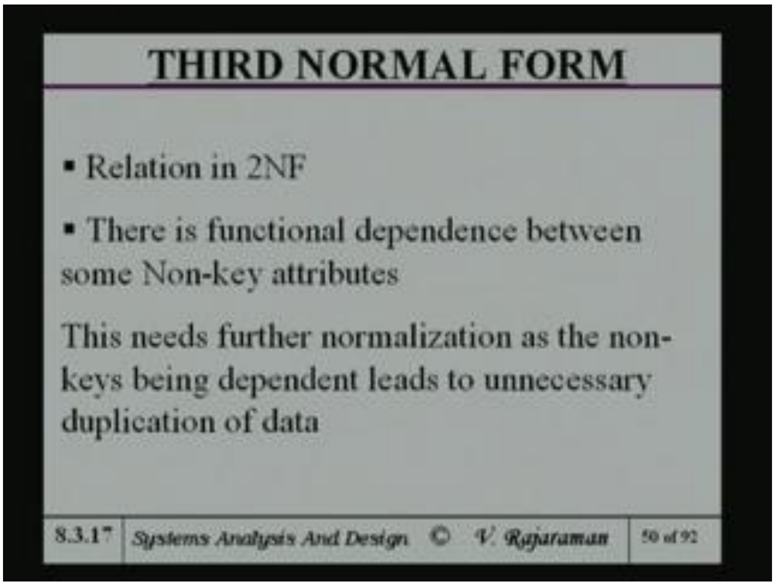
Student(Roll no, name, dept, year, hostelname)

- If students in a given year are all put in one hostel then year and the hostel are functionally dependent
- Year implies hostel-hostel name unnecessarily duplicated
- If all students of the year 1 changed to another hostel many tuples need change

8.3.17 | Systems Analysis And Design © V. Rajaraman | 51 of 92

Then, I spot at the third normal form. Because, it need not be necessary to go to higher normal forms. But, I am not saying that you should stop there, depends up on the problem. So, but third normal form is a very important part. Because, that is the one which from where ((Refer Time: 16:31)) build on to higher normal forms. So, third normal form in order to get something to a third normal form. The relation should be already in second normal form.

(Refer Slide Time: 16:45)



THIRD NORMAL FORM

- Relation in 2NF
- There is functional dependence between some Non-key attributes

This needs further normalization as the non-keys being dependent leads to unnecessary duplication of data

8.3.17 Systems Analysis And Design © V. Rajaraman 50 of 92

And there is no functional dependency between some non key attributes. In second normal form, we taught only about functional dependency between the part of the ((Refer Time: 16:58)) composite key. A non key attributes build a function part of the key attributes. That is what we talked about in the second normal form. In this case, we are talking about the non key attributes. There should not be any dependent ((Refer Time: 17:18)) some of the attributes.

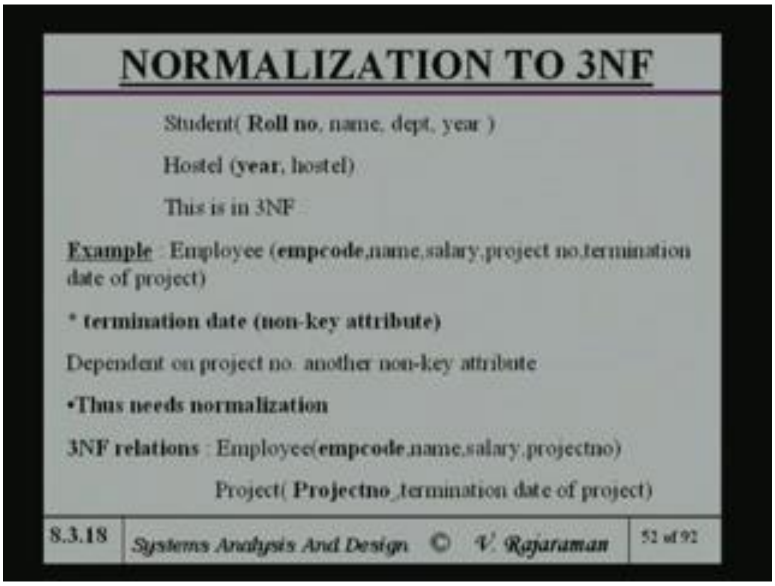
If that happens, there is a some little difficulty. So, it needs further normalization, if you find there is a dependency. But, in some non key attributes. Because, there is unnecessary duplicated. Let me, give an example ((Refer Time: 17:36)) of what I mean, the unnecessary duplication of day time. And what is meant by, the functional dependency between non key attributes. Take a relation student and giving the important parts of this, important attributes handling.

There will be many more attributes. But I am not concerning example with all the other attributes. So, students the key ((Refer Time: 18:12)) is role number. And I mean department, year and hostel name. Now, if all the students in say first year or put in hostel a. And all students in second year are put in hostel b. And all students in third year put in hostel c and so on.

Then it turns out, that the hostel name and year are functionally related. That means, the students happens to be second year. You know about implication that he will be in hostel number b. If in first year you know he will be in hostel a. So Year implies the hostel and hostel name unnecessarily duplicated. If all students in the first year are changed to another hostel. Many tuples need to change. Other words, we would decide to build many hostel, hostel d.

Here, all incoming new students. They ((Refer Time: 19:24)) put in new hostel. Then, you have to kind of go through the entire tuples set to change everywhere. Where first year new hostel appears to first year in d hostel. So, the lot of search and unnecessary work is goes on. So, you change it into true relation that is you split up their sending relation into co relation. One is student roll number, name, department and year.

(Refer Slide Time: 20:00)



NORMALIZATION TO 3NF

Student(**Roll no**, name, dept, year)

Hostel (**year**, hostel)

This is in 3NF

Example : Employee (**empcode**,name,salary,project no,termination date of project)

*** termination date (non-key attribute)**

Dependent on project no: another non-key attribute

•Thus needs normalization

3NF relations : Employee(**empcode**,name,salary,projectno)

Project(**Projectno** ,termination date of project)

8.3.18 | Systems Analysis And Design © V. Rajaraman | 52 of 92

Now, observe that the student relation has the non key attributes or not functionally related. Because, the year of the student is not depended on the ((Refer Time: 20:25)). There are students of third year in computer science department, also the student in mechanical department and so on. They are not actually functionally depended. So obviously, name is not functionally dependent on either department or year. There may multiple ramasamy in the class.

And the unique identifier is the roll number. So, you might the name is related to roll number. But it is not related to any other non key attributes. So, when you note that first relation, where the all non key attributes are depended only on the key attributes. Or non key attributes are not depended on one another. And then, you create another relation hostel, where year become is the key norm. And hostel name becomes the attributes.

So, everybody in the first year is going to hostel a. It will be just 1 and a. ((Refer Time: 21:37)) suppose there are in a college there are 4 years. This relation will have only 4 year entries. We got project number also then you would have 5 entries. If, the fifth hostel mentioned. I am making assumption here, that all students in a given year are put in one hostel. If any some situations, students they have mixed up.

And the year of the student does not uniquely determine the hostel in which he stays. Then of course, this normalization step is not necessary. Because, the fact that the year and hostel will not be functionally depended. Because, it is arbitrary, the hostel the person goes to he has to, have look at the entire relation. Because, a mass students one of being shifted. Implicate an example in a employ relation the employ code is the key ((Refer Time: 22:45)) for this relation.

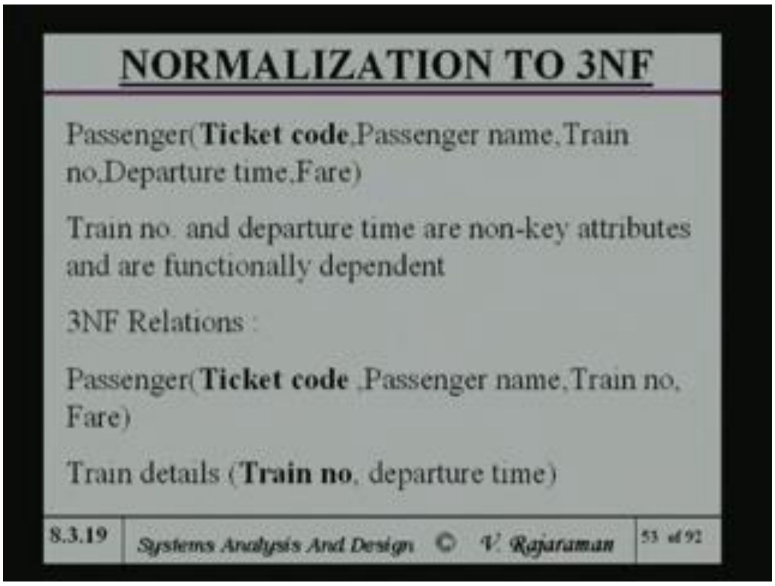
And their name, salary, project number, some project in which the employee working. So, company or something like that, there is number of project and each project has been given by a number. And the particular project you come to termination date, may be project terminates. The employ may be shifted to some other project or the project terminates the employ also be terminated, depends up on the policy of the company.

Whatever days, if you look at the termination date of project it is functional dependent on the project number. Because, any given project has the unique termination date. So, non key attributes in the termination date. And the non key attributes, if you have project number are actually related. So, two non key attributes are functionally dependent. So, this is the normalization. So, there is a third normal form, where you put employee, employee code name, salary and project number.

And then separately, you create a project relation where the project number and termination date of the project is given. The advantage of this is suppose the project termination date is changed. You only touch the project relation you do not do anything to the employ relation. And there is advantage. And you can somewhat independent it becomes and suppose there is primarily the date is changed.

If it is not normalized then you have to go through the entire database. And every ((Refer Time: 24:46)) project number appears, we have to change termination date. Whereas, in this case you searched only one relation, you do not touch the other relation.

(Refer Slide Time: 25:04)



NORMALIZATION TO 3NF

Passenger(**Ticket code**,Passenger name,Train no,Departure time,Fare)

Train no. and departure time are non-key attributes and are functionally dependent

3NF Relations :

Passenger(**Ticket code** ,Passenger name,Train no, Fare)

Train details (**Train no**, departure time)

8.3.19 Systems Analysis And Design © V. Rajaraman 53 of 92

If you take another example, Passenger relation ticket codes, if the passenger buy the ticket, the ticket code like a number which is given by railways called train no or whatever. And that ticket code and there is any identify the ticket. In passenger name, train number, departure time and fare. Now, once see that train number is related to the departure time. So, non-key attribute train number and the departure time are related.

So, you suppose the departure time of the specific train is changed. Then, in this relation you had go through all the tickets with the train number. And change all the places where the departure time is mentioned. So, you know the actually, developed into two parts, ticket number, passenger name and fare fate quite. Because, ((Refer Time: 26:20)) it is not completely relation. Because, if you buy a ticket, you have a origin station and designation station and the fare. All the fact also is normally given.

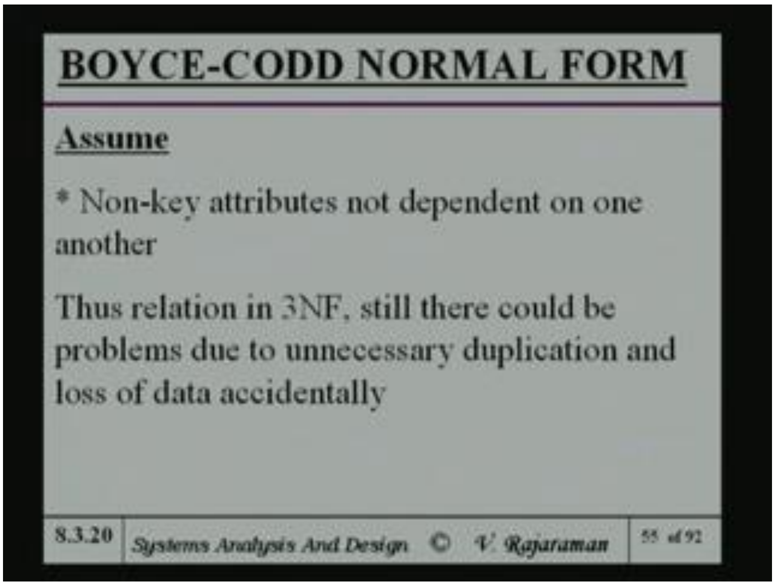
That, for say just an illustration, I am not putting the whole details of a ticket. I am just abstracting out some part, some important parts. Of course, this to illustrate the fact that a non key attribute like train number and the departure time they are related.

And so, if departure time is changed, you had only such that. And so you got, if a ((Refer Time: 27:05)) similarly, if the passengers cancels the ticket. If the only one passenger is traveling the particular train, it happens the particular origin station.

Then the information of the departure time may be lost. Because, ((Refer Time: 27:27)) case. So, I will not lose that is when passengers cancels the ticket, unnecessarily some required data may not be lost accidentally. That is what we showed could happen in previous example. Here also, same thing. So, if I put train number and departure time, in the appropriate relation. So, there is no possibility of using the data that is the advantage.

Now, the third normal form is as I said, even in a good place where many people stop. That as I said, it is not correct thing to do ((Refer Time: 28:23)). You must look at further possibilities and further normalization step this may become necessary.

(Refer Slide Time: 28:36)



BOYCE-CODD NORMAL FORM

Assume

* Non-key attributes not dependent on one another

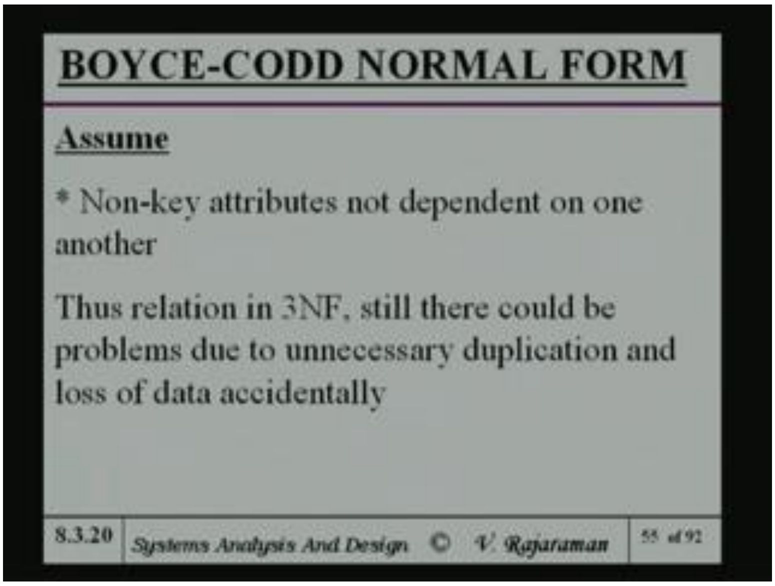
Thus relation in 3NF, still there could be problems due to unnecessary duplication and loss of data accidentally

8.3.20 Systems Analysis And Design © V. Rajaraman 55 of 92

And among them, the Boyce codd normal form or BCNF is not important one. And assume that, a relation has more than one possible key. And the keys are composite and composite keys have common attributes, over assumption. Only in such cases, we required the normalization steps of BCNF. Other words, relation has the possibilities more than one possible key and keys are composite. And composite keys have some common attribute.

For example, I will give an example non key attributes not depended one another. Of course, because already in 3NF. So, non key attributes are independent of one another. So, there could be a problem due to unnecessary duplication and loss of data accidentally, even though is in third normal form.

(Refer Slide Time: 29:42)



BOYCE-CODD NORMAL FORM

Assume

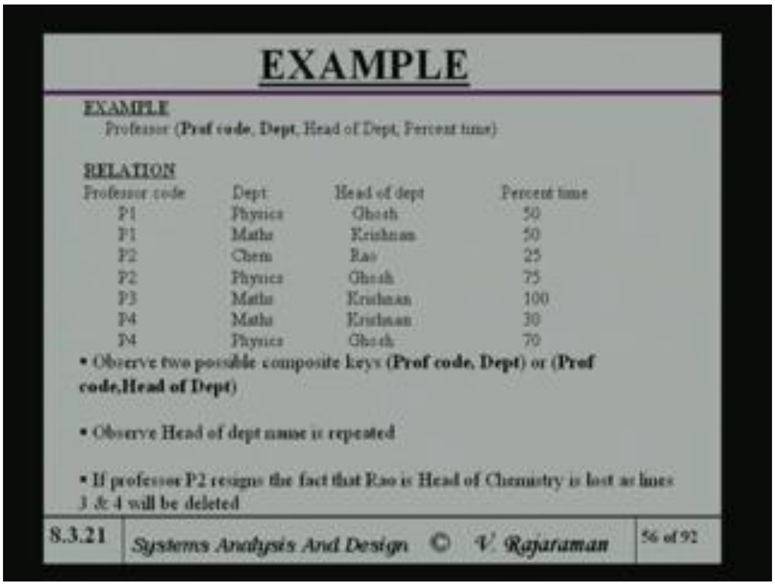
* Non-key attributes not dependent on one another

Thus relation in 3NF, still there could be problems due to unnecessary duplication and loss of data accidentally

8.3.20 *Systems Analysis And Design* © *V. Rajaraman* 55 of 92

Because, of this difficulty ((Refer Time: 29:44)) more than one possible key in case composite. So, take an example.

(Refer Slide Time: 29:48)



EXAMPLE

EXAMPLE

Professor (**Prof code, Dept**, Head of Dept, Percent time)

RELATION

Professor code	Dept	Head of dept	Percent time
P1	Physics	Ghosh	50
P1	Maths	Krishnan	50
P2	Chem	Rao	25
P2	Physics	Ghosh	75
P3	Maths	Krishnan	100
P4	Maths	Krishnan	30
P4	Physics	Ghosh	70

- Observe two possible composite keys **(Prof code, Dept)** or **(Prof code,Head of Dept)**
- Observe Head of dept name is repeated
- If professor P2 resigns the fact that Rao is Head of Chemistry is lost as lines 3 & 4 will be deleted

8.3.21 *Systems Analysis And Design* © *V. Rajaraman* 56 of 92

So, these again as an example with composite key, professor code, department code or the head of the department and there is a percentile. The particular professor ((Refer Time: 30:07)) in the department. Many universities have so called joint appointments. There is a teacher, teacher is more than one department. Part time in one department, part time in another department and so on. These examples, assume that such situations arises.

So, you look at professor code and there is a department name and another department and percentile. So, if I put everything in one relation. I will get this and because the non key items. And another department and percentile of the non key attributes. They are not related. So, it is already in third normal form. So, in this case, there is a professor code, department code is one composite key.

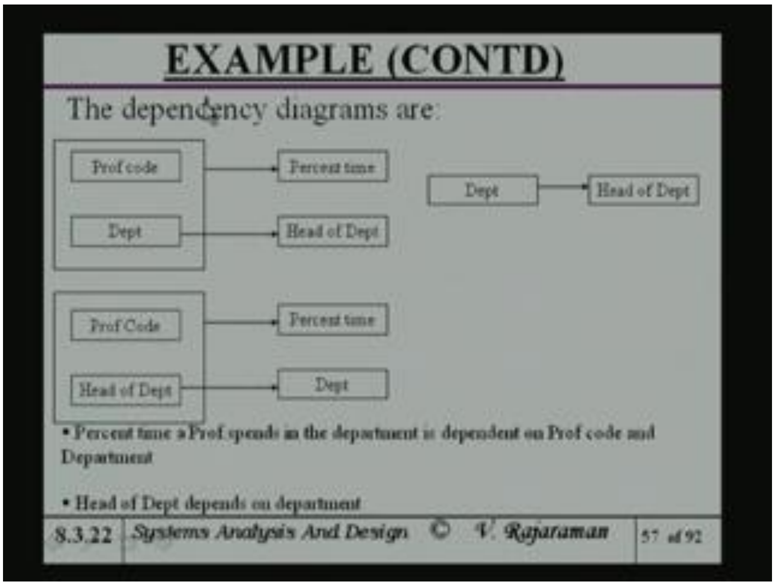
Is also another composite key, because every department has normally got only one department. Normally, departments do not have multiple heads for obvious reason. There should be one person, where the department. So, the department and head of the department are related. In other words, every department is could be a key also. And so, there are two possible composites keys. These 3NF arises only when this situation arises.

Namely, you can take another non key attributes. And it can, it is carry take key attributes in a composite key. That is the terminology people use carry take possible candidate for a big part of a composite key. Observe in this case, that the head of the department name is repeated, like professor P 1 teacher in physic with 50 percentage time. And professor P 2 is in physics for 75 percentage time. And Ghosh is the head of the department. So, Ghosh gets repeated.

Similarly, P 1 P 3 and P 4 or in the math’s department and 50 percent. And there is a professor P 1 is in for 50 percent. Whereas, P 3 100 percent and that is the P 4 is only 30 percent and 70 percent in physics. So, Krishnan there is in three places. In order to, if professor P 2 resigns there is suppose professor P 2 resigns. Then what you do, only these two tuple from the database. That is, if you delete this tuple and you delete this tuple.

If you delete these two the fact that, Rao there was the department head of chemistry is lost. Because, Rao appears in the only one place in ((Refer Time: 33:47)). So, P 2 reveal the fact that, Rao is head of the chemistry is lost as line 3 and 4 will be these two lines to be deleted, because when P 2 resigns.

(Refer Slide Time: 34:05)



EXAMPLE (CONTD)

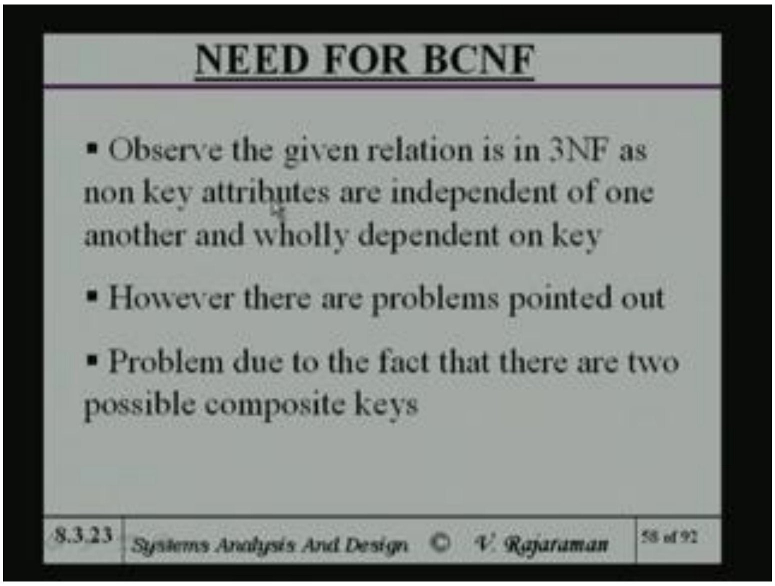
The dependency diagrams are:

- Percent time a Prof spends in the department is dependent on Prof code and Department
- Head of Dept depends on department

8.3.22 Systems Analysis And Design © V. Rajaraman 57 of 92

So, I lose information. So, that if you look at this dependency diagram, professor code, department code. So, head of the department function depended department. Similarly, department is also ((Refer Time: 34:21)) depend, I could have put the head of the department also has a possible composite key item. And so department and head of the department is really one dependency. So, in other words, I want to split, what I am trying to do is that this fact is the one, which P not attention to. And one try and look at eliminate that.

(Refer Slide Time: 35:00)



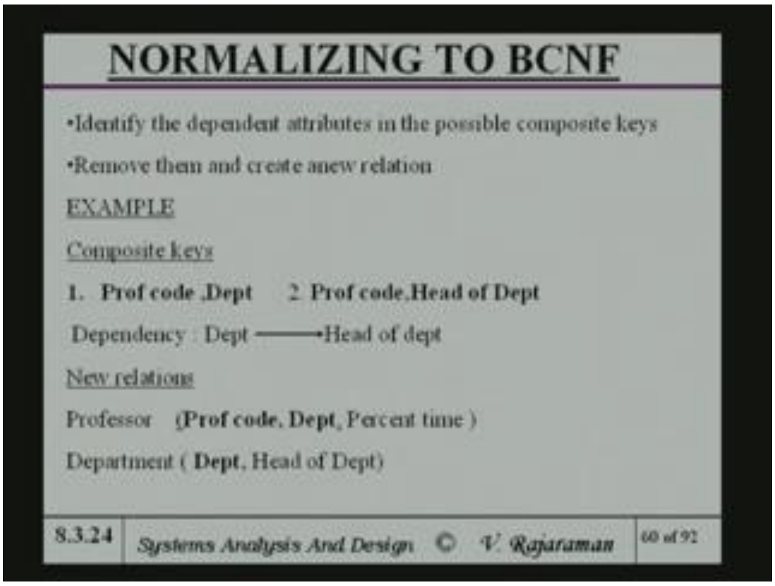
NEED FOR BCNF

- Observe the given relation is in 3NF as non key attributes are independent of one another and wholly dependent on key
- However there are problems pointed out
- Problem due to the fact that there are two possible composite keys

8.3.23 Systems Analysis And Design © V. Rajaraman 58 of 92

So, I cannot with the non key again, we have seen that the given relation in 3NF as non key attributes are there independent one another, over the problem to be find out. Problem due to the fact that, two possible composite keys. Now, one of the composite keys depends on the attribute another possible composite key, this leads the problem indicated.

(Refer Slide Time: 35:32)



NORMALIZING TO BCNF

•Identify the dependent attributes in the possible composite keys

•Remove them and create anew relation

EXAMPLE

Composite keys

1. **Prof code ,Dept** 2 **Prof code,Head of Dept**

Dependency : Dept ——→Head of dept

New relations

Professor (**Prof code, Dept**, Percent time)

Department (**Dept**, Head of Dept)

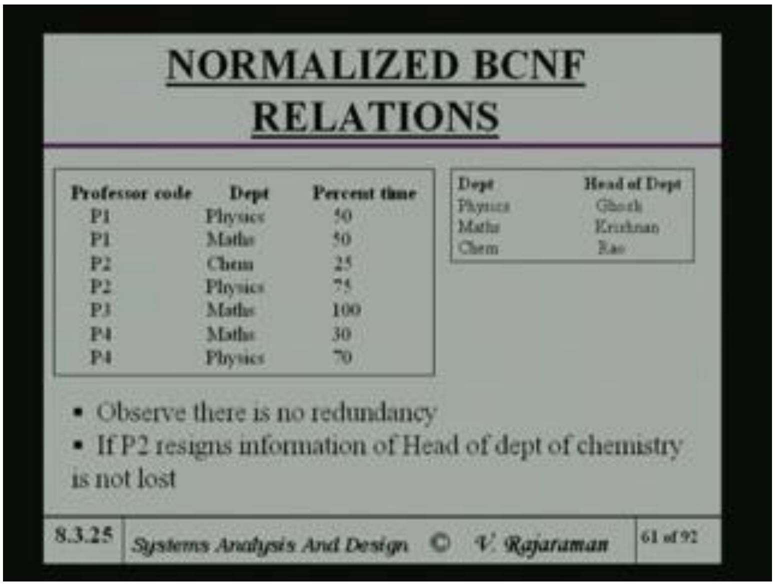
8.3.24 Systems Analysis And Design © V. Rajaraman 60 of 92

And so the normalization BCNF is based on identifying the dependent attributes in the possible composite key. Remove them and create a new relation. So, in this case, the composite keys are professor code, department and professor code and head of the department. And so, dependency with department add a department. So, more relations are I put professor, professor code and percent time. The another department does not appear here.

So, here some particular professor retires or resigns, the fact that the particular person is a head of the department is not lost. Because, the separate relation here, which is department and head of the department. So, this is whenever, a departments had changed only this is changed and will not touched at all. And another whenever, professor changes his percentile in a given department only this relation is touched, this is not touched. So, the whole point of BCNF is that, you start with BCNF.

And if you find out there is dependency, if there is a possibility one of the non key attributes ((Refer Time: 36:53)) a part of a composite key. If it is so, it requires a normalization BCNF and you go to the BCNF normalization.

(Refer Slide Time: 37:12)



NORMALIZED BCNF RELATIONS

Professor code	Dept	Percent time
P1	Physics	50
P1	Maths	50
P2	Chem	25
P2	Physics	75
P3	Maths	100
P4	Maths	30
P4	Physics	70

Dept	Head of Dept
Physics	Ghosh
Maths	Krishnan
Chem	Rao

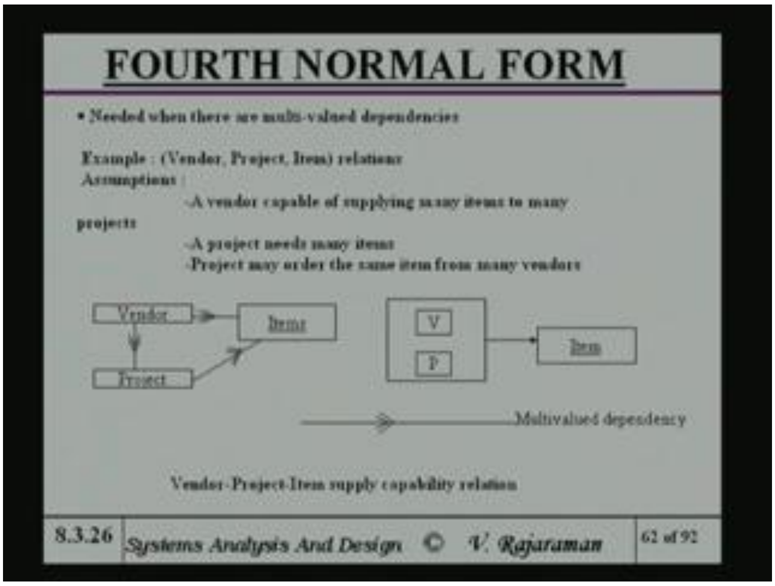
- Observe there is no redundancy
- If P2 resigns information of Head of dept of chemistry is not lost

8.3.25 Systems Analysis And Design © V. Rajaraman 61 of 92

So, this case the, I am just taking the relation ((Refer Time: 37:17)) observe there is no redundancy. If P 2 resigns information of the head of the chemistry is not lost. The same person mainly attributes in many places. And of course, the fact that P 1 appears twice is essentially information. Because, he is getting 50 percent, 50 percent in physics and math's. And P 2 similarly, is ((Refer Time: 37:41)) appear twice this is unique chemistry and physics. And P 3 of course, 100 percent it is only one line. Whereas, in this case there are two lines as far as the professor 2 is concerned.

So, examples have been shown over might say tie examples small examples. But still, we have to illustrate this kind of ((Refer Time: 38:08)) I want to make. If BCNF is, but 1 goes to higher normal forms in case they got multi value dependency. There is a, if I look at fourth normal norm even there are multi valued dependency.

(Refer Slide Time: 38:34)



FOURTH NORMAL FORM

- Needed when there are multi-valued dependencies

Example : (Vendor, Project, Item) relations
Assumptions :
-A vendor capable of supplying many items to many projects
-A project needs many items
-Project may order the same item from many vendors

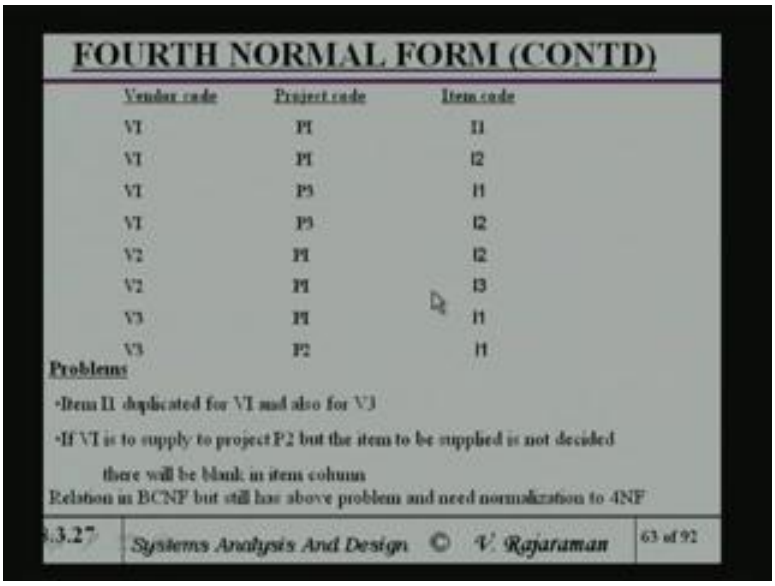
Vendor-Project-Item supply capability relation

8.3.26 Systems Analysis And Design © V. Rajaraman 62 of 92

In fact, fourth and fifth normal forms are required, only when you have multiple valued dependency not otherwise. In this case, I am taking a example of a project, a vendors, and items. Vendors and items are multiple value dependency. In other words, many vendors can apply item for the same project. And also, what I am saying is vendor is giving a supplies many items to many projects. And project means many items. And project order of same item from many vendors.

So, items are supplied on many vendors. Vendors is supplying many items and project of course, obviously. So, multi valued dependency for take once is vendor and project as a composite key, item is depended on that. The vendor and project have been supplied capability relation. This is this vendor is capable of supplying this item to this project result indicates.

(Refer Slide Time: 40:04)



FOURTH NORMAL FORM (CONTD)

Vendor code	Project code	Item code
V1	P1	I1
V1	P1	I2
V1	P3	I1
V1	P3	I2
V2	P1	I2
V2	P1	I3
V3	P1	I1
V3	P2	I1

Problems

- Item I1 duplicated for V1 and also for V3
- If V1 is to supply to project P2 but the item to be supplied is not decided there will be blank in item column

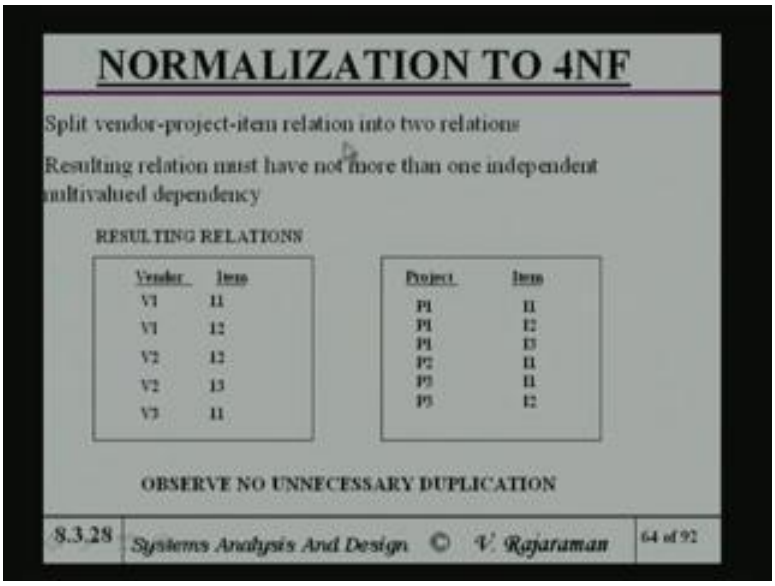
Relation in BCNF but still has above problem and need normalization to 4NF

3.27 Systems Analysis And Design © V. Rajaraman 63 of 92

So, if I look in the whole relation v 1, if I look at V 1 P 1 I 1, V 1 P 1 and I 2 and so on. That means, vendor V 1 came supply I 1 I 2 to the project 1, can supply the project 3. And vendor 2 can supply I 2 and I 3 the project 1. And vendor 3 can supply I 1 to project p 1. And P 3 also supply I 1 this same item. Item I 1 is duplicated for V 1 and also for V 3. V 1 is here and V 3 also it is here. V 1 is supplies the project P 2 at the item is to supply is not decided. In other words, V 1 is to supply to the project P 2. This case it is only supplying to P 1 and p 3.

If suppose P 2 the item supply, what item is to supply is not decided, then there will be a blank here. The relation in BCNF is still ((Refer Time: 41:29)) needs a further normalization. It is a BCNF, because the only one non key attributes, which cannot be a key.

(Refer Slide Time: 41:40)



NORMALIZATION TO 4NF

Split vendor-project-item relation into two relations

Resulting relation must have not more than one independent multivalued dependency

RESULTING RELATIONS

Vendor	Item
V1	I1
V1	I2
V2	I2
V2	I3
V3	I1

Project	Item
P1	I1
P1	I2
P1	I3
P2	I1
P3	I1
P3	I2

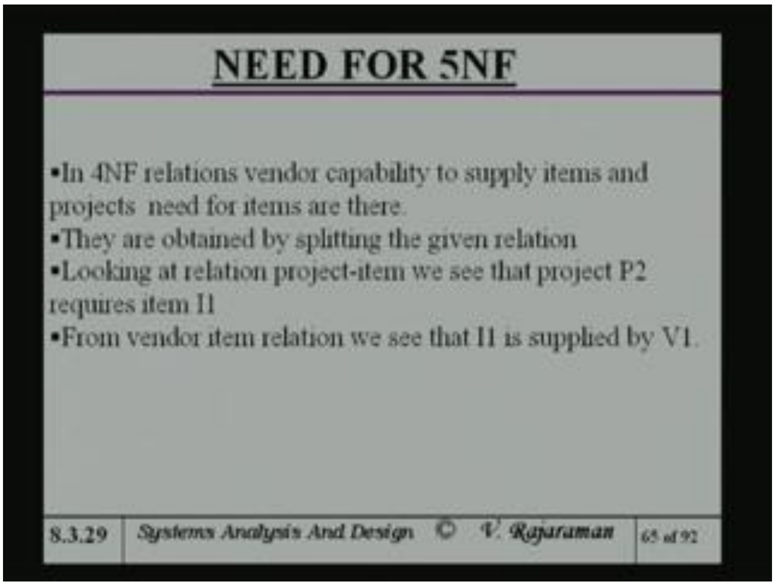
OBSERVE NO UNNECESSARY DUPLICATION

8.3.28 Systems Analysis And Design © V. Rajaraman 64 of 92

If I split the vendor item taking a two relations. Resulting relation may have one and multi valued dependency. In other words, not more than independent multi valued dependency. So, resulting relation is V 1 I 1, V 1 I 2, V 2 I 3 observe unnecessary duplication and project P 1 I 1 I 2 I 3 are required. This shows project requirements, this shows the capability requirements. So, when the capability is there, suppose vendor 1 is cable also supplying I 3. It will appear here, V 1 I 3 appear here. And ((Refer Time: 42:26)) not ((Refer Time: 42:28)) blank.

Similarly, project P 1 requires an item I 4. Then I put I 4 here and there will not be a blank. Because, previously the project need not kind of have this requirements.

(Refer Slide Time: 42:49)



NEED FOR 5NF

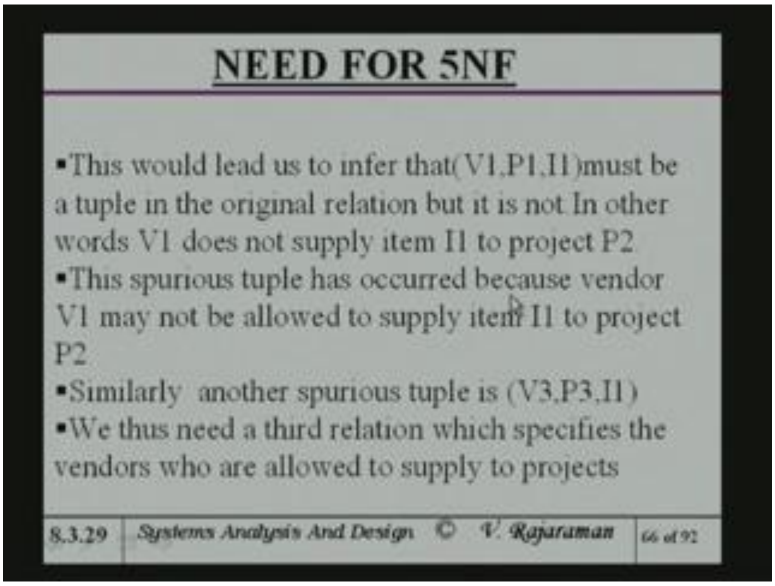
- In 4NF relations vendor capability to supply items and projects need for items are there.
- They are obtained by splitting the given relation
- Looking at relation project-item we see that project P2 requires item I1
- From vendor item relation we see that I1 is supplied by V1.

8.3.29 Systems Analysis And Design © V. Rajaraman 65 of 92

So, this 4NF steps I am not going to detail, because a lot of theory behind it. And you will learn about this in much detail, when course and database management system. And this giving a flower of normalization step. But, the examples I am going to be illustrating with some case examples. Primarily, we stop with 3NF or at the most BCNF. In 4NF relation vender capability to supply items and projects need for items is not there. They are obtained by splitting the given relation.

For vendor item relation we say I 1 is supplied V 1 that mean saw that see. We saw that project items ((Refer Time: 43:45)) we see that ((Refer Time: 43:48)) P 2 requires item I 1. And vendor relation I 1 is supplied by v 1 that is quite clear.

(Refer Slide Time: 44:05)



NEED FOR 5NF

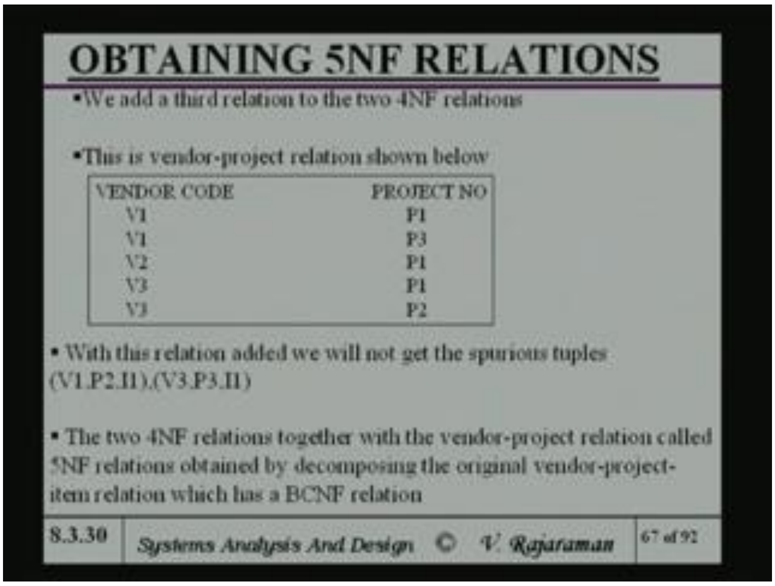
- This would lead us to infer that(V1,P1,I1)must be a tuple in the original relation but it is not In other words V1 does not supply item I1 to project P2
- This spurious tuple has occurred because vendor V1 may not be allowed to supply item I1 to project P2
- Similarly another spurious tuple is (V3,P3,I1)
- We thus need a third relation which specifies the vendors who are allowed to supply to projects

8.3.29 Systems Analysis And Design © V. Rajaraman 66 of 92

This is leads to infer that V 1 P 1 I 1 must be a tuple in the original relation. But, it is not really so. So, ((Refer Time: 44:14)) this is vendor supply to vendor Item relation we say I 1 supply the project p 2. So P 2, ((Refer Time: 44:27)) V 1 P 2 I 1 is not here. P 3 P 2, V 3 P 2 I 1 is there. So, it really means, that the spurious tuple has occurred, because vendor V 1 may not be allowed to supply item I 1 to project P 2. That is not reflected in the relation.

So, another spurious tuple is V 3 P 3 I 1 ((Refer Time: 44:55)), which we can infer V 3 P 3, P 3 requires I 1, V 3 supplies I 1. We may think that V 3 I 1 is the relation, which may not actually be there, because he may not be allowed.

(Refer Slide Time: 45:11)



OBTAINING 5NF RELATIONS

- We add a third relation to the two 4NF relations
- This is vendor-project relation shown below

VENDOR CODE	PROJECT NO
V1	P1
V1	P3
V2	P1
V3	P1
V3	P2

- With this relation added we will not get the spurious tuples (V1.P2.I1).(V3.P3.I1)
- The two 4NF relations together with the vendor-project relation called 5NF relations obtained by decomposing the original vendor-project-item relation which has a BCNF relation

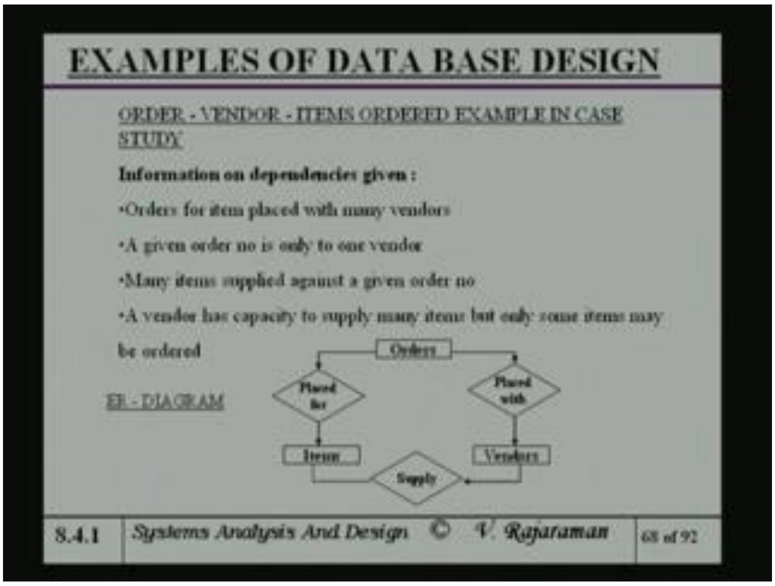
8.3.30 Systems Analysis And Design © V. Rajaraman 67 of 92

We need a third relation, this vendor project relation that is it shows, that vendor can give vendor ((Refer Time: 45:21)) project and not any project. So, to 4NF relations to where form the vendor project relation called 5NF relation. ((Refer Time: 45:30)) now, I have got 3 relations. And because the 3 attributes or all multi valued dependency I split up each one of the multiples value dependency into a separate relation.

So, effectively if you go back to the original here. So, end up ((Refer Time: 45:47)), because the multi valued dependency. I end up with vendor item relation, project item relation and vendor project relation. So, there is a fifth normal form. So, whenever you find, such multiple value dependency then ((Refer Time: 46:13)) it is kind of timed split up. Then we split them into multiple we might say relations. And these relations are supposed to be in fifth normal form.

And as I said, they will not occurring all the time practice, it is a people are kind of it is a multi valued dependency and so on. And vendor capability and so on, it is not a very common occurrence. But, any have you should know these things. You should know before the full theory detail I think should know these things to exists. So, I add one more and get, let me make some more examples of data base design.

(Refer Slide Time: 47:10)



EXAMPLES OF DATA BASE DESIGN

ORDER - VENDOR - ITEMS ORDERED EXAMPLE IN CASE STUDY

Information on dependencies given :

•Orders for item placed with many vendors

•A given order no is only to one vendor

•Many items supplied against a given order no

•A vendor has capacity to supply many items but only some items may be ordered

ER - DIAGRAM

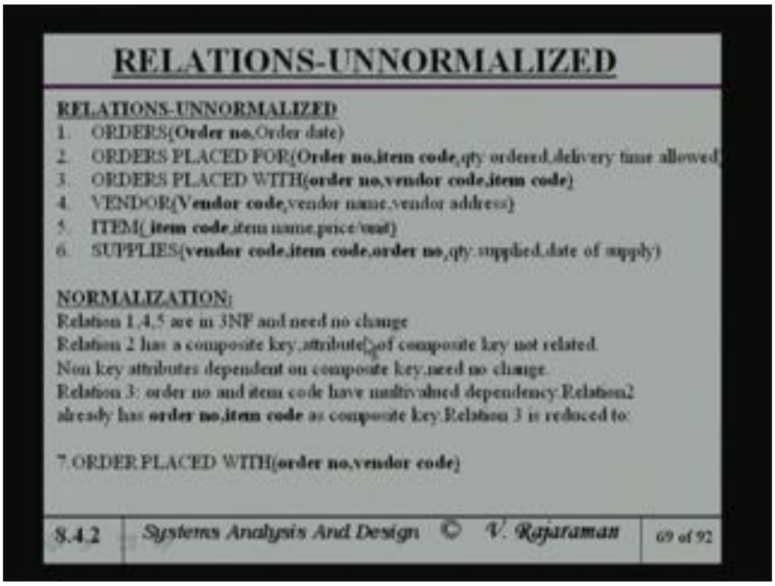
8.4.1 Systems Analysis And Design © V. Rajaraman 68 of 92

The real ((Refer Time: 47:13)) are example and design to regulate the things, which are learnt and to apply our ideas to the examples. Order vendor item ordered examples is lifted case study at beginning of these lectures. Where, company orders to items ((Refer Time: 47:36)) vendor. And the vendor applies item to the company goes to infection office and then purchase and so on. So, ((Refer Time: 47:46)) I complete data base diagram. And look at the entire case study.

Some ((Refer Time: 47:51)) part of the case study, I am looking at the vendor ordered item relationship this case. Orders for items placed with many vendors. Again order number is occurred to only one vendor. Many items applied against a given order. The given order may have many items applied. Vendor ((Refer Time: 48:23)) supply many items applied only some items may be ordered.

So, vendors can supply many items, but I only can sometime you know ((Refer Time: 48:34)). So generally, here the diagram orders placed with vendors ((Refer Time: 48:43)) relationship. And orders placed for items and vendors supplies items.

(Refer Slide Time: 48:49)



RELATIONS-UNNORMALIZED

RELATIONS-UNNORMALIZED

1. ORDERS(**Order no**,Order date)
2. ORDERS PLACED FOR(**Order no,item code**,qty ordered,delivery time allowed)
3. ORDERS PLACED WITH(**order no,vendor code,item code**)
4. VENDOR(**Vendor code**,vendor name,vendor address)
5. ITEM(**item code**,item name,price/unit)
6. SUPPLIES(**vendor code,item code,order no**,qty supplied,date of supply)

NORMALIZATION:

Relation 1,4,5 are in 3NF and need no change

Relation 2 has a composite key,attributes of composite key not related.

Non key attributes dependent on composite key,need no change.

Relation 3: order no and item code have multivalued dependency.Relation2 already has **order no,item code** as composite key.Relation 3 is reduced to:

7.ORDER PLACED WITH(**order no,vendor code**)

8.4.2 Systems Analysis And Design © V. Rajaraman 69 of 92

So, un normalized relations or orders, order no, order date. So, this may looking at this diagram ((Refer Time: 49:00)) I can just can the relation and normalized. There is a starting point, orders order number, order date. Order placed was order number item code, quantity order, delivery time allowed. And order placed with order number vendor code item code. And the vendor code, vendor name, vendor address and item, item code, item name, price per unit. Supplies vendor code, item code, order number ((Refer Time: 49:29)).

Relations 1, 4 and 5 if you look at relation 1, it is a straight forward one. What is the key and there is a norm key. And this is functionally depended on norm key 4 is also a vendor, vendor code is no composite key is here see, single key is there. And non key attributes are depending only on the key attributes. So, I have no problem.

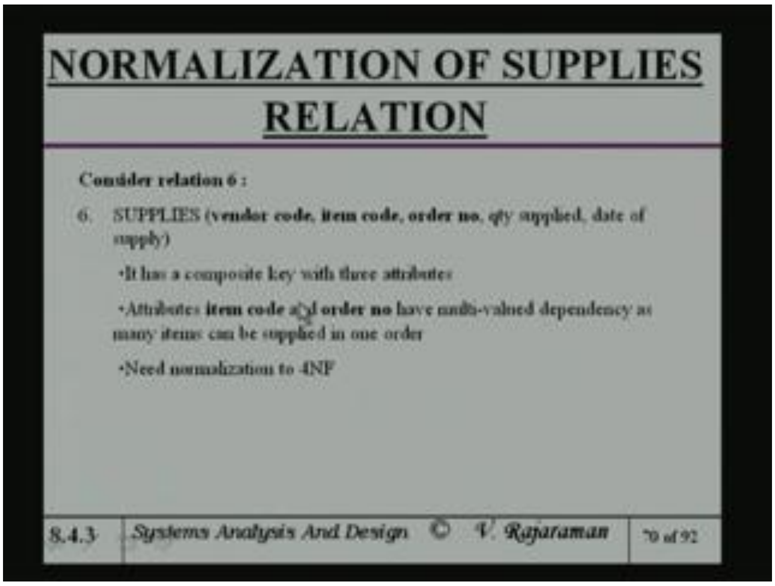
And similarly 5 item code, single key and item is price per unit or depended on the item code. So, there is no problem. So in fact, 1 4 and 5 are already in third normal form. Because, here is only one non key attributes and item code and price per unit are related. Whereas, item name there can be many item, item name is not a possible key at all.

And these two are not really related. Because, this relationship is already is there, only something which is not unique. So, it can be a key also, so it is not functionally depended. Relation 2 has a composite key, order number item code is a composite key. Order number composite key are not related. So, order number item code they are two independent orthogonal.

No key attributes depended on composite these not need for a mistake. Because, if I take a non key attributes, quantity order, delay time and time allowed. None of them is depended on a part the key attribute. And none of these are depended on none another. Quantity order is not related took delivery time allowed these are independent quantities. So, in a functional dependency here. And none of these functional dependent part of this. So, there is no need for any ((Refer Time: 51:54)).

Order number and item code are multivariate dependency. But, the order number will remain to already has order number item code is composite key. So, relation 3 can be reduced, order price for order number vendor code. And because, we look at this one, the item code the 3 not really necessary. And order number one item code is known. And if you know the vendor code, the item code appears right here So, you add that relation.

(Refer Slide Time: 52:37)



NORMALIZATION OF SUPPLIES RELATION

Consider relation 6 :

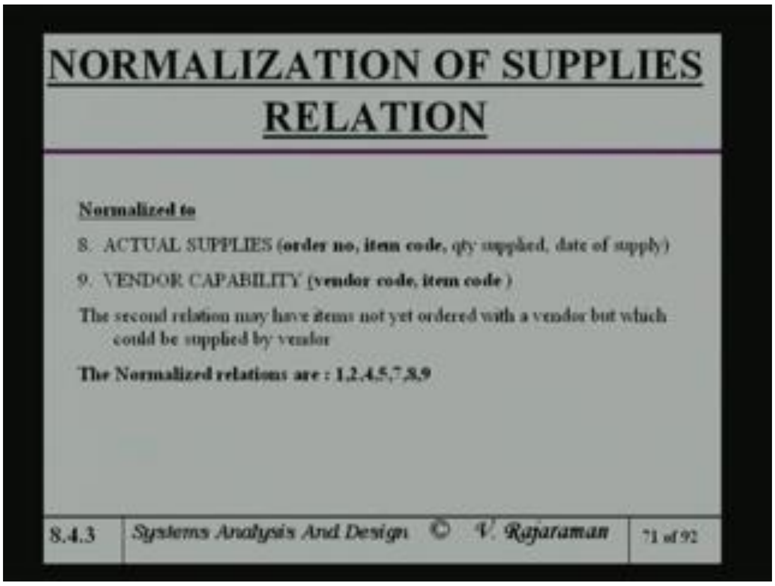
6. SUPPLIES (**vendor code, item code, order no**, *qty* supplied, date of supply)

- It has a composite key with three attributes
- Attributes **item code and order no** have multi-valued dependency as many items can be supplied in one order
- Need normalization to 4NF

8.4.3 | *Systems Analysis And Design* © *V. Rajaraman* | 70 of 92

Supplies again there is the possibility problem here. The three it has a composite key is the three attributes, item code in order no, multivariate dependency has had many items can supply in one order, these normalization into 4 NF.

(Refer Slide Time: 52:56)



NORMALIZATION OF SUPPLIES RELATION

Normalized to

8. ACTUAL SUPPLIES (**order no, item code,** qty supplied, date of supply)

9. VENDOR CAPABILITY (**vendor code, item code**)

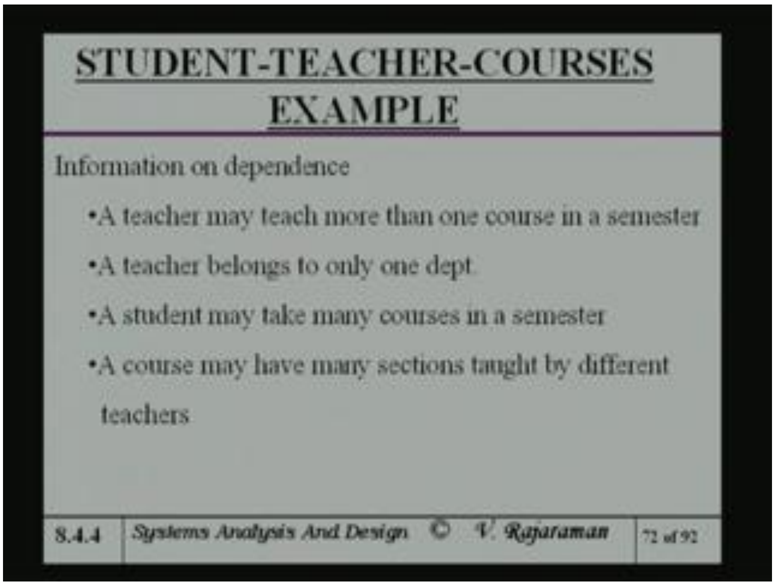
The second relation may have items not yet ordered with a vendor but which could be supplied by vendor

The Normalized relations are : 1,2,4,5,7,8,9

8.4.3 Systems Analysis And Design © V. Rajaraman 71 of 92

So, normalize to 4 NF actual supplies order no item code, quantity supply, dealer supply, vendor capability. So, the split up the multiple value dependency and t separate new relation. So, second relation is one more relation, vendor code, item code is there. So, the normalizations relations are 1, 2, 4, 5 no other problem plus 7, 8 9. So, I replace the old set by this new set.

(Refer Slide Time: 53:34)



STUDENT-TEACHER-COURSES EXAMPLE

Information on dependence

- •A teacher may teach more than one course in a semester
- •A teacher belongs to only one dept.
- •A student may take many courses in a semester
- •A course may have many sections taught by different teachers

8.4.4 Systems Analysis And Design © V. Rajaraman 72 of 92

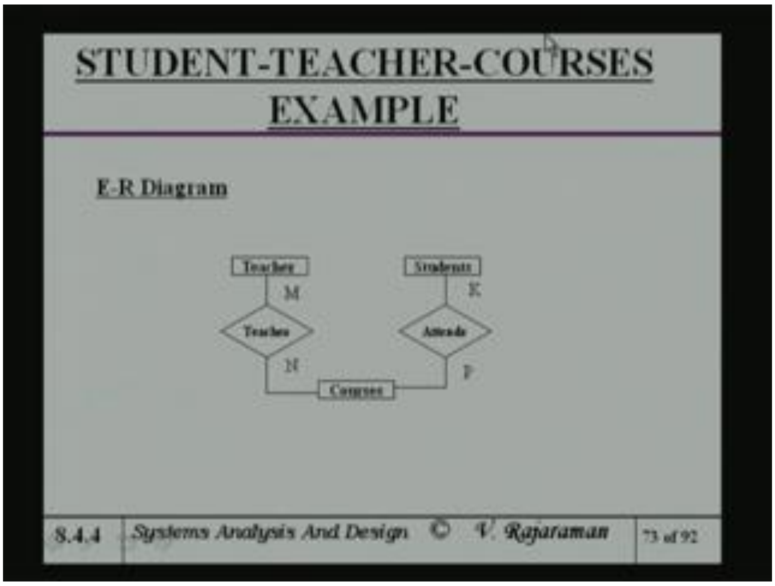
So, these kinds of steps you normally follow, take another example. Student, teacher courses example. Teacher may teach more than one course in a semester. And these are the assumption have made. And there is dependence based on the assumptions and teacher belongs to only one department. I am assuming that it is not a similar situation; there a teacher has part time, various department.

Teacher belongs only one department, assuming that there are many courses in the semester. A course may have many sections for the different teachers. So, suppose ((Refer Time: 54:16)) the class instead of one teacher giving to all the 300 students. Now, we divide up class into 10 sections of 30 students each. And 10 different teacher will be teaching the 10 sections, it is possible I am assuming that. A course may have several sections for several different teachers.

(Refer Slide Time: 54:47)

So, that means is a multivariate dependency Teacher can teach many courses and a course can be taught many teachers. And similarly, and students attends many courses and courses may be attended by the students obviously.

(Refer Slide Time: 55:15)



STUDENT-TEACHER-COURSES EXAMPLE

E-R Diagram

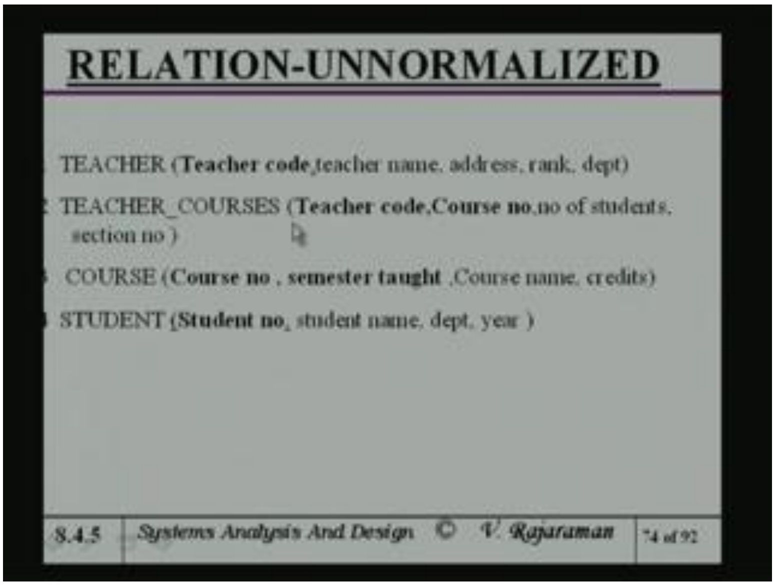
8.4.4 Systems Analysis And Design © V. Rajaraman 73 of 92

So, I make the relation un normalized formed begin with, teacher code, teacher name, address, rank and department. Now, delegate the relation it is valid straight forward relation. The non key attributes are not related to one another. Because, the teacher code determines, the department ((Refer Time: 55:51)). Because, we said that a teacher belongs to only one department not a multiple departments.

Because, name and address and rank depend up on the teacher code. The teacher courses got teacher code, course number, number of students, section. And course, course number, semester taught, course ((Refer Time: 56:11)) credits, these are two possible correlates may arise here. But, not here because student number, student name, department, year here there is no non key attributes or not dependent one another.

Because, department, year and dependent, name and department are not related. So, I think one and four can be kept as they are. And let us look at course 3, look at the third one, course number semester taught, course name and credits, where there is no problem here. Because, course name and credits are not dependent on a partial part is so. In other words, the courses name something which is that really a possible key at all. And privates of course, unique depending on the course.

(Refer Slide Time: 57:18)



RELATION-UNNORMALIZED

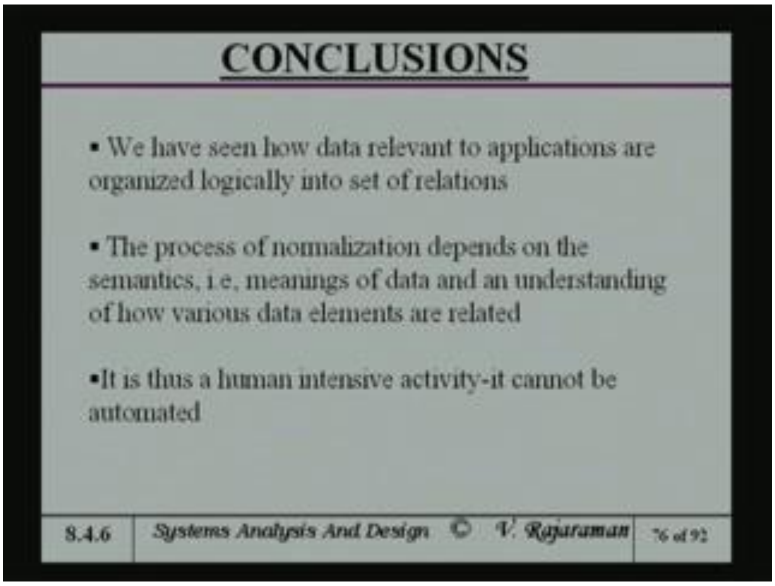
1. TEACHER (**Teacher code**,teacher name, address, rank, dept)
2. TEACHER_COURSES (**Teacher code,Course no**,no of students, section no)
3. COURSE (**Course no , semester taught** ,Course name, credits)
4. STUDENT (**Student no**, student name, dept, year)

8.4.5 | *Systems Analysis And Design* © *V. Rajaraman* | 74 of 92

1, 3 and 4 can be kept as they are. Relation 2 and 5 multi attribute key, which has multi-valued dependency. But, looks although we not need normalization or information on which teacher teach as a given student specified course, cannot be found from relation 1 to 5. So, add a new relation, teacher student, teacher code, student number, course number. This set is normalize. In other words, now I can find out, this teacher, teach gave a student in the specified course.

Because, I said a particular course can be taught by any teachers. So, same student may take the same course for many teachers. So, this something which is not reflected in any of the previous relations ((Refer Time: 58:11)). Because, I cannot infer that from any of these 1 3 4 and 5. So, I need a new relation and I get now this set normalized.

((Refer Slide Time: 58:35))



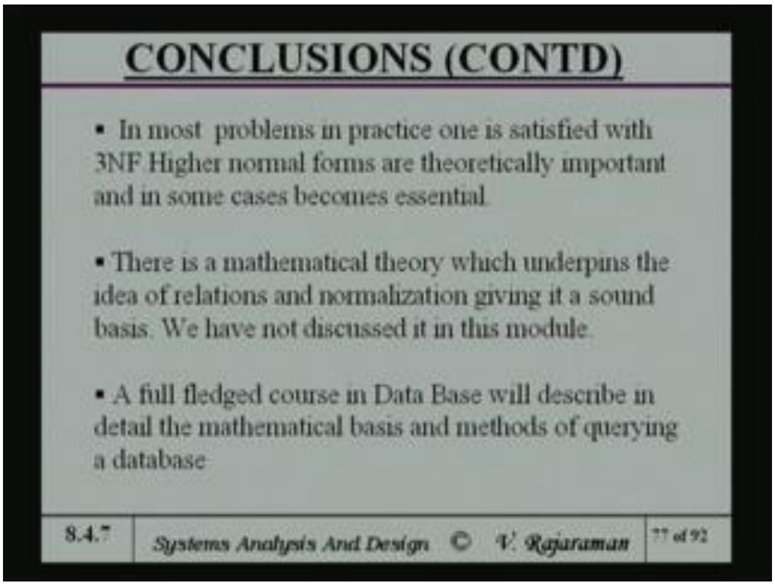
CONCLUSIONS

- We have seen how data relevant to applications are organized logically into set of relations
- The process of normalization depends on the semantics, i.e. meanings of data and an understanding of how various data elements are related
- It is thus a human intensive activity-it cannot be automated

8.4.6 | *Systems Analysis And Design* © *V. Rajaraman* | 76 of 92

So, conclude we have seen how data relevant applications organized logically to several relations. The process of normalization depends on the semantics, meanings. So, human intensive activity it is not computer based.

((Refer Slide Time: 58:49))



CONCLUSIONS (CONTD)

- In most problems in practice one is satisfied with 3NF Higher normal forms are theoretically important and in some cases becomes essential
- There is a mathematical theory which underpins the idea of relations and normalization giving it a sound basis. We have not discussed it in this module.
- A full fledged course in Data Base will describe in detail the mathematical basis and methods of querying a database

8.4.7 | *Systems Analysis And Design* © *V. Rajaraman* | 77 of 92

And one is normally satisfied with 3 NF, as well. So, if we have mathematical theory, which handles all these, which will not talk about. So, next time I will kind of look at DBMS Data Base Management Systems and here I stop the idea of normalization.